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

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or 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 rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, 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 total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or 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 in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

RENESAS

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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 where the specifications for values which are read from or written to the bit are provided in the description.
The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.
Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
    - CPU and System-Control Modules
    - On-Chip Modules

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

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

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

7. 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 versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

10. Index

Preface

This LSI is a RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas Technology-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of 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 into parts on the CPU, system control functions, peripheral functions and electrical characteristics.

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 xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Abbreviations

ALU	Arithmetic Logic Unit
ASID	Address Space Identifier
BGA	Ball Grid Array
CMT	Compare Match Timer (Timer/Counter)
CPG	Clock Pulse Generator
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

MMCIF	Multimedia Card Interface
MMU	Memory Management Unit
MSB	Most Significant Bit
PC	Program Counter
PCI	Peripheral Component Interconnect
PCIC	PCI (local bus) Controller
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

Contents

Section 1 Overview 1
1.1 SH7780 Features 1
1.2 Block Diagram 9
1.3 Pin Arrangement 10
1.4 Pin Functions 11
1.5 Memory Address Map 27
1.6 SuperHyway Bus 30
1.7 SuperHyway Memory (SuperHyway RAM) 31

Section 2 Programming Model 33
2.1 Data Formats 33
2.2 Register Descriptions 34
2.2.1 Privileged Mode and Banks 34
2.2.2 General Registers 37
2.2.3 Floating-Point Registers 38
2.2.4 Control Registers 40
2.2.5 System Registers 42
2.3 Memory-Mapped Registers 46
2.4 Data Formats in Registers 47
2.5 Data Formats in Memory 48
2.6 Processing States 49
2.7 Usage Note 50
2.7.1 Notes on self-modified codes 50

Section 3 Instruction Set 51
3.1 Execution Environment 51
3.2 Addressing Modes 53
3.3 Instruction Set 57

Section 4 Pipelining 73
4.1 Pipelines 73
4.2 Parallel-Executability 84
4.3 Issue Rates and Execution Cycles 87

Section 5 Exception Handling 97
5.1 Summary of Exception Handling 97

5.2 Register Descriptions 97
5.2.1 TRAPA Exception Register (TRA) 98
5.2.2 Exception Event Register (EXPEVT) 99
5.2.3 Interrupt Event Register (INTEVT) 100
5.3 Exception Handling Functions 101
5.3.1 Exception Handling Flow 101
5.3.2 Exception Handling Vector Addresses 101
5.4 Exception Types and Priorities 102
5.5 Exception Flow 104
5.5.1 Exception Flow 104
5.5.2 Exception Source Acceptance 106
5.5.3 Exception Requests and BL Bit 107
5.5.4 Return from Exception Handling 107
5.6 Description of Exceptions 108
5.6.1 Resets 108
5.6.2 General Exceptions 110
5.6.3 Interrupts 124
5.6.4 Priority Order with Multiple Exceptions 125
5.7 Usage Notes 127

Section 6 Floating-Point Unit (FPU) 129
6.1 Features 129
6.2 Data Formats 130
6.2.1 Floating-Point Format 130
6.2.2 Non-Numbers (NaN) 133
6.2.3 Denormalized Numbers 134
6.3 Register Descriptions 135
6.3.1 Floating-Point Registers 135
6.3.2 Floating-Point Status/Control Register (FPSCR) 137
6.3.3 Floating-Point Communication Register (FPUL) 140
6.4 Rounding 141
6.5 Floating-Point Exceptions 142
6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions 142
6.5.2 FPU Exception Sources 142
6.5.3 FPU Exception Handling 142
6.6 Graphics Support Functions 144
6.6.1 Geometric Operation Instructions 144
6.6.2 Pair Single-Precision Data Transfer 145

Section 7 Memory Management Unit (MMU) 147
7.1 Overview of MMU 147
7.1.1 Address Spaces 149
7.2 Register Descriptions 156
7.2.1 Page Table Entry High Register (PTEH) 157
7.2.2 Page Table Entry Low Register (PTEL) 158
7.2.3 Translation Table Base Register (TTB) 159
7.2.4 TLB Exception Address Register (TEA) 159
7.2.5 MMU Control Register (MMUCR) 160
7.2.6 Physical Address Space Control Register (PASCR) 164
7.2.7 Instruction Re-Fetch Inhibit Control Register (IRMCR) 165
7.3 TLB Functions 167
7.3.1 Unified TLB (UTLB) Configuration 167
7.3.2 Instruction TLB (ITLB) Configuration 170
7.3.3 Address Translation Method 171
7.4 MMU Functions 173
7.4.1 MMU Hardware Management 173
7.4.2 MMU Software Management 173
7.4.3 MMU Instruction (LDTLB) 174
7.4.4 Hardware ITLB Miss Handling 175
7.4.5 Avoiding Synonym Problems 176
7.5 MMU Exceptions 177
7.5.1 Instruction TLB Multiple Hit Exception 177
7.5.2 Instruction TLB Miss Exception 178
7.5.3 Instruction TLB Protection Violation Exception 179
7.5.4 Data TLB Multiple Hit Exception 180
7.5.5 Data TLB Miss Exception 180
7.5.6 Data TLB Protection Violation Exception 181
7.5.7 Initial Page Write Exception 182
7.6 Memory-Mapped TLB Configuration 183
7.6.1 ITLB Address Array 184
7.6.2 ITLB Data Array 185
7.6.3 UTLB Address Array 186
7.6.4 UTLB Data Array 187
7.7 32-Bit Address Extended Mode 188
7.7.1 Overview of 32-Bit Address Extended Mode 189
7.7.2 Transition to 32-Bit Address Extended Mode 189
7.7.3 Privileged Space Mapping Buffer (PMB) Configuration 189
7.7.4 PMB Function 191

7.7.5 Memory-Mapped PMB Configuration 192
7.7.6 Notes on Using 32-Bit Address Extended Mode 194

Section 8 Caches 197
8.1 Features 197
8.2 Register Descriptions 200
8.2.1 Cache Control Register (CCR) 201
8.2.2 Queue Address Control Register 0 (QACR0) 203
8.2.3 Queue Address Control Register 1 (QACR1) 204
8.2.4 On-Chip Memory Control Register (RAMCR) 205
8.3 Operand Cache Operation 207
8.3.1 Read Operation 207
8.3.2 Prefetch Operation 208
8.3.3 Write Operation 209
8.3.4 Write-Back Buffer 211
8.3.5 Write-Through Buffer 211
8.3.6 OC Two-Way Mode 211
8.4 Instruction Cache Operation 212
8.4.1 Read Operation 212
8.4.2 Prefetch Operation 213
8.4.3 IC Two-Way Mode 213
8.5 Cache Operation Instruction 214
8.5.1 Coherency between Cache and External Memory 214
8.5.2 Prefetch Operation 215
8.6 Memory-Mapped Cache Configuration 216
8.6.1 IC Address Array 217
8.6.2 IC Data Array 219
8.6.3 OC Address Array 220
8.6.4 OC Data Array 222
8.7 Store Queues 223
8.7.1 SQ Configuration 223
8.7.2 Writing to SQ 223
8.7.3 Transfer to External Memory 224
8.7.4 Determination of SQ Access Exception 225
8.7.5 Reading from SQ 225
8.8 Notes on Using 32-Bit Address Extended Mode 226

Section 9 L Memory 227
9.1 Features 227
9.2 Register Descriptions 228

9.2.1 On-Chip Memory Control Register (RAMCR) 229
9.2.2 L Memory Transfer Source Address Register 0 (LSA0) 230
9.2.3 L Memory Transfer Source Address Register 1 (LSA1) 232
9.2.4 L Memory Transfer Destination Address Register 0 (LDA0) 234
9.2.5 L Memory Transfer Destination Address Register 1 (LDA1) 236
9.3 Operation 238
9.3.1 Access from the CPU and FPU 238
9.3.2 Access from the SuperHyway Bus Master Module 238
9.3.3 Block Transfer 238
9.4 L Memory Protective Functions 240
9.5 Usage Notes 241
9.5.1 Page Conflict 241
9.5.2 L Memory Coherency 241
9.5.3 Sleep Mode 241
9.6 Note on Using 32-Bit Address Extended Mode 241

Section 10 Interrupt Controller (INTC) 243
10.1 Features 243
10.1.1 Interrupt Method 245
10.1.2 Interrupt Types in INTC 246
10.2 Input/Output Pins 250
10.3 Register Descriptions 251
10.3.1 Interrupt Control Register 0 (ICR0) 255
10.3.2 Interrupt Control Register 1 (ICR1) 258
10.3.3 Interrupt Priority Register (INTPRI) 259
10.3.4 Interrupt Source Register (INTREQ) 260
10.3.5 Interrupt Mask Registers (INTMSK0 to INTMSK2) 261
10.3.6 Interrupt Mask Clear Registers (INTMSKCLR0 to INTMSKCLR2) 266
10.3.7 NMI Flag Control Register (NMIFCR) 271
10.3.8 User Interrupt Mask Level Register (USERIMASK) 273
10.3.9 On-chip Module Interrupt Priority Registers (INT2PRI0 to INT2PRI7) 276
10.3.10 Interrupt Source Register (INT2A0: Not affected by Mask States) 277
10.3.11 Interrupt Source Register (INT2A1: Affected by Mask States) 280
10.3.12 Interrupt Mask Register (INT2MSKR) 282
10.3.13 Interrupt Mask Clear Register (INT2MSKCR) 285
10.3.14 On-chip Module Interrupt Source Registers (INT2B0 to INT2B7) 287
10.3.15 GPIO Interrupt Set Register (INT2GPIC) 294
10.4 Interrupt Sources 296
10.4.1 NMI Interrupt 296
10.4.2 IRQ Interrupts 296

10.4.3 IRL Interrupts 297
10.4.4 On-chip Module Interrupts 299
10.4.5 Interrupt Priority Levels of On-chip Module Interrupts 300
10.4.6 Interrupt Exception Handling and Priority 301
10.5 Operation 308
10.5.1 Interrupt Sequence 308
10.5.2 Multiple Interrupts 310
10.5.3 Interrupt Masking by MAI Bit 310
10.6 Interrupt Response Time 311
10.7 Usage Notes 312
10.7.1 To Clear Interrupt Request When Holding Function Selected 312
10.7.2 Notes on Setting IRQ/$\overline{\text{IRL}}$[7:0] Pin Function 313
10.7.3 To clear IRQ and IRL interrupt requests 313

Section 11 Local Bus State Controller (LBSC) 315
11.1 Features 315
11.2 Input/Output Pins 318
11.3 Area Overview 320
11.3.1 Space Divisions 320
11.3.2 Memory Bus Width 324
11.3.3 Data Alignment 325
11.3.4 PCMCIA Support 325
11.4 Register Descriptions 329
11.4.1 Memory Address Map Select Register (MMSELR) 331
11.4.2 Bus Control Register (BCR) 333
11.4.3 CSn Bus Control Register (CSnBCR) 336
11.4.4 CSn Wait Control Register (CSnWCR) 342
11.4.5 CSn PCMCIA Control Register (CSnPCR) 347
11.5 Operation 352
11.5.1 Endian/Access Size and Data Alignment 352
11.5.2 Areas 357
11.5.3 SRAM interface 361
11.5.4 Burst ROM (Clock Asynchronous) Interface 370
11.5.5 PCMCIA Interface 372
11.5.6 MPX Interface 383
11.5.7 Byte Control SRAM Interface 389
11.5.8 Wait Cycles between Accesses 393
11.5.9 Bus Arbitration 395
11.5.10 Bus Release and Acquire Sequence 397
11.5.11 Cooperation between Master and Slave 399

Section 12 DDR-SDRAM Interface (DDRIF) 401
12.1 Features 401
12.2 Input/Output Pins 403
12.3 Address Space, Bus Width, and Data Alignment 404
12.3.1 Address Space of the DDRIF 404
12.3.2 Memory Data Bus Width 405
12.3.3 Data Alignment 406
12.4 Register Descriptions 410
12.4.1 Memory Interface Mode Register (MIM) 412
12.4.2 SDRAM Control Register (SCR) 416
12.4.3 SDRAM Timing Register (STR) 418
12.4.4 SDRAM Row Attribute Register (SDR) 421
12.4.5 SDRAM Mode Register (SDMR) 422
12.4.6 DDR-SDRAM Back-up Register (DBK) 424
12.5 Operation 425
12.5.1 DDR-SDRAM Access 425
12.5.2 DDR-SDRAM Initialization Sequence 425
12.5.3 Supported SDRAM Commands 426
12.5.4 SDRAM Access Mode 427
12.5.5 Power-Down Modes 427
12.5.6 Address Multiplexing 429
12.6 DDR-SDRAM Basic Timing 430
12.7 Usage Notes 440
12.7.1 Operating Frequency 440
12.7.2 Stopping Clock 440
12.7.3 Using SCR to Issue REFA Command (Outside the Initialization Sequence) 440
12.7.4 Timing of Connected SDRAM 440
12.7.5 Setting Auto-Refresh Interval 441

Section 13 PCI Controller (PCIC) 443
13.1 Features 443
13.2 Input/Output Pins 446
13.3 Register Descriptions 449
13.3.1 PCIC Enable Control Register (PCIECR) 455
13.3.2 Configuration Registers 456
13.3.3 Local Register 481
13.4 Operation 522
13.4.1 Supported PCI Commands 522
13.4.2 PCIC Initialization 523
13.4.3 Master Access 524

13.4.4 Target Access 532
13.4.5 Host Bus Bridge Mode 541
13.4.6 Normal mode 544
13.4.7 Power Management 544
13.4.8 PCI Local Bus Basic Interface 545

Section 14 Direct Memory Access Controller (DMAC) 557
14.1 Features 557
14.2 Input/Output Pins 559
14.3 Register Descriptions 561
14.3.1 DMA Source Address Registers 0 to 11 (SAR0 to SAR11) 567
14.3.2 DMA Source Address Registers B0 to B3, B6 to B9 (SARB0 to SARB3, SARB6 to SARB9) 568
14.3.3 DMA Destination Address Registers 0 to 11 (DAR0 to DAR11) 568
14.3.4 DMA Destination Address Registers B0 to B3, B6 to B9 (DARB0 to DARB3, DARB6 to DARB9) 569
14.3.5 DMA Transfer Count Registers 0 to 11 (TCR0 to TCR11) 570
14.3.6 DMA Transfer Count Registers B0 to B3, B6 to B9 (TCRB0 to TCRB3, TCRB6 to TCRB9) 571
14.3.7 DMA Channel Control Registers 0 to 11 (CHCR0 to CHCR11) 572
14.3.8 DMA Operation Register 0, 1 (DMAOR0 and DMAOR1) 581
14.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2) 584
14.4 Operation 588
14.4.1 DMA Transfer Requests 588
14.4.2 Channel Priority 592
14.4.3 DMA Transfer Types 595
14.4.4 DMA Transfer Flow 602
14.4.5 Repeat Mode Transfer 604
14.4.6 Reload Mode Transfer 605
14.4.7 DREQ Pin Sampling Timing 606
14.5 Usage Notes 608
14.5.1 Module Stop 608
14.5.2 Address Error 608
14.5.3 Notes on Burst Mode Transfer 608
14.5.4 DACK output division 609
14.5.5 Clear DMINT Interrupt 609
14.5.6 $\overline{CS}$ Output Settings and Transfer Size Larger than External Bus Width 609
14.5.7 DACK Assertion and DREQ Sampling 609

Section 15 Clock Pulse Generator (CPG) 613
15.1 Features 613
15.2 Input/Output Pins 616
15.3 Clock Operating Modes 617
15.4 Register Descriptions 618
15.4.1 Frequency Control Register (FRQCR) 619
15.4.2 PLL Control Register (PLLCR) 621
15.5 Notes on Board Design 622

Section 16 Watchdog Timer and Reset 625
16.1 Features 625
16.2 Input/Output Pins 627
16.3 Register Descriptions 628
16.3.1 Watchdog Timer Stop Time Register (WDTST) 629
16.3.2 Watchdog Timer Control/Status Register (WDTCSR) 630
16.3.3 Watchdog timer Base Stop Time Register (WDTBST) 631
16.3.4 Watchdog Timer Counter (WDTCNT) 632
16.3.5 Watchdog Timer Base Counter (WDTBCNT) 632
16.4 Operation 633
16.4.1 Reset request 633
16.4.2 Using watchdog timer mode 634
16.4.3 Using Interval timer mode 634
16.4.4 Time for WDT Overflow 635
16.4.5 Clearing WDT Counter 636
16.5 Status Pin Change Timing during Reset 636
16.5.1 Power-On Reset by PRESET 636
16.5.2 Power-On Reset by Watchdog Timer Overflow 638
16.5.3 Manual Reset by Watchdog Timer Overflow 640

Section 17 Power-Down Mode 643
17.1 Features 643
17.1.1 Types of Power-Down Modes 643
17.2 Input/Output Pins 645
17.3 Register Descriptions 645
17.3.1 Standby Control Register (MSTPCR) 646
17.4 Sleep Mode 648
17.4.1 Transition to Sleep mode 648
17.4.2 Cancellation of Sleep Mode 648

17.5 Module Standby State 649
17.5.1 Transition to Module Standby Mode 649
17.5.2 Cancellation of Module Standby Mode and Resume 649
17.6 DDR-SDRAM Power Supply Backup 650
17.6.1 Self-Refresh and Initialization 650
17.6.2 DDR-SDRAM Backup Sequence when Turning Off System Power Supply 651
17.7 RTC Power Supply Backup 653
17.7.1 Transition to RTC Power Supply Backup 653
17.7.2 Cancellation of RTC Power Supply Backup 653
17.8 Mode Transitions 655
17.9 STATUS Pin Change Timing 656
17.9.1 In Reset 656
17.9.2 In Sleep 656

Section 18 Timer Unit (TMU) 657
18.1 Features 657
18.2 Input/Output Pins 659
18.3 Register Descriptions 660
18.3.1 Timer Output Control Register (TOCR) 662
18.3.2 Timer Start Register (TSTR0, TSTR1) 663
18.3.3 Timer Constant Register (TCORn) (n = 0 to 5) 665
18.3.4 Timer Counter (TCNTn) (n = 0 to 5) 665
18.3.5 Timer Control Registers (TCRn) (n = 0 to 5) 666
18.3.6 Input Capture Register 2 (TCPR2) 668
18.4 Operation 669
18.4.1 Counter Operation 669
18.4.2 Input Capture Function 673
18.5 Interrupts 674
18.6 Usage Notes 675
18.6.1 Register Writes 675
18.6.2 Reading from TCNT 675
18.6.3 Reset RTC Frequency Divider Circuit 675
18.6.4 External Clock Frequency 675

Section 19 Compare Match Timer (CMT) 677
19.1 Features 677
19.2 Input/Output Pins 679
19.3 Register Descriptions 679
19.3.1 Configuration Register (CMTCFG) 681
19.3.2 Free-Running Timer (CMTFRT) 684

19.3.3 Control Register (CMTCTL) 684
19.3.4 Interrupt Status Register (CMTIRQS) 688
19.3.5 Channels 0 to 3 Time Registers (CMTCH0T to CMTCH3T) 689
19.3.6 Channels 0 to 1 Stop Time Registers (CMTCH0ST to CMTCH1ST) 689
19.3.7 Channels 0 to 3 Counters (CMTCH0C to CMTCH3C) 690
19.4 Operation 691
19.4.1 Edge Detection 691
19.4.2 32-Bit Timer: Input Capture 692
19.4.3 32-Bit Timer: Output Compare 693
19.4.4 16-Bit Timer: Input Capture 697
19.4.5 16-Bit Timer: Output Compare 699
19.4.6 Counter: Up-counter 701
19.4.7 Counter: Updown-counter 703
19.4.8 Counter: Rotary Switch Operation of Updown-counter 705
19.4.9 Interrupts 706

Section 20 Realtime Clock (RTC) 707
20.1 Features 707
20.1.1 Block Diagram 708
20.2 Input/Output Pins 709
20.3 Register Descriptions 710
20.3.1 64 Hz Counter (R64CNT) 712
20.3.2 Second Counter (RSECCNT) 712
20.3.3 Minute Counter (RMINCNT) 713
20.3.4 Hour Counter (RHRCNT) 713
20.3.5 Day-of-Week Counter (RWKCNT) 714
20.3.6 Day Counter (RDAYCNT) 715
20.3.7 Month Counter (RMONCNT) 716
20.3.8 Year Counter (RYRCNT) 716
20.3.9 Second Alarm Register (RSECAR) 717
20.3.10 Minute Alarm Register (RMINAR) 717
20.3.11 Hour Alarm Register (RHRAR) 718
20.3.12 Day-of-Week Alarm Register (RWKAR) 718
20.3.13 Day Alarm Register (RDAYAR) 719
20.3.14 Month Alarm Register (RMONAR) 720
20.3.15 Year-Alarm Register (RYRAR) 720
20.3.16 RTC Control Register 1 (RCR1) 721
20.3.17 RTC Control Register 2 (RCR2) 723
20.3.18 RTC Control Register (RCR3) 726

20.4 Operation 727
20.4.1 Time Setting Procedures 727
20.4.2 Time Reading Procedures 728
20.4.3 Alarm Function 729
20.5 Interrupts 730
20.6 Usage Notes 730
20.6.1 Register Initialization 730
20.6.2 Crystal Oscillator Circuit 730
20.6.3 Interrupt source and request generating order 732

Section 21 Serial Communication Interface with FIFO (SCIF) 733
21.1 Features 733
21.2 Input/Output Pins 739
21.3 Register Descriptions 740
21.3.1 Receive Shift Register (SCRSR) 742
21.3.2 Receive FIFO Data Register (SCFRDR) 742
21.3.3 Transmit Shift Register (SCTSR) 743
21.3.4 Transmit FIFO Data Register (SCFTDR) 743
21.3.5 Serial Mode Register (SCSMR) 744
21.3.6 Serial Control Register (SCSCR) 747
21.3.7 Serial Status Register n (SCFSR) 751
21.3.8 Bit Rate Register n (SCBRR) 758
21.3.9 FIFO Control Register n (SCFCR) 759
21.3.10 Transmit FIFO Data Count Register n (SCTFDR) 762
21.3.11 Receive FIFO Data Count Register n (SCRFDR) 763
21.3.12 Serial Port Register n (SCSPTR) 764
21.3.13 Line Status Register n (SCLSR) 767
21.3.14 Serial Error Register n (SCRER) 768
21.4 Operation 769
21.4.1 Overview 769
21.4.2 Operation in Asynchronous Mode 772
21.4.3 Operation in Clocked Synchronous Mode 783
21.5 SCIF Interrupt Sources and the DMAC 792
21.6 Usage Notes 794

Section 22 Serial I/O with FIFO (SIOF) 797
22.1 Features 797
22.2 Input/Output Pins 799
22.3 Register Descriptions 800
22.3.1 Mode Register (SIMDR) 802

22.3.2 Clock Select Register (SISCR) 804
22.3.3 Control Register (SICTR) 806
22.3.4 Transmit Data Register (SITDR) 809
22.3.5 Receive Data Register (SIRDR) 810
22.3.6 Transmit Control Data Register (SITCR) 811
22.3.7 Receive Control Data Register (SIRCR) 812
22.3.8 Status Register (SISTR) 813
22.3.9 Interrupt Enable Register (SIIER) 819
22.3.10 FIFO Control Register (SIFCTR) 821
22.3.11 Transmit Data Assign Register (SITDAR) 823
22.3.12 Receive Data Assign Register (SIRDAR) 824
22.3.13 Control Data Assign Register (SICDAR) 825
22.4 Operation 827
22.4.1 Serial Clocks 827
22.4.2 Serial Timing 829
22.4.3 Transfer Data Format 830
22.4.4 Register Allocation of Transfer Data 832
22.4.5 Control Data Interface 834
22.4.6 FIFO 836
22.4.7 Transmit and Receive Procedures 838
22.4.8 Interrupts 843
22.4.9 Transmit and Receive Timing 845

Section 23 Serial Protocol Interface (HSPI) 849
23.1 Features 849
23.2 Input/Output Pins 851
23.3 Register Descriptions 851
23.3.1 Control Register (SPCR) 852
23.3.2 Status Register (SPSR) 854
23.3.3 System Control Register (SPSCR) 857
23.3.4 Transmit Buffer Register (SPTBR) 859
23.3.5 Receive Buffer Register (SPRBR) 860
23.4 Operation 861
23.4.1 Operation Overview without DMA (FIFO Mode Disabled) 861
23.4.2 Operation Overview with DMA 862
23.4.3 Operation with FIFO Mode Enabled 862
23.4.4 Timing Diagrams 863
23.4.5 HSPI Software Reset 864
23.4.6 Clock Polarity and Transmit Control 864
23.4.7 Transmit and Receive Routines 864

Section 24 Multimedia Card Interface (MMCIF) 865
24.1 Features 865
24.2 Input/Output Pins 866
24.3 Register Descriptions 867
24.3.1 Command Registers 0 to 5 (CMDR0 to CMDR5) 871
24.3.2 Command Start Register (CMDSTRT) 872
24.3.3 Operation Control Register (OPCR) 873
24.3.4 Card Status Register (CSTR) 875
24.3.5 Interrupt Control Registers 0 to 2 (INTCR0 to INTCR2) 877
24.3.6 Interrupt Status Registers 0 to 2 (INTSTR0 to INTSTR2) 880
24.3.7 Transfer Clock Control Register (CLKON) 885
24.3.8 Command Timeout Control Register (CTOCR) 886
24.3.9 Transfer Byte Number Count Register (TBCR) 887
24.3.10 Mode Register (MODER) 888
24.3.11 Command Type Register (CMDTYR) 889
24.3.12 Response Type Register (RSPTYR) 890
24.3.13 Transfer Block Number Counter (TBNCR) 894
24.3.14 Response Registers 0 to 16, D (RSPR0 to RSPR16, RSPRD) 895
24.3.15 Data Timeout Register (DTOUTR) 897
24.3.16 Data Register (DR) 898
24.3.17 FIFO Pointer Clear Register (FIFOCLR) 899
24.3.18 DMA Control Register (DMACR) 900
24.4 Operation 901
24.4.1 Operations in MMC Mode 901
24.5 MMCIF Interrupt Sources 931
24.6 Operations when Using DMA 932
24.6.1 Operation in Read Sequence 932
24.6.2 Operation in Write Sequence 942
24.7 Register Accesses with Little Endian Specification 953

Section 25 Audio Codec Interface (HAC) 955
25.1 Features 955
25.2 Input/Output Pins 956
25.3 Register Descriptions 957
25.3.1 Control and Status Register (HACCR) 958
25.3.2 Command/Status Address Register (HACCSAR) 960
25.3.3 Command/Status Data Register (HACCSDR) 962
25.3.4 PCM Left Channel Register (HACPCML) 963
25.3.5 PCM Right Channel Register (HACPCMR) 965
25.3.6 TX Interrupt Enable Register (HACTIER) 966

25.3.7 TX Status Register (HACTSR) 967
25.3.8 RX Interrupt Enable Register (HACRIER) 969
25.3.9 RX Status Register (HACRSR) 970
25.3.10 HAC Control Register (HACACR) 971
25.4 AC 97 Frame Slot Structure 973
25.5 Operation 974
25.5.1 Receiver 974
25.5.2 Transmitter 975
25.5.3 DMA 975
25.5.4 Interrupts 975
25.5.5 Initialization Sequence 976
25.5.6 Notes 981
25.5.7 Reference 981

Section 26 Serial Sound Interface (SSI) Module 983
26.1 Features 983
26.2 Input/Output Pins 984
26.3 Register Descriptions 985
26.3.1 Control Register (SSICR) 986
26.3.2 Status Register (SSISR) 992
26.3.3 Transmit Data Register (SSITDR) 997
26.3.4 Receive Data Register (SSIRDR) 997
26.4 Operation 998
26.4.1 Bus Format 998
26.4.2 Non-Compressed Modes 999
26.4.3 Compressed Modes 1008
26.4.4 Operation Modes 1011
26.4.5 Transmit Operation 1012
26.4.6 Receive Operation 1015
26.4.7 Serial Clock Control 1018
26.5 Usage Note 1019
26.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation 1019

Section 27 NAND Flash Memory Controller (FLCTL) 1021
27.1 Features 1021
27.2 Input/Output Pins 1024
27.3 Register Descriptions 1025
27.3.1 Common Control Register (FLCMNCR) 1026
27.3.2 Command Control Register (FLCMDCR) 1028
27.3.3 Command Code Register (FLCMCDR) 1030

27.3.4 Address Register (FLADR) 1030
27.3.5 Data Counter Register (FLDTCNTR) 1032
27.3.6 Data Register (FLDATAR) 1033
27.3.7 Interrupt DMA Control Register (FLINTDMACR) 1034
27.3.8 Ready Busy Timeout Setting Register (FLBSYTMR) 1039
27.3.9 Ready Busy Timeout Counter (FLBSYCNT) 1040
27.3.10 Data FIFO Register (FLDTFIFO) 1041
27.3.11 Control Code FIFO Register (FLECFIFO) 1042
27.3.12 Transfer Control Register (FLTRCR) 1043
27.4 Operation 1044
27.4.1 Operating Modes 1044
27.4.2 Command Access Mode 1044
27.4.3 Sector Access Mode 1046
27.4.4 ECC Error Correction 1048
27.4.5 Status Read 1049
27.5 Example of Register Setting 1050
27.6 Interrupt Sources 1053
27.7 DMA Transfer Specifications 1053

Section 28 General Purpose I/O (GPIO) 1055
28.1 Features 1055
28.2 Register Descriptions 1060
28.2.1 Port A Control Register (PACR) 1063
28.2.2 Port B Control Register (PBCR) 1064
28.2.3 Port C Control Register (PCCR) 1066
28.2.4 Port D Control Register (PDCR) 1067
28.2.5 Port E Control Register (PECR) 1069
28.2.6 Port F Control Register (PFCR) 1070
28.2.7 Port G Control Register (PGCR) 1072
28.2.8 Port H Control Register (PHCR) 1074
28.2.9 Port J Control Register (PJCR) 1075
28.2.10 Port K Control Register (PKCR) 1077
28.2.11 Port L Control Register (PLCR) 1079
28.2.12 Port M Control Register (PMCR) 1080
28.2.13 Port A Data Register (PADR) 1081
28.2.14 Port B Data Register (PBDR) 1081
28.2.15 Port C Data Register (PCDR) 1082
28.2.16 Port D Data Register (PDDR) 1082
28.2.17 Port E Data Register (PEDR) 1083
28.2.18 Port F Data Register (PFDR) 1084

28.2.19 Port G Data Register (PGDR) 1084
28.2.20 Port H Data Register (PHDR) 1085
28.2.21 Port J Data Register (PJDR) 1085
28.2.22 Port K Data Register (PKDR) 1086
28.2.23 Port L Data Register (PLDR) 1086
28.2.24 Port M Data Register (PMDR) 1087
28.2.25 Port E Pull-Up Control Register (PEPUPR) 1087
28.2.26 Port H Pull-Up Control Register (PHPUPR) 1088
28.2.27 Port J Pull-Up Control Register (PJPUPR) 1089
28.2.28 Port K Pull-Up Control Register (PKPUPR) 1090
28.2.29 Port M Pull-Up Control Register (PMPUPR) 1091
28.2.30 Input-Pin Pull-Up Control Register 1 (PPUPR1) 1092
28.2.31 Input-Pin Pull-Up Control Register 2 (PPUPR2) 1093
28.2.32 On-chip Module Select Register (OMSELR) 1094
28.3 Usage Example 1097
28.3.1 Port Output Function 1097
28.3.2 Port Input function 1098
28.3.3 On-chip Module Function 1099

Section 29 User Break Controller (UBC) 1101
29.1 Features 1101
29.2 Register Descriptions 1103
29.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1) 1105
29.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1) 1111
29.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1) 1113
29.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1) 1114
29.2.5 Match Data Setting Register 1 (CDR1) 1115
29.2.6 Match Data Mask Setting Register 1 (CDMR1) 1116
29.2.7 Execution Count Break Register 1 (CETR1) 1117
29.2.8 Channel Match Flag Register (CCMFR) 1118
29.2.9 Break Control Register (CBCR) 1119
29.3 Operation Description 1120
29.3.1 Definition of Words Related to Accesses 1120
29.3.2 User Break Operation Sequence 1121
29.3.3 Instruction Fetch Cycle Break 1122
29.3.4 Operand Access Cycle Break 1123
29.3.5 Sequential Break 1124
29.3.6 Program Counter Value to be Saved 1126
29.4 User Break Debugging Support Function 1127
29.5 User Break Examples 1128

29.6 Usage Notes 1132

Section 30 User Debugging Interface (H-UDI) 1135
30.1 Features 1135
30.2 Input/Output Pins 1137
30.3 Boundary Scan TAP Controllers
(IDCODE, EXTEST, SAMPLE/PRELOAD, and BYPASS) 1138
30.4 Register Descriptions 1140
30.4.1 Instruction Register (SDIR) 1141
30.4.2 Interrupt Source Register (SDINT) 1142
30.4.3 Bypass Register (SDBPR) 1142
30.4.4 Boundary Scan Register (SDBSR) 1143
30.5 Operation 1152
30.5.1 TAP Control 1152
30.5.2 H-UDI Reset 1153
30.5.3 H-UDI Interrupt 1153
30.6 Usage Notes 1154

Section 31 Electrical Characteristics 1155
31.1 Absolute Maximum Ratings 1155
31.2 DC Characteristics 1156
31.3 AC Characteristics 1159
31.3.1 Clock and Control Signal Timing 1160
31.3.2 Control Signal Timing 1163
31.3.3 Bus Timing 1164
31.3.4 DDRIF Signal Timing 1182
31.3.5 INTC Module Signal Timing 1186
31.3.6 PCIC Module Signal Timing 1188
31.3.7 DMAC Module Signal Timing 1190
31.3.8 TMU Module Signal Timing 1191
31.3.9 CMT Module Signal Timing 1192
31.3.10 SCIF Module Signal Timing 1193
31.3.11 SIOF Module Signal Timing 1195
31.3.12 HSPI Module Signal Timing 1199
31.3.13 MMCIF Module Signal Timing 1201
31.3.14 HAC Interface Module Signal Timing 1203
31.3.15 SSI Interface Module Signal Timing 1205
31.3.16 FLCTL Module Signal Timing 1207
31.3.17 GPIO Signal Timing 1211
31.3.18 H-UDI Module Signal Timing 1212

31.4 AC Characteristic Test Conditions 1214
31.5 Change in Delay Time Based on Load Capacitance 1215

Appendix 1217
A. CPU Operation Mode Register (CPUOPM) 1217
B. Instruction Prefetching and Its Side Effects 1219
C. Speculative Execution for Subroutine Return 1220
D. Register Address Map 1221
E. Package Dimensions 1255
F. Mode Pin Settings 1256
G. Pin Functions 1258
G.1 Pin States 1258
G.2 Handling of Unused Pins 1267
H. Turning On and Off Power Supply 1275
I. Version Registers (PVR, PRR) 1276
J. Part Number List 1277

Index 1279

Figures

Section 1 Overview
Figure 1.1 SH7780 Block Diagram 9
Figure 1.2 SH7780 Pin Arrangement 10
Figure 1.3 Physical Address Space of SH7780 28
Figure 1.4 Relationship between AREASEL Bits and Memory Address Map 29

Section 2 Programming Model
Figure 2.1 Data Formats 33
Figure 2.2 CPU Register Configuration in Each Processing Mode 36
Figure 2.3 General Registers 37
Figure 2.4 Floating-Point Registers 39
Figure 2.5 Relationship between SZ bit and Endian 45
Figure 2.6 Formats of Byte Data and Word Data in Register 47
Figure 2.7 Data Formats in Memory 48
Figure 2.8 Processing State Transitions 49

Section 4 Pipelining
Figure 4.1 Basic Pipelines 73
Figure 4.2 Instruction Execution Patterns (1) 75
Figure 4.2 Instruction Execution Patterns (2) 76
Figure 4.2 Instruction Execution Patterns (3) 77
Figure 4.2 Instruction Execution Patterns (4) 78
Figure 4.2 Instruction Execution Patterns (5) 79
Figure 4.2 Instruction Execution Patterns (6) 80
Figure 4.2 Instruction Execution Patterns (7) 81
Figure 4.2 Instruction Execution Patterns (8) 82
Figure 4.2 Instruction Execution Patterns (9) 83

Section 5 Exception Handling
Figure 5.1 Instruction Execution and Exception Handling 105
Figure 5.2 Example of General Exception Acceptance Order 106

Section 6 Floating-Point Unit (FPU)
Figure 6.1 Format of Single-Precision Floating-Point Number 130
Figure 6.2 Format of Double-Precision Floating-Point Number 130
Figure 6.3 Single-Precision NaN Bit Pattern 133
Figure 6.4 Floating-Point Registers 136
Figure 6.5 Relation between SZ Bit and Endian 139

Section 7 Memory Management Unit (MMU)

Figure 7.1 Role of MMU 149
Figure 7.2 Virtual Address Space (AT in MMUCR = 0) 150
Figure 7.3 Virtual Address Space (AT in MMUCR = 1) 151
Figure 7.4 P4 Area 153
Figure 7.5 Physical Address Space 154
Figure 7.6 UTLB Configuration 167
Figure 7.7 Relationship between Page Size and Address Format 169
Figure 7.8 ITLB Configuration 170
Figure 7.9 Flowchart of Memory Access Using UTLB 171
Figure 7.10 Flowchart of Memory Access Using ITLB 172
Figure 7.11 Operation of LDTLB Instruction 175
Figure 7.12 Memory-Mapped ITLB Address Array 184
Figure 7.13 Memory-Mapped ITLB Data Array 185
Figure 7.14 Memory-Mapped UTLB Address Array 187
Figure 7.15 Memory-Mapped UTLB Data Array 188
Figure 7.16 Physical Address Space (32-Bit Address Extended Mode) 188
Figure 7.17 PMB Configuration 190
Figure 7.18 Memory-Mapped PMB Address Array 193
Figure 7.19 Memory-Mapped PMB Data Array 193

Section 8 Caches

Figure 8.1 Configuration of Operand Cache (OC) 198
Figure 8.2 Configuration of Instruction Cache (IC) 199
Figure 8.3 Configuration of Write-Back Buffer 211
Figure 8.4 Configuration of Write-Through Buffer 211
Figure 8.5 Memory-Mapped IC Address Array 218
Figure 8.6 Memory-Mapped IC Data Array 219
Figure 8.7 Memory-Mapped OC Address Array 221
Figure 8.8 Memory-Mapped OC Data Array 222
Figure 8.9 Store Queue Configuration 223

Section 10 Interrupt Controller (INTC)

Figure 10.1 Block Diagram of INTC 244
Figure 10.2 Example of IRL Interrupt Connection 297
Figure 10.3 On-chip Module Interrupt Priority 301
Figure 10.4 Interrupt Operation Flowchart 309
Figure 10.5 Example of Interrupt Handling Routine 312

Section 11 Local Bus State Controller (LBSC)

Figure 11.1 LBSC Block Diagram 317

Figure 11.2 Correspondence between Virtual Address Space and External Memory Space of LBSC 321
Figure 11.3 External Memory Space Allocation (29-bit address mode) 323
Figure 11.4 Basic Timing of SRAM Interface 362
Figure 11.5 Example of 32-Bit Data-Width SRAM Connection 363
Figure 11.6 Example of 16-Bit Data-Width SRAM Connection 364
Figure 11.7 Example of 8-Bit Data-Width SRAM Connection 365
Figure 11.8 SRAM Interface Wait Timing (Software Wait Only) 366
Figure 11.9 SRAM Interface Wait Timing (Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal, $\overline{\text{RDY}}$ Signal is synchronous input) 367
Figure 11.10 SRAM Interface Wait Timing (Read-Strobe Negate Timing Setting) 369
Figure 11.11 Burst ROM Basic Timing 371
Figure 11.12 Burst ROM Wait Timing 371
Figure 11.13 Burst ROM Wait Timing 372
Figure 11.14 $\overline{\text{CExx}}$ and $\overline{\text{DACK}}$ Output of ATA Complement Mode in DMA Transfer 374
Figure 11.15 Example of PCMCIA Interface 377
Figure 11.16 Basic Timing for PCMCIA Memory Card Interface 378
Figure 11.17 Wait Timing for PCMCIA Memory Card Interface 379
Figure 11.18 Basic Timing for PCMCIA I/O Card Interface 380
Figure 11.19 Wait Timing for PCMCIA I/O Card Interface 381
Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface 382
Figure 11.21 Example of 32-Bit Data Width MPX Connection 384
Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External Wait) 384
Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait Inserted) 385
Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No External Wait) 385
Figure 11.25 MPX Interface Timing 4 (Single Write Cycle, IW = 1, One External Wait Inserted) 386
Figure 11.26 MPX Interface Timing 5 (Burst Read Cycle, IW = 0, No External Wait, 32-Byte Data Transfer) 386
Figure 11.27 MPX Interface Timing 6 (Burst Read Cycle, IW = 0, External Wait Control, 32-Byte Data Transfer) 387
Figure 11.28 MPX Interface Timing 7 (Burst Write Cycle, IW = 0, No External Wait, 32-Byte Data Transfer) 387
Figure 11.29 MPX Interface Timing 8 (Burst Write Cycle, IW = 1, External Wait Control, 32-Byte Data Transfer) 388
Figure 11.30 Example of 32-Bit Data-Width Byte-Control SRAM 389
Figure 11.31 Byte-Control SRAM Basic Read Cycle (No Wait) 390
Figure 11.32 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cycle) 391
Figure 11.33 Byte-Control SRAM Basic Read Cycle (One Internal Wait + One External Wait) 392

Figure 11.34 Wait Cycles between Access Cycles 394
Figure 11.35 Arbitration Sequence 396
Figure 11.36 Example of the Bus Release Restraint by the DMAC CHCR LCKN bit 398

Section 12 DDR-SDRAM Interface (DDRIF)

Figure 12.1 DDRIF Block Diagram 402
Figure 12.2 Physical Address Space of This LSI 405
Figure 12.3 Data Alignment in DDR-SDRAM and DDRIF 409
Figure 12.4 Relationship between Write Values in SDMR and Output Signals to Memory Pins 423
Figure 12.5 DDRIF Basic Timing (1-/2-/4-/8-Byte Single Burst Read without Auto Precharge) 430
Figure 12.6 DDRIF Basic Timing (1-/2-/4-/8-Byte Single Burst Write without Auto Precharge) 431
Figure 12.7 DDRIF Basic Timing (1-/2-/4-/8-Byte Single Burst Read with Auto Precharge) 432
Figure 12.8 DDRIF Basic Timing (1-/2-/4-/8-Byte Single Burst Write with Auto Precharge) 433
Figure 12.9 DDRIF Basic Timing (4 Burst Read: 32-byte without Auto Precharge) 434
Figure 12.10 DDRIF Basic Timing (4 Burst Write: 32-byte without Auto Precharge) 435
Figure 12.11 DDRIF Basic Timing (from Precharging All Banks to Bank Activation) 436
Figure 12.12 DDRIF Basic Timing (Mode Register Setting) 437
Figure 12.13 DDRIF Basic Timing (Enter Auto-Refresh/Exit to Bank Activation) 438
Figure 12.14 DDRIF Basic Timing (Enter Self-Refresh/Exit to Command Issuing) 439

Section 13 PCI Controller (PCIC)

Figure 13.1 PCIC Block Diagram 445
Figure 13.2 SuperHyway Bus to PCI Local Bus Access 525
Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 0) 526
Figure 13.4 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 1) 527
Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 2) 527
Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/O) 528
Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus (Non-Byte Swapping: TBS = 0) 530
Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus (Byte Swapping: TBS = 1) 531
Figure 13.9 PCI local bus to SuperHyway bus Memory Map 532
Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation (Local Address Space 0/1) 534

Figure 13.11 PCI Local Bus to SuperHyway Bus Address Translation (PCIC I/O Space) 535
Figure 13.12 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 0) 537
Figure 13.13 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 1) 538
Figure 13.14 Cache Flush/Purge Execution Flow for PCI local Bus to SuperHyway Bus 540
Figure 13.15 Address Generation for Type 0 Configuration Access 542
Figure 13.16 PCI Local Bus Power Down State Transition 545
Figure 13.17 Master Write Cycle in Host Bus Bridge Mode (Single) 546
Figure 13.18 Master Read Cycle in Host Bus Bridge Mode (Single) 547
Figure 13.19 Master Write Cycle in Normal Mode (Burst) 548
Figure 13.20 Master Read Cycle in Normal Mode (Burst) 549
Figure 13.21 Target Read Cycle in Normal Mode (Single) 551
Figure 13.22 Target Write Cycle in Normal Mode (Single) 552
Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burst) 553
Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst) 554
Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with stepping) 555
Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with stepping) 556

Section 14 Direct Memory Access Controller (DMAC)

Figure 14.1 Block Diagram of DMAC 558
Figure 14.2 Round-Robin Mode (example of channel 0 to 5) 593
Figure 14.3 Changes in Channel Priority in Round-Robin Mode (example of channel 0 to 5) 594
Figure 14.4 Data Flow of Dual Address Mode 595
Figure 14.5 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory) 596
Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection) 597
Figure 14.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection) 598
Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection) 598
Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection) 599
Figure 14.10 Bus State when Multiple Channels are Operating 602
Figure 14.11 DMA Transfer Flowchart 603
Figure 14.12 Reload Mode Transfer 605
Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection 606
Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection 606
Figure 14.15 Example of DREQ Input Detection in Burst Mode Edge Detection 607
Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detection 607

Section 15 Clock Pulse Generator (CPG)
Figure 15.1 Block Diagram of CPG 614
Figure 15.2 Points for Attention when Using Crystal Resonator 622
Figure 15.3 Points for Attention when Using PLL and DLL Circuit 623

Section 16 Watchdog Timer and Reset
Figure 16.1 Block Diagram of WDT 626
Figure 16.2 WDT Counting Up Operation 635
Figure 16.3 STATUS Output during Power-on 637
Figure 16.4 STATUS Output by Reset input during Normal Operation 637
Figure 16.5 STATUS Output by Reset input during Sleep Mode 638
Figure 16.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation 639
Figure 16.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode 639
Figure 16.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation 640
Figure 16.9 STATUS Output by Watchdog timer overflow Manual Reset during Sleep Mode 641

Section 17 Power-Down Mode
Figure 17.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off 650
Figure 17.2 Sequence for Turning Off System Power Supply in Self-Refresh Mode 652
Figure 17.3 Sequence for Turning System Power Supply On/Off 654
Figure 17.4 Mode Transition Diagram 655
Figure 17.5 Status Pins Output from Sleep to Interrupt 656

Section 18 Timer Unit (TMU)
Figure 18.1 Block Diagram of TMU 658
Figure 18.2 Example of Count Operation Setting Procedure 670
Figure 18.3 TCNT Auto-Reload Operation 671
Figure 18.4 Count Timing when Operating on Internal Clock 671
Figure 18.5 Count Timing when Operating on External Clock 672
Figure 18.6 Count Timing when Operating on on-chip RTC output Clock 672
Figure 18.7 Operation Timing when Using Input Capture Function 673

Section 19 Timer/Counter (CMT)
Figure 19.1 Block Diagram of CMT 678
Figure 19.2 Edge Detection (example of rising edge) 691
Figure 19.3 32-Bit Timer Mode: Input Capture (channel 1 and channel 0) 692
Figure 19.4 32-bit Timer mode: Input Capture Operation Timing 692

Figure 19.5 CMT_CTRn Assert Timing (channel 0 and 1) 694
Figure 19.6 32-Bit Timer Mode: Output Compare (channel 1 and channel 0) 694
Figure 19.7 32-bit Timer Mode: Output Compare Operation Timing
(Example of High output in Active and Not Active by CMTCHnST) 695
Figure 19.8 32-bit Timer Mode: Output Compare Operation Timing
(Example of High output in Active and Not Active by CMTFRT) 695
Figure 19.9 16-Bit Timer Mode: Input Capture (channel 1 and channel 0) 697
Figure 19.10 16-Bit Timer Mode: Input Capture Operation Timing 698
Figure 19.11 16-Bit Timer Mode: Output Compare
(CMT_CTR pins are available for channel 1 and channel 0) 699
Figure 19.12 16-Bit Timer Mode: Output Compare Operation Timing 700
Figure 19.13 Up-Counter Mode (channel 1 and channel 0) 701
Figure 19.14 Up-counter Mode Operation Timing 701
Figure 19.15 Updown-Counter Mode (only channel 0) 703
Figure 19.16 Updown-Counter Mode: Countdown Operation Timing (only channel 0) 703
Figure 19.17 Rotary Switch Operation Count-Up Timing 705
Figure 19.18 Rotary Switch Operation Count-Down Timing 705

Section 20 Realtime Clock (RTC)

Figure 20.1 Block Diagram of RTC 708
Figure 20.2 Examples of Time Setting Procedures 727
Figure 20.3 Examples of Time Reading Procedures 728
Figure 20.4 Example of Use of Alarm Function 729
Figure 20.5 Example of Crystal Oscillator Circuit Connection 731
Figure 20.6 Interrupt Request Signal Generation Timing of Complex Sources 732

Section 21 Serial Communication Interface with FIFO (SCIF)

Figure 21.1 Block Diagram of SCIF 735
Figure 21.2 $\overline{\text{SCIF0\_RTS}}$ Pin (Only in Channel 0) 736
Figure 21.3 $\overline{\text{SCIF0\_CTS}}$ Pin (Only in Channel 0) 736
Figure 21.4 SCIFn_SCK Pin (n = 0, 1) 737
Figure 21.5 SCIFn_TXD Pin (n = 0, 1) 737
Figure 21.6 SCIFn_RXD Pin (n = 0, 1) 738
Figure 21.7 Data Format in Asynchronous Communication
(Example with 8-Bit Data, Parity, and Two Stop Bits) 772
Figure 21.8 Sample SCIF Initialization Flowchart 775
Figure 21.9 Sample Serial Transmission Flowchart 776
Figure 21.10 Sample SCIF Transmission Operation
(Example with 8-Bit Data, Parity, One Stop Bit) 778
Figure 21.11 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_CTS}}$) (Only in Channel 0) 778
Figure 21.12 Sample Serial Reception Flowchart (1) 779

Figure 21.12 Sample Serial Reception Flowchart (2) 780
Figure 21.13 Sample SCIF Receive Operation
(Example with 8-Bit Data, Parity, One Stop Bit) 782
Figure 21.14 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_RTS}}$) (Only in Channel 0) 782
Figure 21.15 Data Format in Clocked Synchronous Communication 783
Figure 21.16 Sample SCIF Initialization Flowchart 785
Figure 21.17 Sample Serial Transmission Flowchart 786
Figure 21.18 Sample SCIF Transmission Operation in Clocked Synchronous Mode 787
Figure 21.19 Sample Serial Reception Flowchart (1) 788
Figure 21.19 Sample Serial Reception Flowchart (2) 789
Figure 21.20 Sample SCIF Reception Operation in Clocked Synchronous Mode 790
Figure 21.21 Sample Simultaneous Serial Transmission and Reception Flowchart 791
Figure 21.22 Receive Data Sampling Timing in Asynchronous Mode 795
Figure 21.23 Example of Synchronization Clock Transfer by DMAC 796

Section 22 Serial I/O with FIFO (SIOF)

Figure 22.1 Block Diagram of SIOF 798
Figure 22.2 Serial Clock Supply 827
Figure 22.3 Serial Data Synchronization Timing 829
Figure 22.4 SIOF Transmit/Receive Timing 830
Figure 22.5 Transmit/Receive Data Bit Alignment 832
Figure 22.6 Control Data Bit Alignment 833
Figure 22.7 Control Data Interface (Slot Position) 834
Figure 22.8 Control Data Interface (Secondary FS) 835
Figure 22.9 Example of Transmit Operation in Master Mode 838
Figure 22.10 Example of Receive Operation in Master Mode 839
Figure 22.11 Example of Transmit Operation in Slave Mode 840
Figure 22.12 Example of Receive Operation in Slave Mode 841
Figure 22.13 Transmit and Receive Timing (8-Bit Monaural Data (1)) 845
Figure 22.14 Transmit and Receive Timing (8-Bit Monaural Data (2)) 845
Figure 22.15 Transmit and Receive Timing (16-Bit Monaural Data) 846
Figure 22.16 Transmit and Receive Timing (16-Bit Stereo Data (1)) 846
Figure 22.17 Transmit and Receive Timing (16-Bit Stereo Data (2)) 847
Figure 22.18 Transmit and Receive Timing (16-Bit Stereo Data (3)) 847
Figure 22.19 Transmit and Receive Timing (16-Bit Stereo Data (4)) 848
Figure 22.20 Transmit and Receive Timing (16-Bit Stereo Data) 848

Section 23 Serial Protocol Interface (HSPI)

Figure 23.1 Block Diagram of HSPI 850
Figure 23.2 Operational Flowchart 861
Figure 23.3 Timing Conditions when FBS = 0 863

Figure 23.4 Timing Conditions when FBS = 1 864

Section 24 Multimedia Card Interface (MMCIF)

Figure 24.1 Block Diagram of MMCIF 866
Figure 24.2 DR Access Example 899
Figure 24.3 Example of Command Sequence for Commands Not Requiring Command Response 903
Figure 24.4 Example of Operational Flow for Commands Not Requiring Command Response 904
Figure 24.5 Example of Command Sequence for Commands without Data Transfer (No Data Busy State) 905
Figure 24.6 Example of Command Sequence for Commands without Data Transfer (with Data Busy State) 906
Figure 24.7 Example of Operational Flow for Commands without Data Transfer 907
Figure 24.8 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size) 909
Figure 24.9 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size) 910
Figure 24.10 Example of Command Sequence for Commands with Read Data (Multiple Block Transfer) 911
Figure 24.11 Example of Command Sequence for Commands with Read Data (Stream Transfer) 912
Figure 24.12 Example of Operational Flow for Commands with Read Data (Single Block Transfer) 913
Figure 24.13 Example of Operational Flow for Commands with Read Data (1) (Open-ended Multiple Block Transfer) 914
Figure 24.13 Example of Operational Flow for Commands with Read Data (2) (Open-ended Multiple Block Transfer) 915
Figure 24.13 Example of Operational Flow for Commands with Read Data (3) (Pre-defined Multiple Block Transfer) 916
Figure 24.13 Example of Operational Flow for Commands with Read Data (4) (Pre-defined Multiple Block Transfer) 917
Figure 24.14 Example of Operational Flow for Commands with Read Data (Stream Transfer) 918
Figure 24.15 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size) 921
Figure 24.16 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size) 922
Figure 24.17 Example of Command Sequence for Commands with Write Data (Multiple Block Transfer) 923

Figure 24.18 Example of Command Sequence for Commands with Write Data (Stream Transfer) 924
Figure 24.19 Example of Operational Flow for Commands with Write Data (Single Block Transfer) 925
Figure 24.20 Example of Operational Flow for Commands with Write Data (1) (Open-ended Multiple Block Transfer) 926
Figure 24.20 Example of Operational Flow for Commands with Write Data (2) (Open-ended Multiple Block Transfer) 927
Figure 24.20 Example of Operational Flow for Commands with Write Data (3) (Pre-defined Multiple Block Transfer) 928
Figure 24.20 Example of Operational Flow for Commands with Write Data (4) (Pre-defined Multiple Block Transfer) 929
Figure 24.21 Example of Operational Flow for Commands with Write Data (Stream Transfer) 930
Figure 24.22 Example of Read Sequence Flow (Single Block Transfer) 934
Figure 24.23 Example of Read Sequence Flow (1) (Open-ended Multiple Block Transfer) 935
Figure 24.23 Example of Read Sequence Flow (2) (Open-ended Multiple Block Transfer) 936
Figure 24.23 Example of Read Sequence Flow (3) (Pre-defined Multiple Block Transfer) 937
Figure 24.23 Example of Read Sequence Flow (4) (Pre-defined Multiple Block Transfer) 938
Figure 24.24 Example of Operational Flow for Stream Read Transfer 939
Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (1) 940
Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (2) 941
Figure 24.26 Example of Write Sequence Flow (1) (Single Block Transfer) 944
Figure 24.26 Example of Write Sequence Flow (2) (Single Block Transfer) 945
Figure 24.27 Example of Write Sequence Flow (1) (Open-ended Multiple Block Transfer) 946
Figure 24.27 Example of Write Sequence Flow (2) (Open-ended Multiple Block Transfer) 947
Figure 24.27 Example of Write Sequence Flow (3) (Pre-defined Multiple Block Transfer) 948
Figure 24.27 Example of Write Sequence Flow (4) (Pre-defined Multiple Block Transfer) 949
Figure 24.28 Example of Operational Flow for Stream Write Transfer 950
Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (1) 951
Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (2) 952

Section 25 Audio Codec Interface (HAC)

Figure 25.1 Block Diagram 956
Figure 25.2 AC97 Frame Slot Structure 973
Figure 25.3 Initialization Sequence 976
Figure 25.4 Sample Flowchart for Off-Chip Codec Register Write 977

Figure 25.5 Sample Flowchart for Off-Chip Codec Register Read (1) 978
Figure 25.6 Sample Flowchart for Off-Chip Codec Register Read (2) 979
Figure 25.7 Sample Flowchart for Off-Chip Codec Register Read (3) 980

Section 26 Serial Sound Interface (SSI) Module

Figure 26.1 Block Diagram of SSI Module 984
Figure 26.2 Philips Format (with no Padding) 1000
Figure 26.3 Philips Format (with Padding) 1000
Figure 26.4 Sony Format (with Serial Data First, Followed by Padding Bits) 1001
Figure 26.5 Matsushita Format (with Padding Bits First, Followed by Serial Data) 1001
Figure 26.6 Multi-channel Format (4 Channels, No Padding) 1003
Figure 26.7 Multi-channel Format (6 Channels with High Padding) 1003
Figure 26.8 Multi-channel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding) 1004
Figure 26.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length) 1005
Figure 26.10 Inverted Clock 1005
Figure 26.11 Inverted Word Select 1006
Figure 26.12 Inverted Padding Polarity 1006
Figure 26.13 Padding Bits First, Followed by Serial Data, with Delay 1006
Figure 26.14 Padding Bits First, Followed by Serial Data, without Delay 1007
Figure 26.15 Serial Data First, Followed by Padding Bits, without Delay 1007
Figure 26.16 Parallel Right Aligned with Delay 1007
Figure 26.17 Mute Enabled 1008
Figure 26.18 Compressed Data Format, Slave Transmitter, Burst Mode Disabled 1009
Figure 26.19 Compressed Data Format, Slave Transmitter, and Burst Mode Enabled 1009
Figure 26.20 Transition Diagram between Operation Modes 1011
Figure 26.21 Transmission Using DMA Controller 1013
Figure 26.22 Transmission using Interrupt Data Flow Control 1014
Figure 26.23 Reception using DMA Controller 1016
Figure 26.24 Reception using Interrupt Data Flow Control 1017

Section 27 NAND Flash Memory Controller (FLCTL)

Figure 27.1 FLCTL Block Diagram 1023
Figure 27.2 Read Operation Timing for NAND-Type Flash Memory (1) 1044
Figure 27.3 Programming Operation Timing for NAND-Type Flash Memory (1) 1045
Figure 27.4 Programming Operation Timing for NAND-Type Flash Memory (2) 1045
Figure 27.5 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer 1046
Figure 27.6 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory 1047

Figure 27.7 Sector Access when Unusable Sector Exists in Continuous Sectors 1048
Figure 27.8 NAND Flash Command Access (Block Erase) 1050
Figure 27.9 NAND Flash Sector Access (Flash Write) Using DMA 1051
Figure 27.10 NAND Flash Command Access (Flash Read) 1052

Section 28 General Purpose I/O (GPIO)
Figure 28.1 Port Data Output Timing (Example of Port A) 1097
Figure 28.2 Port Data input Timing (Example of Port A) 1098

Section 29 User Break Controller (UBC)
Figure 29.1 Block Diagram of UBC 1102
Figure 29.2 Flowchart of User Break Debugging Support Function 1127

Section 30 User Debugging Interface (H-UDI)
Figure 30.1 H-UDI Block Diagram 1136
Figure 30.2 Sequence for Switching from Boundary-Scan TAP Controller to H-UDI 1139
Figure 30.3 TAP Controller State Transitions 1152
Figure 30.4 H-UDI Reset 1153

Section 31 Electrical Characteristics
Figure 31.1 EXTAL Clock Input Timing 1161
Figure 31.2 CLKOUT Clock Output Timing (1) 1161
Figure 31.3 CLKOUT Clock Output Timing (2) 1161
Figure 31.4 Power-On Oscillation Settling Time 1162
Figure 31.5 MODE pins Setup/Hold Timing 1162
Figure 31.6 PLL Synchronization Settling Time 1163
Figure 31.7 Control Signal Timing 1163
Figure 31.8 SRAM Bus Cycle: Basic Bus Cycle (No Wait) 1165
Figure 31.9 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait) 1166
Figure 31.10 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait + One External Wait) 1167
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) 1168
Figure 31.12 Burst ROM Bus Cycle (No Wait) 1169
Figure 31.13 Burst ROM Bus Cycle (1st Data: One Internal Wait + One External Wait ; 2nd/3rd/4th Data: One Internal Wait) 1170
Figure 31.14 Burst ROM Bus Cycle (No Wait, No Address Setup/ Hold Time Insertion, RDS = 1, RDH = 0) 1171
Figure 31.15 Burst ROM Bus Cycle (One Internal Wait + One External Wait) 1172
Figure 31.16 PCMCIA Memory Bus Cycle 1173
Figure 31.17 PCMCIA I/O Bus Cycle 1174

Figure 31.18 PCMCIA I/O Bus Cycle (TEDx = 1, THEx = 1, IW/PCIW = 1, One Internal Wait, Dynamic Bus Sizing) 1175
Figure 31.19 MPX Basic Bus Cycle: Read 1176
Figure 31.20 MPX Basic Bus Cycle: Write 1177
Figure 31.21 MPX Bus Cycle: Burst Read 1178
Figure 31.22 MPX Bus Cycle: Burst Write 1179
Figure 31.23 Byte Control SRAM Bus Cycle 1180
Figure 31.24 Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0) 1181
Figure 31.25 MCLK Output Timing 1183
Figure 31.26 Read Timing of DDR-SDRAM (2 Burst Read) 1184
Figure 31.27 Write Timing of DDR-SDRAM (2 Burst Write) 1185
Figure 31.28 NMI Input Timing 1186
Figure 31.29 IRQ/IRL, GPIO Interrupt Input and IRQOUT Output Timing 1187
Figure 31.30 PCI Clock Input Timing 1189
Figure 31.31 Output Signal Timing 1189
Figure 31.32 Input Signal Timing 1189
Figure 31.33 $\overline{\text{DREQ}}$ and $\overline{\text{DRAK}}$ Timing 1190
Figure 31.34 TCLK Input Timing 1191
Figure 31.35 CMT Timing (1) 1192
Figure 31.36 CMT Timing (2) 1192
Figure 31.37 SCIFn_SCK Input Clock Timing (n = 0, 1) 1193
Figure 31.38 SCIF Channel n I/O Synchronous Mode Clock Timing (n = 0, 1) 1194
Figure 31.39 SIOF_MCLK Input Timing 1195
Figure 31.40 SIOF Transmission/Reception Timing (Master Mode 1, Fall Sampling) 1196
Figure 31.41 SIOF Transmission/Reception Timing (Master Mode 1, Rise Sampling) 1196
Figure 31.42 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sampling) 1197
Figure 31.43 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sampling) 1197
Figure 31.44 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2) 1198
Figure 31.45 HSPI Data Output/Input Timing 1200
Figure 31.46 MMCIF Transmit Timing 1201
Figure 31.47 MMCIF Receive Timing 1202
Figure 31.48 HAC Cold Reset Timing 1203
Figure 31.49 HAC SYNC Output Timing 1203
Figure 31.50 HAC Clock Input Timing 1203
Figure 31.51 HAC Interface Module Signal Timing 1204
Figure 31.52 SSI Clock Input/Output Timing 1205
Figure 31.53 SSI Transmit Timing (1) 1205
Figure 31.54 SSI Transmit Timing (2) 1206
Figure 31.55 SSI Receive Timing (1) 1206

Figure 31.56 SSI Receive Timing (2) 1206
Figure 31.57 Command Issue Timing of NAND-type Flash Memory 1208
Figure 31.58 Address Issue Timing of NAND-type Flash Memory 1209
Figure 31.59 Data Read Timing of NAND-type Flash Memory 1209
Figure 31.60 Data Write Timing of NAND-type Flash Memory 1210
Figure 31.61 Status Read Timing of NAND-type Flash Memory 1210
Figure 31.62 GPIO Timing 1211
Figure 31.63 TCK Input Timing 1212
Figure 31.64 $\overline{\text{PRESET}}$ Hold Timing 1213
Figure 31.65 H-UDI Data Transfer Timing 1213
Figure 31.66 $\overline{\text{ASEBRK}}$ Pin Break Timing 1213
Figure 31.67 Output Load Circuit 1214
Figure 31.68 Load Capacitance-Delay Time 1215

Appendix

Figure B.1 Instruction Prefetch 1219
Figure E.1 Package Dimensions (449-Pin BGA) 1255
Figure H.1 Sequence of Turning On and Off Power Supply 1275

Tables

Section 1 Overview
Table 1.1 SH7780 Features 2
Table 1.2 Pin Functions 11

Section 2 Programming Model
Table 2.1 Initial Register Values 35
Table 2.2 Bit Allocation for FPU Exception Handling 45

Section 3 Instruction Set
Table 3.1 Execution Order of Delayed Branch Instructions 51
Table 3.2 Addressing Modes and Effective Addresses 53
Table 3.3 Notation Used in Instruction List 57
Table 3.4 Fixed-Point Transfer Instructions 59
Table 3.5 Arithmetic Operation Instructions 61
Table 3.6 Logic Operation Instructions 63
Table 3.7 Shift Instructions 64
Table 3.8 Branch Instructions 65
Table 3.9 System Control Instructions 66
Table 3.10 Floating-Point Single-Precision Instructions 69
Table 3.11 Floating-Point Double-Precision Instructions 70
Table 3.12 Floating-Point Control Instructions 70
Table 3.13 Floating-Point Graphics Acceleration Instructions 71

Section 4 Pipelining
Table 4.1 Representations of Instruction Execution Patterns 74
Table 4.2 Instruction Groups 84
Table 4.3 Combination of Preceding and Following Instructions 86
Table 4.4 Issue Rates and Execution Cycles 88

Section 5 Exception Handling
Table 5.1 Register Configuration 97
Table 5.2 States of Register in Each Operating Mode 97
Table 5.3 Exceptions 102

Section 6 Floating-Point Unit (FPU)
Table 6.1 Floating-Point Number Formats and Parameters 131
Table 6.2 Floating-Point Ranges 132
Table 6.3 Bit Allocation for FPU Exception Handling 140

Section 7 Memory Management Unit (MMU)

Table 7.1 Register Configuration 156
Table 7.2 Register States in Each Processing State 156

Section 8 Caches

Table 8.1 Cache Features 197
Table 8.2 Store Queue Features 197
Table 8.3 Register Configuration 200
Table 8.4 Register States in Each Processing State 200

Section 9 L Memory

Table 9.1 L Memory Addresses 227
Table 9.2 Register Configuration 228
Table 9.3 Register Status in Each Processing State 228
Table 9.4 Protective Function Exceptions to Access L Memory 240

Section 10 Interrupt Controller (INTC)

Table 10.1 Interrupt Types 246
Table 10.2 INTC Pin Configuration 250
Table 10.3 INTC Register Configuration 251
Table 10.4 Register States in Each Operating Mode 253
Table 10.5 Interrupt Request Sources and INT2PRI0 to INT2PRI7 276
Table 10.6 Correspondence between Bits in INT2A0 and Sources 277
Table 10.7 Correspondence between Bits in INT2A1 and Sources 280
Table 10.8 Correspondence between Bits in INT2MSKR and Interrupt Masking 283
Table 10.9 Correspondence between Bits in INT2MSKCR and Interrupt Mask Clearing 285
Table 10.10 Correspondence between Interrupt Input Pins and Bits in INT2GPIC 295
Table 10.11 IRL[3:0], IRL[7:4] Pins and Interrupt Levels 298
Table 10.12 Interrupt Exception Handling and Priority 302
Table 10.13 Interrupt Response Time 311
Table 10.14 Switching Sequence of IRQ/$\overline{\text{IRL}}$[7:0] Pin Function 313

Section 11 Local Bus State Controller (LBSC)

Table 11.1 Pin Configuration 318
Table 11.2 LBSC External Memory Space Map 322
Table 11.3 Correspondence Between External Pins (MODE4 and MODE3) 324
Table 11.4 Correspondence Between External Pin (MODE5) and Endian 325
Table 11.5 PCMCIA Interface Features 325
Table 11.6 PCMCIA Support Interface 326
Table 11.7 Register Configuration 329
Table 11.8 Register State in Each Processing Mode 330
Table 11.9 32-Bit External Device/Big-Endian Access and Data Alignment 353

Table 11.10 16-Bit External Device/Big-Endian Access and Data Alignment 353
Table 11.11 8-Bit External Device/Big-Endian Access and Data Alignment 354
Table 11.12 32-Bit External Device/Little-Endian Access and Data Alignment 355
Table 11.13 16-Bit External Device/Little-Endian Access and Data Alignment 355
Table 11.14 8-Bit External Device/Little-Endian Access and Data Alignment 356
Table 11.15 Relationship between Address and CE When Using PCMCIA Interface 375
Table 11.16 Relationship between D31 to D29 and Access Size in Address Phase 383

Section 12 DDR-SDRAM Interface (DDRIF)

Table 12.1 Pin Configuration 403
Table 12.2 Access and Data Alignment in Little Endian Mode 406
Table 12.3 Access and Data Alignment in Big Endian Mode 408
Table 12.4 Register Configuration 410
Table 12.5 Register States in Each Operating Mode 411
Table 12.6 SDRAM Commands Issuable by DDRIF 426
Table 12.7 Relationship between SPLIT Bits and Address Multiplexing 429

Section 13 PCI Controller (PCIC)

Table 13.1 Input/Output Pins 446
Table 13.2 List of PCIC Registers 449
Table 13.3 Register States in Each Operating Mode 452
Table 13.4 Supported Bus Commands 522
Table 13.5 PCIC Address Map 524
Table 13.6 Interrupt Priority 543

Section 14 Direct Memory Access Controller (DMAC)

Table 14.1 Pin Configuration 559
Table 14.2 Register Configuration of DMAC 561
Table 14.3 Register States in Each Processing Mode 564
Table 14.4 Transfer Request Sources 587
Table 14.5 Selecting External Request Detection with DL, DS Bits 589
Table 14.6 Selecting External Request Detection with DO Bit 589
Table 14.7 Peripheral Module Request Modes 591
Table 14.8 DMA Transfer Matrix in Auto-Request Mode (all channels) 599
Table 14.9 DMA Transfer Matrix in External Request Mode (only channels 0 to 3) 600
Table 14.10 DMA Transfer Matrix in Peripheral module Request Mode 601
Table 14.11 Register Settings for SRAM, Burst ROM, Byte Control SRAM Interface 611
Table 14.12 Register Settings for PCMCIA Interface 612
Table 14.13 Register Settings for MPX Interface (Read Access) 612
Table 14.14 Register Settings for MPX Interface (Write Access) 612

Section 15 Clock Pulse Generator (CPG)
Table 15.1 CPG Pin Configuration 616
Table 15.2 Clock Operating Modes 617
Table 15.3 Register configuration 618
Table 15.4 Register States of CPG in Each Processing Mode 618

Section 16 Watchdog Timer and Reset
Table 16.1 Pin Configuration 627
Table 16.2 Register Configuration 628
Table 16.3 Register States in Each Processing Mode 628

Section 17 Power-Down Mode
Table 17.1 Power-Down Modes 644
Table 17.2 Pin Configuration 645
Table 17.3 Register configuration 645
Table 17.4 Register States in Each Processing Mode 645
Table 17.5 Pin Configuration 653

Section 18 Timer Unit (TMU)
Table 18.1 Pin Configuration 659
Table 18.2 Register Configuration 660
Table 18.3 Register States in Each Processing Mode 661
Table 18.4 TMU Interrupt Sources 674

Section 19 Timer/Counter (CMT)
Table 19.1 Pin Configuration 679
Table 19.2 Register Configuration 679
Table 19.3 Register States of CMT in Each Processing Mode 680
Table 19.4 32-bit Timer Mode: Example of Input Capture Setting 693
Table 19.5 32-bit Timer Mode: Example of Output Compare Setting 696
Table 19.6 16-bit Timer Mode: Example of Input Capture Setting 698
Table 19.7 16-bit Timer Mode: Example of Output Compare Setting 700
Table 19.8 Setting Example of Up-counter Mode 702
Table 19.9 Setting Example of Updown-counter Mode 704
Table 19.10 Setting Example of Updown-counter Mode 706
Table 19.11 CMT Interrupt Setting 706

Section 20 Realtime Clock (RTC)
Table 20.1 RTC Pins 709
Table 20.2 RTC Registers 710
Table 20.3 Register States of RTC in Each Processing Mode 711
Table 20.4 Crystal Oscillator Circuit Constants (Recommended Values) 730

Section 21 Serial Communication Interface with FIFO (SCIF)
Table 21.1 Pin Configuration ... 739
Table 21.2 Register Configuration ... 740
Table 21.3 Register States of SCIF in Each Processing Mode ... 741
Table 21.4 SCSMR Settings ... 758
Table 21.5 SCSMR Settings for Serial Transfer Format Selection ... 770
Table 21.6 SCSMR and SCSCR Settings for SCIF Clock Source Selection ... 771
Table 21.7 Serial Transfer Formats (Asynchronous Mode) ... 773
Table 21.8 SCIF Interrupt Sources ... 793

Section 22 Serial I/O with FIFO (SIOF)
Table 22.1 Pin Configuration ... 799
Table 22.2 Register Configuration of SIOF ... 800
Table 22.3 Register States of SIOF in Each Processing Mode ... 801
Table 22.4 Operation in Each Transfer Mode ... 804
Table 22.5 SIOF Serial Clock Frequency ... 828
Table 22.6 Serial Transfer Modes ... 830
Table 22.7 Frame Length ... 831
Table 22.8 Audio Mode Specification for Transmit Data ... 833
Table 22.9 Audio Mode Specification for Receive Data ... 833
Table 22.10 Setting Number of Channels in Control Data ... 834
Table 22.11 Conditions to Issue Transmit Request ... 836
Table 22.12 Conditions to Issue Receive Request ... 836
Table 22.13 Transmit and Receive Reset ... 842
Table 22.14 SIOF Interrupt Sources ... 843

Section 23 Serial Protocol Interface (HSPI)
Table 23.1 Pin Configuration ... 851
Table 23.2 Register Configuration ... 851
Table 23.3 Register States of HSPI in Each Processing Mode ... 851

Section 24 Multimedia Card Interface (MMCIF)
Table 24.1 Pin Configuration ... 866
Table 24.2 Register Configuration ... 867
Table 24.3 Register States of HSPI in Each Processing Mode ... 869
Table 24.4 CMDR Configuration ... 871
Table 24.5 Correspondence between Commands and Settings of CMDTYR and RSPTYR ... 892
Table 24.6 Correspondence between Command Response Byte Number and RSPR ... 895
Table 24.7 MMCIF Interrupt Sources ... 931

Section 25 Audio Codec Interface (HAC)
Table 25.1 Pin Configuration ... 956
Table 25.2 Register Configuration ... 957
Table 25.3 Register States of HAC in Each Processing Mode ... 957
Table 25.4 AC97 Transmit Frame Structure ... 973
Table 25.5 AC97 Receive Frame Structure ... 974

Section 26 Serial Sound Interface (SSI) Module
Table 26.1 Pin Configuration ... 984
Table 26.2 Register Configuration ... 985
Table 26.3 Register States of SSI in Each Processing Mode ... 985
Table 26.4 Bus Formats of SSI Module ... 998
Table 26.5 Number of Padding Bits for Each Valid Configuration ... 1002

Section 27 NAND Flash Memory Controller (FLCTL)
Table 27.1 Pin Configuration ... 1024
Table 27.2 Register Configuration of FLCTL ... 1025
Table 27.3 Register States of FLCTL in Each Processing Mode ... 1025
Table 27.4 Status Read of NAND-Type Flash Memory ... 1049
Table 27.5 FLCTL Interrupt Requests ... 1053
Table 27.6 DMA Transfer Specifications ... 1053

Section 28 General Purpose I/O (GPIO)
Table 28.1 Multiplexed Pins Controlled by Port Control Registers ... 1056
Table 28.2 Register Configuration ... 1060
Table 28.3 Register States of GPIO in Each Processing Mode ... 1062

Section 29 User Break Controller (UBC)
Table 29.1 Register Configuration ... 1103
Table 29.2 Register Status in Each Processing State ... 1104
Table 29.3 Settings for Match Data Setting Register ... 1116
Table 29.4 Relation between Operand Sizes and Address Bits to be Compared ... 1123

Section 30 User Debugging Interface (H-UDI)
Table 30.1 Pin Configuration ... 1137
Table 30.2 Commands Supported by Boundary-Scan TAP Controller ... 1139
Table 30.3 Register Configuration (1) ... 1140
Table 30.4 Register Configuration (2) ... 1140
Table 30.5 Register Status in Each Processing State ... 1140
Table 30.6 SDBSR Configuration ... 1143

Section 31 Electrical Characteristics
Table 31.1 Absolute Maximum Ratings ... 1155

Table 31.2 DC Characteristics (T_a = –20 to 75°C / –40 to 85°C) 1156
Table 31.3 Permissible Output Currents 1159
Table 31.4 Clock Timing 1159
Table 31.5 Clock and Control Signal Timing 1160
Table 31.6 Control Signal Timing 1163
Table 31.7 Bus Timing 1164
Table 31.8 DDRIF Signal Timing 1182
Table 31.9 INTC Module Signal Timing 1186
Table 31.10 PCIC Signal Timing (in PCIREQ/PCIGNT Non-Port Mode) (1) 1188
Table 31.11 DMAC Module Signal Timing 1190
Table 31.12 TMU Module Signal Timing 1191
Table 31.13 CMT Module Signal Timing 1192
Table 31.14 SCIF Module Signal Timing 1193
Table 31.15 SIOF Module Signal Timing 1195
Table 31.16 HSPI Module Signal Timing 1199
Table 31.17 MMCIF Module Signal Timing 1201
Table 31.18 HAC Interface Module Signal Timing 1203
Table 31.19 SSI Interface Module Signal Timing 1205
Table 31.20 FLCTL Module Signal Timing 1207
Table 31.21 GPIO Signal Timing 1211
Table 31.22 H-UDI Module Signal Timing 1212

Appendix

Table F.1 Clock Operating Modes with External Pin Combination 1256
Table F.2 Area 0 Memory Map and Bus Width 1256
Table F.3 Endian 1256
Table F.4 PCI Mode 1257
Table F.5 Clock Input 1257
Table F.6 Mode Control 1257
Table G.1 Pin states in Reset, Power-Down State, and Bus-Released State 1258
Table G.2 Treatment of Unused Pins 1267
Table I.1 Register Configuration 1276
Table J.1 SH7780 Product Lineup 1277

Section 1 Overview

1.1 SH7780 Features

The SH7780 is an integrated system-on-a-chip microprocessor that is designed as a high performance, embedded, stand-alone Host Processor aimed at the multimedia, infotainment and consumer networking market. The SH7780 features a DDR-SDRAM interface that can be coupled to the DDR320* or 266 SDRAM. Also, because of its built-in functions, such as a PCI bus controller, a DMA controller, timers, and serial communications functions with an audio interface, as required for multimedia, network, and OA equipment, use of the SH7780 enables a high performance and high integrated system.

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

Note: "DDR320" indicates the DDR-SDRAM bus interface which operates at a frequency of 160 MHz in this manual.

The features of the SH7780 are summarized in table 1.1.

Table 1.1 SH7780 Features

Item	Features
LSI	• Operating frequency: 400 MHz • Performance: 720MIPS, 2.8 GFLOPS • Voltage: 1.25 V (internal), 2.5 V (DDR-SDRAM interface), 3.3 V (I/O) • Superscalar architecture: Parallel execution of two instructions • Packages: 449-pin BGA (Size: 21 × 21 mm, pin pitch: 0.8 mm) • Local bus interface (External bus): — Separate 26-bit address and 32-bit data buses — 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 registers) — Seven 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatible with the SH-1, SH-2, SH-3 and SH-4 microcomputers) — Instruction length: 16-bit fixed length for improved code efficiency — Load/store architecture — Delayed branch instructions — Instructions executed with conditions — Instruction set based on the C language • Super scalar which executes two instructions simultaneously including 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

Item	Features
FPU	• On-chip floating-point coprocessor • Supports single-precision (32 bits) and double-precision (64 bits) • Supports IEEE754-compliant data types and exceptions • Two rounding modes: Round to Nearest and Round to Zero • Handling of denormalized numbers: Truncation to zero or interrupt generation for IEEE754 compliance • Floating-point registers: 32 bits × 16 words × 2 banks (single-precision × 16 words or double-precision × 8 words) × 2 banks • 32-bit CPU-FPU floating-point communication register (FPUL) • 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 (double-precision) — Latency (FMAC/ FMUL): 5 cycles (single-precision), 7 cycles (double-precision) — Pitch (FADD/FSUB): 1 cycle (single-precision/double-precision) — Pitch (FMAC/FMUL): 1 cycle (single-precision), 3 cycles (double-precision) Note: FMAC is supported for single-precision only. • 3-D graphics instructions (single-precision only): — 4-dimensional vector conversion and matrix operations (FTRV): 4 cycles (pitch), 8 cycles (latency) — 4-dimensional vector (FIPR) inner product: 1 cycle (pitch), 5 cycles (latency) • Ten-stage pipeline

Item	Features
Memory management unit (MMU)	• 4 Gbytes of physical address space, 256 address space identifiers (address space identifier ASID: 8 bits) • Supports single virtual memory mode and multiple virtual memory mode • Supports multiple page sizes: 1 Kbyte, 4 Kbytes, 64 Kbytes, or 1 Mbyte • 4-entry full associative TLB for instructions • 64-entry full associative TLB for instructions and operands • Supports software selection of replacement method and random-counter replacement algorithms • Contents of TLB are directly accessible through address mapping
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 bus 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 bus master • 32-Kbyte capacity

Item	Features
Interrupt controller (INTC)	• Nine independent external interrupts: NMI and IRQ7 to IRQ0 — NMI: Fall/rise selectable — IRQ: Fall/rise/high level/low level selectable • 15-level signed external interrupts: $\overline{IRL3}$ to $\overline{IRL0}$, or $\overline{IRL7}$ to $\overline{IRL4}$ • On-chip module interrupts: Priority level can be set for each module The following modules can issue on-chip module interrupts: TMU, RTC, SCIF, WDT, H-UDI, DMAC, CMT, HAC, PCIC, SIOF, HSPI, MMCIF, SSI, FLCTL, and GPIO
User break controller (UBC)	• Supports debugging by means of user break interrupts • Two break channels • Address, data value, access type, and data size are available as break 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 64 Mbytes, with the following parameters settable for each area: — Bus size (8, 16, or 32 bits) — Number of wait cycles (hardware wait function also supported) — 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(Internal bus) • Supports a 4-bank DDR-SDRAM • Supports a burst length of 2 • Connectable memory size: 256-Mbit, 512-Mbit, 1-Gbit, and 2-Gbit

Item	Features
PCI bus controller (PCIC)	• PCI bus controller (subset of revision 2.2) — 32-bit bus — 33 MHz/66 MHz support • PCI master/target support • PCI host function support — Built-in bus arbiter • Interrupt requests can be sent to CPU • Up to 512-Mbyte memory can be connected for PCI memory space
Direct memory access controller (DMAC)	• 12-channel physical address DMA controller • 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 (channel 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

Item	Features
Timer unit (TMU)	• 6-channel auto-reload 32-bit timer • Input-capture function (only channel 2) • Choice of seven types counter input clocks (external and peripheral clocks)
Compare Match Timer (CMT)	• 4-channel auto-reload 32-bit timers • Choice of 16 or 32 bits • Choice of 1-shot or free-running operation • Choice of an interrupt source or DMA transfer request from compare match or overflow
Realtime clock (RTC)	• On-chip clock and calendar functions • Built-in 32 kHz crystal oscillator with maximum 1/256 second resolution (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 generator • On-chip modem control function ($\overline{\text{SCIF0\_RTS}}$ and $\overline{\text{SCIF0\_CTS}}$) for 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 3.1 • Supports MMC mode • Interface with MCCLK output for transfer clock output, MCCMD I/O for command output/response input, MCDAT I/O (data I/O) • Four interrupt sources

Item	Features
Audio codec interface (HAC)	• Digital interface for audio codec • Supports transfer for slot 1 to slot 4 • Choice of 16- or 20-bit DMA transfer • Supports various sampling rates by adjusting slot data • Generates interrupt: data ready, data request, overflow, and underrun
Serial sound interface (SSI)	• 1-channel bi-directional transfer • Support compressed-data and non-compressed-data transfer — 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 format can be used in the compressed and non-compressed 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)

1.2 Block Diagram

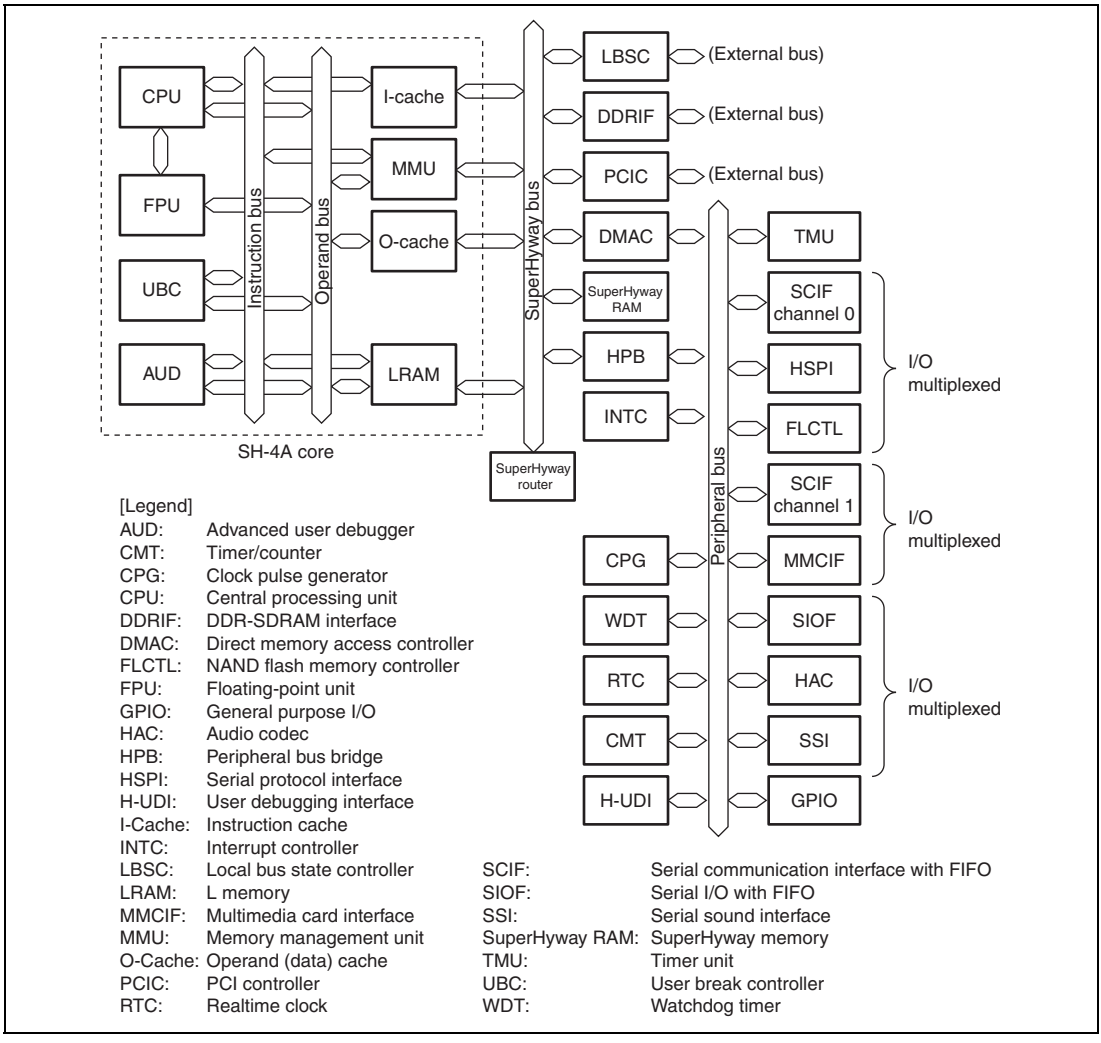


Figure 1.1 SH7780 Block Diagram

1.3 Pin Arrangement

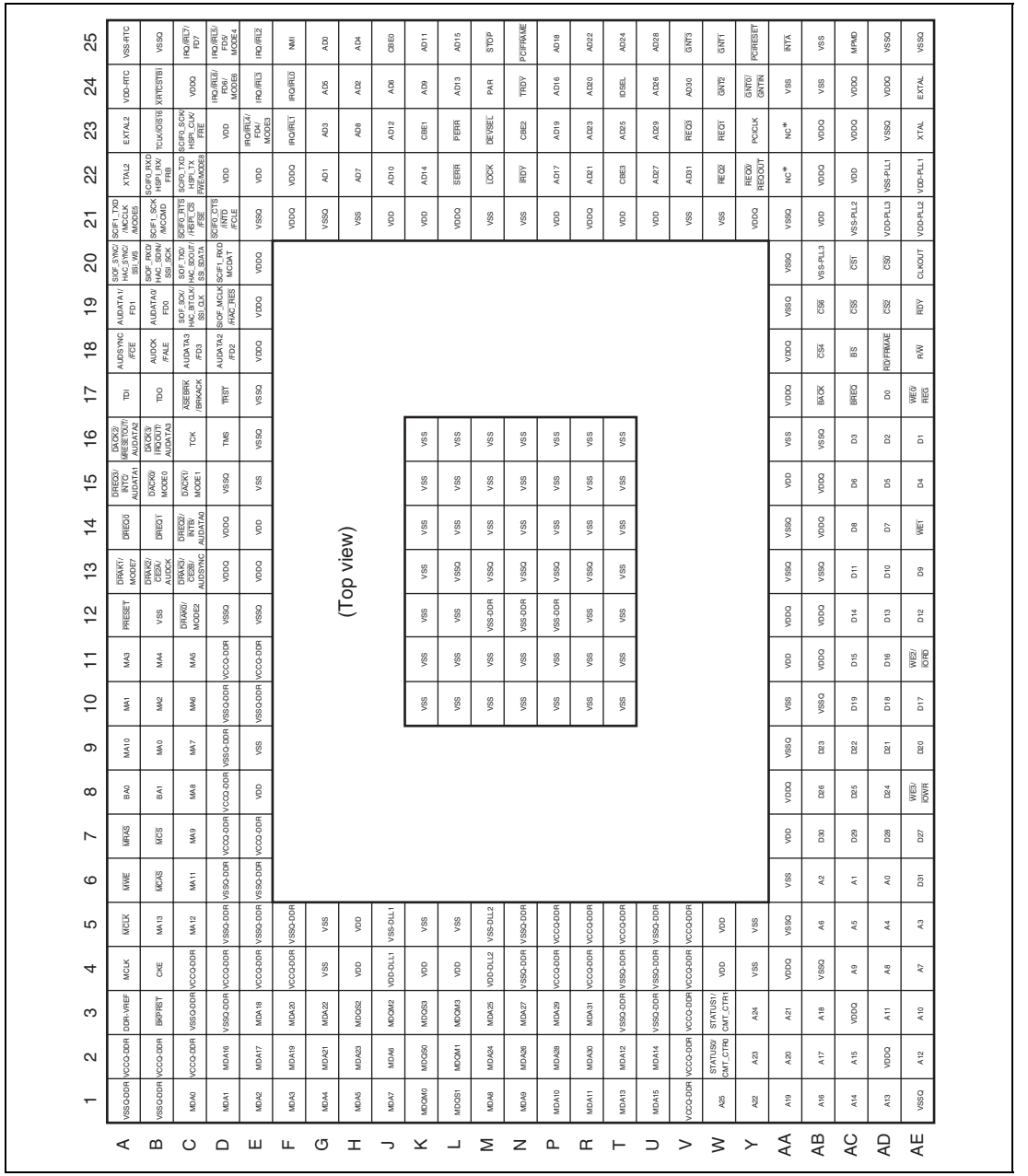


Figure 1.2 SH7780 Pin Arrangement

1.4 Pin Functions

Table 1.2 lists the pin functions of the SH7780. In the I/O column, I, O, and IO indicate input, output, and input/output, respectively. In the GPIO column, for example, A0 indicates the port A0, which also functions as a general I/O port (input/output).

Table 1.2 Pin Functions

No.	Pin No.	Pin Name	I/O	Function	GPIO*
1	A1	VSSQ-DDR	—	DDR I/O GND	
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	L0(O)
14	A14	$\overline{\text{DREQ0}}$	I	DMA channel 0 request	K7
15	A15	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	I/I/O	DMA channel 3 request/ PCI interrupt C/H-UDI emulator	K4*
16	A16	$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2	O/O/O	DMA channel 2 bus acknowledgment/manual reset output/ H-UDI emulator	K3
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 select	J3

No.	Pin No.	Pin Name	I/O	Function	GPIO*
21	A21	SCIF1_TXD/MCCLK/MODE5	O/O/I	SCIF 1 transmit data/ card clock output/mode control 5	H6(O)
22	A22	XTAL2	O	RTC clock	
23	A23	EXTAL2	I	RTC crystal resonator	
24	A24	VDD-RTC	—	RTC VDD	
25	A25	VSS-RTC	—	RTC GND	
26	B1	VSSQ-DDR	—	DDR I/O GND	
27	B2	VCCQ-DDR	—	DDR I/O VCC	
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	K1(O)
39	B14	$\overline{\text{DREQ1}}$	I	DMA channel 1 request	K6
40	B15	$\overline{\text{DACK0}}$/MODE0	O/I	DMA channel 0 bus acknowledgement/mode control 0	L3(O)
41	B16	$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3	O/O/O	DMA channel 3 bus acknowledgement/ interrupt request output/H-UDI emulator	K2
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 clock	J4
46	B21	SCIF1_SCK/MCCMD	IO/IO	SCIF1 serial clock/ MMCIF command response	H7

No.	Pin No.	Pin Name	I/O	Function	GPIO*
47	B22	SCIF0_RXD/HSPI_RX/FRB	I/I/I	SCIF receive data/HSPI receive data input/NAND flash ready or busy	H2
48	B23	TCLK/$\overline{\text{IOIS16}}$	IO/I	TMU clock/PCMCIA IOIS16	J0*
49	B24	$\overline{\text{XRTCSTBI}}$	I	RTC standby	
50	B25	VSSQ	—	I/O GND	
51	C1	MDA0	IO	DDR data	
52	C2	VCCQ-DDR	—	DDR I/O VCC	
53	C3	VSSQ-DDR	—	DDR I/O GND	
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	L1(O)
63	C13	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC	O/O/O	DMA channel 3 request acknowledgment/PCMCIA CE2/ H-UDI emulator	K0(O)
64	C14	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	I/I/O	DMA channel 2 request/PCI interrupt B/H-UDI emulator	K5*
65	C15	$\overline{\text{DACK1}}$/MODE1	O/I	DMA channel 1 bus acknowledgement/mode control 1	L2(O)
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 clock	J1
70	C20	SIOF_TXD/HAC_SDOUT/ SSI_SDATA	O/O/IO	SIOF transmit data/HAC serial data/ SSI serial data	J5
71	C21	$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CS}}$/$\overline{\text{FSE}}$	IO/IO/O	SCIF modem control/HSPI chip selection/NAND flash spare area enable	H0*

No.	Pin No.	Pin Name	I/O	Function	GPIO*
72	C22	SCIF0_TXD/HSPI_TX/$\overline{\text{FWE}}$/MODE8	O/O/O/I	SCIF0 transmit data/HSPI transmit data/NAND flash write enable/ mode control 8	H3(O)
73	C23	SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$	IO/IO/O	SCIF0 serial clock/HSPI serial clock/NAND flash read enable	H4
74	C24	VDDQ	—	I/O VDD	
75	C25	IRQ/$\overline{\text{IRL7}}$/FD7	I/IO	IRL IRQ interrupt request 7/ NAND flash data	E6*
76	D1	MDA1	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	J2
95	D20	SCIF1_RXD/MCDAT	I/IO	SCIF1 receive data/MMCIF data	H5
96	D21	$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE	IO/I/O	SCIF modem control/PCI interrupt D /NAND flash command latch enable	H1*
97	D22	VDD	—	Internal VDD	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
98	D23	VDD	—	Internal VDD	
99	D24	IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	I/IO/I	IRL IRQ interrupt request 6/NAND flash data/mode control 6	
100	D25	IRQ/$\overline{\text{IRL5}}$/FD5/MODE4	I/IO/I	IRL IRQ interrupt request 5/NAND flash data/mode control 4	
101	E1	MDA2	IO	DDR data	
102	E2	MDA17	IO	DDR data	
103	E3	MDA18	IO	DDR data	
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
126	F1	MDA3	IO	DDR data	
127	F2	MDA19	IO	DDR data	
128	F3	MDA20	IO	DDR data	
129	F4	VCCQ-DDR	—	DDR I/O VCC	
130	F5	VSSQ-DDR	—	DDR I/O GND	
131	F21	VDDQ	—	I/O VDD	
132	F22	VDDQ	—	I/O VDD	
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	D1
143	G23	AD3	IO	PCI address/data	D3
144	G24	AD5	IO	PCI address/data	D5
145	G25	AD0	IO	PCI address/data	D0
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	D7
153	H23	AD8	IO	PCI address/data	C0
154	H24	AD2	IO	PCI address/data	D2
155	H25	AD4	IO	PCI address/data	D4

No.	Pin No.	Pin Name	I/O	Function	GPIO*
156	J1	MDA7	IO	DDR data	
157	J2	MDA6	IO	DDR data	
158	J3	MDQM2	O	DDR data mask	
159	J4	VDD-DLL1	—	DLL1 VDD	
160	J5	VSS-DLL1	—	DLL1 GND	
161	J21	VDD	—	Internal VDD	
162	J22	AD10	IO	PCI address/data	C2
163	J23	AD12	IO	PCI address/data	C4
164	J24	AD6	IO	PCI address/data	D6
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	C6
180	K23	CBE1	IO	PCI command/byte enable	
181	K24	AD9	IO	PCI address/data	C1
182	K25	AD11	IO	PCI address/data	C3
183	L1	MDQS1	IO	DDR data strobe	
184	L2	MDQM1	O	DDR data mask	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
185	L3	MDQM3	O	DDR data mask	
186	L4	VDD	—	Internal VDD	
187	L5	VSS	—	Internal GND	
188	L10	VSS	—	Internal GND	
189	L11	VSS	—	Internal GND	
190	L12	VSS	—	Internal GND	
191	L13	VSSQ	—	I/O GND	
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	C5
199	L25	AD15	IO	PCI address/data	C7
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
215	M24	PAR	IO	PCI parity	
216	M25	$\overline{\text{STOP}}$	IO	PCI transaction stop	
217	N1	MDA9	IO	DDR data	
218	N2	MDA26	IO	DDR data	
219	N3	MDA27	IO	DDR data	
220	N4	VSSQ-DDR	—	DDR I/O GND	
221	N5	VSSQ-DDR	—	DDR I/O GND	
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
245	P16	VSS	—	Internal GND	
246	P21	VDD	—	Internal VDD	
247	P22	AD17	IO	PCI address/data	B1
248	P23	AD19	IO	PCI address/data	B3
249	P24	AD16	IO	PCI address/data	B0
250	P25	AD18	IO	PCI address/data	B2
251	R1	MDA11	IO	DDR data	
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	B5
265	R23	AD23	IO	PCI address/data	B7
266	R24	AD20	IO	PCI address/data	B4
267	R25	AD22	IO	PCI address/data	B6
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
275	T12	VSS	—	Internal GND	
276	T13	VSS	—	Internal GND	
277	T14	VSS	—	Internal GND	
278	T15	VSS	—	Internal GND	
279	T16	VSS	—	Internal GND	
280	T21	VDD	—	Internal VDD	
281	T22	CBE3	IO	PCI command/byte enable	
282	T23	AD25	IO	PCI address/data	A1
283	T24	IDSEL	I	PCI configuration device select	
284	T25	AD24	IO	PCI address/data	A0
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	A3
292	U23	AD29	IO	PCI address/data	A5
293	U24	AD26	IO	PCI address/data	A2
294	U25	AD28	IO	PCI address/data	A4
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	A7
302	V23	$\overline{\text{REQ3}}$	I	Bus request (PCI host)	E3*
303	V24	AD30	IO	PCI address/data	A6
304	V25	$\overline{\text{GNT3}}$	O	PCI bus grant	E0*

No.	Pin No.	Pin Name	I/O	Function	GPIO*
305	W1	A25	O	Address bus	
306	W2	STATUS0/CMT_CTR0	O/IO	Status0/CMT0 timer counter	
307	W3	STATUS1/CMT_CTR1	O/IO	Status1/CMT1 timer counter	
308	W4	VDD	—	Internal VDD	
309	W5	VDD	—	Internal VDD	
310	W21	VSS	—	Internal GND	
311	W22	$\overline{\text{REQ2}}$	I	Bus request (PCI host)	E4*
312	W23	$\overline{\text{REQ1}}$	I	Bus request (PCI host)	E5*
313	W24	$\overline{\text{GNT2}}$	O	PCI bus grant	E1*
314	W25	$\overline{\text{GNT1}}$	O	PCI bus grant	E2*
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
334	AA10	VSS	—	Internal GND	
335	AA11	VDD	—	Internal VDD	
336	AA12	VDDQ	—	I/O VDD	
337	AA13	VSSQ	—	I/O GND	
338	AA14	VSSQ	—	I/O GND	
339	AA15	VDD	—	Internal VDD	
340	AA16	VSS	—	Internal 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	F6
357	AB8	D26	IO	Data bus	F2
358	AB9	D23	IO	Data bus	G7
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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
364	AB15	VDDQ	—	I/O VDD	
365	AB16	VSSQ	—	I/O GND	
366	AB17	$\overline{\text{BACK}}$	O	Bus acknowledgement	M0
367	AB18	$\overline{\text{CS4}}$	O	Chip select 4	
368	AB19	$\overline{\text{CS6}}$	O	Chip select 6	
369	AB20	VSS-PLL3	—	PLL3 GND	
370	AB21	VDD	—	Internal 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	F5
382	AC8	D25	IO	Data bus	F1
383	AC9	D22	IO	Data bus	G6
384	AC10	D19	IO	Data bus	G3
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	M1
392	AC18	$\overline{\text{BS}}$	O	Bus start	
393	AC19	$\overline{\text{CS5}}$	O	Chip select 5	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
394	AC20	$\overline{CS1}$	O	Chip select 1	
395	AC21	VSS-PLL2	—	PLL2 GND	
396	AC22	VDD	—	Internal VDD	
397	AC23	VDDQ	—	I/O VDD	
398	AC24	VDDQ	—	I/O VDD	
399	AC25	MPMD	I	Mode control	
400	AD1	A13	O	Address bus	
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	F4
407	AD8	D24	IO	Data bus	F0
408	AD9	D21	IO	Data bus	G5
409	AD10	D18	IO	Data bus	G2
410	AD11	D16	IO	Data bus	G0
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	$\overline{RD}/\overline{FRAME}$	O	Read strobe/MPX interface cycle frame	
418	AD19	$\overline{CS2}$	O	Chip select 2	
419	AD20	$\overline{CS0}$	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	

No.	Pin No.	Pin Name	I/O	Function	GPIO*
424	AD25	VSSQ	—	I/O GND	
425	AE1	VSSQ	—	I/O GND	
426	AE2	A12	O	Address bus	
427	AE3	A10	O	Address bus	
428	AE4	A7	O	Address bus	
429	AE5	A3	O	Address bus	
430	AE6	D31	IO	Data bus	F7
431	AE7	D27	IO	Data bus	F3
432	AE8	$\overline{WE3}$/$\overline{IOWR}$	O/O	Selection signal for D31 to D24	
433	AE9	D20	IO	Data bus	G4
434	AE10	D17	IO	Data bus	G1
435	AE11	$\overline{WE2}$/$\overline{IORD}$	O/O	Selection signal for D23 to D16/PCMCIA 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	

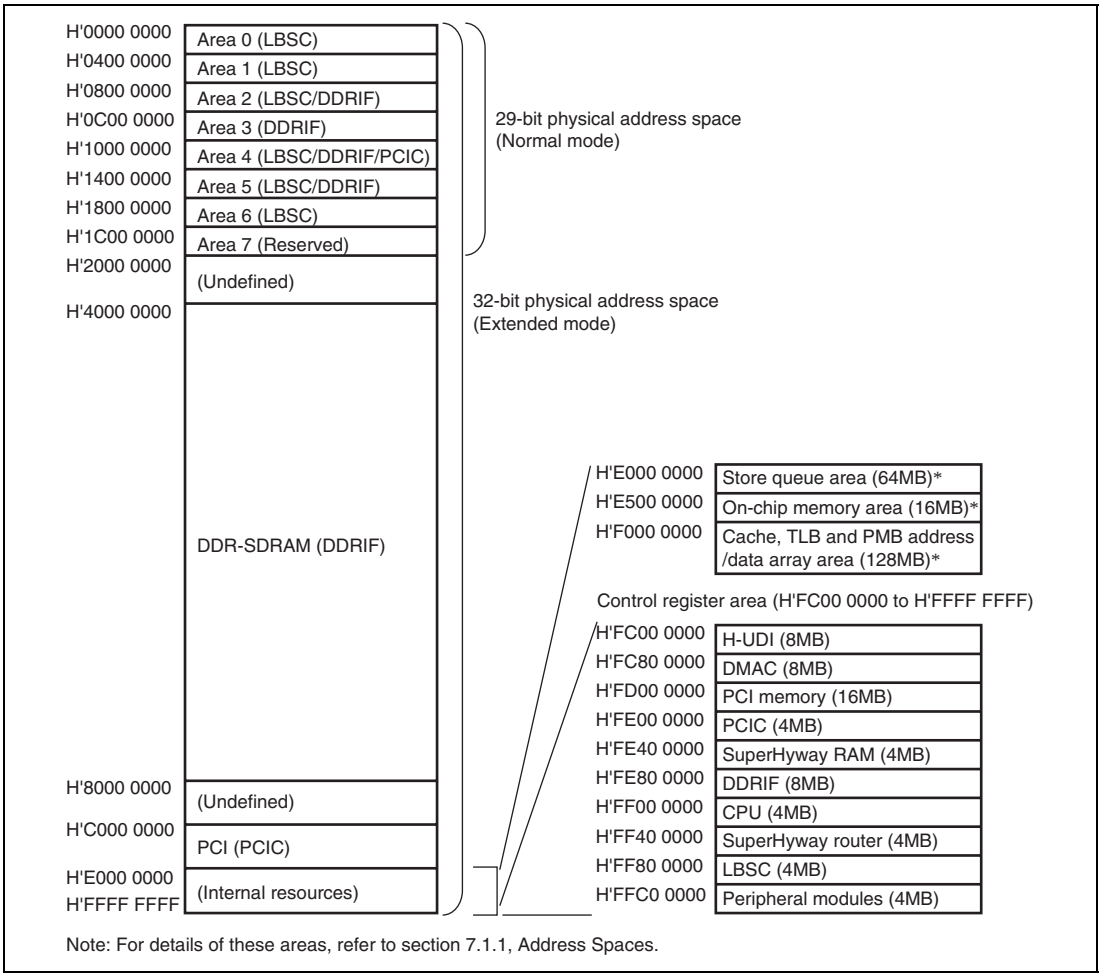
Note: * Can be used as a GPIO interrupt pin. (O) Only outputs.

1.5 Memory Address Map

The SH7780 supports 32-bit virtual address space, and supports both 29-bit and 32-bit physical address spaces (normal mode and extended mode). For details of mappings from the virtual address space to the physical address spaces, see section 7, Memory Management Unit (MMU).

The external memory space of the SH7780 consists of the LBSC space, DDRIF space and PCIC space. The LBSC has up to 384 Mbytes, the DDRIF has up to 256 Mbytes and the PCIC has up to 512 Mbytes external memory space individually and the SH7780 can control the external memory space up to 1152 Mbytes totally. Areas 0, 1, and 6 are controlled by the LBSC. Areas 2, 4, and 5 are controlled by the LBSC, DDRIF or PCIC that depends on the setting of the memory address map select register (MMSELR) of the LBSC. Note that area 3 is for the DDRIF. For details, see section 11, Local Bus Sate Controller (LBSC), section 12, DDR-SDRAM Interface (DDRIF) or section 13, PCI Controller (PCIC).

Figure 1.3 shows the physical address space of the SH7780. Figure 1.4 shows the relationship between the AREASEL bits and the memory address map. The 32-bit physical address space 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	MMSELR.AREASEL[2:0]*	B'000	B'001	B'010	B'011	B'100
H'0000 0000	Area 0 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'0400 0000	Area 1 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'0800 0000	Area 2 (LBSC/DDRIF)	LBSC	LBSC	DDRIF-0	DDRIF-0	DDRIF-0
H'0C00 0000	Area 3 (DDRIF)	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'1000 0000	Area 4 (LBSC/DDRIF/PCIC)	LBSC	PCIC	LBSC	PCIC	DDRIF-2
H'1400 0000	Area 5 (LBSC/DDRIF)	LBSC	LBSC	LBSC	LBSC	DDRIF-3
H'1800 0000	Area 6 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'1C00 0000	Area 7 (Reserved area)					
H'2000 0000	(Undefined)					
H'4000 0000	DDR-SDRAM (DDRIF)	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
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		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		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 – H'FFFF FFFF	(Internal resources)					

[ ] : Shadow

H'0000 0000 – H'1FFF FFFF: 29-bit physical address space (Normal mode)

H'0000 0000 – H'FFFF FFFF: 32-bit physical address space (Extended mode)

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 Map

1.6 SuperHyway Bus

The SH7780 is implemented with the SuperHyway bus as the system bus.

The SuperHyway bus is a 32-bit-address, 64-bit-data internal bus capable of up to 200 MHz operation that is connected to on-chip modules to allow high speed communication.

Each module that is connected to the SuperHyway bus operates as an initiator (i.e. , bus master) that issues a transfer request or a target that replies with a response to the request. The transaction is controlled by the dedicated SuperHyway router.

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

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

An actual transaction on the SuperHyway bus is started from a request issued by the initiator module according to a read/write command sent to the SuperHyway bus address (physical address), and then the target module replies with a response to the request (LOAD/STORE transaction). In addition, a transaction that controls the cache coherency occurs if necessary (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 (except for the INTC and DMAC modules).

1.7 SuperHyway Memory (SuperHyway RAM)

The SH7780 includes an on-chip SuperHyway memory which stores instructions or data. The SuperHyway memory has the following features.

- Capacity

  Total SuperHyway memory capacity is 32 Kbytes (512 words × 256 bits × 2 pages).
- Memory address map

  The SuperHyway memory is allocated within the physical address H'FE41 0000 to H'FE41 3FFF and H'FE42 0000 to H'FE42 3FFF.
- Ports

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

  The SuperHyway memory is always accessed by the SuperHyway bus master module, 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. After that 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 cycles
  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 memory in sleep mode.

Section 2 Programming Model

The programming model of this LSI is explained in this section. This LSI has registers and data formats as shown below.

2.1 Data Formats

The data formats supported in this LSI are shown in figure 2.1.

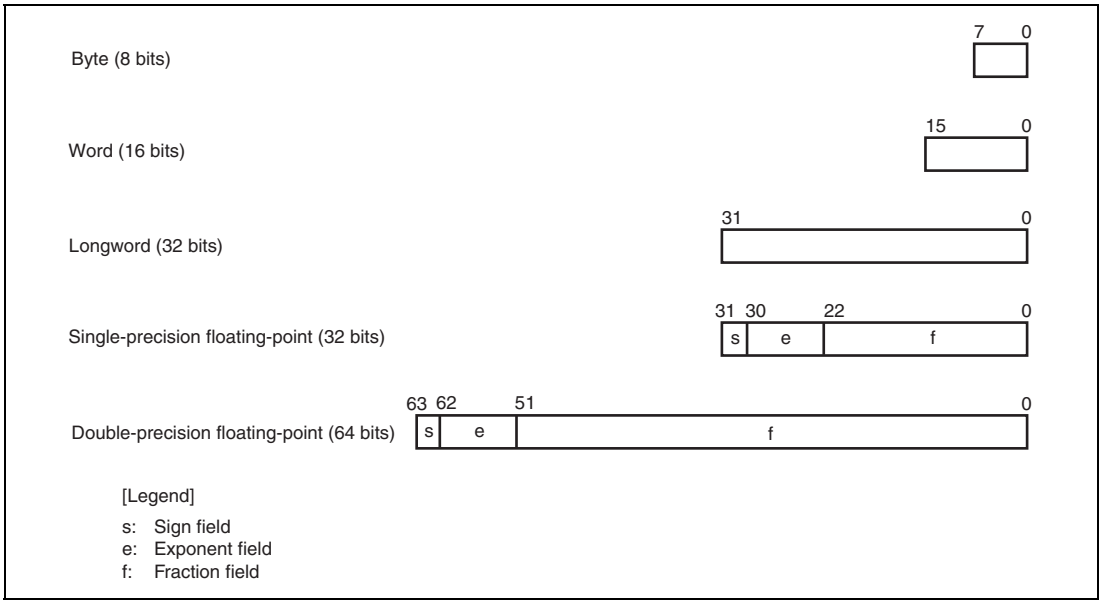


Figure 2.1 Data Formats

2.2 Register Descriptions

2.2.1 Privileged Mode and Banks

Processing Modes: This LSI has two processing modes, user mode and privileged mode. This LSI normally operates in user mode, and switches to privileged mode when an exception occurs or an interrupt is accepted. There are four kinds of registers—general registers, system registers, control registers, and floating-point registers—and the registers that can be accessed differ in the two processing modes.

General Registers: There are 16 general registers, designated R0 to R15. General registers R0 to 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 which banked register set is accessed as general registers, and which set is accessed only through the load 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 bank 1 general registers R0_BANK1 to R7_BANK1 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 are accessed by the LDC/STC instructions.
  When the RB bit is 0 (that is, when bank 0 is selected), the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 are accessed by the LDC/STC instructions.
- User mode

  In user mode, the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. The eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 cannot be accessed.

Control Registers: Control registers comprise the global base register (GBR) and status register (SR), which can be accessed in both processing modes, and the saved status register (SSR), saved program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and debug base register (DBR), which can only be accessed in privileged mode. Some bits of the status 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 these registers does not depend on the processing mode.

Floating-Point Registers and System Registers Related to FPU: There are thirty-two floating-point registers, FR0–FR15 and XF0–XF15. FR0–FR15 and XF0–XF15 can be assigned to either of two banks (FPR0_BANK0–FPR15_BANK0 or FPR0_BANK1–FPR15_BANK1).

FR0–FR15 can be used as the eight registers DR0/2/4/6/8/10/12/14 (double-precision floating-point registers, or pair registers) or the four registers FV0/4/8/12 (register vectors), while XF0–XF15 can be used as the eight registers XD0/2/4/6/8/10/12/14 (register pairs) or register matrix XMTRX.

System registers related to the FPU comprise the floating-point communication register (FPUL) and the floating-point status/control register (FPSCR). These registers are used for communication between the FPU and the CPU, and the exception handling setting.

Register values after a reset are shown in table 2.1.

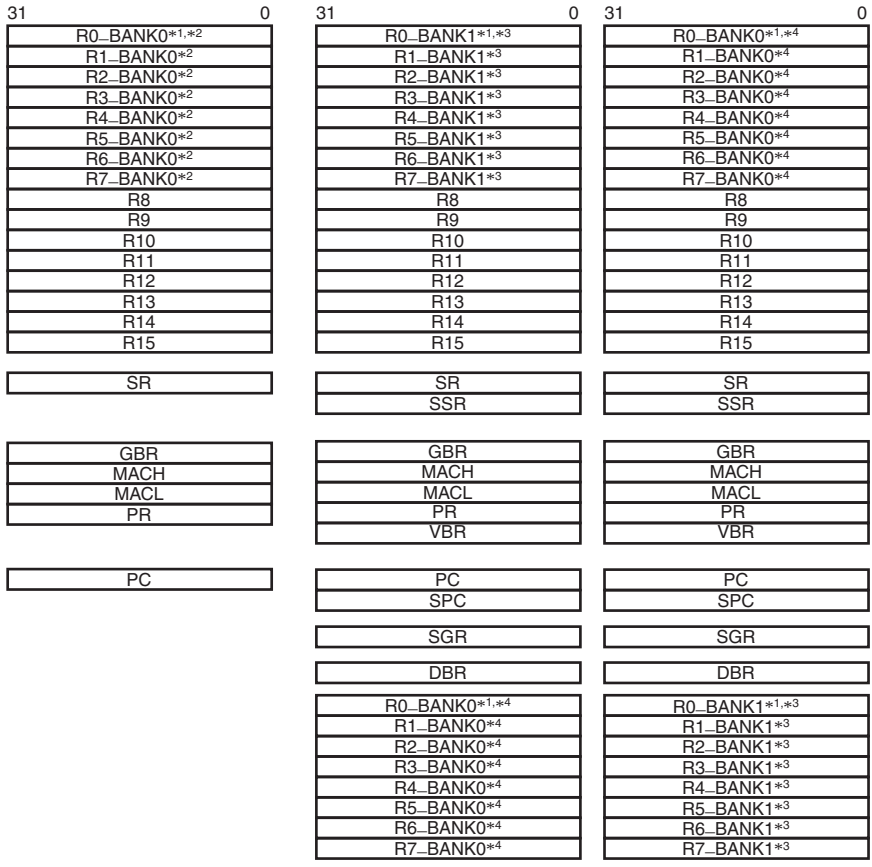
Table 2.1 Initial Register Values

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 bit = 0, 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 status register.



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

2.2.2 General Registers

Figure 2.3 shows the relationship between the processing modes and general registers. This LSI has twenty-four 32-bit general registers (R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, and R8 to R15). However, only 16 of these can be accessed as general registers R0 to R15 in one processing mode. This LSI has two processing modes, user mode and privileged mode.

- R0_BANK0 to R7_BANK0
  Allocated to R0 to R7 in user mode (SR.MD = 0)
  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.

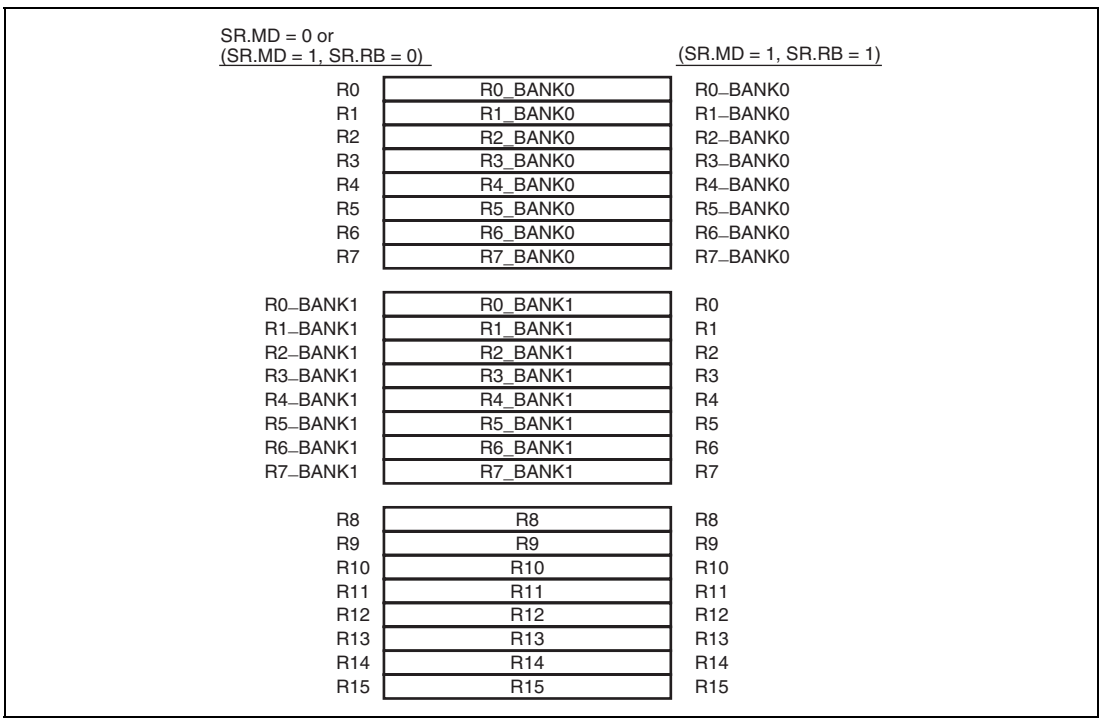


Figure 2.3 General Registers

Note on Programming: As the user's R0 to R7 are assigned to R0_BANK0 to R7_BANK0, and after an exception or interrupt R0 to R7 are assigned to R0_BANK1 to R7_BANK1, it is not necessary for the interrupt handler to save and restore the user's R0 to R7 (R0_BANK0 to R7_BANK0).

2.2.3 Floating-Point Registers

Figure 2.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers, FPR0_BANK0 to FPR15_BANK0, AND FPR0_BANK1 to FPR15_BANK1, comprising two banks. These registers are referenced as FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, or XMTRX. Reference names of each register 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_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are assigned to FPR0_BANK1 to FPR15_BANK1.

3. Double-precision floating-point registers or single-precision floating-point registers, DRi (8 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, FR15}

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises 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_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are assigned to FPR0_BANK0 to FPR15_BANK0.

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 = {XF14, XF15}

7. Single-precision floating-point extended register matrix, XMTRX: XMTRX comprises all 16 XF registers.

$$XMTRX = \begin{bmatrix} XF0 & XF4 & XF8 & XF12 \\ XF1 & XF5 & XF9 & XF13 \\ XF2 & XF6 & XF10 & XF14 \\ XF3 & XF7 & XF11 & XF15 \end{bmatrix}$$

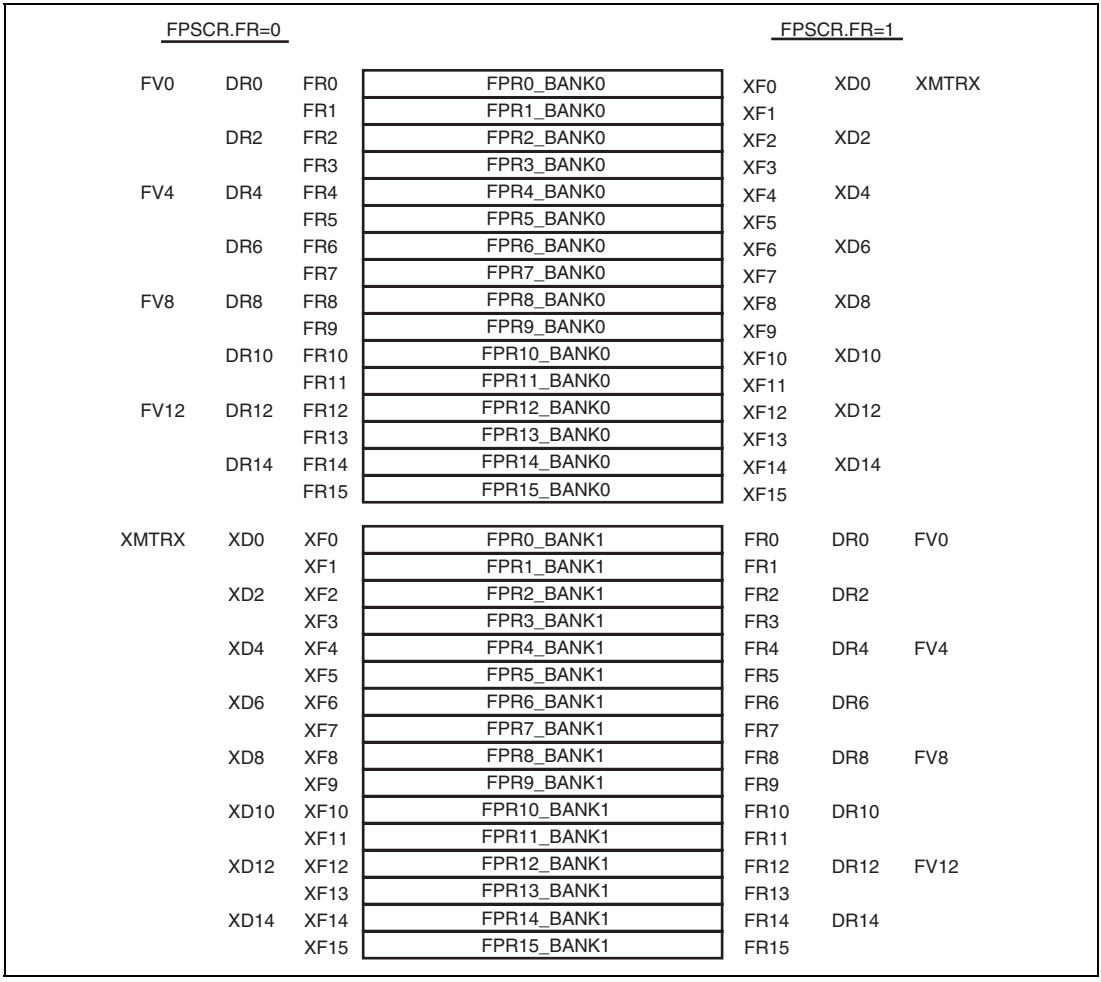


Figure 2.4 Floating-Point Registers

2.2.4 Control Registers

Status Register (SR):

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MD	RB	BL	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	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	1	0
	FD	—	—	—	—	—	M	Q	IMASK				—	—	S	T
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
30	MD	1	R/W	Processing Mode Selects the processing mode. 0: User mode (Some instructions cannot be executed 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 Specification Bit 0: R0_BANK0 to R7_BANK0 are accessed as general registers R0 to R7 and R0_BANK1 to R7_BANK1 can be accessed using LDC/STC instructions 1: R0_BANK1 to R7_BANK1 are accessed as general registers R0 to R7 and R0_BANK0–R7_BANK0 can be accessed using LDC/STC instructions 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 an interrupt. While this bit is set to 1, an interrupt request is masked. In this case, this processor enters the reset state when a general exception other than a user break occurs.

Bit	Bit Name	Initial Value	R/W	Description
27 to 16	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
15	FD	0	R/W	FPU Disable Bit When this bit is set to 1 and an FPU instruction is not in a delay slot, a general FPU disable exception occurs. When this bit is set to 1 and an FPU instruction is in a delay slot, a slot FPU disable exception occurs. (FPU instructions: H'F*** instructions and LDS (.L)/STS(.L) instructions using FPUL/FPSCR)
14 to 10	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
9	M	0	R/W	M Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
8	Q	0	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	IMASK	All 1	R/W	Interrupt Mask Level Bits An interrupt whose priority is equal to or less than the value of the IMASK bits is masked. It can be chosen by CPU operation mode register (CPUOPM) whether the level of IMASK is changed to accept an interrupt or not when an interrupt is occurred. For details, see Appendix A, CPU Operation Mode Register (CPUOPM).
3, 2	—	All 0	R	Reserved For details on reading/writing this bit, see General 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, or overflow/underflow. For details, see section 3, Instruction Set.

Saved Status Register (SSR) (32 bits, Privileged Mode, Initial Value = Undefined): The contents of SR are saved to SSR in the event of an exception or interrupt.

Saved Program Counter (SPC) (32 bits, Privileged Mode, Initial Value = Undefined): The address of an instruction at which an interrupt or exception occurs is saved to SPC.

Global Base Register (GBR) (32 bits, Initial Value = Undefined): GBR is referenced as the base address of addressing @(disp,GBR) and @(R0,GBR).

Vector Base Register (VBR) (32 bits, Privileged Mode, Initial Value = H'00000000): VBR is referenced as the branch destination base address in the event of an exception or interrupt. For details, see section 5, Exception Handling.

Saved General Register 15 (SGR) (32 bits, Privileged Mode, Initial Value = Undefined): The 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): When the user break debugging function is enabled (CBCR.UBDE = 1), DBR is referenced as the branch 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 to store the operation result of a MAC or MUL instruction.

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

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

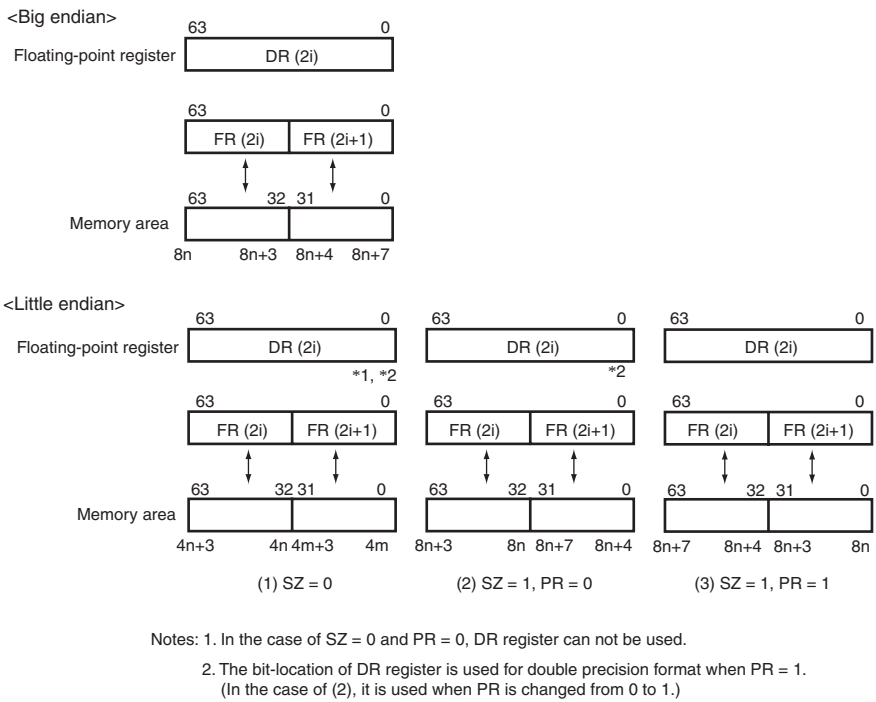
Floating-Point Status/Control Register (FPSCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	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	1	0
	Cause				Enable (EN)					Flag					RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 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 register pair (64 bits) For relationship between the SZ bit, PR bit, and endian, 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 support instructions are undefined) For relationship between the SZ bit, PR bit, and endian, 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

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable (EN)	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU exception occurs, the bits corresponding to FPU exception cause field and flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. 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)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Floating-Point Communication Register (FPUL) (32 bits, Initial Value = Undefined):
Information is transferred between the FPU and CPU via FPUL.

2.3 Memory-Mapped Registers

Some control registers are mapped to the following memory areas. Each of the mapped registers has two addresses.

H'1C00 0000 to H'1FFF FFFF
H'FC00 0000 to H'FFFF FFFF

These two areas are used as follows.

- H'1C00 0000 to H'1FFF FFFF

  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 access to a memory-mapped register.

  The operation of an access to this area without using the address translation function of the 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 error. Memory-mapped registers can be referenced in user mode by means of access that involves address translation.

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

2.4 Data Formats in Registers

Register operands are always longwords (32 bits). When a memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register.

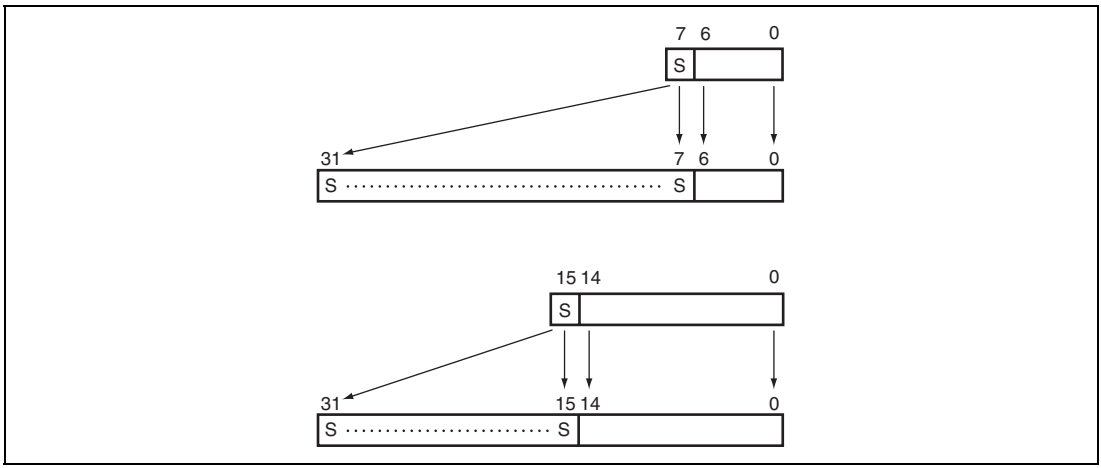


Figure 2.6 Formats of Byte Data and Word Data in Register

2.5 Data Formats in Memory

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

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

Big endian or little endian byte order can be selected for the data format. The endian should be set with the external pin after a power-on reset. The endian cannot be changed dynamically. Bit positions are numbered left to right from most-significant to least-significant. Thus, in a 32-bit longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is the least significant bit.

The data format in memory is shown in figure 2.7.

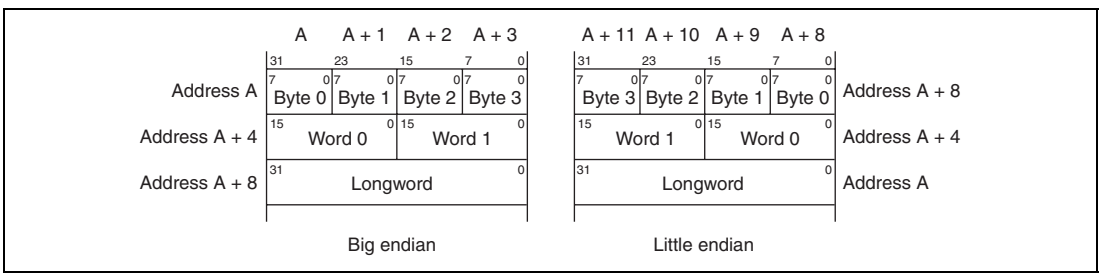


Figure 2.7 Data Formats in Memory

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

2.6 Processing States

This LSI has major three processing states: the reset state, instruction execution state, and power-down state.

Reset State: In this state the CPU is reset. The reset state is divided into the power-on reset state and the manual reset.

In the power-on reset state, the internal state of the CPU and the on-chip peripheral module registers are initialized. In the manual reset state, the internal state of the CPU and some registers of on-chip peripheral modules are initialized. For details, see register descriptions for each section.

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

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

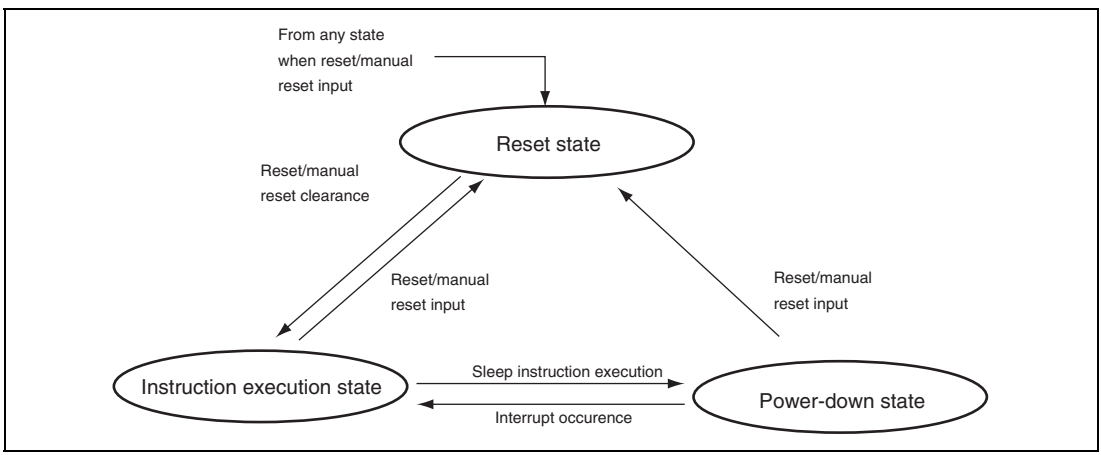


Figure 2.8 Processing State Transitions

2.7 Usage Note

2.7.1 Notes on self-modified codes*

This LSI prefetches instructions more drastically than conventional SH-4 to accelerate the processing speed. Therefore if the instruction in the memory is modified and it is executed immediately, then the pre-modified code that is prefetched are likely to be executed. In order to execute the modified code definitely, one of the following sequences should be executed between the execution of modifying codes and modified codes.

(1) In case the modified codes are in non-cacheable area

```
SYNCO
ICBI  @Rn
```

The target for the ICBI instruction can be any address within the range where no address error 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 by the ICBI instruction. The ICBI instruction should be issued to each cache line. One cache line is 32 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 the 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 OCBP, OCBWB and ICBI instruction should be issued to each cache line. One cache line is 32 bytes.

Note: * Processes executed while changing the instructions on the memory dynamically.

Section 3 Instruction Set

This LSI's instruction set is implemented with 16-bit fixed-length instructions. This LSI can use byte (8-bit), word (16-bit), longword (32-bit), and quadword (64-bit) data sizes for memory access. Single-precision floating-point data (32 bits) can be moved to and from memory using longword or quadword size. Double-precision floating-point data (64 bits) can be moved to and from memory using longword size. When this LSI moves byte-size or word-size data from memory to a register, the data is sign-extended.

3.1 Execution Environment

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

Load-Store Architecture: This LSI has a load-store architecture in which operations are basically executed using registers. Except for bit-manipulation operations such as logical AND that are executed directly in memory, operands in an operation that requires memory access are loaded 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 following the branch is executed before the branch destination instruction.

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

Table 3.1 Execution Order of Delayed Branch Instructions

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

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

T Bit: The T bit in SR is used to show the result of a compare operation, and is referenced by a conditional branch instruction. An example of the use of a conditional branch instruction is shown below.

```
ADD     #1, R0      ; T bit is not changed by ADD operation
CMP/EQ  R1, R0      ; If R0 = R1, T bit is set to 1
BT      TARGET      ; Branches to TARGET if T bit = 1 (R0 = R1)
```

In an RTE delay slot, the SR bits are referenced as follows. In instruction access, the MD bit is used before modification, and in data access, the MD bit is accessed after modification. The other bits—S, T, M, Q, FD, BL, and RB—after modification are used for 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 an immediate value. 16-bit and 32-bit constant values can be defined as literal constant values in memory, and can be referenced by a PC-relative load instruction.

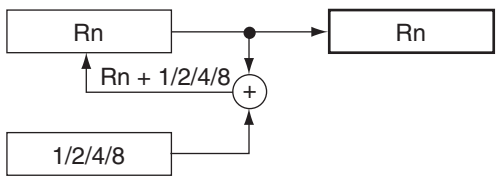
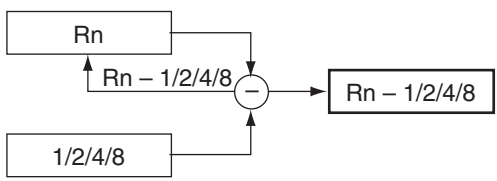
```
MOV.W   @(disp, PC), Rn
MOV.L   @(disp, PC), Rn
```

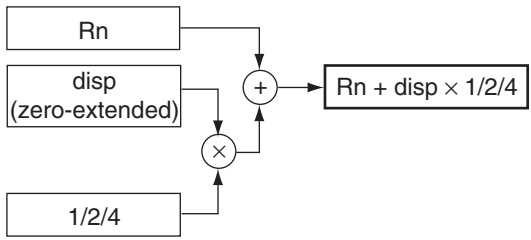
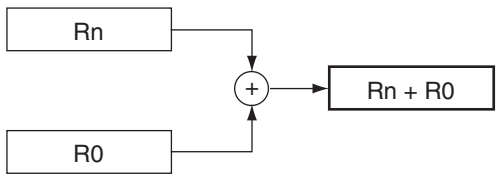
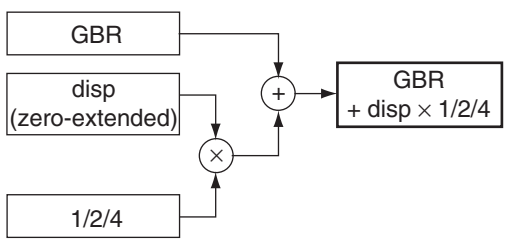
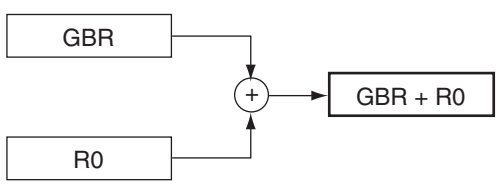
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

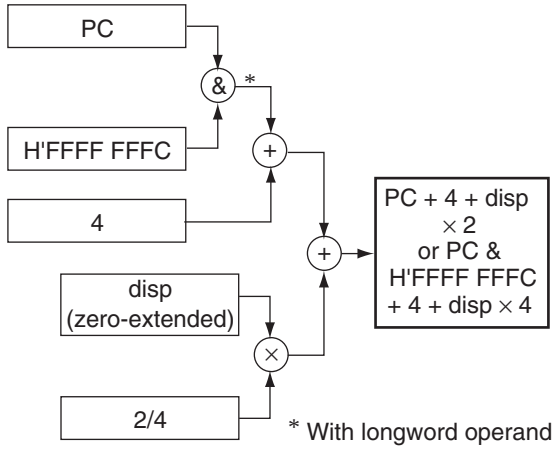
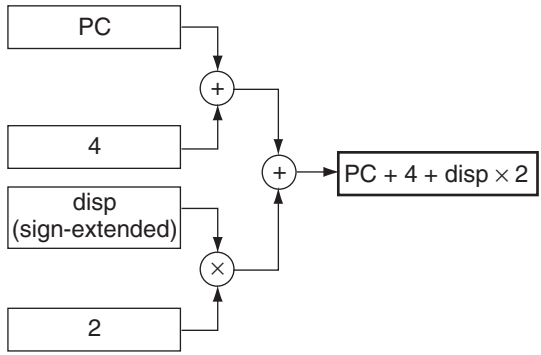
3.2 Addressing Modes

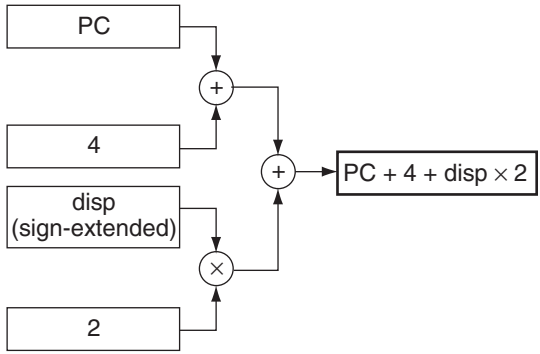
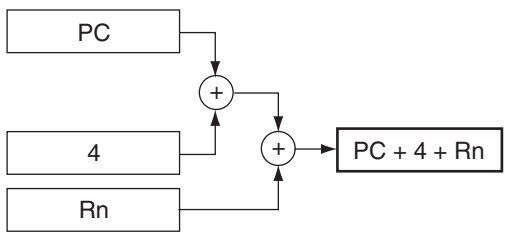
Addressing modes and effective address calculation methods are shown in table 3.2. When a location in virtual memory space is accessed (AT in MMUCR = 1), the effective address is translated into a physical memory address. If multiple virtual memory space systems are selected (SV in MMUCR = 0), the least significant bit of PTEH is also referenced as the access ASID. For details, see section 7, Memory Management Unit (MMU).

Table 3.2 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn → EA (EA: effective address)
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 → EA After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn Quadword: Rn + 8 → Rn
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand, 8 for a quadword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn Quadword: Rn – 8 → Rn Rn → EA (Instruction executed with Rn after calculation)

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

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is PC + 4 with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: PC + 4 + disp × 2 → EA Longword: PC & H'FFFF FFFC + 4 + disp × 4 → 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 + 4 + disp × 2 → Branch-Target

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative	disp:12	Effective address is PC + 4 with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
	Rn	Effective address is sum of PC + 4 and Rn. 	PC + 4 + Rn → Branch-Target
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 descriptions in this manual show the value before scaling (×1, ×2, or ×4) is performed according to the 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

3.3 Instruction Set

Table 3.3 shows the notation used in the SH instruction lists shown in tables 3.4 to 3.13.

Table 3.3 Notation Used in Instruction List

Item	Format	Description	
Instruction mnemonic	OP.Sz SRC, DEST	OP:	Operation code
		Sz:	Size
		SRC:	Source operand
		DEST:	Source and/or destination operand
		Rm:	Source register
		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 bits
		~	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_BANK)
		nnn:	Register number (DRn, XDn, Rn_BANK)
		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

Item	Format	Description
Privileged mode		"Privileged" means the instruction can only be executed in privileged mode.
T bit	Value of T bit after instruction execution	—: No change
New	—	"New" means the instruction which is newly added in this LSI.

Note: Scaling (×1, ×2, ×4, or ×8) is executed according to the size of the instruction operand.

Table 3.4 Fixed-Point Transfer Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV	#imm,Rn	imm → sign extension → Rn	1110nnnniiiiiiii	—	—	—
MOV.W	@(disp*,PC),Rn	(disp × 2 + PC + 4) → sign extension → Rn	1001nnnndddddddd	—	—	—
MOV.L	@(disp*,PC),Rn	(disp × 4 + PC & H'FFFF FFFC + 4) → Rn	1101nnnndddddddd	—	—	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	—	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	—	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	—	—
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	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—	—
MOV.B	R0,@(disp*,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	—	—
MOV.W	R0,@(disp*,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	—	—
MOV.L	R0,@(disp*,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	—	—
MOV.B	@(disp*,GBR),R0	(disp + GBR) → sign extension → R0	11000100dddddddd	—	—	—
MOV.W	@(disp*,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101dddddddd	—	—	—
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	—	LDST	New
MOVLI.L	@Rm,R0	1 → LDST (Rm) → R0 When interrupt/exception occurred 0 → LDST	0000mmmm01100011	—	—	New
MOVUA.L	@Rm,R0	(Rm) → R0 Load non-boundary alignment data	0100mmmm10101001	—	—	New
MOVUA.L	@Rm+,R0	(Rm) → R0, Rm + 4 → Rm Load non-boundary alignment data	0100mmmm11101001	—	—	New
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).

Table 3.5 Arithmetic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	—	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	—	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, carry → T	0011nnnnmmmm1110	—	Carry	—
ADDV	Rm,Rn	Rn + Rm → Rn, overflow → T	0011nnnnmmmm1111	—	Overflow	—
CMP/EQ	#imm,R0	When R0 = imm, 1 → T Otherwise, 0 → T	10001000iiiiiiii	—	Comparison result	—
CMP/EQ	Rm,Rn	When Rn = Rm, 1 → T Otherwise, 0 → T	0011nnnnmmmm0000	—	Comparison result	—
CMP/HS	Rm,Rn	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0010	—	Comparison result	—
CMP/GE	Rm,Rn	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0011	—	Comparison result	—
CMP/HI	Rm,Rn	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0110	—	Comparison result	—
CMP/GT	Rm,Rn	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0111	—	Comparison result	—
CMP/PZ	Rn	When Rn ≥ 0, 1 → T Otherwise, 0 → T	0100nnnn00010001	—	Comparison result	—
CMP/PL	Rn	When Rn > 0, 1 → T Otherwise, 0 → T	0100nnnn00010101	—	Comparison result	—
CMP/STR	Rm,Rn	When any bytes are equal, 1 → T Otherwise, 0 → T	0010nnnnmmmm1100	—	Comparison result	—
DIV1	Rm,Rn	1-step division (Rn ÷ Rm)	0011nnnnmmmm0100	—	Calculation result	—
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^Q → T	0010nnnnmmmm0111	—	Calculation result	—
DIV0U		0 → M/Q/T	0000000000011001	—	0	—
DMULS.L	Rm,Rn	Signed, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm1101	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
DMULU.L	Rm,Rn	Unsigned, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm0101	—	—	—
DT	Rn	Rn – 1 → Rn; when Rn = 0, 1 → T When Rn ≠ 0, 0 → T	0100nnnn00010000	—	Comparison result	—
EXTS.B	Rm,Rn	Rm sign-extended from byte → Rn	0110nnnnmmmm1110	—	—	—
EXTS.W	Rm,Rn	Rm sign-extended from word → Rn	0110nnnnmmmm1111	—	—	—
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	—	Underflow	—

Table 3.6 Logic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
AND	Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	—	—
AND	#imm,R0	R0 & imm → R0	11001001iiiiiiii	—	—	—
AND.B	#imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	—	—
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	—	—
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	—	—
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	—	—
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	—	Test result	—
TST	Rm,Rn	Rn & Rm; when result = 0, 1 → T Otherwise, 0 → T	0010nnnnmmmm1000	—	Test result	—
TST	#imm,R0	R0 & imm; when result = 0, 1 → T Otherwise, 0 → T	11001000iiiiiiii	—	Test result	—
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; when result = 0, 1 → T Otherwise, 0 → T	11001100iiiiiiii	—	Test result	—
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	—	—	—

Table 3.7 Shift Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
ROTL	Rn	T ← Rn ← MSB	0100nnnn00000100	—	MSB	—
ROTR	Rn	LSB → Rn → T	0100nnnn00000101	—	LSB	—
ROTCL	Rn	T ← Rn ← T	0100nnnn00100100	—	MSB	—
ROTCR	Rn	T → Rn → T	0100nnnn00100101	—	LSB	—
SHAD	Rm,Rn	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> Rm → [MSB → Rn]	0100nnnnmmmm1100	—	—	—
SHAL	Rn	T ← Rn ← 0	0100nnnn00100000	—	MSB	—
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	—	—	—

Table 3.8 Branch Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
BF	label	When T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001011dddddddd	—	—	—
BF/S	label	Delayed branch; when T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001111dddddddd	—	—	—
BT	label	When T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001001dddddddd	—	—	—
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	—	—	—

Table 3.9 System Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
CLRMAC		0 → MACH, MACL	0000000000101000	—	—	—
CLRS		0 → S	0000000001001000	—	—	—
CLRT		0 → T	0000000000001000	—	0	—
ICBI	@Rn	Invalidates instruction cache block indicated by virtual address	0000nnnn11100011	—	—	New
LDC	Rm,SR	Rm → SR	0100mmmm00001110	Privileged	LSB	—
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	—	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	Privileged	—	—
LDC	Rm,SGR	Rm → SGR	0100mmmm00111010	Privileged	—	New
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	LSB	—
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	—	New
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	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
NOP		No operation	0000000000001001	—	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—	—
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—	—
PREFI	@Rn	Reads 32-byte instruction block into instruction cache	0000nnnn11010011	—	—	New
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	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	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
STS	PR,Rn	PR → Rn	0000nnnn00101010	—	—	—
STS.L	MACH,@-Rn	Rn – 4 → Rn, MACH → (Rn)	0100nnnn00000010	—	—	—
STS.L	MACL,@-Rn	Rn – 4 → Rn, MACL → (Rn)	0100nnnn00010010	—	—	—
STS.L	PR,@-Rn	Rn – 4 → Rn, PR → (Rn)	0100nnnn00100010	—	—	—
SYNCO		Prevents the next instruction from being issued until instructions issued before this instruction have been completed.	0000000010101011	—	—	New
TRAPA	#imm	PC + 2 → SPC, SR → SSR, #imm << 2 → TRA, H'160 → EXPEVT, VBR + H'0100 → PC	11000011iiiiiiii	—	—	—

Table 3.10 Floating-Point Single-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FLDI0	FRn	H'0000 0000 → FRn	1111nnnn10001101	—	—	—
FLDI1	FRn	H'3F80 0000 → FRn	1111nnnn10011101	—	—	—
FMOV	FRm,FRn	FRm → FRn	1111nnnnmmmm1100	—	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnmmmm1000	—	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnmmmm0110	—	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnmmmm1001	—	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnmmmm1010	—	—	—
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	—	Comparison result	—
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Comparison result	—
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	—	—	—

Table 3.11 Floating-Point Double-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FABS	DRn	DRn & H'7FFF FFFF FFFF FFFF → DRn	1111nnn001011101	—	—	—
FADD	DRm,DRn	DRn + DRm → DRn	1111nnn0mmm00000	—	—	—
FCMP/EQ	DRm,DRn	When DRn = DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00100	—	Comparison result	—
FCMP/GT	DRm,DRn	When DRn > DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00101	—	Comparison result	—
FDIV	DRm,DRn	DRn /DRm → DRn	1111nnn0mmm00011	—	—	—
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	New
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	—	—	—

Table 3.13 Floating-Point Graphics Acceleration Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FMOV	DRm,XDn	DRm → XDn	1111nnn1mmm01100	—	—	—
FMOV	XDm,DRn	XDm → DRn	1111nnn0mmm11100	—	—	—
FMOV	XDm,XDn	XDm → XDn	1111nnn1mmm11100	—	—	—
FMOV	@Rm,XDn	(Rm) → XDn	1111nnn1mmmm1000	—	—	—
FMOV	@Rm+,XDn	(Rm) → XDn, Rm + 8 → Rm	1111nnn1mmmm1001	—	—	—
FMOV	@(R0,Rm),XDn	(R0 + Rm) → XDn	1111nnn1mmmm0110	—	—	—
FMOV	XDm,@Rn	XDm → (Rn)	1111nnnnmmm11010	—	—	—
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	—	—	New
FSRRA	FRn	1/sqrt (FRn)* → FRn	1111nnnn01111101	—	—	New
FSCA	FPUL,DRn	sin(FPUL) → FRn cos(FPUL) → FR[n + 1]	1111nnn011111101	—	—	New

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

Section 4 Pipelining

This LSI is a 2-ILP (instruction-level-parallelism) superscalar pipelining microprocessor. Instruction execution is pipelined, and two instructions can be executed in parallel.

4.1 Pipelines

Figure 4.1 shows the basic pipelines. Normally, a pipeline consists of seven stages: instruction sfetch (I1/I2), decode and register read (ID), execution (E1/E2/E3), and write-back (WB). An instruction is executed as a combination of basic pipelines.

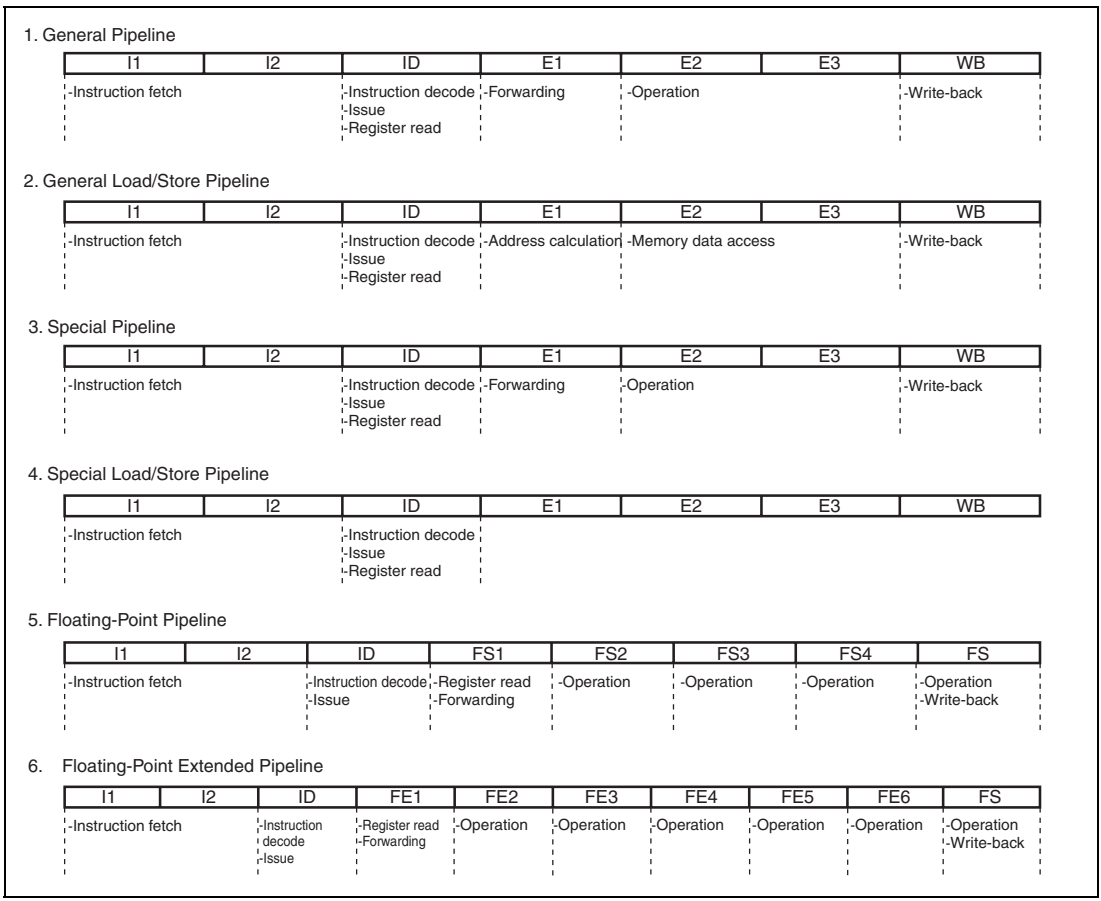


Figure 4.1 Basic Pipelines

Figure 4.2 shows the instruction execution patterns. Representations in figure 4.2 and their descriptions are listed in table 4.1.

Table 4.1 Representations of Instruction Execution Patterns

Representation	Description
E1 E2 E3 WB	CPU EX pipe is occupied
S1 S2 S3 WB	CPU LS pipe is occupied (with memory access)
s1 s2 s3 WB	CPU LS pipe is occupied (without memory access)
E1/S1	Either CPU EX pipe or CPU LS pipe is occupied
E1S1 , E1s1	Both CPU EX pipe and CPU LS pipe are occupied
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

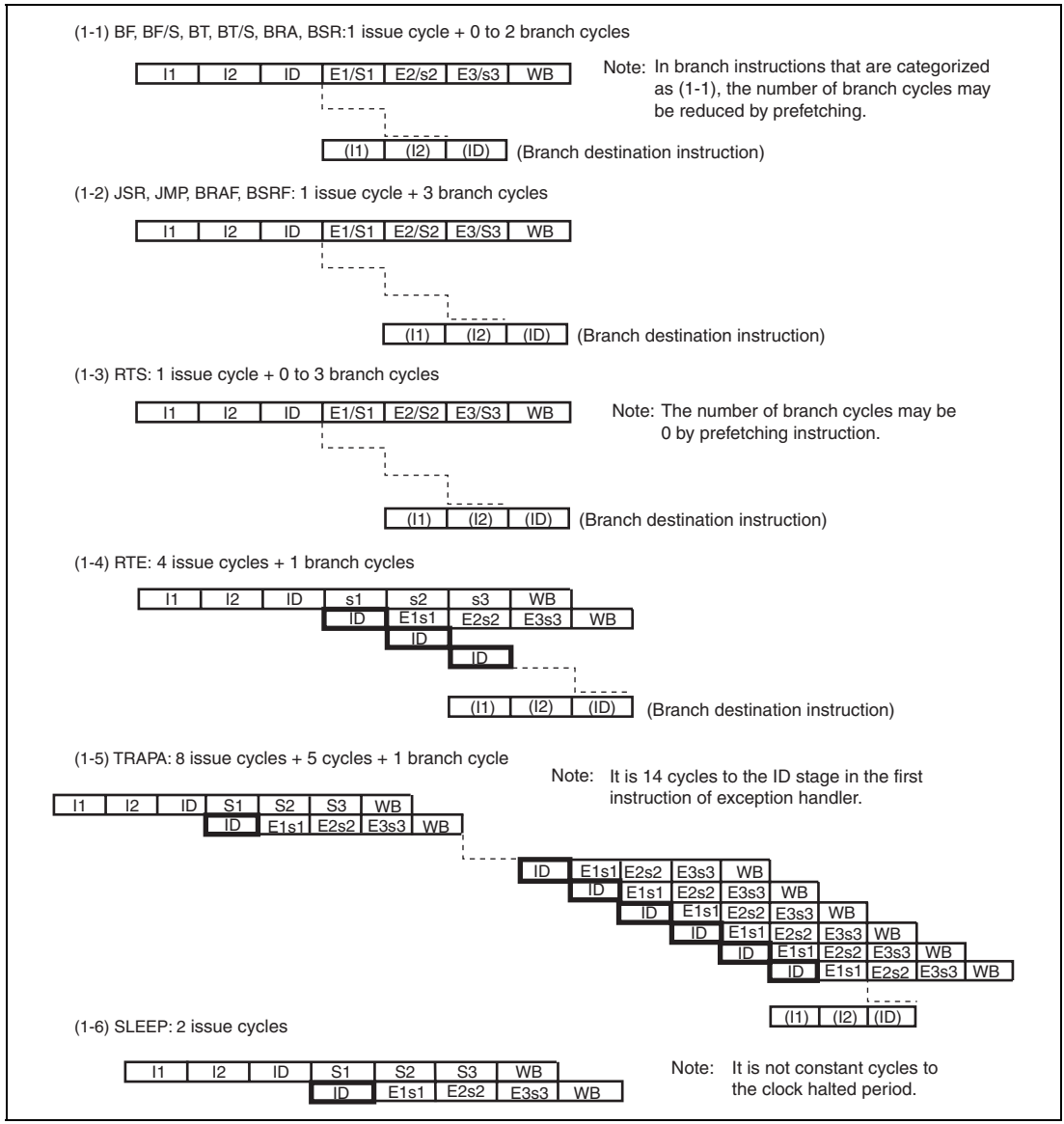


Figure 4.2 Instruction Execution Patterns (1)

(2-1) 1-step operation (EX type): 1 issue cycle

EXT[SU].[BW], MOVT, SWAP, XTRCT, ADD*, CMP*, DIV*, DT, NEG*, SUB*, AND, AND#, NOT, OR, OR#, TST, TST#, XOR, XOR#, ROT*, SHA*, SHL*, CLRS, CLRT, SETS, SETT

Note: Except for AND#, OR#, TST#, and XOR# instructions using GBR relative addressing mode

I1	I2	ID	E1	E2	E3	WB

(2-2) 1-step operation (LS type): 1 issue cycle

MOVA

I1	I2	ID	s1	s2	s3	WB

(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)

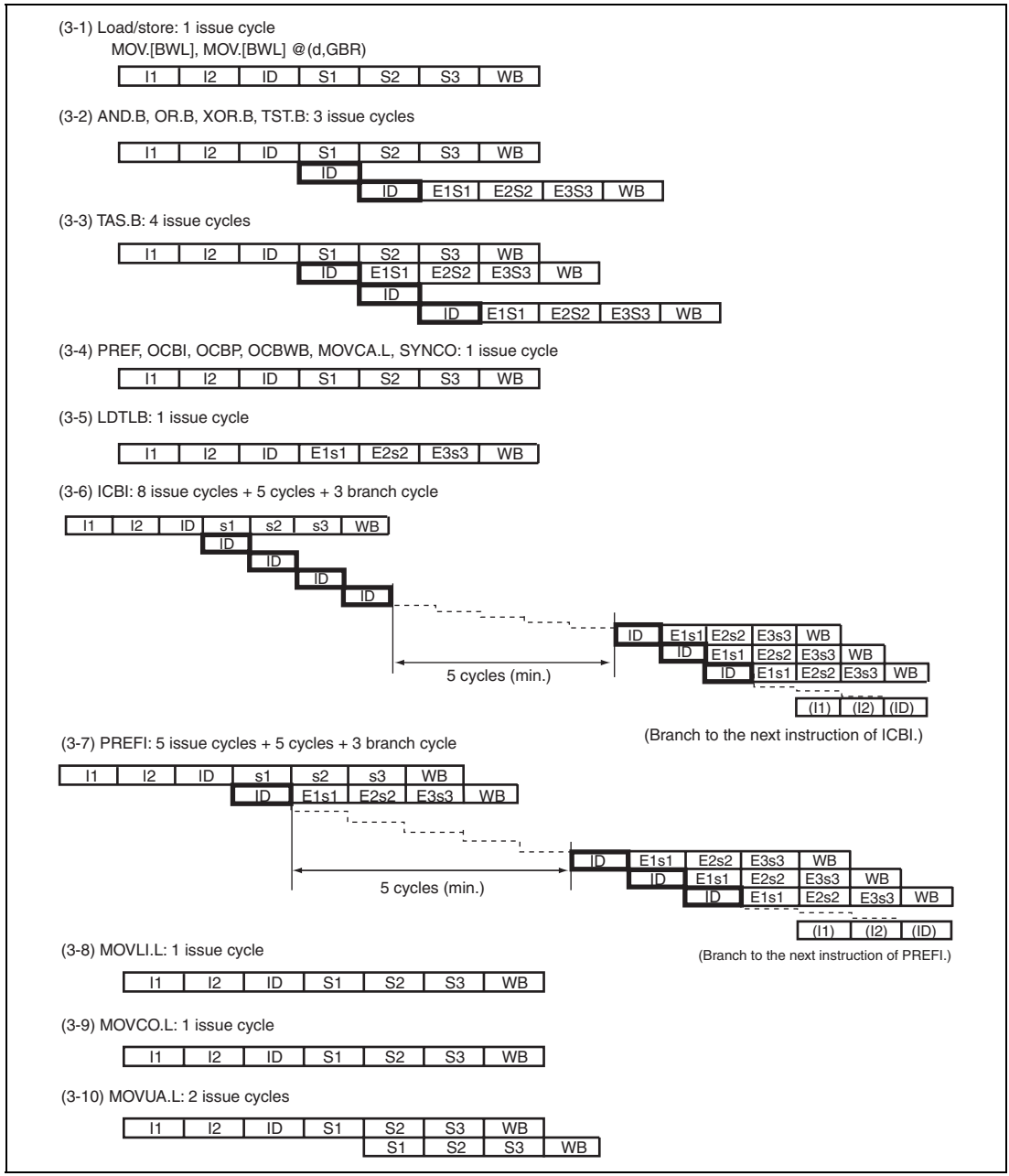


Figure 4.2 Instruction Execution Patterns (3)

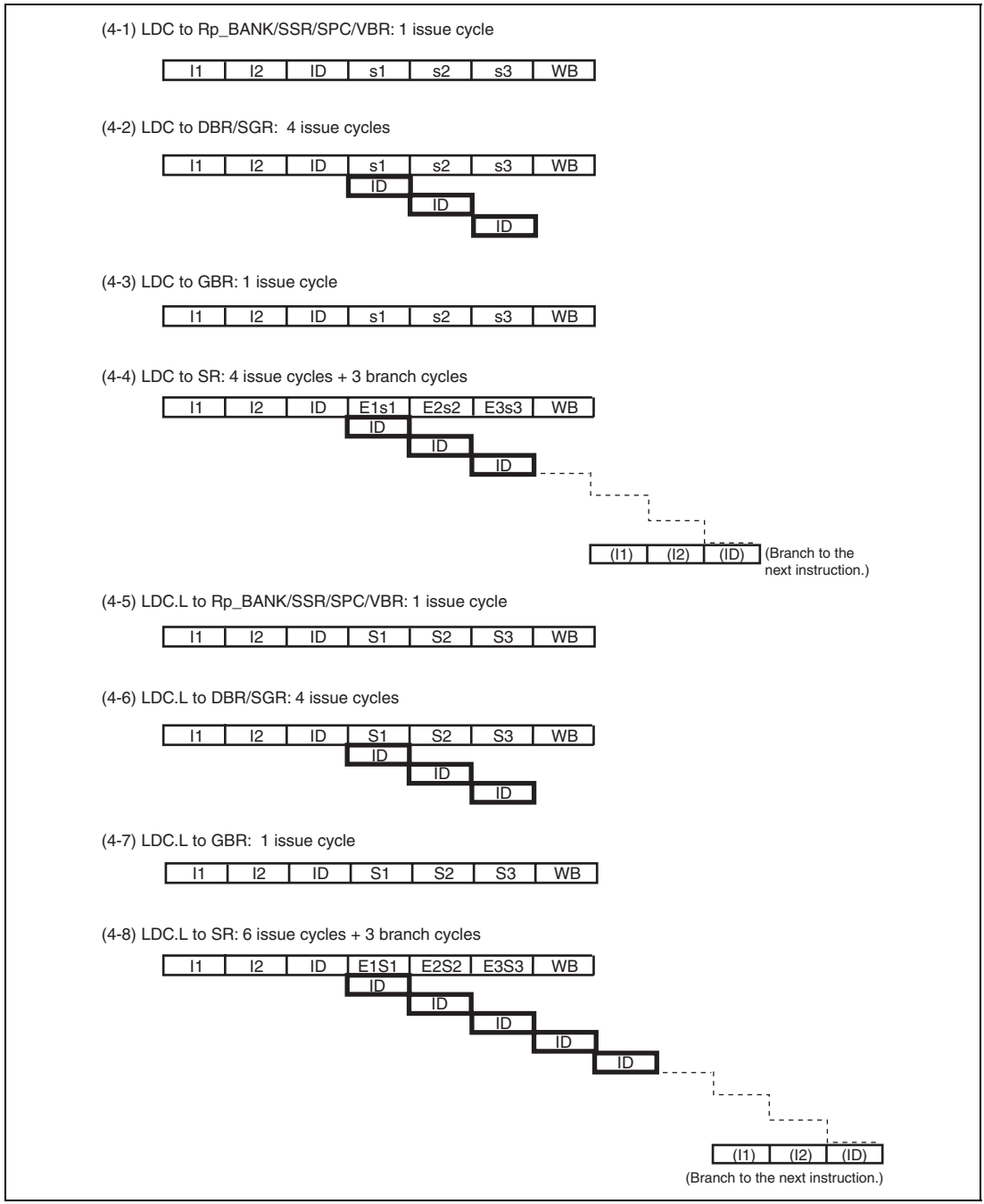


Figure 4.2 Instruction Execution Patterns (4)

(4-9) STC from DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR: 1 issue cycle

I1	I2	ID	s1	s2	s3	WB

(4-10) STC from SR: 1 issue cycle

I1	I2	ID	E1s1	E2s2	E3s3	WB

(4-11) STC.L from DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR: 1 issue cycle

I1	I2	ID	S1	S2	S3	WB

(4-12) STC.L from SR: 1 issue cycle

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)

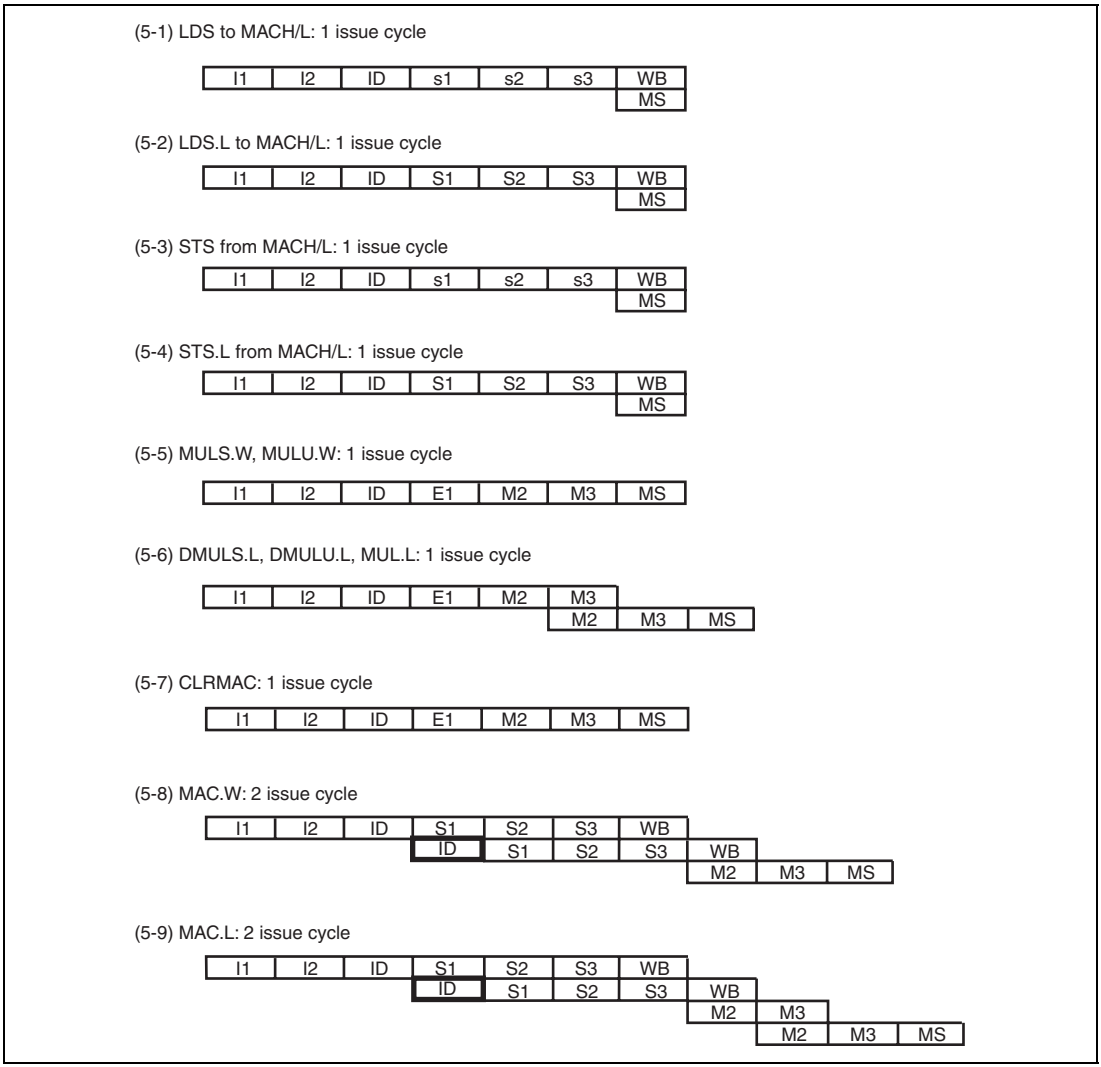
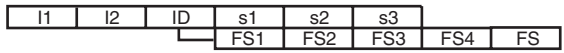
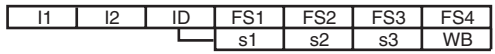


Figure 4.2 Instruction Execution Patterns (6)

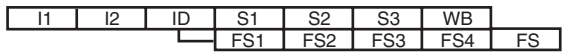
(6-1) LDS to FPUL: 1 issue cycle



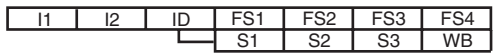
(6-2) STS from FPUL: 1 issue cycle



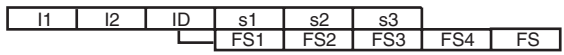
(6-3) LDS.L to FPUL: 1 issue cycle



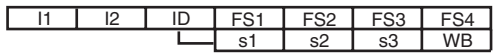
(6-4) STS.L from FPUL: 1 issue cycle



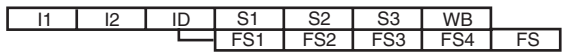
(6-5) LDS to FPSCR: 1 issue cycle



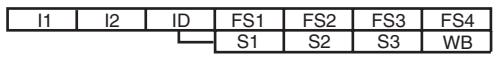
(6-6) STS from FPSCR: 1 issue cycle



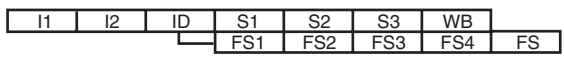
(6-7) LDS.L to FPSCR: 1 issue cycle



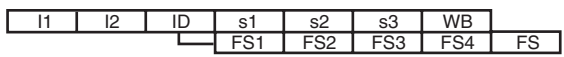
(6-8) STS.L from FPSCR: 1 issue cycle



(6-9) FPU load/store instruction FMOV: 1 issue cycle



(6-10) FLDS: 1 issue cycle



(6-11) FSTS: 1 issue cycle

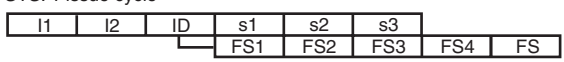
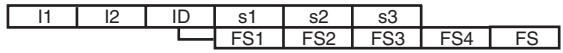
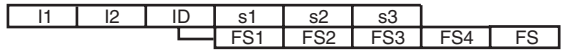


Figure 4.2 Instruction Execution Patterns (7)

(6-12) Single-precision FABS, FNEG/double-precision FABS, FNEG: 1 issue cycle

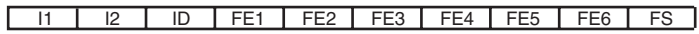


(6-13) FLDI0, FLDI1: 1 issue cycle

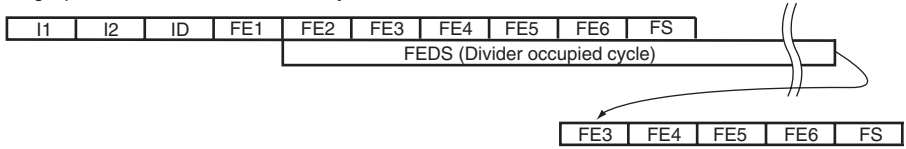


(6-14) Single-precision floating-point computation: 1 issue cycle

FCMP/EQ, FCMP/GT, FADD, FLOAT, FMAC, FMUL, FSUB, FTRC, FRCHG, FSCHG, FPCHG

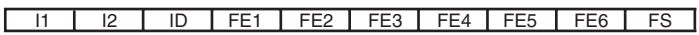


(6-15) Single-precision FDIV/FSQRT: 1 issue cycle



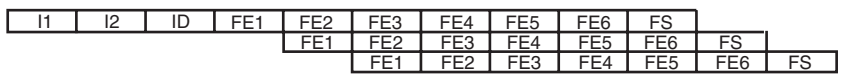
(6-16) Double-precision floating-point computation: 1 issue cycle

FCMP/EQ, FCMP/GT, FADD, FLOAT, FSUB, FTRC, FCNVSD, FCNVDS



(6-17) Double-precision floating-point computation: 1 issue cycle

FMUL



(6-18) Double-precision FDIV/FSQRT: 1 issue cycle

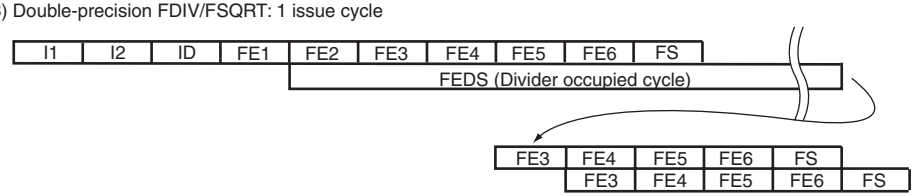


Figure 4.2 Instruction Execution Patterns (8)

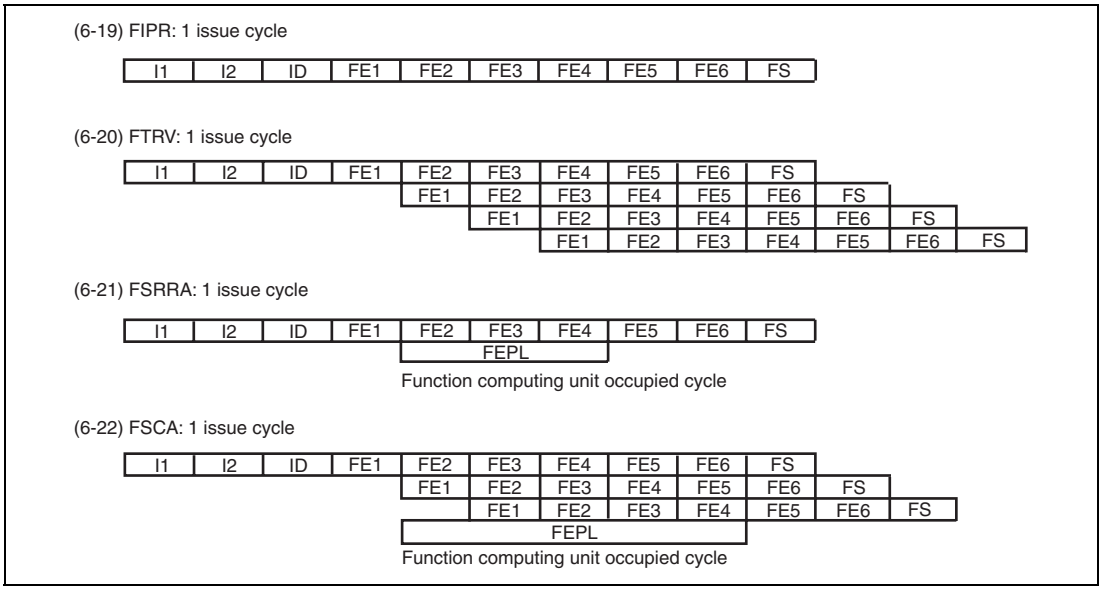


Figure 4.2 Instruction Execution Patterns (9)

4.2 Parallel-Executability

Instructions are categorized into six groups according to the internal function blocks used, as shown in table 4.2. Table 4.3 shows the parallel-executability of pairs of instructions in terms of groups. For example, ADD in the EX group and BRA in the BR group can be executed in parallel.

Table 4.2 Instruction Groups

Instruction Group	Instruction			
EX	ADD	DT	ROTL	SHLR8
	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,R0
	CLRT	NEG	SHLD	TST Rm,Rn
	CMP	NEGC	SHLL	XOR #imm,R0
	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,@-Rn
	FLDI0	LDC Rm,CR1	MOVA	STS SR2,Rn
	FLDI1	LDC.L @Rm+,CR1	MOVCA.L	STS.L SR2,@-Rn
	FLDS	LDS Rm,SR1	MOVUA	STS SR1,Rn
	FMOV @adr,FR	LDS Rm,SR2	OCBI	STS.L SR1,@-Rn
	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	

Instruction Group	Instruction			
FE	FADD	FDIV	FRCHG	FSCA
	FSUB	FIPR	FSCHG	FSRRA
	FCMP (S/D)	FLOAT	FSQRT	FPCHG
	FCNVDS	FMAC	FTRC	
	FCNVSD	FMUL	FTRV	
CO	AND.B #imm,@(R0,GBR)	LDC.L @Rm+,SR	PREFI	TRAPA
	ICBI	LDTLB	RTE	TST.B #imm,@(R0,GBR)
	LDC Rm,DBR	MAC.L	SLEEP	XOR.B #imm,@(R0,GBR)
	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 within the minimum page size (1 Kbyte).
2. The execution of these two instructions is supported in table 4.3, Combination of Preceding and Following Instructions.
3. Data used by an instruction of addr does not conflict with data used by a previous instruction
4. Data used by an instruction of addr+2 does not conflict with data used by a previous instruction
5. Both instructions are valid

Table 4.3 Combination of Preceding and Following Instructions

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

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

		Preceding Instruction (addr)							
		EX	**MT**	**BR**	**LS**	**FLSR**	**FLSM**	**FE**	**CO**
Following Instruction (addr+2)	EX	No	Yes	Yes	Yes	Yes	Yes	Yes	No
	MT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	BR	Yes	Yes	No	Yes	Yes	Yes	Yes	No
	LS	Yes	Yes	Yes	No	Yes	No	Yes	No
	FLSR	Yes	Yes	Yes	Yes	No	No*	No	No
	FLSM	Yes	Yes	Yes	No	No*	No	Yes	No
	FE	Yes	Yes	Yes	Yes	No	Yes	No	No
	CO	No	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+,SR2, 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 are stalled in the FPU.

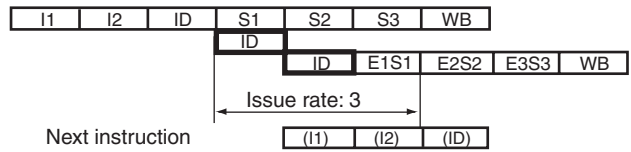
4.3 Issue Rates and Execution Cycles

Instruction execution cycles are summarized in table 4.4. Instruction Group in the table 4.4 corresponds to the category in the table 4.2. Penalty cycles due to a pipeline stall are not considered in the issue rates and execution cycles in this section.

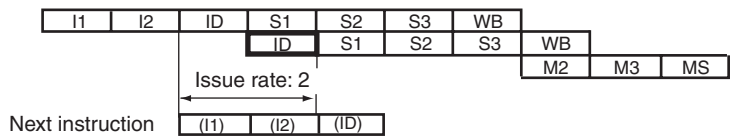
1. Issue Rate

Issue rates indicates the issue period between one instruction and next instruction.

E.g. AND.B instruction



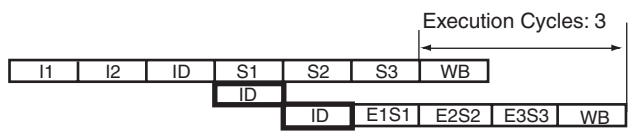
E.g. MAC.W instruction



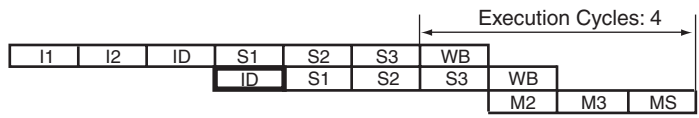
2. Execution Cycles

Execution cycles indicates the cycle counts an instruction occupied the pipeline based on the next rules.

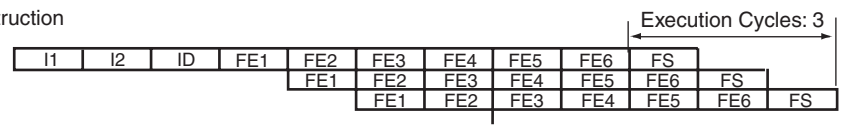
CPU instruction
E.g. AND.B instruction



E.g. MAC.W instruction



FPU instruction
E.g. FMUL instruction



E.g. FDIV instruction

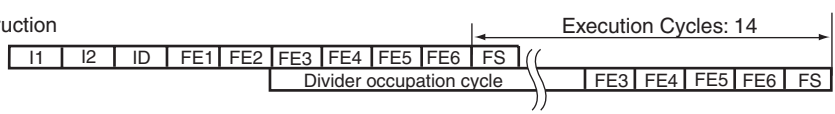


Table 4.4 Issue Rates and Execution Cycles

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Data transfer instructions	1	EXTS.B	Rm,Rn	EX	1	1	2-1
	2	EXTS.W	Rm,Rn	EX	1	1	2-1
	3	EXTU.B	Rm,Rn	EX	1	1	2-1
	4	EXTU.W	Rm,Rn	EX	1	1	2-1
	5	MOV	Rm,Rn	MT	1	1	2-4
	6	MOV	#imm,Rn	MT	1	1	2-3
	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

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Data transfer instructions	30	MOV.L	Rm,@-Rn	LS	1	1	3-1
	31	MOV.B	R0,@(disp,Rn)	LS	1	1	3-1
	32	MOV.W	R0,@(disp,Rn)	LS	1	1	3-1
	33	MOV.L	Rm,@(disp,Rn)	LS	1	1	3-1
	34	MOV.B	Rm,@(R0,Rn)	LS	1	1	3-1
	35	MOV.W	Rm,@(R0,Rn)	LS	1	1	3-1
	36	MOV.L	Rm,@(R0,Rn)	LS	1	1	3-1
	37	MOV.B	R0,@(disp,GBR)	LS	1	1	3-1
	38	MOV.W	R0,@(disp,GBR)	LS	1	1	3-1
	39	MOV.L	R0,@(disp,GBR)	LS	1	1	3-1
	40	MOVCA.L	R0,@Rn	LS	1	1	3-4
	41	MOVCO.L	R0,@Rn	CO	1	1	3-9
	42	MOVLI.L	@Rm,R0	CO	1	1	3-8
	43	MOVUA.L	@Rm,R0	LS	2	2	3-10
	44	MOVUA.L	@Rm+,R0	LS	2	2	3-10
	45	MOVT	Rn	EX	1	1	2-1
	46	OCBI	@Rn	LS	1	1	3-4
	47	OCBP	@Rn	LS	1	1	3-4
	48	OCBWB	@Rn	LS	1	1	3-4
	49	PREF	@Rn	LS	1	1	3-4
	50	SWAP.B	Rm,Rn	EX	1	1	2-1
	51	SWAP.W	Rm,Rn	EX	1	1	2-1
	52	XTRCT	Rm,Rn	EX	1	1	2-1
Fixed-point arithmetic instructions	53	ADD	Rm,Rn	EX	1	1	2-1
	54	ADD	#imm,Rn	EX	1	1	2-1
	55	ADDC	Rm,Rn	EX	1	1	2-1
	56	ADDV	Rm,Rn	EX	1	1	2-1
	57	CMP/EQ	#imm,R0	EX	1	1	2-1
	58	CMP/EQ	Rm,Rn	EX	1	1	2-1
	59	CMP/GE	Rm,Rn	EX	1	1	2-1
	60	CMP/GT	Rm,Rn	EX	1	1	2-1

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Fixed-point arithmetic instructions	61	CMP/HI	Rm,Rn	EX	1	1	2-1
	62	CMP/HS	Rm,Rn	EX	1	1	2-1
	63	CMP/PL	Rn	EX	1	1	2-1
	64	CMP/PZ	Rn	EX	1	1	2-1
	65	CMP/STR	Rm,Rn	EX	1	1	2-1
	66	DIV0S	Rm,Rn	EX	1	1	2-1
	67	DIV0U		EX	1	1	2-1
	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

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Logical instructions	92	TST.B	#imm,@(R0,GBR)	CO	3	3	3-2
	93	XOR	Rm,Rn	EX	1	1	2-1
	94	XOR	#imm,R0	EX	1	1	2-1
	95	XOR.B	#imm,@(R0,GBR)	CO	3	3	3-2
Shift instructions	96	ROTL	Rn	EX	1	1	2-1
	97	ROTR	Rn	EX	1	1	2-1
	98	ROTCL	Rn	EX	1	1	2-1
	99	ROTCR	Rn	EX	1	1	2-1
	100	SHAD	Rm,Rn	EX	1	1	2-1
	101	SHAL	Rn	EX	1	1	2-1
	102	SHAR	Rn	EX	1	1	2-1
	103	SHLD	Rm,Rn	EX	1	1	2-1
	104	SHLL	Rn	EX	1	1	2-1
	105	SHLL2	Rn	EX	1	1	2-1
	106	SHLL8	Rn	EX	1	1	2-1
	107	SHLL16	Rn	EX	1	1	2-1
	108	SHLR	Rn	EX	1	1	2-1
	109	SHLR2	Rn	EX	1	1	2-1
	110	SHLR8	Rn	EX	1	1	2-1
	111	SHLR16	Rn	EX	1	1	2-1
Branch instructions	112	BF	disp	BR	1+0 to 2	1	1-1
	113	BF/S	disp	BR	1+0 to 2	1	1-1
	114	BT	disp	BR	1+0 to 2	1	1-1
	115	BT/S	disp	BR	1+0 to 2	1	1-1
	116	BRA	disp	BR	1+0 to 2	1	1-1
	117	BRAF	Rm	BR	1+3	1	1-2
	118	BSR	disp	BR	1+0 to 2	1	1-1
	119	BSRF	Rm	BR	1+3	1	1-2
	120	JMP	@Rn	BR	1+3	1	1-2
	121	JSR	@Rn	BR	1+3	1	1-2
	122	RTS		BR	1+0 to 3	1	1-3

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
System control instructions	123	NOP		MT	1	1	2-3
	124	CLRMAC		EX	1	1	5-7
	125	CLRS		EX	1	1	2-1
	126	CLRT		EX	1	1	2-1
	127	ICBI	@Rn	CO	8+5+3	13	3-6
	128	SETS		EX	1	1	2-1
	129	SETT		EX	1	1	2-1
	130	PREFI		CO	5+5+3	10	3-7
	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

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
System control instructions	154	LDS	Rm,PR	LS	1	1	4-13
	155	LDS.L	@Rm+,MACH	LS	1	1	5-2
	156	LDS.L	@Rm+,MACL	LS	1	1	5-2
	157	LDS.L	@Rm+,PR	LS	1	1	4-14
	158	STC	DBR,Rn	LS	1	1	4-9
	159	STC	SGR,Rn	LS	1	1	4-9
	160	STC	GBR,Rn	LS	1	1	4-9
	161	STC	Rp_BANK,Rn	LS	1	1	4-9
	162	STC	SR,Rn	CO	1	1	4-10
	163	STC	SSR,Rn	LS	1	1	4-9
	164	STC	SPC,Rn	LS	1	1	4-9
	165	STC	VBR,Rn	LS	1	1	4-9
	166	STC.L	DBR,@-Rn	LS	1	1	4-11
	167	STC.L	SGR,@-Rn	LS	1	1	4-11
	168	STC.L	GBR,@-Rn	LS	1	1	4-11
	169	STC.L	Rp_BANK,@-Rn	LS	1	1	4-11
	170	STC.L	SR,@-Rn	CO	1	1	4-12
	171	STC.L	SSR,@-Rn	LS	1	1	4-11
	172	STC.L	SPC,@-Rn	LS	1	1	4-11
	173	STC.L	VBR,@-Rn	LS	1	1	4-11
	174	STS	MACH,Rn	LS	1	1	5-3
	175	STS	MACL,Rn	LS	1	1	5-3
	176	STS	PR,Rn	LS	1	1	4-15
	177	STS.L	MACH,@-Rn	LS	1	1	5-4
	178	STS.L	MACL,@-Rn	LS	1	1	5-4
	179	STS.L	PR,@-Rn	LS	1	1	4-16
Single-precision floating-point instructions	180	FLDI0	FRn	LS	1	1	6-13
	181	FLDI1	FRn	LS	1	1	6-13
	182	FMOV	FRm,FRn	LS	1	1	6-9
	183	FMOV.S	@Rm,FRn	LS	1	1	6-9
	184	FMOV.S	@Rm+,FRn	LS	1	1	6-9

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Single-precision floating-point instructions	185	FMOV.S	@(R0,Rm),FRn	LS	1	1	6-9
	186	FMOV.S	FRm,@Rn	LS	1	1	6-9
	187	FMOV.S	FRm,@-Rn	LS	1	1	6-9
	188	FMOV.S	FRm,@(R0,Rn)	LS	1	1	6-9
	189	FLDS	FRm,FPUL	LS	1	1	6-10
	190	FSTS	FPUL,FRn	LS	1	1	6-11
	191	FABS	FRn	LS	1	1	6-12
	192	FADD	FRm,FRn	FE	1	1	6-14
	193	FCMP/EQ	FRm,FRn	FE	1	1	6-14
	194	FCMP/GT	FRm,FRn	FE	1	1	6-14
	195	FDIV	FRm,FRn	FE	1	14	6-15
	196	FLOAT	FPUL,FRn	FE	1	1	6-14
	197	FMAC	FR0,FRm,FRn	FE	1	1	6-14
	198	FMUL	FRm,FRn	FE	1	1	6-14
	199	FNEG	FRn	LS	1	1	6-12
	200	FSQRT	FRn	FE	1	30	6-15
	201	FSUB	FRm,FRn	FE	1	1	6-14
	202	FTRC	FRm,FPUL	FE	1	1	6-14
	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	6-12
	211	FADD	DRm,DRn	FE	1	1	6-16
	212	FCMP/EQ	DRm,DRn	FE	1	1	6-16
	213	FCMP/GT	DRm,DRn	FE	1	1	6-16
	214	FCNVDS	DRm,FPUL	FE	1	1	6-16
	215	FCNVSD	FPUL,DRn	FE	1	1	6-16

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Double-precision floating-point instructions	216	FDIV	DRm,DRn	FE	1	14	6-18
	217	FLOAT	FPUL,DRn	FE	1	1	6-16
	218	FMUL	DRm,DRn	FE	1	3	6-17
	219	FNEG	DRn	LS	1	1	6-12
	220	FSQRT	DRn	FE	1	30	6-18
	221	FSUB	DRm,DRn	FE	1	1	6-16
	222	FTRC	DRm,FPUL	FE	1	1	6-16
FPU system control instructions	223	LDS	Rm,FPUL	LS	1	1	6-1
	224	LDS	Rm,FPSCR	LS	1	1	6-5
	225	LDS.L	@Rm+,FPUL	LS	1	1	6-3
	226	LDS.L	@Rm+,FPSCR	LS	1	1	6-7
	227	STS	FPUL,Rn	LS	1	1	6-2
	228	STS	FPSCR,Rn	LS	1	1	6-6
	229	STS.L	FPUL,@-Rn	LS	1	1	6-4
	230	STS.L	FPSCR,@-Rn	LS	1	1	6-8
Graphics acceleration instructions	231	FMOV	DRm,XDn	LS	1	1	6-9
	232	FMOV	XDm,DRn	LS	1	1	6-9
	233	FMOV	XDm,XDn	LS	1	1	6-9
	234	FMOV	@Rm,XDn	LS	1	1	6-9
	235	FMOV	@Rm+,XDn	LS	1	1	6-9
	236	FMOV	@(R0,Rm),XDn	LS	1	1	6-9
	237	FMOV	XDm,@Rn	LS	1	1	6-9
	238	FMOV	XDm,@-Rn	LS	1	1	6-9
	239	FMOV	XDm,@(R0,Rn)	LS	1	1	6-9
	240	FIPR	FVm,FVn	FE	1	1	6-19
	241	FRCHG		FE	1	1	6-14
	242	FSCHG		FE	1	1	6-14
	243	FPCHG		FE	1	1	6-14
	244	FSRRA	FRn	FE	1	1	6-21
	245	FSCA	FPUL,DRn	FE	1	3	6-22
	246	FTRV	XMTRX,FVn	FE	1	4	6-20

Section 5 Exception Handling

5.1 Summary of Exception Handling

Exception handling processing is handled by a special routine which is executed by a reset, general exception handling, or interrupt. For example, if the executing instruction ends abnormally, appropriate action must be taken in order to return to the original program sequence, or report the abnormality before terminating the processing. The process of generating an exception handling request in response to abnormal termination, and passing control to a user-written exception handling routine, in order to support such functions, is given the generic name of exception handling.

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

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*	Access Size
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32

Note: * P4 is the address when virtual address space P4 area is used. Area 7 is the address 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	Sleep
TRAPA exception register	TRA	Undefined	Undefined	Retained
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Retained
Interrupt event register	INTEVT	Undefined	Undefined	Retained

5.2.1 TRAPA Exception Register (TRA)

The TRAPA exception register (TRA) consists of 8-bit immediate data (imm) for the TRAPA instruction. TRA is set automatically by hardware when a TRAPA instruction is executed. TRA can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

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

5.2.2 Exception Event Register (EXPEVT)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	EXPCODE											
Initial value:	0	0	0	0	0	0	0	0	0	0	0/1	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	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 value should always be 0.
11 to 0	EXPCODE	H'000 or H'020	R/W	Exception Code The exception code for a reset or general exception is set. For details, see table 5.3.

5.2.3 Interrupt Event Register (INTEVT)

The interrupt event register (INTEVT) consists of a 14-bit exception code. The exception code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

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

5.3 Exception Handling Functions

5.3.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC), status register (SR), and R15 are saved in the saved program counter (SPC), saved status register (SSR), and saved general register15 (SGR), and the CPU starts execution of the appropriate exception handling routine according to the vector address. An exception handling routine is a program written by the user to handle a specific exception. The exception handling routine is terminated and control returned to the original program by executing a return-from-exception instruction (RTE). This instruction restores the PC and SR contents and returns control to the normal processing routine at the point at which the exception occurred. The SGR contents are not written back to R15 with an RTE 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 (EXPEVT) or interrupt event register (INTEVT).
7. The CPU branches to the determined exception handling vector address, and the exception handling routine begins.

5.3.2 Exception Handling Vector Addresses

The reset vector address is fixed at H'A0000000. Exception and interrupt vector addresses are determined by adding the offset for the specific event to the vector base address, which is set by software in the vector base register (VBR). In the case of the TLB miss exception, for example, the offset is H'00000400, so if H'9C080000 is set in VBR, the exception handling vector address will be H'9C080400. If a further exception occurs at the exception handling vector address, a duplicate exception will result, and recovery will be difficult; therefore, addresses that are not to be converted (in P1 and P2 areas) should be specified for vector addresses.

5.4 Exception Types and Priorities

Table 5.3 shows the types of exceptions, with their relative priorities, vector addresses, and exception/interrupt codes.

Table 5.3 Exceptions

Exception Category	Execution Mode	Exception	Priority Level*2	Priority Order*2	Exception Transition Direction*3 Vector Address	Exception Transition Direction*3 Offset	Exception Code*4
Reset	Abort type	Power-on reset	1	1	H'A000 0000	—	H'000
		Manual reset	1	2	H'A000 0000	—	H'020
		H-UDI reset	1	1	H'A000 0000	—	H'000
		Instruction TLB multiple-hit exception	1	3	H'A000 0000	—	H'140
		Data TLB multiple-hit exception	1	4	H'A000 0000	—	H'140
General exception	Re-execution type	User break before instruction execution*1	2	0	(VBR/DBR)	H'100/—	H'1E0
		Instruction address error	2	1	(VBR)	H'100	H'0E0
		Instruction TLB miss exception	2	2	(VBR)	H'400	H'040
		Instruction TLB protection violation exception	2	3	(VBR)	H'100	H'0A0
		General illegal instruction exception	2	4	(VBR)	H'100	H'180
		Slot illegal instruction exception	2	4	(VBR)	H'100	H'1A0
		General FPU disable exception	2	4	(VBR)	H'100	H'800
		Slot FPU disable exception	2	4	(VBR)	H'100	H'820
		Data address error (read)	2	5	(VBR)	H'100	H'0E0
		Data address error (write)	2	5	(VBR)	H'100	H'100
		Data TLB miss exception (read)	2	6	(VBR)	H'400	H'040
		Data TLB miss exception (write)	2	6	(VBR)	H'400	H'060
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100	H'0A0
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100	H'0C0
		FPU exception	2	8	(VBR)	H'100	H'120
		Initial page write exception	2	9	(VBR)	H'100	H'080

Exception Category	Execution Mode	Exception	Priority Level*2	Priority Order*2	Exception Transition Direction*3		Exception Code*4
					Vector Address	Offset	
General exception	Completion type	Unconditional trap (TRAPA)	2	4	(VBR)	H'100	H'160
		User break after instruction execution*1	2	10	(VBR/DBR)	H'100/—	H'1E0
Interrupt	Completion type	Nonmaskable interrupt	3	—	(VBR)	H'600	H'1C0
		General interrupt request	4	—	(VBR)	H'600	—

Notes: 1. When UBDE in CBCR = 1, PC = DBR. In other cases, PC = VBR + H'100.
2. Priority is first assigned by priority level, then by priority order within each level (the lowest number represents the highest priority).
3. Control passes to H'A000 0000 in a reset, and to [VBR + offset] in other cases.
4. Stored in EXPEVT for a reset or general exception, and in INTEVT for an interrupt.

5.5 Exception Flow

5.5.1 Exception Flow

Figure 5.1 shows an outline flowchart of the basic operations in instruction execution and exception handling. For the sake of clarity, the following description assumes that instructions are executed sequentially, one by one. Figure 5.1 shows the relative priority order of the different kinds of exceptions (reset, general exception, and interrupt). Register settings in the event of an exception are shown only for SSR, SPC, SGR, EXPEVT/INTEVT, SR, and PC. However, other registers may be set automatically by hardware, depending on the exception. For details, see section 5.6, Description of Exceptions. Also, see section 5.6.4, Priority Order with Multiple Exceptions, for exception handling during execution of a delayed branch instruction and a delay slot instruction, or in the case of instructions in which two data accesses are performed.

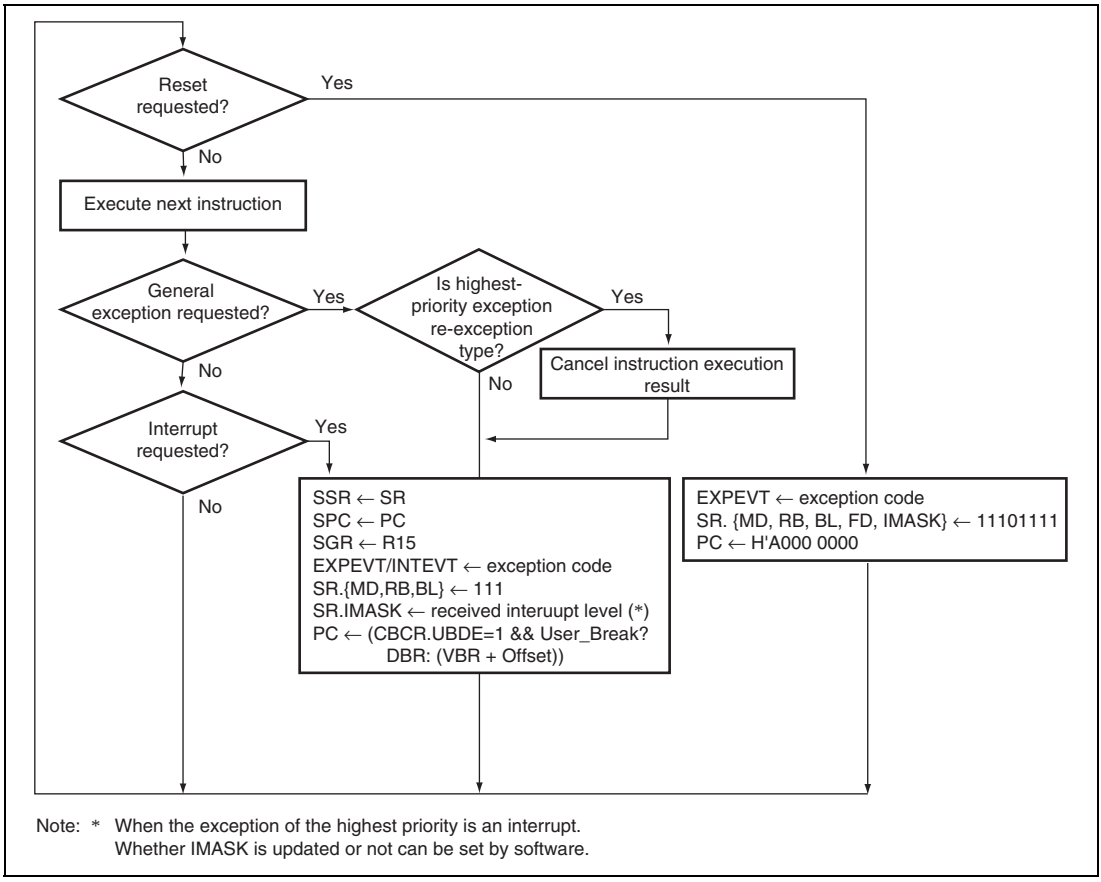


Figure 5.1 Instruction Execution and Exception Handling

5.5.2 Exception Source Acceptance

A priority ranking is provided for all exceptions for use in determining which of two or more simultaneously generated exceptions should be accepted. Five of the general exceptions—general illegal instruction exception, slot illegal instruction exception, general FPU disable exception, slot FPU disable exception, and unconditional trap exception—are detected in the process of instruction decoding, and do not occur simultaneously in the instruction pipeline. These exceptions therefore all have the same priority. General exceptions are detected in the order of instruction execution. However, exception handling is performed in the order of instruction flow (program order). Thus, an exception for an earlier instruction is accepted before that for a later instruction. An example of the order of acceptance for general exceptions is shown in figure 5.2.

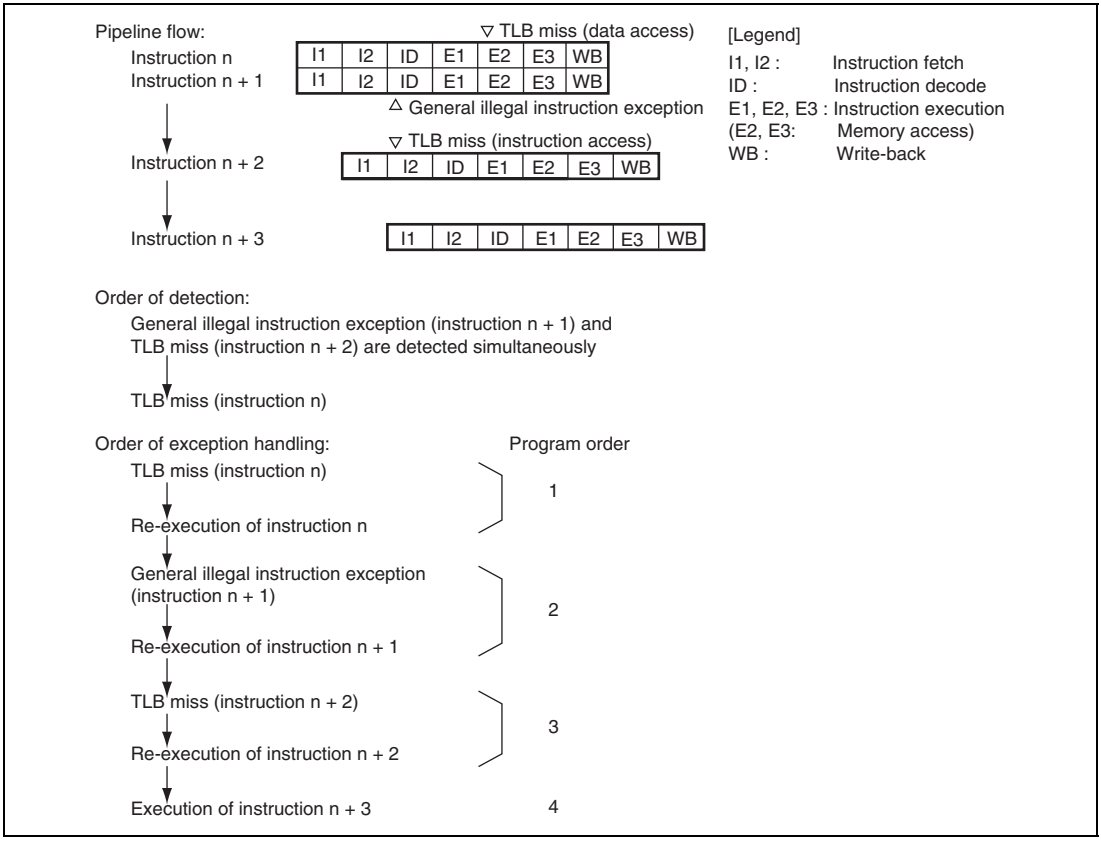


Figure 5.2 Example of General Exception Acceptance Order

5.5.3 Exception Requests and BL Bit

When the BL bit in SR is 0, exceptions and interrupts are accepted.

When the BL bit in SR is 1 and an exception other than a user break is generated, the CPU's internal registers and the registers of the other modules are set to their states following a manual reset, and the CPU branches to the same address as in a reset (H'A0000000). For the operation in the event of a user break, see section 29, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

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

5.5.4 Return from Exception Handling

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

5.6 Description of Exceptions

The various exception handling operations explained here are exception sources, transition address on the occurrence of exception, and processor operation when a transition is made.

5.6.1 Resets

Power-On Reset:

- Condition:
  Power-on reset request
- Operations:
  Exception code H'000 is set in EXPEVT, initialization of the CPU and on-chip peripheral module is carried out, and then a branch is made to the reset vector (H'A0000000). For details, see the register descriptions in the relevant sections. A power-on reset should be executed 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 peripheral module is carried out, and then a branch is made to the branch vector (H'A0000000). The registers initialized by a power-on reset and manual reset are different. For details, see the 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, and a branch is made to PC = H'A0000000.
  CPU and on-chip peripheral module initialization is performed. For details, see the register descriptions in the relevant sections.

Instruction TLB Multiple-Hit Exception:

- Source: Multiple ITLB 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 indicates the ASID when this exception occurred.

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

  CPU and on-chip peripheral module initialization is performed in the same way as in a manual 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 indicates the ASID when this exception occurred.

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

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

5.6.2 General Exceptions

Data TLB Miss Exception:

- Source: Address mismatch in UTLB address comparison
- Transition address: VBR + H'00000400
- 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 indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC 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 EXPEVT. The 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;
}
```

Instruction TLB Miss Exception:

- Source: Address mismatch in ITLB address comparison
- Transition address: VBR + H'00000400
- 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 indicates the ASID when this exception occurred.

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

  Exception code H'40 is set in EXPEVT. The 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.

```
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;
}
```

Initial Page Write Exception:

- Source: TLB is hit in a store access, but dirty bit D = 0
- 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 indicates the ASID when this exception occurred.

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

  Exception code H'080 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a 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;
}
```

Data TLB Protection Violation Exception:

- Source: The access does not accord with the UTLB protection information (PR bits) shown below.

PR	Privileged Mode	User Mode
00	Only read access possible	Access not possible
01	Read/write access possible	Access not possible
10	Only read access possible	Only read access possible
11	Read/write access possible	Read/write access possible

- 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 indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC 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 EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
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;
}
```

Instruction TLB Protection Violation Exception:

- Source: The access does not accord with the ITLB protection information (PR bits) shown below.

PR	Privileged Mode	User Mode
0	Access possible	Access not possible
1	Access possible	Access possible

- 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 indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC 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, and a 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;
}
```

Data Address Error:

- Sources:
  - — Word data access from other than a word boundary (2n +1)
  - — Longword data access from other than a longword data boundary (4n +1, 4n + 2, or 4n +3)
  - — Quadword data access from other than a quadword data boundary (8n +1, 8n + 2, 8n +3, 8n + 4, 8n + 5, 8n + 6, or 8n + 7)
  - — Access to area H'80000000 to H'FFFFFFFF in user mode

    Areas H'E0000000 to H'E3FFFFFF and H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 7, Memory Management Unit (MMU) and section 9, L 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 indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC 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 EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For 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;
}
```

Instruction Address Error:

- Sources:
  - Instruction fetch from other than a word boundary (2n +1)
  - Instruction fetch from area H'80000000 to H'FFFFFFFF in user mode

    Area H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 9, L Memory.
- 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 indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in the 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, and a branch is made to PC = VBR + H'0100. For details, see section 7, Memory Management Unit (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;
}
```

Unconditional Trap:

- Source: Execution of TRAPA instruction
- Transition address: VBR + H'00000100
- Transition operations:

  As this is a processing-completion-type exception, the PC contents for the instruction following the TRAPA instruction are saved in SPC. The value of SR and R15 when the TRAPA instruction is executed are saved in SSR and SGR. The 8-bit immediate value in the TRAPA instruction is multiplied by 4, and the result is set in TRA [9:0]. Exception code H'160 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'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;
}
```

General Illegal Instruction Exception:

- Sources:
  - Decoding of an undefined instruction not in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - Decoding in user mode of a privileged instruction not in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
- Transition address: VBR + H'00000100
- Transition operations:
  The PC and SR contents for the instruction at which this exception occurred are saved in SPC 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, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other 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;
}
```

Slot Illegal Instruction Exception:

- Sources:
  - — Decoding of an undefined instruction in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - — Decoding of an instruction that modifies PC in a delay slot
    Instructions that modify PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm,SR, LDC.L @Rm+,SR, ICBI, PREFI
  - — Decoding in user mode of a privileged instruction in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC 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 SR and 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 SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other 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;
}
```

General FPU Disable Exception:

- Source: Decoding of an FPU instruction* not in a delay slot with SR.FD =1
- Transition address: VBR + H'00000100
- Transition operations:

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

  Exception code H'800 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

Note: * FPU instructions are instructions in which the first 4 bits of the instruction code are F (but excluding undefined instruction H'FFFD), and the LDS, STS, LDS.L, and STS.L 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;
}
```

Slot FPU Disable Exception:

- Source: Decoding of an FPU instruction in a delay slot with SR.FD =1
- Transition address: VBR + H'00000100
- Transition operations:

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

  Exception code H'820 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Slot_fpu_disable_exception()
{
    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;
}
```

Pre-Execution User Break/Post-Execution User Break:

- Source: Fulfilling of a break condition set in the user break controller
- Transition address: VBR + H'00000100, or DBR
- Transition operations:

  In the case of a post-execution break, the PC contents for the instruction following the instruction at which the breakpoint is set are set in SPC. In the case of a pre-execution break, the PC contents for the instruction at which the breakpoint is set are set in SPC.

  The SR and R15 contents when the break occurred are saved in SSR and SGR. Exception code H'1E0 is set in EXPEVT.

  The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. It is also possible to branch to PC = DBR.

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

```
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);
}
```

FPU Exception:

- Source: Exception due to execution of a floating-point operation
- Transition address: VBR + H'00000100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR . The R15 contents at this time are saved in SGR. Exception code H'120 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

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

5.6.3 Interrupts

NMI (Nonmaskable Interrupt):

- Source: NMI pin edge detection
- Transition address: VBR + H'00000600
- Transition operations:

  The PC and SR contents for the instruction immediately after this exception is accepted are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'1C0 is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0600. When the BL bit in SR is 0, this interrupt is not masked by the interrupt mask bits in SR, and is accepted at the highest priority level. When the BL bit in SR is 1, a software setting can specify whether this interrupt is to be masked or 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;
}
```

General Interrupt Request:

- Source: The interrupt mask level bits setting in SR is smaller than the interrupt level of interrupt request, and the BL bit in SR is 0 (accepted at instruction boundary).
- Transition address: VBR + H'00000600
- Transition operations:

  The PC contents immediately after the instruction at which the interrupt is accepted are set in SPC. The SR and R15 contents at the time of acceptance are set in SSR and SGR.

  The code corresponding to the each interrupt source is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'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, multiple exceptions occur. Care is required in these cases, as the exception priority order differs from the normal order.

- Instructions that make two accesses to memory

  With MAC instructions, memory-to-memory arithmetic/logic instructions, TAS instructions, and MOVUA instructions, two data transfers are performed by a single instruction, and an exception will be detected for each of these data transfers. In these cases, therefore, the following order is used to determine priority.

  1. Data address error in first data transfer
  2. TLB miss in first data transfer
  3. TLB protection violation in first data transfer
  4. Initial page write exception in first data transfer
  5. Data address error in second data transfer
  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, they are treated as a single instruction. Consequently, the priority order for exceptions that occur in these instructions differs from the usual priority order. The priority order shown below is for 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 priority levels 1 and 2 in the delayed branch instruction.
  2. A check is performed for the interrupt type and re-execution type exceptions of priority 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 delayed branch instruction.
  4. A check is performed for the completion type exception of priority level 2 in the delay slot instruction.
  5. A check is performed for priority level 3 in the delayed branch instruction and priority level 3 in the delay slot instruction. (There is no priority ranking between these two.)
  6. A check is performed for priority level 4 in the delayed branch instruction and priority level 4 in the delay slot instruction. (There is no priority ranking between these two.)

  If the delay slot instruction has a second data transfer, two checks are performed in step 2, as in 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 → PR operation performed in a BSR, BSRF, or JSR instruction) is not disabled. Note that in this case, the contents of PR register are not guaranteed.

5.7 Usage Notes

1. Return from exception handling
   A. Check the BL bit in SR with software. If SPC and SSR have been saved to memory, set the BL bit in SR to 1 before restoring them.
   B. Issue an RTE instruction. When RTE is executed, the SPC contents are saved in PC, the SSR contents are saved in SR, and branch is made to the SPC address to return from the exception handling routine.
2. If an exception or interrupt occurs when BL bit in SR = 1
   A. Exception

      When an exception other than a user break occurs, a manual reset is executed. The value in 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 accepted after the BL bit in SR has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

      In sleep mode, however, an interrupt is accepted even if the BL bit in SR is set to 1.
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, and the instruction is re-executed after returning from the exception handling routine. If an exception occurs in a delay slot instruction, however, the PC value for the delayed branch instruction is saved in SPC regardless of whether or not the preceding delay slot instruction condition is satisfied.
   B. Completion type exception or interrupt

      The PC value for the instruction following that at which the exception occurred is set in SPC. If an exception occurs in a branch instruction with delay slot, however, the PC value 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 value saved in SSR has been restored to SR. The acceptance of the exception related to the instruction access is determined depending on SR before restoring, while the acceptance of other exceptions is determined depending on the processing mode by SR after restoring or the BL bit. The completion type exception is accepted before branching to the destination of RTE instruction. However, if the re-execution type exception is occurred, the operation cannot be guaranteed.
   B. The user break is not accepted by the instruction in the delay slot of the RTE instruction.

5. Changing the SR register value and accepting exception
   A. When the MD or BL bit in the SR register is changed by the LDC instruction, the acceptance of the exception is determined by the changed SR value, starting from the next instruction.* In the completion type exception, an exception is accepted after the next instruction has been executed. However, an interrupt of completion type exception is accepted before the next instruction is executed.

Note: * When the LDC instruction for SR is executed, following instructions are fetched again and the instruction fetch exception is evaluated again by the changed SR.

Section 6 Floating-Point Unit (FPU)

6.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 32 single-precision floating-point registers (can also be referenced as 16 double-precision registers)
- Two rounding modes: Round to Nearest and Round to Zero
- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Underflow, and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, and system 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 FPU instruction will cause an FPU disable exception (general FPU disable exception or slot FPU disable exception).

6.2 Data Formats

6.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign bit (s)
- Exponent field (e)
- Fraction field (f)

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

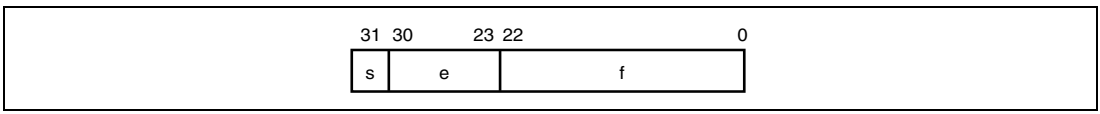


Figure 6.1 Format of Single-Precision Floating-Point Number

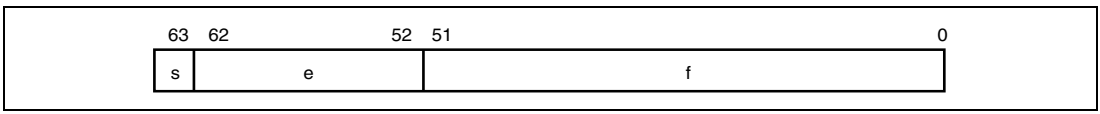


Figure 6.2 Format of Double-Precision Floating-Point Number

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 $E_{max} + 1$ are 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-number (NaN). Table 6.1 shows floating-point formats and parameters.

Table 6.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
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 signaling non-number and quiet non-number, see section 6.2.2, Non-Numbers (NaN). For the denormalized number, see section 6.2.3, Denormalized Numbers.

Table 6.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to 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 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
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 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

6.2.2 Non-Numbers (NaN)

Figure 6.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

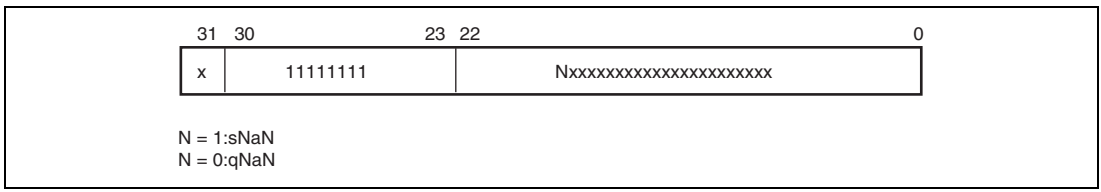


Figure 6.3 Single-Precision NaN Bit Pattern

An sNaN is assumed to be the input data in an operation, except the transfer instructions between 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. In this 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 not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit 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

See SH-4A Software Manual for details of floating-point operations when a non-number (NaN) is input.

6.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

When the DN bit in FPSCR of the FPU is 1, a denormalized number (source operand or operation result) is always positive or negative zero in a floating-point operation that generates a value (an operation other than transfer instructions between registers, FNEG, or FABS).

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

6.3 Register Descriptions

6.3.1 Floating-Point Registers

Figure 6.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers comprised with two banks: FPR0_BANK0 to FPR15_BANK0, and FPR0_BANK1 to FPR15_BANK1. These thirty-two registers are referenced as FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, and XMTRX. Corresponding registers to FPR0_BANK0 to FPR15_BANK0, and FPR0_BANK1 to FPR15_BANK1 are determined according to the FR bit of FPSCR.

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_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are allocated to FPR0_BANK1 to FPR15_BANK1.

3. Double-precision floating-point registers, DRi (8 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, FR15}

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises 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_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are allocated to FPR0_BANK0 to FPR15_BANK0.

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 = {XF14, XF15}

7. Single-precision floating-point extended register matrix, XMTRX: XMTRX comprises all 16 XF registers.

$$XMTRX = \begin{bmatrix} XF0 & XF4 & XF8 & XF12 \\ XF1 & XF5 & XF9 & XF13 \\ XF2 & XF6 & XF10 & XF14 \\ XF3 & XF7 & XF11 & XF15 \end{bmatrix}$$

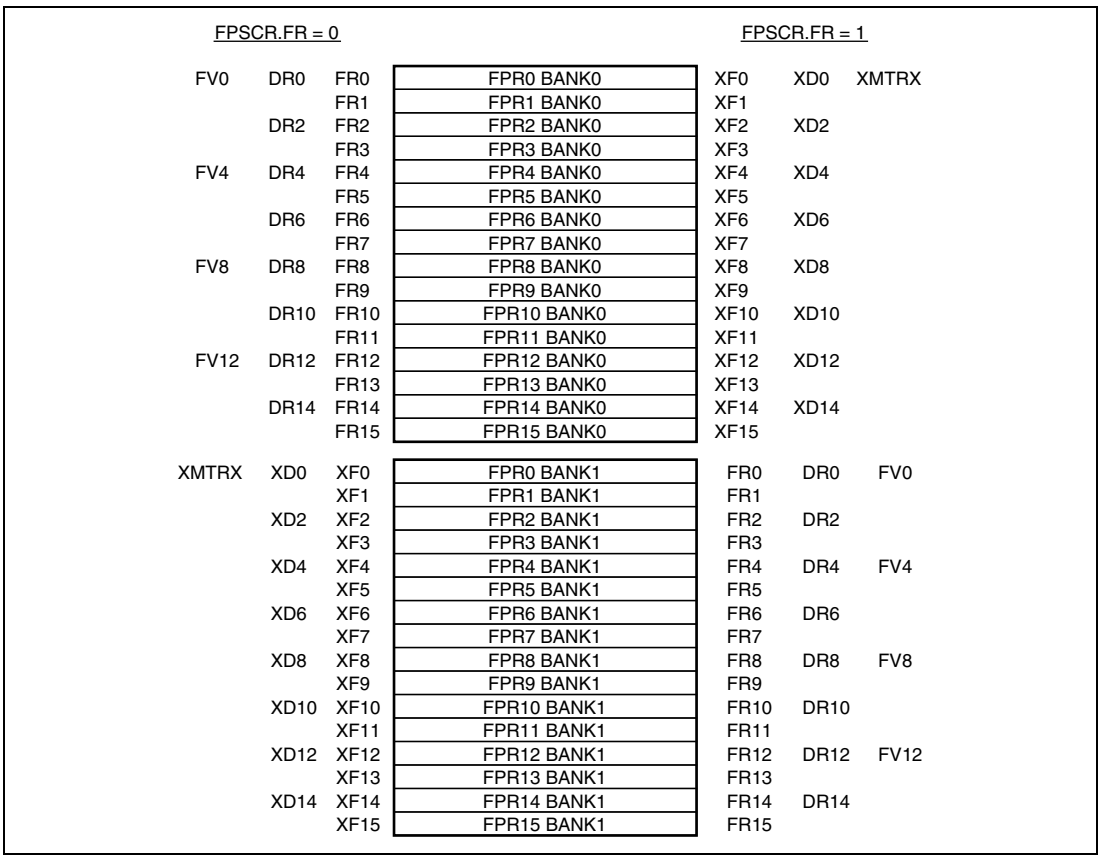


Figure 6.4 Floating-Point Registers

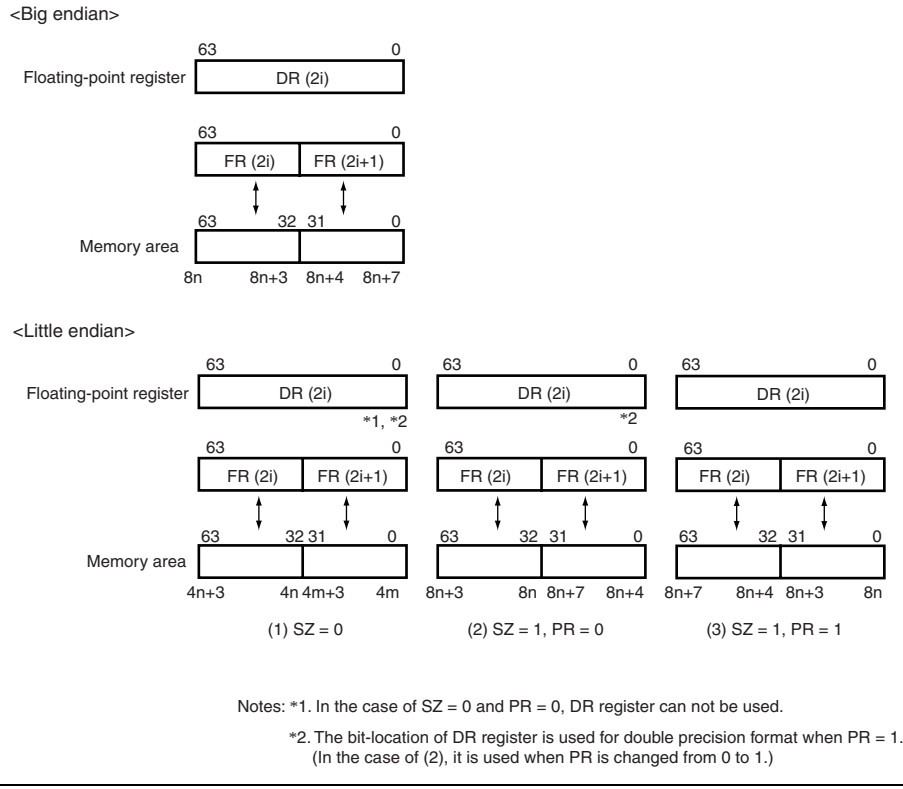
6.3.2 Floating-Point Status/Control Register (FPSCR)

bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	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	1	0
	Cause				Enable (EN)					Flag					RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 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 register pair (64 bits) For relations between endian and the SZ and PR bits, see figure 6.5.
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 support instructions are undefined) For relations between endian and the SZ and PR bits, see figure 6.5.
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field FPU Exception Enable Field FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU exception occurs, the bits corresponding to FPU exception cause field and flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. For bit allocations of each field, see table 6.3.
11 to 7	Enable	All 0	R/W	
6 to 2	Flag	All 0	R/W	
1	RM1	0	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved
0	RM0	1	R/W	



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 6.5 Relation between SZ Bit and Endian

Table 6.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

6.3.3 Floating-Point Communication Register (FPUL)

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

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

6.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC, FTRV, and FIPR will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

Round to Nearest: The operation result is rounded to the nearest expressible value. If there are 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 sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 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 will become the maximum expressible absolute value with the same sign as unrounded value.

6.5 Floating-Point Exceptions

6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions

FPU-related exceptions are occurred when an FPU instruction is executed with SR.FD set to 1. When the FPU instruction is in other than delayed slot, the general FPU disable exception is occurred. When the FPU instruction is in the delay slot, the slot FPU disable exception is occurred.

6.5.2 FPU Exception Sources

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 sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding 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 set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the 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 operation)
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor or the input of FSRRA is zero
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow

- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

All exception events that originate in the FPU are assigned as the same exception event. The meaning of an exception is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed by any FPU exception handling operation.

If the FPU exception sources except for above are generated, the bit corresponding to source V, Z, 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 as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  When FPSCR.DN = 0, a denormalized number with the same sign as the unrounded value, or 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.

6.6 Graphics Support Functions

The SH-4A supports two kinds of graphics functions: new instructions for geometric operations, and pair single-precision transfer instructions that enable high-speed data transfer.

6.6.1 Geometric Operation Instructions

Geometric operation instructions perform approximate-value computations. To enable high-speed computation with a minimum of hardware, the SH-4A ignores comparatively small values in the partial computation results of four multiplications. Consequently, the error shown below is produced in the result of the computation:

$$\text{Maximum error} = \text{MAX (individual multiplication result} \times 2^{-\text{MIN (number of multiplier significant digits}-1,\ \text{number of multiplicand significant digits}-1)}) + \text{MAX (result value} \times 2^{-23},\ 2^{-149})$$

The number of significant digits is 24 for a normalized number and 23 for a denormalized number (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 between different processor cores is not guaranteed.

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

- Inner product ($m \neq n$):
  This operation is generally used for surface/rear surface determination for polygon surfaces.
- 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 (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FIPR instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed.

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

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

- Matrix (4 × 4) × matrix (4 × 4):
  This operation requires the execution of four FTRV instructions.

Since an inexact exception is not detected by an FIRV instruction, the inexact exception (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FTRV instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed. It is not possible to check all data types in the registers beforehand when executing an FTRV instruction. If the V bit is set in the FPU exception enable field, FPU exception handling will be executed.

FRCHG: This instruction modifies banked registers. For example, when the FTRV instruction is executed, matrix elements must be set in an array in the background bank. However, to create the actual elements of a translation matrix, it is easier to use registers in the foreground bank. When the LDS instruction is used on FPSCR, this instruction takes four to five cycles in order to maintain the FPU state. With the FRCHG instruction, the FR bit in FPSCR can be changed in one cycle.

6.6.2 Pair Single-Precision Data Transfer

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

When the SZ bit is 1, the SH-4A can perform data transfer by means of pair single-precision data 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; that is, the transfer performance of these instructions is doubled.

- FSCHG

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

Section 7 Memory Management Unit (MMU)

This LSI supports an 8-bit address space identifier, a 32-bit virtual address space, and a 29-bit physical address space. Address translation from virtual addresses to physical addresses is enabled by the memory management unit (MMU) in this LSI. The MMU performs high-speed address translation by caching user-created address translation table information in an address translation buffer (translation lookaside buffer: TLB).

This LSI has four instruction TLB (ITLB) entries and 64 unified TLB (UTLB) entries. UTLB copies are stored in the ITLB by hardware. A paging system is used for address translation, with four page sizes (1, 4, and 64 Kbytes, and 1 Mbyte) supported. It is possible to set the virtual address space access right and implement memory protection independently for privileged mode and user mode.

7.1 Overview of MMU

The MMU was conceived as a means of making efficient use of physical memory. As shown in (0) in figure 7.1, when a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory, but if the process increases in size to the point where it does not fit into physical memory, it becomes necessary to divide the process into smaller parts, and map the parts requiring execution onto physical memory as occasion arises ((1) in figure 7.1). Having this mapping onto physical memory executed consciously by the process itself imposes a heavy burden on the process. The virtual memory system was devised as a means of handling all physical memory mapping to reduce this burden ((2) in figure 7.1). With a virtual memory system, the size of the available virtual memory is much larger than the actual physical memory, and processes are mapped onto this virtual memory. Thus processes only have to consider their operation in virtual memory, and mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally managed by the OS, and physical memory switching is carried out so as to enable the virtual memory required by a process to be mapped smoothly onto physical 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 time sharing system (TSS) that allows a number of processes to run simultaneously ((3) in figure 7.1). Running a number of processes in a TSS did not increase efficiency since each process had to take account of physical memory mapping. Efficiency is improved and the load on each process reduced by the use of a virtual memory system ((4) in figure 7.1). In this virtual memory system, virtual memory is allocated to each process. The task of the MMU is to map a number of virtual memory areas onto physical memory in an efficient manner. It is also provided with memory protection functions to prevent a process from inadvertently accessing another process's physical memory.

When address translation from virtual memory to physical memory is performed using the MMU, it may happen that the translation information has not been recorded in the MMU, or the virtual memory of a different process is accessed by mistake. In such cases, the MMU will generate an exception, change the physical memory mapping, and record the new address translation information.

Although the functions of the MMU could be implemented by software alone, having address translation performed by software each time a process accessed physical memory would be very inefficient. For this reason, a buffer for address translation (the translation lookaside buffer: TLB) is provided by hardware, and frequently used address translation information is placed here. The TLB can be described as a cache for address translation information. However, unlike a cache, if address translation fails—that is, if an exception occurs—switching of the address translation information is normally performed by software. Thus memory management can be performed in a flexible manner by software.

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

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

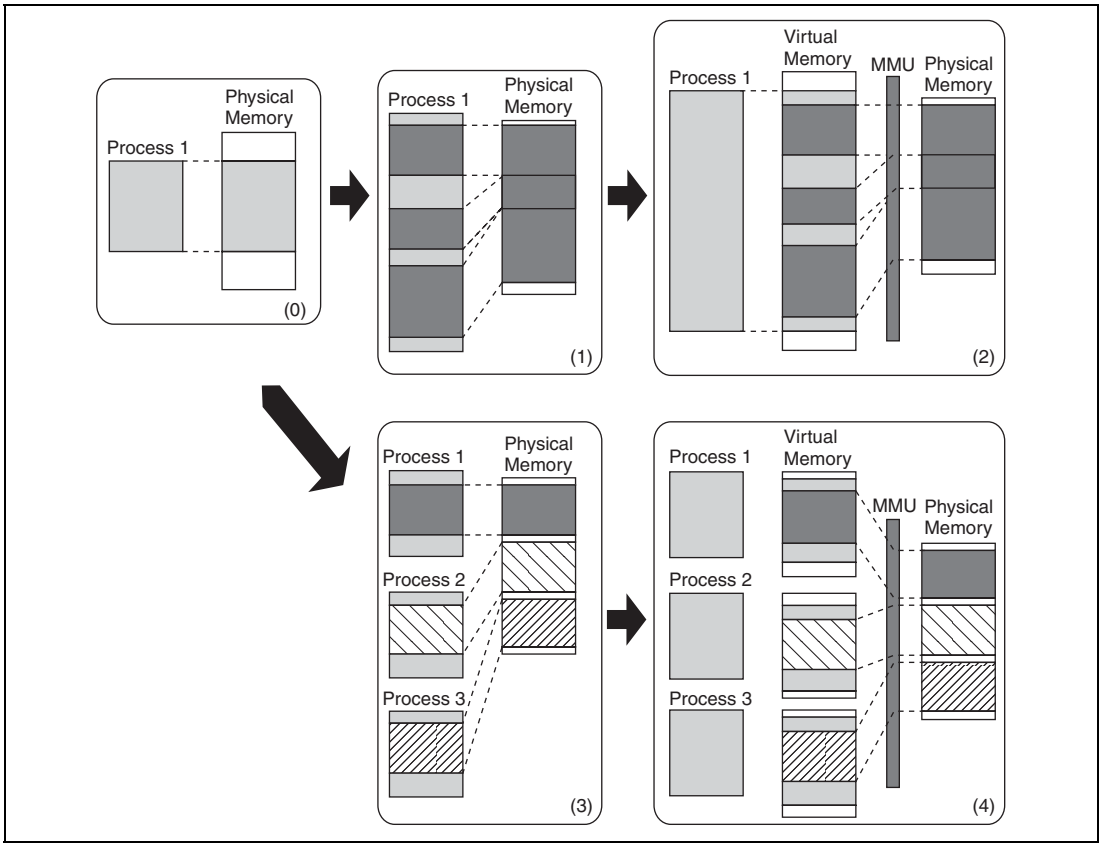


Figure 7.1 Role of MMU

7.1.1 Address Spaces

Virtual Address Space: This LSI supports a 32-bit virtual address space, and can access a 4-Gbyte address space. The virtual address space is divided into a number of areas, as shown in figures 7.2 and 7.3. In privileged mode, the 4-Gbyte space from the P0 area to the P4 area can be accessed. In user mode, a 2-Gbyte space in the U0 area can be accessed. When the SQMD bit in the MMU control register (MMUCR) is 0, a 64-Mbyte space in the store queue area can be accessed. When the RMD bit in the on-chip memory control register (RAMCR) is 1, a 16-Mbyte space in on-chip memory area can be accessed. Accessing areas other than the U0 area, store 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 areas can be mapped onto any physical address space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units. By using an 8-bit address space identifier, the P0, P3, and U0 areas can be increased to a maximum of 256.

Mapping from the virtual address space to the 29-bit physical address space is carried out using the TLB.

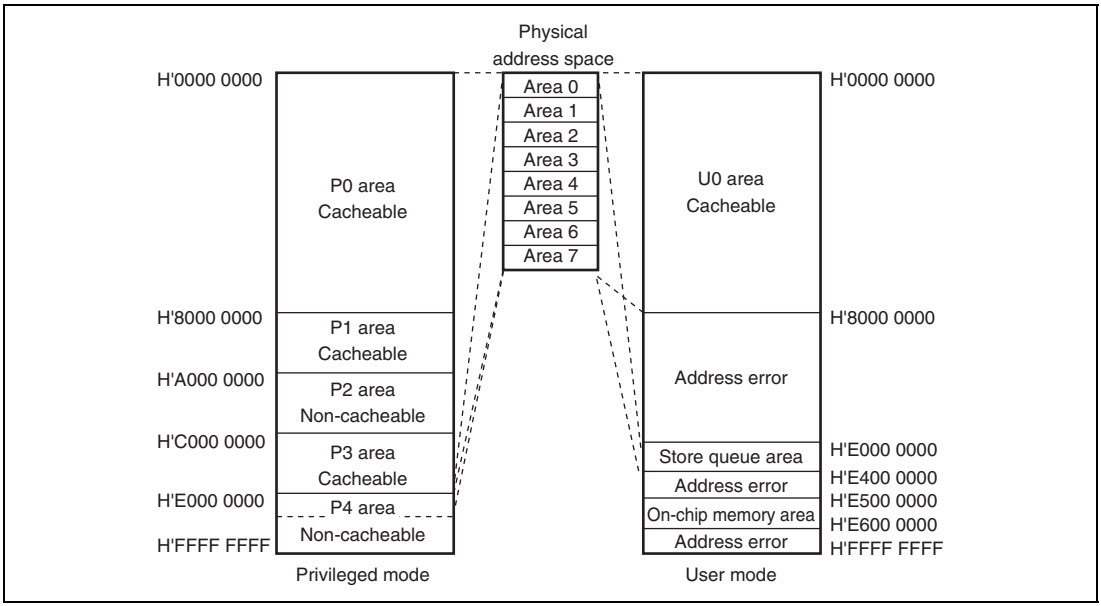


Figure 7.2 Virtual Address Space (AT in MMUCR = 0)

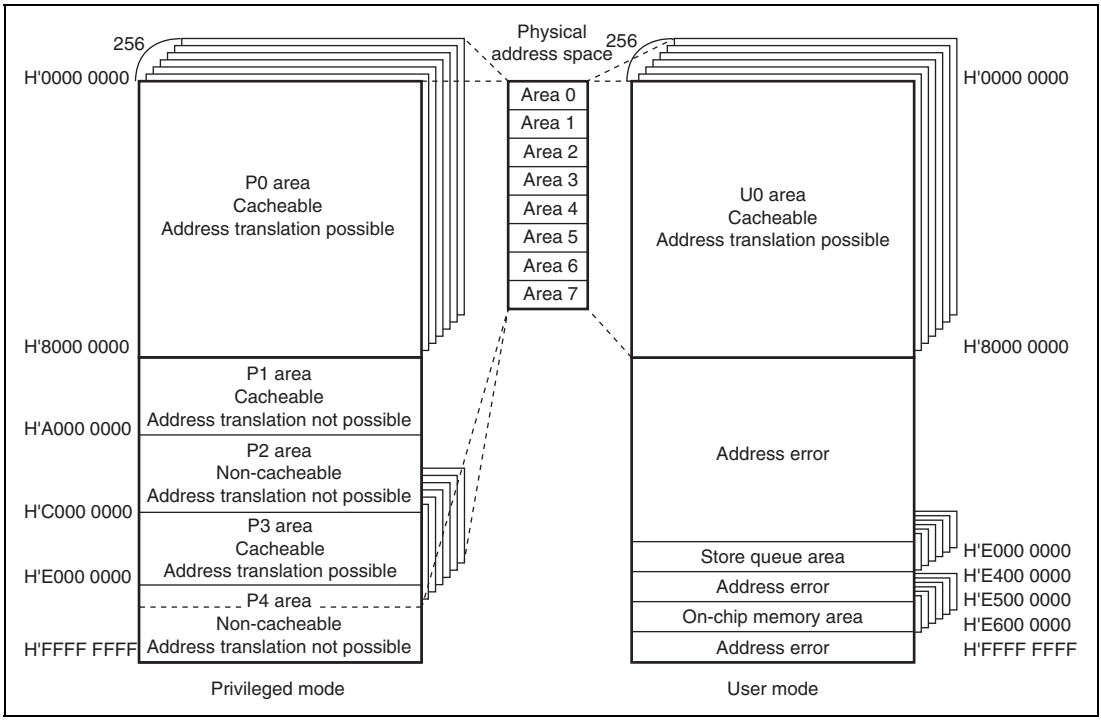


Figure 7.3 Virtual Address Space (AT in MMUCR = 1)

- P0, P3, and U0 Areas:

  The P0, P3, and U0 areas allow address translation using the TLB and access using the cache.

  When the MMU is disabled, replacing the upper 3 bits of an address with 0s gives the corresponding physical address. Whether or not the cache is used is determined by the CCR setting. When the cache is used, switching between the copy-back method and the write-through method for write accesses is specified by the WT bit in CCR.

  When the MMU is enabled, these areas can be mapped onto any physical address space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units using the TLB. When CCR is in the cache enabled state and the C bit for the corresponding page of the TLB entry is 1, accesses can be performed using the cache. When the cache is used, switching between the copy-back method and the 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 allocated in the area 7 in physical address space by means of the TLB, the C bit for the corresponding page must be cleared to 0.

- P1 Area:

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

  Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of an address to 0 gives the corresponding physical address. Whether or not the cache is used is determined by the CCR setting. When the cache is used, switching between the copy-back method and the 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 cache.

  Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of an address 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 store queue and on-chip memory areas does not allow address translation using the TLB. This area cannot be accessed using the cache. The P4 area is shown in detail in figure 7.4.

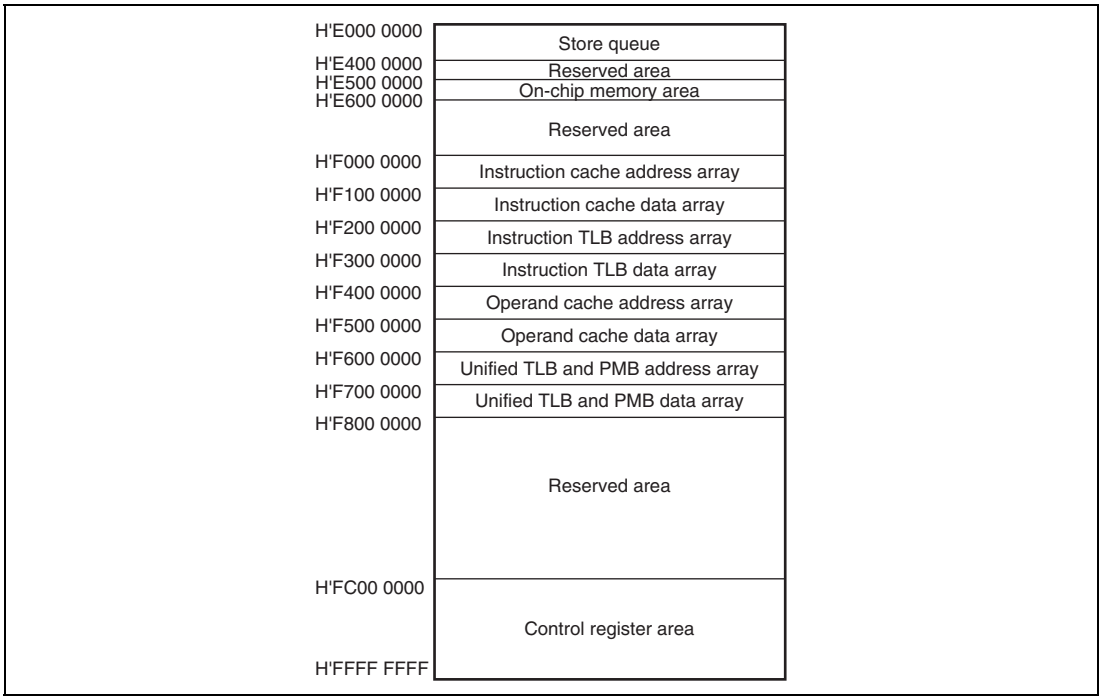


Figure 7.4 P4 Area

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

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

The area from H'F000 0000 to H'F0FF FFFF is used for direct access to the instruction cache 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 cache data 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 TLB 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 data array. For details, see section 7.6.2, ITLB Data Array.

The area from H'F400 0000 to H'F4FF FFFF is used for direct access to the operand cache address array. For details, see section 8.6.3, OC Address Array.

The area from H'F500 0000 to H'F5FF FFFF is used for direct access to the operand cache data array. For details, see section 8.6.4, OC Data Array.

The area from H'F600 0000 to H'F60F FFFF is used for direct access to the unified TLB address array. For details, see section 7.6.3, UTLB Address Array.

The area from H'F700 0000 to H'F70F FFFF is used for direct access to unified TLB data array. For details, see section 7.6.4, UTLB Data Array.

The area from H'F610 0000 to H'F61F FFFF is used for direct access to the PMB address array. 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 array. For 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 register area. For details, see register descriptions in each section.

Physical Address Space: This LSI supports a 29-bit physical address space. The physical address 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'1C00 0000 to H'1FFF FFFF of area 7 are not designated as a reserved area, but are equivalent to the control register area in the P4 area in the virtual address space.

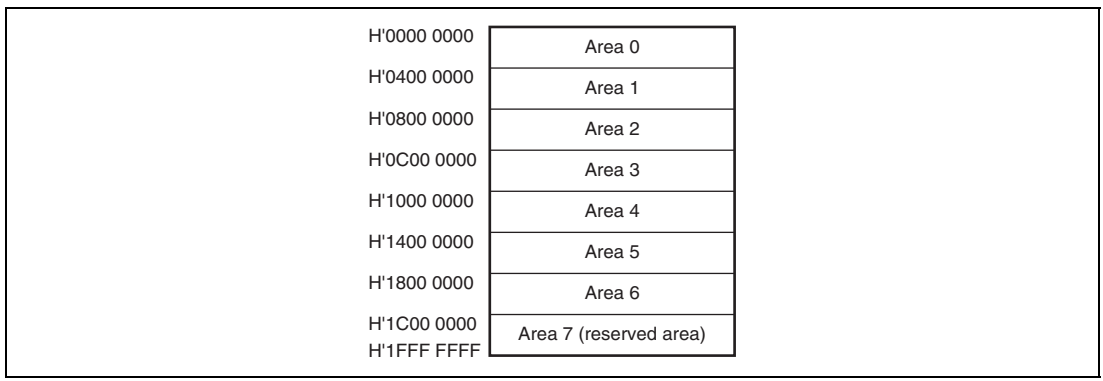


Figure 7.5 Physical Address Space

Address Translation: When the MMU is used, the virtual address space is divided into units called pages, and translation to physical addresses is carried out in these page units. The address translation table in external memory contains the physical addresses corresponding to virtual addresses and additional information such as memory protection codes. Fast address translation is achieved by caching the contents of the address translation table located in external memory into the TLB. In this LSI, basically, the ITLB is used for instruction accesses and the UTLB for data accesses. In the event of an access to an area other than the P4 area, the accessed virtual address is translated to a physical address. If the virtual address belongs to the P1 or P2 area, the physical address is uniquely determined without accessing the TLB. If the virtual address belongs to the P0, U0, or P3 area, the TLB is searched using the virtual address, and if the virtual address is recorded in the TLB, a TLB hit is made and the corresponding physical address is read from the TLB. If the accessed virtual address is not recorded in the TLB, a TLB miss exception is generated and processing switches to the TLB miss exception handling routine. In the TLB miss exception handling routine, the address translation table in external memory is searched, and the corresponding physical address and page management information are recorded in the TLB. After the return from the exception handling routine, the instruction which caused the TLB miss exception is re-executed.

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

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

In single virtual memory mode, ASID is used to provide memory protection for multiple processes 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 ASID must not be set in the TLB simultaneously in single virtual memory mode.

7.2 Register Descriptions

The following registers are related to MMU processing.

Table 7.1 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
Page table entry low register	PTEL	R/W	H'FF00 0004	H'1F00 0004	32
Translation table base register	TTB	R/W	H'FF00 0008	H'1F00 0008	32
TLB exception address register	TEA	R/W	H'FF00 000C	H'1F00 000C	32
MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
Physical address space control register	PASCR	R/W	H'FF00 0070	H'1F00 0070	32
Instruction re-fetch inhibit control register	IRMCR	R/W	H'FF00 0078	H'1F00 0078	32

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

Table 7.2 Register States in Each Processing State

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

7.2.1 Page Table Entry High Register (PTEH)

PTEH consists of the virtual page number (VPN) and address space identifier (ASID). When an MMU exception or address error exception occurs, the VPN of the virtual address at which the exception occurred is set in the VPN bit by hardware. VPN varies according to the page size, but the VPN set by hardware when an exception occurs consists of the upper 22 bits of the virtual address which caused the exception. VPN setting can also be carried out by software. The number of the currently executing process is set in the ASID bit by software. ASID is not updated by hardware. VPN and ASID are recorded in the UTLB by means of the LDTLB instruction.

After the ASID field in PTEH has been updated, execute one of the following three methods before an access (including an instruction fetch) to the P0, P3, or U0 area that uses the updated ASID value is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, 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 specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after the ASID field has been updated.

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	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	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	—	R/W	Virtual Page Number
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ASID	—	R/W	Address Space Identifier

7.2.2 Page Table Entry Low Register (PTEL)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PPN												
Initial value:	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPN						—	V	SZ1	PR1	PR0	SZ0	C	D	SH	WT
Initial value:	—	—	—	—	—	—	0	—	—	—	—	—	—	—	—	—
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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	V	—	R/W	Page Management Information
7	SZ1	—	R/W	The meaning of each bit is same as that of corresponding bit in Common TLB (UTLB). For details, see section 7.3, TLB Functions.
6	PR1	—	R/W	
5	PR0	—	R/W	
4	SZ0	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
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 contents of TTB are not changed unless a software directive is issued. This register can be used freely by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	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	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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	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 the 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	17	16
	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	R/W	R/W

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

7.2.5 MMU Control Register (MMUCR)

The individual bits perform MMU settings as shown below. Therefore, MMUCR rewriting should be performed by a program in the P1 or P2 area.

After MMUCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the P0, P3, U0, or store queue area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, 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 MMUCR, 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 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 is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LRUI						—	—	URB						—	—
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	R	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	1	0
	URC						SQMD	SV	—	—	—	—	—	TI	—	AT
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	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	LRUI	All 0	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. X means that updating is not performed. 000XXX: ITLB entry 0 is used 1XX00X: ITLB entry 1 is used X1X1X0: ITLB entry 2 is used XX1X11: ITLB entry 3 is used XXXXXX: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited 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 value should always be 0.
23 to 18	URB	All 0	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary at which replacement is to be performed. Valid only when URB ≠ 0.

Bit	Bit Name	Initial Value	R/W	Description
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 10	URC	All 0	R/W	UTLB Replace Counter These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction. This bit is incremented each time the UTLB is accessed. If URB > 0, URC is cleared to 0 when the condition URC = URB is satisfied. Also note that if a value is written to URC by software which results in the condition of URC > URB, incrementing is first performed in excess of URB until URC = H'3F. URC is not incremented by an LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Bit Specifies the right of access to the store queues. 0: User/privileged access possible 1: Privileged access possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching Bit When this bit is changed, ensure that 1 is also written 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 value should always be 0.
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	AT	0	R/W	Address Translation Enable Bit These bits enable or disable the MMU. 0: MMU disabled 1: MMU enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

7.2.6 Physical Address Space Control Register (PASCR)

PASCR controls the operation in the physical address space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UB							
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/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 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mbytes) When writing is performed without using the cache or in the cache write-through mode, these bits specify whether the next bus access from the CPU waits for the end of writing for each area. 0: The CPU does not wait for the end of writing bus access and starts the next bus access 1: The CPU waits for the end of writing bus access and starts the next bus access UB[7]: Corresponding to the control register area 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

7.2.7 Instruction Re-Fetch Inhibit Control Register (IRMCR)

When the specific resource is changed, IRMCR controls whether the instruction fetch is performed again for the next instruction. The specific resource means the part of control registers, TLB, and cache.

In the initial state, the instruction fetch is performed again for the next instruction after changing the resource. However, the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction every time the resource is changed. Therefore, it is recommended that each bit in IRMCR is set to 1 and the specific instruction should be executed after all necessary resources have been changed prior to execution of the program which uses 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	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT	MT	MC
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/W	R/W	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 value should always be 0.
4	R2	0	R/W	Re-Fetch Inhibit 2 after Register Change When MMUCR, PASCR, CCR, PTEH, or RAMCR is 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'FF200000 to H'FF2FFFFF is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed

Bit	Bit Name	Initial Value	R/W	Description
2	LT	0	R/W	Re-Fetch Inhibit after LDTLB Execution This bit controls whether re-fetch is performed for the next instruction after the LDTLB instruction has been executed. 0: Re-fetch is performed 1: Re-fetch is not performed
1	MT	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped TLB This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped ITLB/UTLB while the AT bit in MMUCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed
0	MC	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped IC This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped IC while the ICE bit in CCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

7.3 TLB Functions

7.3.1 Unified TLB (UTLB) Configuration

The UTLB is used for the following two purposes:

1. To translate a virtual address to a physical address in a data access
2. As a table of address translation information to be recorded in the ITLB in the event of an ITLB miss

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

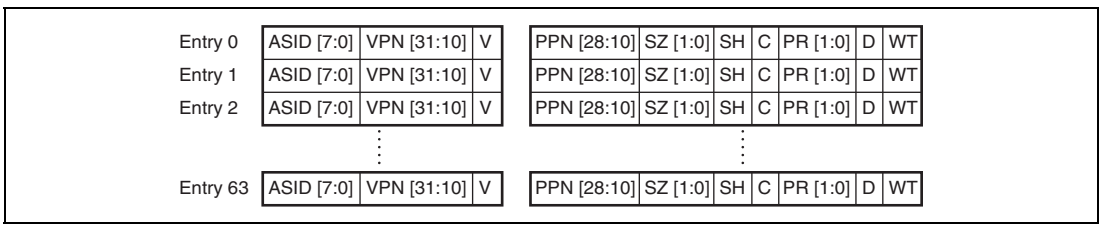


Figure 7.6 UTLB Configuration

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

- SH: Share status bit
  When 0, pages are not shared by processes.
  When 1, pages are shared by processes.

- SZ[1:0]: Page size bits
  Specify the page size.
  00: 1-Kbyte page
  01: 4-Kbyte page
  10: 64-Kbyte page
  11: 1-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 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 7.4.5, 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

1: Cacheable

When the control register area is mapped, this bit must be cleared to 0.

- D: Dirty bit

  Indicates whether a write has been performed to a page.

  0: Write has not been performed

  1: Write has been performed

- WT: Write-through bit

  Specifies the cache write mode.

  0: Copy-back mode

  1: Write-through mode

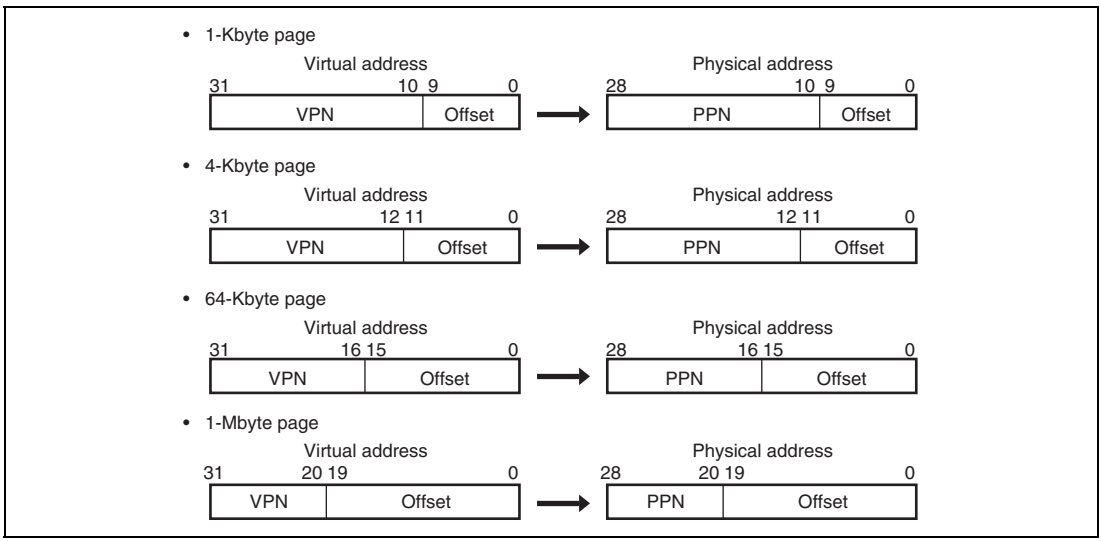


Figure 7.7 Relationship between Page Size and Address Format

7.3.2 Instruction TLB (ITLB) Configuration

The ITLB is used to translate a virtual address to a physical address in an instruction access. Information in the address translation table located in the UTLB is cached into the ITLB. Figure 7.8 shows the ITLB configuration. The ITLB consists of four fully-associative type entries.

Entry 0	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR
Entry 1	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR
Entry 2	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR
Entry 3	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR

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

7.3.3 Address Translation Method

Figure 7.9 shows a flowchart of a memory access using the UTLB.

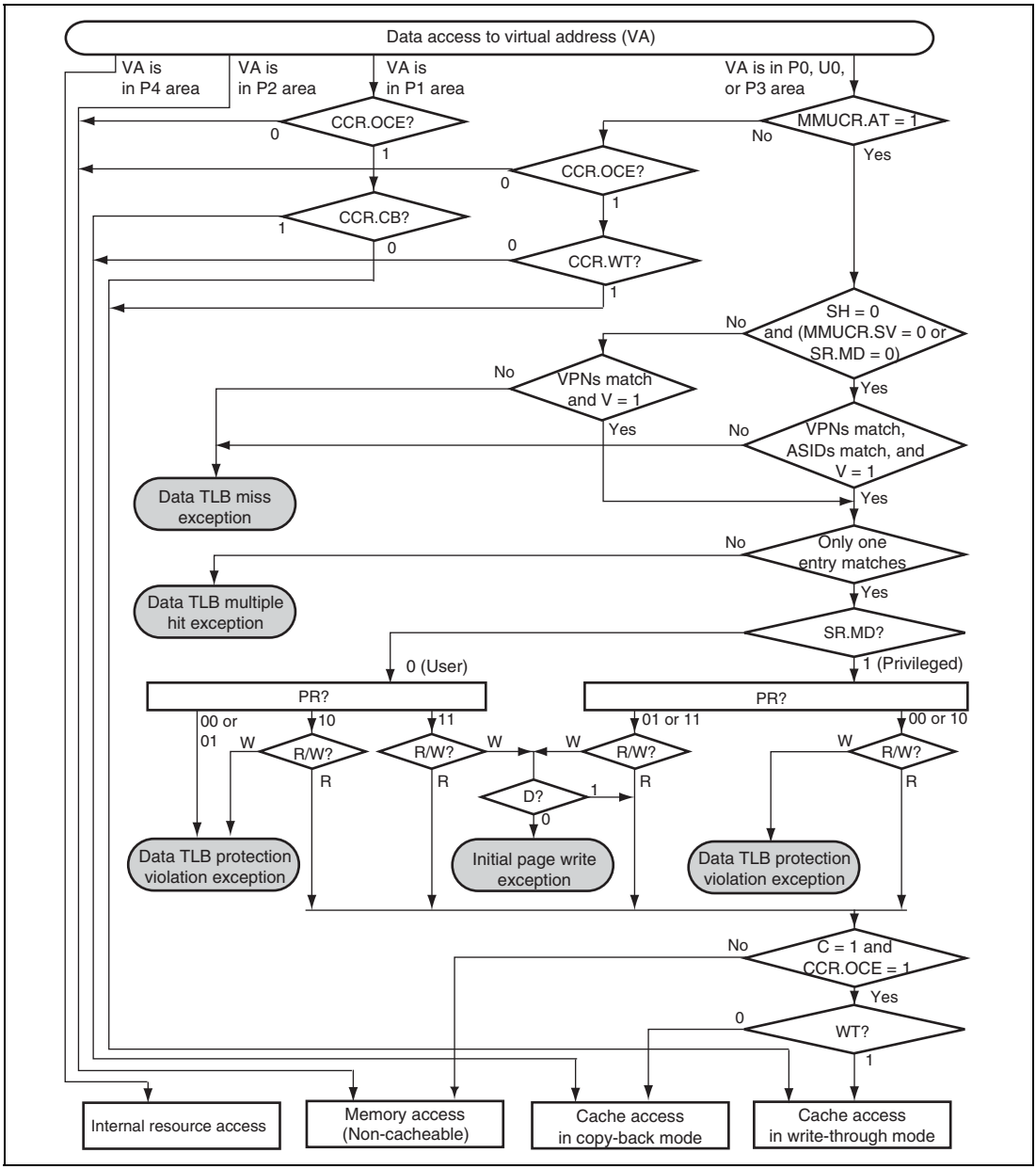


Figure 7.9 Flowchart of Memory Access Using UTLB

Figure 7.10 shows a flowchart of a memory access using the ITLB.

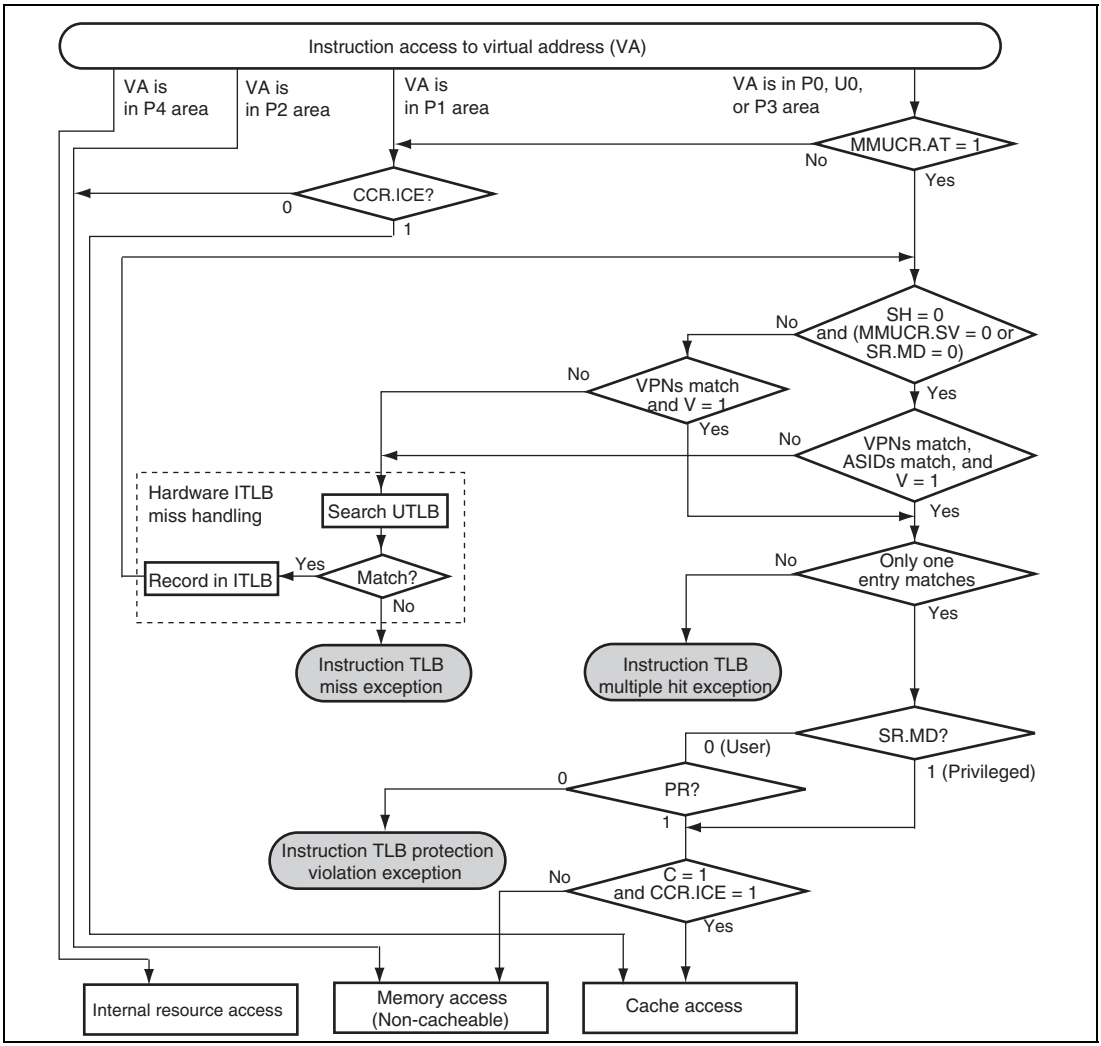


Figure 7.10 Flowchart of Memory Access Using ITLB

7.4 MMU Functions

7.4.1 MMU Hardware Management

This LSI supports the following MMU functions.

1. The MMU decodes the virtual address to be accessed by software, and performs address translation by controlling the UTLB/ITLB in accordance with the MMUCR settings.
2. The MMU determines the cache access status on the basis of the page management information read during address translation (C and WT bits).
3. If address translation cannot be performed normally in a data access or instruction access, the MMU notifies software by means of an MMU exception.
4. If address translation information is not recorded in the ITLB in an instruction access, the MMU searches the UTLB. If the necessary address translation information is recorded in the UTLB, the MMU copies this information into the ITLB in accordance with the LRUI bit 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 hardware automatically.
2. Recording, deletion, and reading of TLB entries. There are two methods of recording UTLB entries: by using the LDTLB instruction, or by writing directly to the memory-mapped UTLB. ITLB entries can only be recorded by writing directly to the memory-mapped ITLB. Deleting or reading UTLB/ITLB entries is enabled by accessing the memory-mapped UTLB/ITLB.
3. MMU exception handling. When an MMU exception occurs, processing is performed based on information set by hardware.

7.4.3 MMU Instruction (LDTLB)

A TLB load instruction (LDTLB) is provided for recording UTLB entries. When an LDTLB instruction is issued, this LSI copies the contents of PTEH and PTEL to the UTLB entry indicated by the URC bit in MMUCR. ITLB entries are not updated by the LDTLB instruction, and therefore address translation information purged from the UTLB entry may still remain in the ITLB entry. As the LDTLB instruction changes address translation information, ensure that it is issued by a program in the P1 or P2 area.

After the LDTLB instruction has been executed, execute one of the following three methods before an access (include an instruction fetch) the area where TLB is used to translate the address is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the 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, 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 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 is recommended that the method 1 or 2 should be used for being compatible with the future SuperH series.

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

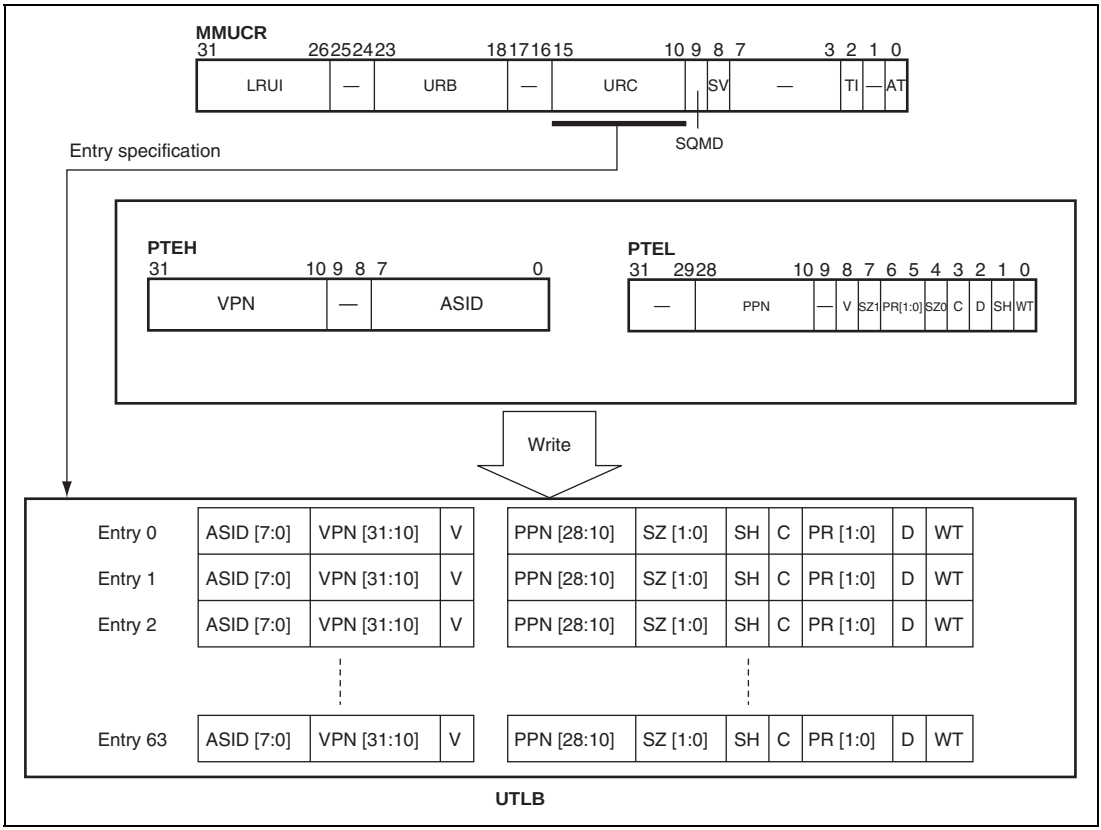


Figure 7.11 Operation of LDTLB Instruction

7.4.4 Hardware ITLB Miss Handling

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

7.4.5 Avoiding Synonym Problems

When 1- or 4-Kbyte pages are recorded in TLB entries, a synonym problem may arise. The problem is that, when a number of virtual addresses are mapped onto a single physical address, the same physical address data is recorded in a number of cache entries, and it becomes impossible to guarantee data integrity. This problem does not occur with the instruction TLB and instruction cache because data is only read in these cases. In this LSI, entry specification is performed using bits 12 to 5 of the virtual address in order to achieve fast operand cache operation. However, bits 12 to 10 of the virtual address in the case of a 1-Kbyte page, and bit 12 of the virtual address in the case of a 4-Kbyte page, are subject to address translation. As a result, bits 12 to 10 of the physical address after translation may differ from bits 12 to 10 of the virtual address.

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

- When address translation information whereby a number of 1-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, ensure that the VPN[12:10] values are the same.
- When address translation information whereby a number of 4-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, ensure that the VPN[12] value is the same.
- Do not use 1-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.
- Do not use 4-Kbyte page UTLB entry physical addresses with UTLB entries of a different 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 memory to 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 address translation information of different page sizes.

7.5 MMU Exceptions

There are seven MMU exceptions: instruction TLB multiple hit exception, instruction TLB miss exception, instruction TLB protection violation exception, data TLB multiple hit exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception. Refer to figures 7.9 and 7.10 for the conditions under which each of these exceptions occurs.

7.5.1 Instruction TLB Multiple Hit Exception

An instruction TLB multiple hit exception occurs when more than one ITLB entry matches the virtual address to which an instruction access has been made. If multiple hits occur when the UTLB is searched by hardware in hardware ITLB miss handling, an instruction TLB multiple hit exception will result.

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

Hardware Processing: In the event of an instruction TLB multiple hit exception, hardware carries 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 exception are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

7.5.2 Instruction TLB Miss Exception

An instruction TLB miss exception occurs when address translation information for the virtual address to which an instruction access is made is not found in the UTLB entries by the hardware ITLB miss handling routine. The instruction TLB miss exception processing carried out by hardware and software is shown below. This is the same as the processing for a data TLB miss exception.

Hardware Processing: In the event of an instruction TLB miss exception, hardware 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'040 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are 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 VBR, and starts the instruction TLB miss exception handling routine.

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

1. Write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry recorded in the external memory address translation table.
2. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be 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 exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

7.5.3 Instruction TLB Protection Violation Exception

An instruction TLB protection violation exception occurs when, even though an ITLB entry contains address translation information matching the virtual address to which an instruction access is made, the actual access type is not permitted by the access right specified by the PR bit. The instruction TLB protection violation exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of an instruction TLB protection violation exception, hardware 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 EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are 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 VBR, and starts the instruction TLB protection violation exception handling routine.

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

7.5.4 Data TLB Multiple Hit Exception

A data TLB multiple hit exception occurs when more than one UTLB entry matches the virtual address to which a data access has been made.

When a data TLB multiple hit exception occurs, a reset is executed, and cache coherency is not guaranteed. The contents of PPN in the UTLB prior to the exception may also be corrupted.

Hardware Processing: In the event of a data TLB multiple hit exception, hardware carries 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 UTLB entries which caused the multiple hit exception 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 address to which a data access is made is not found in the UTLB entries. The data TLB miss exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of a data TLB miss exception, hardware 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'040 in the case of a read, or H'060 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are 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 VBR, and starts the data TLB miss exception handling routine.

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

1. Write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry recorded in the external memory address translation table.
2. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be 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 UTLB.
4. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued 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 contains address translation information matching the virtual address to which a data access is made, the actual access type is not permitted by the access right specified by the PR bit. The data TLB protection violation exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of a data TLB protection violation exception, hardware 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 EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are 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 VBR, and starts the data TLB protection violation exception handling routine.

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

7.5.7 Initial Page Write Exception

An initial page write exception occurs when the D bit is 0 even though a UTLB entry contains address translation information matching the virtual address to which a data access (write) is made, and the access is permitted. The initial page write exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of an initial page write exception, hardware 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'080 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are 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 VBR, and starts the initial page write exception handling routine.

Software Processing (Initial Page Write Exception Handling Routine): Software is responsible 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 table entry recorded in external memory.

4. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.
5. Execute the LDTLB instruction and write the contents of PTEH and PTEL to the UTLB.
6. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

7.6 Memory-Mapped TLB Configuration

To enable the ITLB and UTLB to be managed by software, their contents are allowed to be read from and written to by a program in the P2 area with a MOV instruction in privileged mode. 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 methods before an access (including an instruction fetch) to an area other than the P2 area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be an area 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 TLB, 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 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 is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

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

7.6.1 ITLB Address Array

The ITLB address array is allocated to addresses H'F200 0000 to H'F2FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:24] have the value H'F2 indicating the ITLB address array and the entry is specified by bits [9:8]. As only longword access is used, 0 should be specified for address field bits [1:0].

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

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 the entry 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 corresponding to the entry set in the address field.

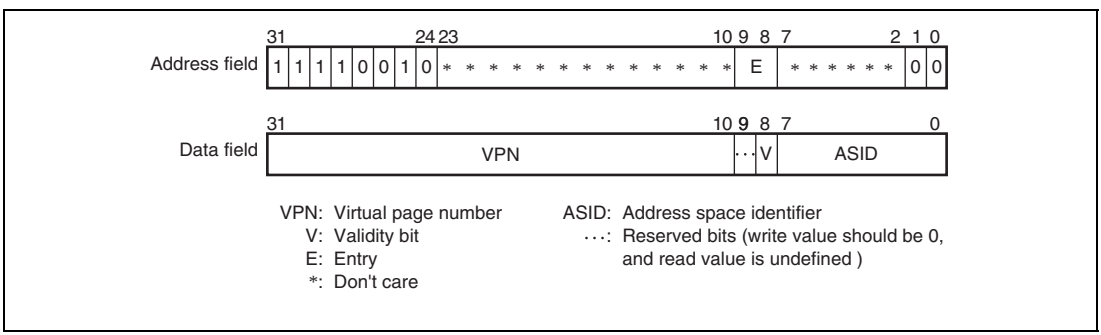


Figure 7.12 Memory-Mapped ITLB Address Array

7.6.2 ITLB Data Array

The ITLB data array is allocated to addresses H'F300 0000 to H'F37F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and PPN, V, SZ, PR, C, and SH to be written to the data array are specified in the data field.

In the address field, bits [31:23] have the value H'F30 indicating ITLB data array and the entry is specified by bits [9:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bit [6] 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 corresponding to 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.

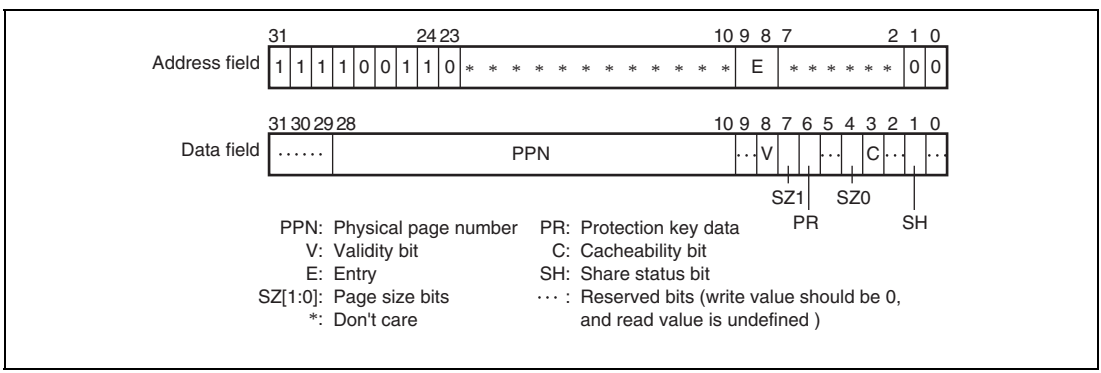


Figure 7.13 Memory-Mapped ITLB Data Array

7.6.3 UTLB Address Array

The UTLB address array is allocated to addresses H'F600 0000 to H'F60F FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, D, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:20] have the value H'F60 indicating the UTLB address array and the entry is specified by bits [13:8]. Bit [7] that is the association bit (A bit) in the address field specifies whether address comparison is performed in a write to the UTLB address array.

In the data field, bits [31:10] indicate VPN, bit [9] indicates D, bit [8] indicates V, and bits [7:0] 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 to the entry set in the address field. In a read, associative operation is not performed regardless of 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 corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.
3. UTLB address array write (associative)

   When a write is performed with the A bit in the address field set to 1, comparison of all the UTLB entries is carried out using the VPN specified in the data field and ASID in PTEH. The usual address comparison rules are followed, but if a UTLB miss occurs, the result is no operation, and an exception is not generated. If the comparison identifies a UTLB entry corresponding to the VPN specified in the data field, D and V specified in the data field are written to that entry. This associative operation is simultaneously carried out on the ITLB, and 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 matching 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.

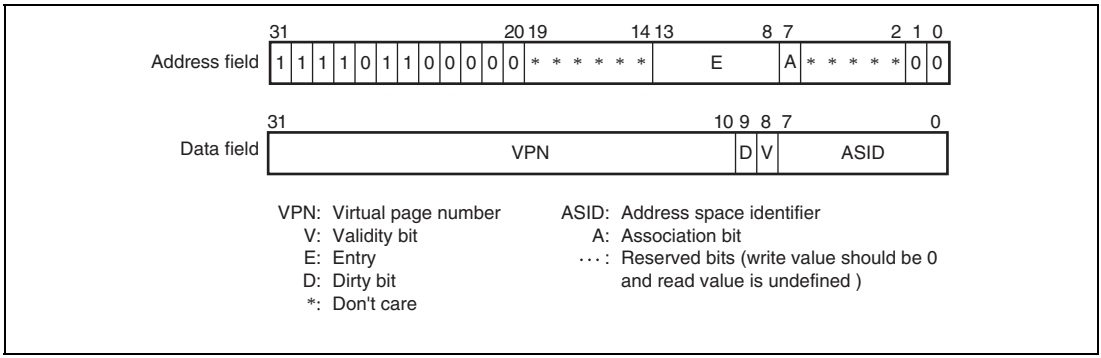


Figure 7.14 Memory-Mapped UTLB Address Array

7.6.4 UTLB Data Array

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

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

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bits [6:5] indicate PR, bit [3] indicates C, bit [2] indicates D, bit [1] indicates SH, and bit [0] indicates 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 UTLB entry corresponding to the entry set in the address field.

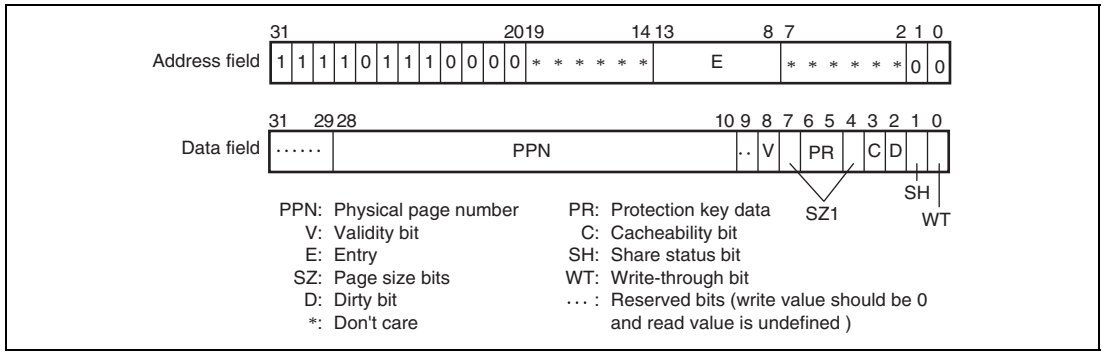


Figure 7.15 Memory-Mapped UTLB Data Array

7.7 32-Bit Address Extended Mode

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

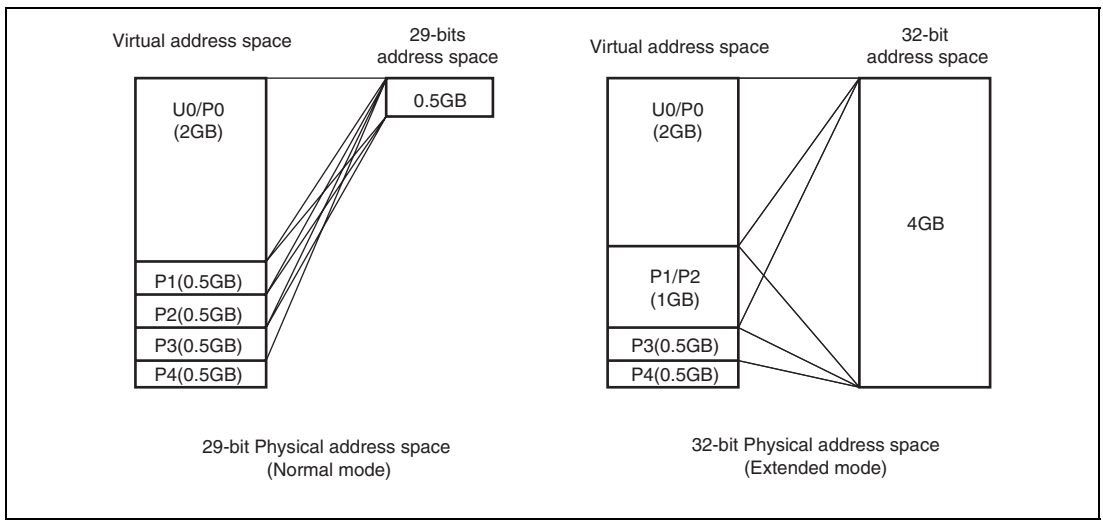


Figure 7.16 Physical Address Space (32-Bit Address Extended Mode)

7.7.1 Overview of 32-Bit Address Extended Mode

In 32-bit address extended mode, the privileged space mapping buffer (PMB) is introduced. The PMB maps virtual addresses in the P1 or P2 area which are not translated in 29-bit address mode to the 32-bit physical address space. In areas which are target for address translation of the TLB (UTLB/ITLB), upper three bits in the PPN field of the UTLB or ITLB are extended and then addresses after the TLB translation can handle the 32-bit physical addresses.

As for the cache operation, P1 area is cacheable and P2 area is non-cacheable in the case of 29-bit address mode, but the cache operation of both P1 and P2 area are determined by the C bit and WT bit in the PMB in the case of 32-bit address 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 address extended mode by setting the SE bit in PASCR to 1. In 32-bit address extended mode, the MMU operates as follows.

1. When the AT bit in MMUCR is 0, virtual addresses in the U0, P0, or P3 area become 32-bit physical addresses. Addresses in the P1 or 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 PMB in order to indicate P1 or P2 area. The operation is not guaranteed when the value except B'10 is set to these bits.
2. When the AT bit in MMUCR is 1, virtual addresses in the U0, P0, or P3 area are translated to 32-bit physical addresses according to the TLB conversion information. Addresses in the P1 or 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 PMB in order to indicate P1 or P2 area. The operation is not guaranteed when the value except B'10 is set to these bits.
3. Regardless of the setting of the AT bit in MMUCR, bits 31 to 29 in physical addresses become B'111 in the control register area (addresses H'FC00 0000 to H'FFFF FFFF). When the control register area is recorded in the UTLB and accessed, B'111 should be set to PPN[31:29].

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 according to the PMB mapping information. The PMB has 16 entries and configuration of each entry is as follows.

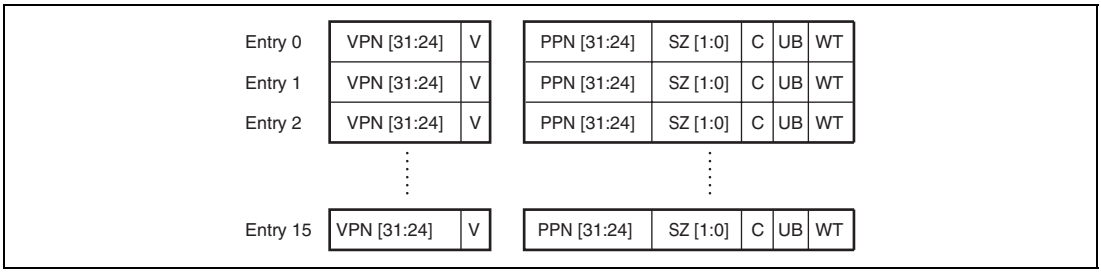


Figure 7.17 PMB Configuration

[Legend]

- 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 area.

- 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.

With a 512-Mbyte page, PPN[31:29] are valid.

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

- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  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 for the write to complete.)
  1: Unbuffered write (Data access of subsequent processing is stalled until the write has 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 instruction cannot be used to write to the PMB.
2. Software must ensure that every accessed P1 or P2 address has a corresponding PMB entry before the access occurs. When an access to an address in the P1 or P2 area which is not recorded in the PMB is made, this LSI is reset by the TLB. In this case, the accessed address in the P1 or P2 area which causes the TLB reset is stored in the TEA and code H′140 in the EXPEVT.
3. This LSI does not guarantee the operation when multiple hit occurs in the PMB. Special care 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. The address translation target of the PMB is the P1 or P2 address. In user mode access, an address error exception occurs.
6. Both entries from the UTLB and PMB are mixed and recorded in the ITLB by means of the hardware ITLB miss handling. However, these entries can be identified by checking whether

VPN[31:30] is 10 or not. When an entry from the PMB is recorded in the ITLB, H′00, 01, and 1 are recorded in the ASID, PR, and SH fields which do not exist in the PMB, respectively.

7.7.5 Memory-Mapped PMB Configuration

To enable the PMB to be managed by software, its contents are allowed to be read from and written to by a P1 or P2 area program with a MOV instruction in privileged mode. The PMB address array is allocated to addresses H'F610 0000 to H'F61F FFFF in the P4 area and the PMB data array to addresses H'F710 0000 to H'F71F FFFF in the P4 area. VPN and V in the PMB can be accessed as an address array, PPN, V, SZ, C, WT, and UB as a data array. V can be accessed from both the address array side and the data array side. A program which executes a PMB memory-mapped access should be placed in the page area at which the C bit in PMB is cleared to 0.

1. PMB address array read

   When memory reading is performed while bits 31 to 20 in the address field are specified as H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as an entry, 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 specified as H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as an entry, and bits 31 to 24 in the data field are specified as VPN and bit 8 in the data field as V, data is 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 specified as H'F71 which indicates the PMB data array and bits 11 to 8 in the address field as an entry, bits 31 to 24 in the data field are read as PPN, bit 9 in the data field as UB, bit 8 in the data field as 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 field as WT.
4. PMB data array write

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

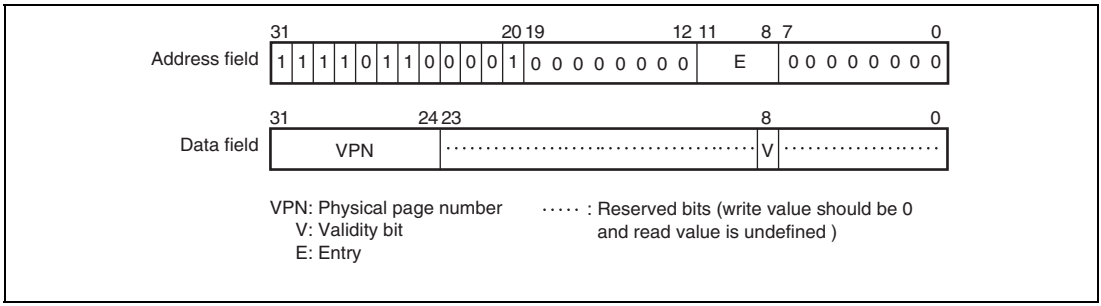


Figure 7.18 Memory-Mapped PMB Address Array

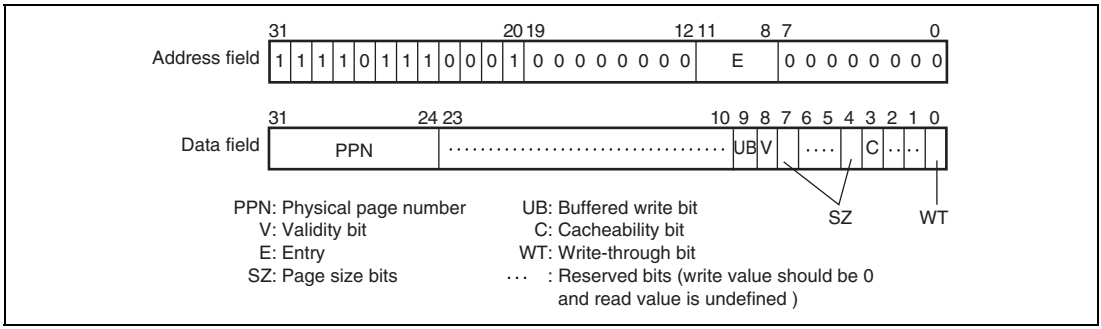


Figure 7.19 Memory-Mapped PMB Data Array

7.7.6 Notes on Using 32-Bit Address Extended Mode

When using 32-bit address extended mode, note that the items described in this section are extended or changed as follows.

PASCR: The SE bit is added in bit 31 in the control register (PASCR). The bits 6 to 0 of the UB in the PASCR are invalid (Note that the bit 7 of the UB is still valid). When writing to the P1 or P2 area, the UB bit in the PMB controls whether a buffered write is performed or not. When the MMU is enabled, the UB bit in the TLB controls writing to the P0, P3, or U0 area. When the MMU is disabled, writing to the P0, P3, or U0 area is always performed as a buffered write.

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 these bits, see description in General Precautions on Handling of Product.
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mbytes) When writing is performed without using the cache or in the cache write-through mode, these bits specify whether the CPU waits for the end of writing for each area. 0: The CPU does not wait for the end of writing 1: The CPU stalls and waits for the end of writing UB[7]: Corresponding to the control register area UB[6:0]: These bits are invalid in 32-bit address extended mode.

ITLB: The PPN field in the ITLB is extended to bits 31 to 10.

UTLB: The PPN field in the UTLB is extended to bits 31 to 10. The same UB bit as that in the PMB is added in each entry of the UTLB.

- UB: Buffered write bit
  Specifies whether a buffered write is performed.
  0: Buffered write (Subsequent processing proceeds without waiting for the write to complete.)
  1: Unbuffered write (Subsequent processing is stalled until the write has completed.)

In a memory-mapped TLB access, the UB bit can be read from or written to by bit 9 in the data array.

PTEL: The same UB bit as that in the PMB is added in bit 9 in PTEL. This UB bit is written to the UB bit in the UTLB by the LDTLB instruction. The PPN field is extended to bits 31 to 10.

CCR.CB: The CB bit in CCR is invalid. Whether a cacheable write for the P1 area is performed 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 to AREA0[7:2]/AREA1[7:2] corresponding to physical address [31:26]. See section 8.2.2, Queue Address Control Register 0 (QACR0) and 8.2.3, Queue Address Control Register 1 (QACR1).

LSA0, LSA1, LDA0, LDA1: L0SADR, L1SADR, L0DADR and L1DADR fields are extended to bits 31 to 10. See section 9.2.2, L Memory Transfer Source Address Register 0 (LSA0), section 9.2.3, L Memory Transfer Source Address Register 1 (LSA1), section 9.2.4, L Memory Transfer Destination Address Register 0 (LDA0), and section 9.2.5, L Memory Transfer Destination 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 disabled 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 target of the PMB address translation. Therefore, the area should be recorded in the PMB before switching the SE bit. An address which may be accessed in the P1 or P2 area such as the exception handler should also be recorded in the PMB.
3. When an external memory access occurs by an operand memory access located before the MOV.L instruction which switches the SE bit, external memory space addresses accessed in both address modes should be the same.

4. Note that the V bit is mapped to both address array and data array in PMB registration. That is, first write 0 to the V bit in one of arrays and then write 1 to the V bit in another array.

Section 8 Caches

This LSI has an on-chip 32-Kbyte instruction cache (IC) for instructions and an on-chip 32-Kbyte operand cache (OC) for data.

8.1 Features

The features of the cache are shown in table 8.1.

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external 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, virtual address index/physical address tag
Line size	32 bytes	32 bytes
Entries	256 entries/way	256 entries/way
Write method	—	Copy-back/write-through selectable
Replacement method	LRU (least-recently-used) algorithm	LRU (least-recently-used) algorithm

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

The operand cache of this LSI uses the 4-way set-associative, each way comprising 256 cache lines. Figure 8.1 shows the configuration of the operand cache.

The instruction cache is 4-way set-associative, each way is comprising 256 cache lines. Figure 8.2 shows the configuration of the instruction cache.

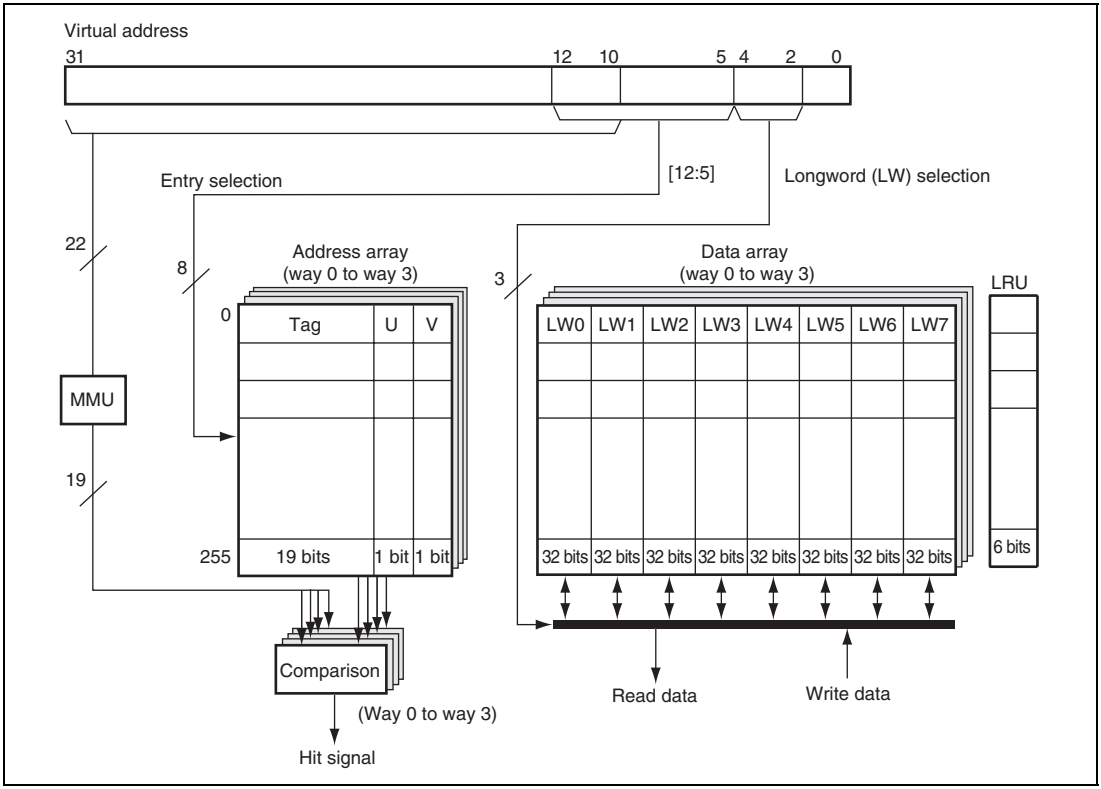


Figure 8.1 Configuration of Operand Cache (OC)

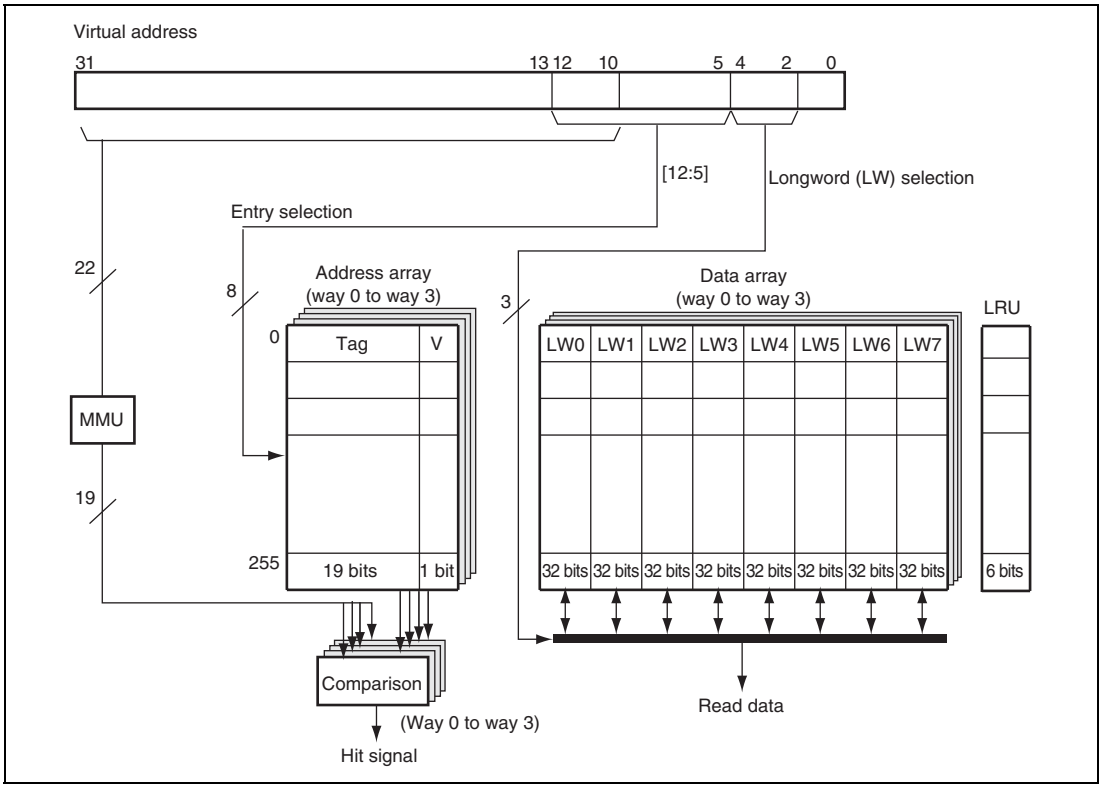


Figure 8.2 Configuration of Instruction Cache (IC)

- Tag

  Stores the upper 19 bits of the 29-bit physical address of the data line to be cached. The tag is 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 data is valid. The V bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- 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 in copy-back mode. That is, the U bit indicates a mismatch between the data in the cache line and the data in external memory. The U bit is never set to 1 while the cache is being used in write-through mode, unless it is modified by accessing the memory-mapped cache (see section 8.6, Memory-Mapped Cache Configuration). The U bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- Data array

  The data field holds 32 bytes (256 bits) of data per cache line. The data array is not initialized by a power-on or manual reset.

- LRU

  In a 4-way set-associative method, up to 4 items of data can be registered in the cache at each entry address. When an entry is registered, the LRU bit indicates which of the 4 ways it is to be registered in. The LRU mechanism uses 6 bits of each entry, and its usage is controlled by hardware. The LRU (least-recently-used) algorithm is used for way selection, and selects the less recently accessed way. The LRU bits are initialized to 0 by a power-on reset but not by a manual reset. The LRU bits cannot be read from or written to by software.

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*	Access Size
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32
Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

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

Table 8.4 Register States in Each Processing State

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

8.2.1 Cache Control Register (CCR)

CCR controls the cache operating mode, the cache write mode, and invalidation of all cache entries.

CCR modifications must only be made by a program in the non-cacheable P2 area. After CCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the cacheable 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 CCR, 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 again for the next instruction after CCR has been updated.

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB	WT	OCE
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/W	R	R	R/W	R	R	R	R	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 value should always be 0.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	IC Enable Bit Selects whether the IC is used. Note however when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC not used 1: IC used
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of all OC entries are cleared to 0. This bit is always read as 0.
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 mode. When address translation is performed, the value of the WT bit in the page management information has priority. 0: Copy-back mode 1: Write-through mode
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note however when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC not used 1: OC used

8.2.2 Queue Address Control Register 0 (QACR0)

QACR0 specifies the area maped which store queue 0 (SQ0) is mapped when the MMU is disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA0			—	—
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/W	R/W	R/W	R	R

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

8.2.3 Queue Address Control Register 1 (QACR1)

QACR1 specifies the area onto which store queue 1 (SQ1) is mapped when the MMU is disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	AREA1			—	—
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/W	R/W	R/W	R	R

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

8.2.4 On-Chip Memory Control Register (RAMCR)

RAMCR controls the number of ways in the IC and OC.

RAMCR modifications must only be made by a program in the non-cacheable P2 area. After RAMCR has been updated, execute one of the following three methods before an access (including an instruction fetch) to the cacheable area or the L memory area is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the non-cacheable area or the L memory 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 RAMCR, 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 again for the next instruction after RAMCR has been updated.

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	—	—	—	—	—	—
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/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RMD	0	R/W	On-Chip Memory Access Mode Bit For details, see section 9.4, L Memory Protective Functions.
8	RP	0	R/W	On-Chip Memory Protection Enable Bit For details, see section 9.4, L Memory Protective 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 Mode.
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 Mode.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.3 Operand Cache Operation

8.3.1 Read Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is read from a 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 by virtual address bits [12:5].
2. The tag read from the each way is compared with bits [28:10] of the physical address resulting 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 which is 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 which is 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 line on the hitted way in accordance with the access size. 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 physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data(8 bytes) including the cache-missed data. When the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. Then the LRU bit is updated to indicate the way is latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data, and when the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit, and 0 to the U bit. And the LRU bits are updated to indicate the way is latest one. The data in the write-back buffer is then written back to external memory.

8.3.2 Prefetch Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is prefetched from a 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 by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting 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 which is 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 which is 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 physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU doesn't wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. And the LRU bit is updated to indicate the way is latest one.

5. Cache miss (with write-back)

   The tag and data field of the cache line on the way which is selected to replace are saved in the write-back buffer. Then data is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU doesn't wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. And the LRU bits are updated to indicate the way is latest one. The data in the write-back buffer is then written back to external memory.

8.3.3 Write Operation

When the Operand cache (OC) is enabled (OCE = 1 in CCR) and data is written to a 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 by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting 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 No. 4 for write-through.
- I If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for write-through.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for write-through.

3. Cache hit (copy-back)

   A data write in accordance with the access size is performed for the data field on the hit way which is indexed by virtual address bits [4:0]. Then 1 is written to the U bit. The LRU bits are 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 hit way which is indexed by virtual address bits [4:0]. A write is also performed to external memory corresponding to the virtual address. Then the LRU bits are updated to indicate the way is the latest one. In this case, the U bit isn't updated.

5. Cache miss (copy-back, no write-back)

   A data write in accordance with the access size is performed for the data of the 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 on the way which is selected to replace from the physical address space corresponding to the virtual address.

   Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way. Then the LRU bit is updated to indicate the way is latest one.

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 saved in the write-back buffer. Then a data write in accordance with the access size is performed for the 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 on the way which is selected to replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. While the remaining data on the cache line is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way. Then the LRU bit is updated to indicate the way is latest one. Then the data in the write-back buffer is then written back to external memory.

7. Cache miss (write-through)

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

8.3.4 Write-Back Buffer

In order to give priority to data reads to the cache and improve performance, this LSI has a write-back buffer which holds the relevant cache entry when it becomes necessary to purge a dirty cache entry into external memory as the result of a cache miss. The write-back buffer contains one cache line of data and the physical address of the purge destination.

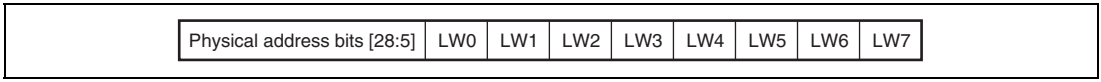


Figure 8.3 Configuration of Write-Back Buffer

8.3.5 Write-Through Buffer

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

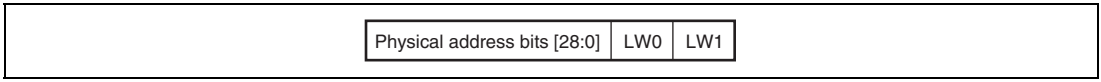


Figure 8.4 Configuration of Write-Through Buffer

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 and way 1 in the OC is entered. Thus, power consumption can be reduced. In this mode, only way 0 and 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 line has already been recorded in the OC, data should be written back by software, if necessary, 1 should be written to the OCI bit in CCR, and all entries in the OC should be invalid before modifying the OC2W bit.

8.4 Instruction Cache Operation

8.4.1 Read Operation

When the IC is enabled (ICE = 1 in CCR) and instruction fetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, U bit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting 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 field on 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 from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data, and when the corresponding data arrives in the cache, the read data is returned to the CPU as an instruction. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, and 1 is written to the V bit, the LRU bits are updated to indicate the way is the latest one.

8.4.2 Prefetch Operation

When the IC is enabled (ICE = 1 in CCR) and instruction prefetches are performed from a cacheable area, the instruction cache operates as follows:

1. The tag, V bit, Ubit and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address resulting 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 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 replace from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch opreration, the CPU doesn't wait the data arrived. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, and 1 is written to the V bit, the LRU bits is updated to indicate the way is the latest 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 way 1 in the IC is entered. Thus, power consumption can be reduced. In this mode, only way 0 and way 1 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 line has already been recorded in the IC, 1 should be written to the ICI bit in CCR and all entries in the IC should be invalid before modifying the IC2W bit.

8.5 Cache Operation Instruction

8.5.1 Coherency between Cache and External Memory

Coherency between cache and external memory should be assured by software. In this LSI, the following six instructions are supported for cache operations. Details of these instructions are given in the Software Manual.

- Operand cache invalidate instruction: OCBI @Rn
  Operand cache invalidation (no write-back)
- 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 bus to control the cache coherency. Since the address used by the PURGE and FLUSH transaction is a physical address, the following restrictions occur to avoid cache synonym problem in MMU enable mode.

- 1 Kbyte page size cannot be used.

PURGE transaction: When the operand cache is enabled, the PURGE transaction checks the operand cache and invalidates the hit entry. If the invalidated entry is dirty, the data is written back to the external memory. If the transaction is not hit to the cache, it is no-operation.

FLUSH transaction: When the operand cache is enabled, the FLUSH transaction checks the operand cache and if the hit line is dirty, then the data is written back to the external memory.

If the transaction is not hit to the cache or the hit entry is not dirty, it is no-operation.

8.5.2 Prefetch Operation

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

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

8.6 Memory-Mapped Cache Configuration

To enable the IC and OC to be managed by software, the IC contents can be read from or written to by a program in the P2 area by means of a MOV instruction in privileged mode. Operation is not guaranteed if access is made from a program in another area. In this case, execute one of the following three methods for executing a branch to the P0, U0, P1, or P3 area.

1. Execute a branch using the RTE instruction.
2. Execute a branch to the P0, U0, P1, or P3 area after executing the ICBI instruction for any address (including non-cacheable area).
3. If the MC bit in IRMCR is 0 (initial value) before making an access to the memory-mapped 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 again for 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 is recommended that the method 1 or 2 should be used for being compatible with the future SuperH Series.

In privileged mode, the OC contents can be read from or written to by a program in the P1 or P2 area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. The IC and OC are allocated to the P4 area in the virtual address space. Only data accesses can be used on both the IC address array and data array and the OC address array and data array, and accesses are always longword-size. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified and the read value is undefined.

8.6.1 IC Address Array

The IC address array is allocated to addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification 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 way and entry set in the address field. In a read, associative operation is not performed regardless of 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 to the way and entry set in the address field. The A bit in the address field should be cleared to 0.
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 way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit in the way is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an 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 definitely by handling ITLB miss and reporting ITLB miss exception.

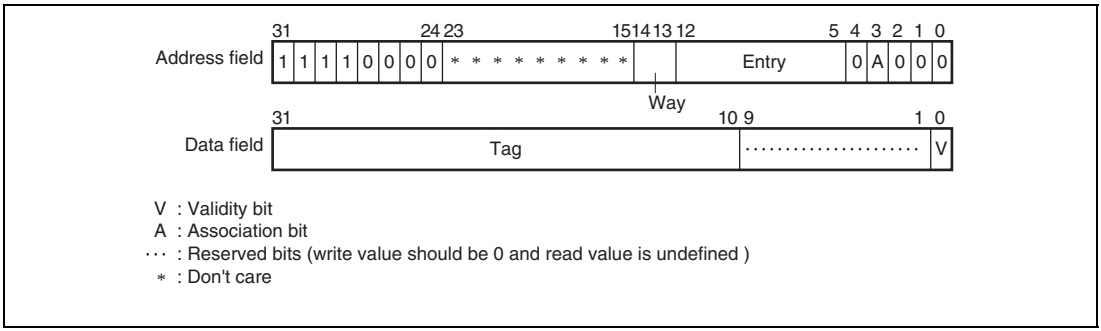


Figure 8.5 Memory-Mapped IC Address Array

8.6.2 IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

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 specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.
2. IC data array write

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

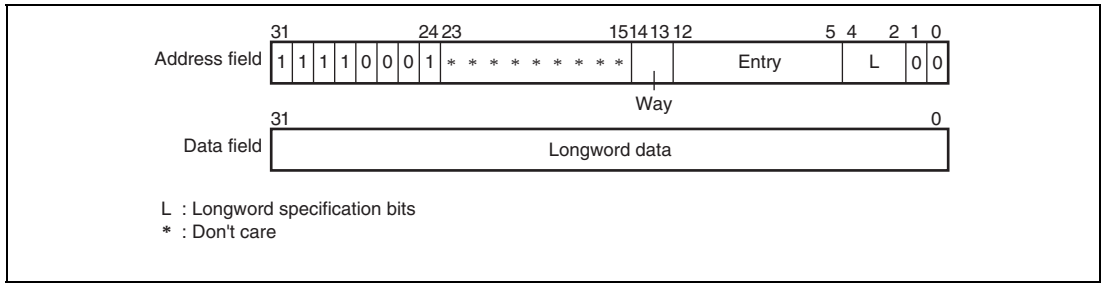


Figure 8.6 Memory-Mapped IC Data Array

8.6.3 OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC address array, and the way is specified by bits [14:13] and the entry by bits [12:5]. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address 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 to the way and entry set in the address field. In a read, associative operation is not performed 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 corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-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 way stored in the entry specified in the address field is compared with the tag specified in the data field. The way numbers of bits [14:13] in the address field are not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit in the way is 1, the U bit and V bit specified in the data field are written into the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an 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 OCBI, OCBP, or OCBWB instruction should be used to operate the OC definitely by reporting data TLB miss exception.

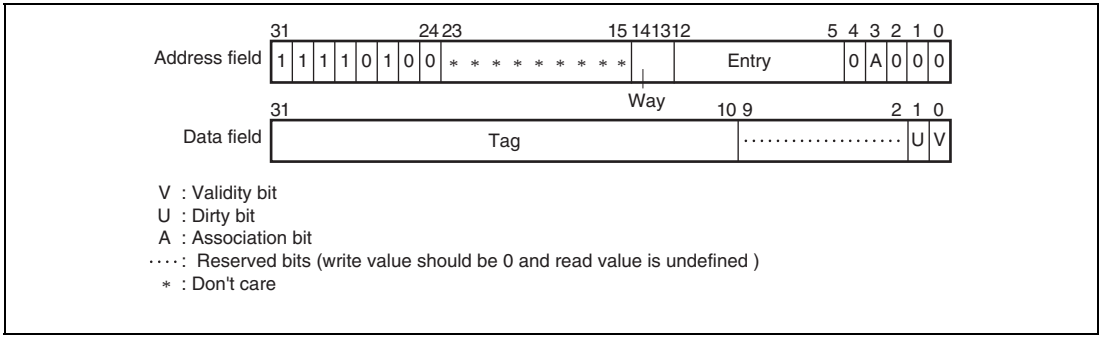


Figure 8.7 Memory-Mapped OC Address Array

8.6.4 OC Data Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, and the way is specified by bits [14:13] and the entry by bits [12:5]. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

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 specification bits in the address field in the OC entry corresponding to the way and entry set in the address field.
2. OC data array write

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

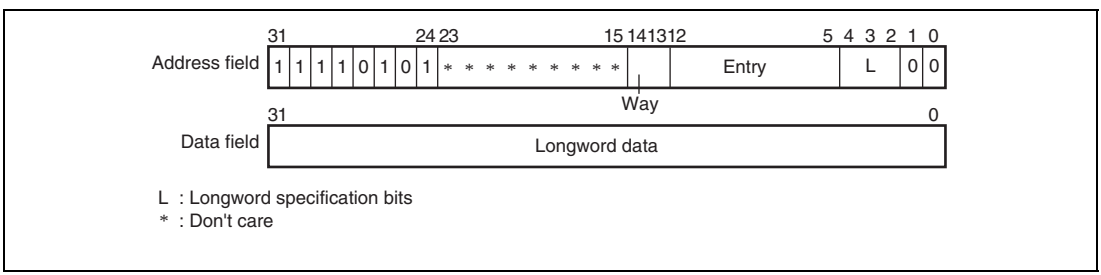


Figure 8.8 Memory-Mapped OC Data Array

8.7 Store Queues

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory.

8.7.1 SQ Configuration

There are two 32-byte store queues, SQ0 and SQ1, as shown in figure 8.9. These two store queues can be set independently.

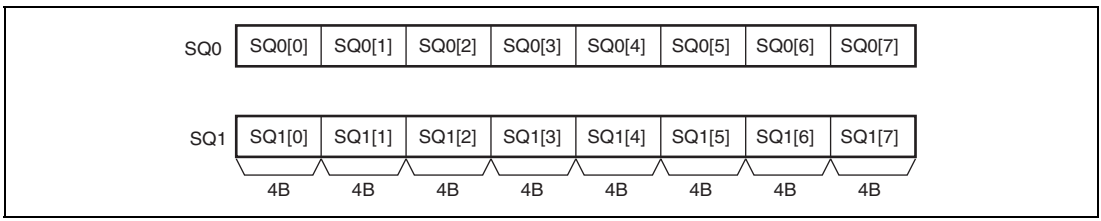


Figure 8.9 Store Queue Configuration

8.7.2 Writing to SQ

A write to the SQs can be performed using a store instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area. A longword or quadword access size can be used. The meanings of 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

8.7.3 Transfer to External Memory

Transfer from the SQs to external memory can be performed with a prefetch instruction (PREF). Issuing a PREF instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area starts a transfer from the SQs to external memory. The transfer length is fixed at 32 bytes, and the start address is always at a 32-byte boundary. While the contents of one SQ are being transferred to external memory, the other SQ can be written to without a penalty cycle. However, writing to the SQ involved in the transfer to external memory is kept waiting until the transfer is completed.

The physical address bits [28:0] of the SQ transfer destination are specified as shown below, according to whether the MMU is enabled or disabled.

- 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 transfer destination physical address in PPN. The ASID, V, SZ, SH, PR, and D bits have the same meaning as for normal address translation, but the C and WT bits have no meaning with regard to this page. When a prefetch instruction is issued for the SQ area, address translation is performed and physical address bits [28:10] are generated in accordance with the SZ bit 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] are fixed 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 PREF 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 generated 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.

8.7.4 Determination of SQ Access Exception

Determination of an exception in a write to an SQ or transfer to external memory (PREF instruction) is performed as follows according to whether the MMU is enabled or disabled. If an exception occurs during a write to an SQ, the SQ contents before the write are retained. If an exception occurs in a data transfer from an SQ to external memory, the transfer to external memory will be aborted.

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

  Operation is in accordance with the address translation information recorded in the UTLB, and the SQMD bit in MMUCR. Write type exception judgment is performed for writes to the SQs, and read type exception judgment for transfer from the SQs to external memory (using a PREF instruction). As a result, a TLB miss exception or protection violation exception is generated as required. However, if SQ access is enabled in privileged mode only by the SQMD bit in MMUCR, an address error will occur even if address translation is successful in user mode.

- 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, an address 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 means of a load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longword access 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

8.8 Notes on Using 32-Bit Address Extended Mode

In 32-bit address extended mode, the items described in this section are extended as follows.

1. The tag bits [28:10] (19 bits) in the IC and OC are extended to bits [31:10] (22 bits).
2. An instruction which operates the IC (a memory-mapped IC access and writing to the ICI bit in CCR) should be located in the P1 or P2 area. The cacheable bit (C bit) in the corresponding entry in the PMB should be 0.
3. Bits [4:2] (3 bits) for the AREA0 bit in QACR0 and the AREA1 bit in QACR1 are extended to bits [7:2] (6 bits).

Section 9 L Memory

This LSI includes on-chip L-memory which stores instructions or data.

9.1 Features

- Capacity
- Total L memory capacity is 16 Kbytes.
  The L memory is divided into two pages (pages 0 and 1).
- Memory map
  The L memory is allocated in the addresses shown in table 9.1 in both the virtual address space 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 instruction bus is used when L memory is accessed through instruction fetch. The operand bus is used when L memory is accessed through operand access. The SuperHyway bus is used for L memory access from the SuperHyway bus master module.
- Priority
  In the event of simultaneous accesses to the same page from different buses, the access requests are processed according to priority. The priority order is: SuperHyway bus > operand bus > instruction bus.

9.2 Register Descriptions

The following registers are related to L memory.

Table 9.2 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
On-chip memory control register	RAMCR	R/W	H'FF000074	H'1F000074	32
L memory transfer source address register 0	LSA0	R/W	H'FF000050	H'1F000050	32
L memory transfer source address register 1	LSA1	R/W	H'FF000054	H'1F000054	32
L memory transfer destination address register 0	LDA0	R/W	H'FF000058	H'1F000058	32
L memory transfer destination address register 1	LDA1	R/W	H'FF00005C	H'1F00005C	32

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

Table 9.3 Register Status in Each Processing State

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

9.2.1 On-Chip Memory Control Register (RAMCR)

RAMCR controls the protective functions in the L memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	—	—	—	—	—	—
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/W	R/W	R/W	R/W	R	R	R	R	R	R

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

9.2.2 L Memory Transfer Source Address Register 0 (LSA0)

When MMUCR.AT = 0 or RAMCR.RP = 0, the LSA0 specifies the transfer source physical address for block transfer to page 0 of the L memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L0SADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W	R/W

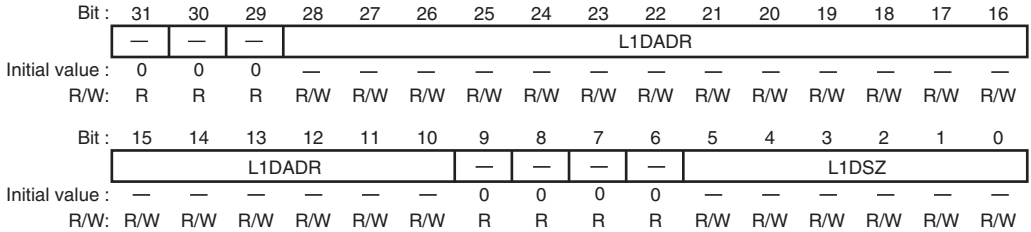
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0SADR						—	—	—	—	L0SSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	L0SSZ	Undefined	R/W	L Memory Page 0 Block Transfer Source Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L0SADR values are used as bits 15 to 10 of the transfer source physical address for block transfer to the L memory. L0SSZ[5:0] correspond to the transfer source physical addresses[15:10]. 0: The operand address is used as the transfer source physical address. 1: The L0SADR value is used as the transfer source physical address. Settable values: 111111: Transfer source physical address is specified in 1-Kbyte units. 111110: Transfer source physical address is specified in 2-Kbyte units. 111100: Transfer source physical address is specified in 4-Kbyte units. 111000: Transfer source physical address is specified in 8-Kbyte units. 110000: Transfer source physical address is specified in 16-Kbyte units. 100000: Transfer source physical address is specified in 32-Kbyte units. 000000: Transfer source physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.3 L Memory Transfer Source Address Register 1 (LSA1)

When MMUCR.AT = 0 or RAMCR.RP = 0, the LSA1 specifies the transfer source physical address for block transfer to page 1 in the L memory.



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

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	L1SSZ	Undefined	R/W	L Memory Page 1 Block Transfer Source Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L1SADR values are used as bits 15 to 10 of the transfer source physical address for block transfer to page 1 in the L memory. L1SSZ bits [5:0] correspond to the transfer source physical addresses [15:10]. 0: The operand address is used as the transfer source physical address. 1: The L1SADR value is used as the transfer source physical address. Settable values: 111111: Transfer source physical address is specified in 1-Kbyte units. 111110: Transfer source physical address is specified in 2-Kbyte units. 111100: Transfer source physical address is specified in 4-Kbyte units. 111000: Transfer source physical address is specified in 8-Kbyte units. 110000: Transfer source physical address is specified in 16-Kbyte units. 100000: Transfer source physical address is specified in 32-Kbyte units. 000000: Transfer source physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.4 L Memory Transfer Destination Address Register 0 (LDA0)

When MMUCR.AT = 0 or RAMCR.RP = 0, LDA0 specifies the transfer destination physical address for block transfer to page 0 of the L memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L0SADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0SADR						—	—	—	—	L0SSZ					
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

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

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	L0DSZ	Undefined	R/W	L Memory Page 0 Block Transfer Destination Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L0DADR values are used as bits 15 to 10 of the transfer destination physical address for block transfer to page 0 in the L memory. L0DSZ bits [5:0] correspond to the transfer destination physical address bits [15:10]. 0: The operand address is used as the transfer destination physical address. 1: The L0DADR value is used as the transfer destination physical address. Settable values: 111111: Transfer destination physical address is specified in 1-Kbyte units. 111110: Transfer destination physical address is specified in 2-Kbyte units. 111100: Transfer destination physical address is specified in 4-Kbyte units. 111000: Transfer destination physical address is specified in 8-Kbyte units. 110000: Transfer destination physical address is specified in 16-Kbyte units. 100000: Transfer destination physical address is specified in 32-Kbyte units. 000000: Transfer destination physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.2.5 L Memory Transfer Destination Address Register 1 (LDA1)

When MMUCR.AT = 0 or RAMCR.RP = 0, LDA1 specifies the transfer destination physical address for block transfer to page 1 in the L memory.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L1SADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L1SADR						—	—	—	—	L1SSZ					
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

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

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	L1DSZ	Undefined	R/W	L Memory Page 1 Block Transfer Destination Address Select When MMUCR.AT = 0 or RAMCR.RP = 0, these bits select whether the operand addresses or L1DADR values are used as bits 15 to 10 of the transfer destination physical address for block transfer to page 1 in the L memory. L1DSZ bits [5:0] correspond to the transfer destination physical addresses [15:10]. 0: The operand address is used as the transfer destination physical address. 1: The L1DADR value is used as the transfer destination physical address. Settable values: 111111: Transfer destination physical address is specified in 1-Kbyte units. 111110: Transfer destination physical address is specified in 2-Kbyte units. 111100: Transfer destination physical address is specified in 4-Kbyte units. 111000: Transfer destination physical address is specified in 8-Kbyte units. 110000: Transfer destination physical address is specified in 16-Kbyte units. 100000: Transfer destination physical address is specified in 32-Kbyte units. 000000: Transfer destination physical address is specified in 64-Kbyte units. Settings other than the ones given above are prohibited.

9.3 Operation

9.3.1 Access from the CPU and FPU

L memory access from the CPU and FPU is direct via the instruction bus and operand bus by means of the virtual address. As long as there is no conflict on the page, the L memory is accessed in one cycle.

9.3.2 Access from the SuperHyway Bus Master Module

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

9.3.3 Block Transfer

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

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

In either case, transfer rate is fixed to 32 bytes. Since the start address is always limited to a 32-byte boundary, the lower five bits of the address indicated by Rn are ignored, and are always dealt with as all 0s. In either case, other pages and cache can be accessed during block transfer, but the 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 L 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 memory area is specified to the UTLB VPN field, and to the physical address of the transfer source (in the case of the PREF instruction) or the transfer destination (in the case of the OCBWB instruction) to the PPN field. The ASID, V, SZ, SH, PR, and D bits have the same meaning as normal address conversion; however, the C and WT bits have no meaning in this page.

When the PREF instruction is issued to the L memory area, address conversion is performed in order to generate the physical address bits [28:10] in accordance with the SZ bit specification. The physical address bits [9:5] are generated from the virtual address prior to address conversion. The physical address bits [4:0] are fixed to 0. Block transfer is performed to the L memory from the external memory which is specified by these physical addresses.

When the OCBWB instruction is issued to the L memory area, address conversion is performed in order to generate the physical address bits [28:10] in accordance with the SZ bit specification. The physical address bits [9:5] are generated from the virtual address prior to address conversion. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the L memory to the external memory specified by these physical addresses.

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

When MMU is Disabled (MMUCR.AT = 0) or RAMCR.RP = 0: The transfer source physical address in block transfer to page 0 in the L memory is set in the L0SADR bits of the LSA0 register. And the L0SSZ bits in the LSA0 register choose either the virtual addresses specified through the PRFF instruction or the L0SADR values as bits 15 to 10 of the transfer source physical address. In other words, the transfer source area can be specified in units of 1 Kbyte to 64 Kbytes.

The transfer destination physical address in block transfer from page 0 in the L memory is set in the L0DADR bits of the LDA0 register. And the L0DSZ bits in the LDA0 register choose either the virtual addresses specified through the OCBWB instruction or the L0DADR values as bits 15 to 10 of the transfer destination physical address. In other words, the transfer source area can be 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 L memory.

When the PREF instruction is issued to the L memory area, the physical address bits [28:10] are generated in accordance with the LSA0 or LSA1 specification. The physical address bits [9:5] are generated from the virtual address. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the external memory specified by these physical addresses to the L memory.

When the OCBWB instruction is issued to the L memory area, the physical address bits [28:10] are generated in accordance with the LDA0 or LDA1 specification. The physical address bits [9:5] are generated from the virtual address. The physical address bits [4:0] are fixed to 0. Block transfer is performed from the L memory to the external memory specified by these physical addresses.

9.4 L Memory Protective Functions

This LSI implements the following protective functions to the L memory by using the on-chip memory access mode bit (RMD) and the on-chip memory protection enable bit (RP) in the on-chip memory control register (RAMCR).

- Protective functions for access from the CPU and FPU

  When RAMCR.RMD = 0, and the L memory is accessed in user mode, it is determined to be an address error exception.

  When MMUCR.AT = 1 and RAMCR.RP = 1, MMU exception and address error exception are 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 Occurring 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 exception
		1	*	—	MMU exception

Note: * : Don't care

9.5 Usage Notes

9.5.1 Page Conflict

In the event of simultaneous access to the same page from different buses, page conflict occurs. Although each access is completed correctly, this kind of conflict tends to lower L memory accessibility. Therefore it is advisable to provide all possible preventative software measures. For example, conflicts will not occur if each bus accesses different pages.

9.5.2 L Memory Coherency

In order to allocate instructions in the L memory, write an instruction to the L memory, execute 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 may be possible) within the range where no address error exception occurs, and cache hit/miss is possible.

9.5.3 Sleep Mode

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

9.6 Note on Using 32-Bit Address Extended Mode

In 32-bit address extended mode, L0SADR fields in LSA0, L1SADR fields in LSA1, L0DADR fields in LDA0, and L1DADR fields in LDA1 are extended from 19-bit [28:10] to 22-bit [31:10].

Section 10 Interrupt Controller (INTC)

The interrupt controller (INTC) determines the priority of interrupt sources and controls the flow of interrupt requests to the CPU (SH-4A). The INTC has registers for setting the priority of each of the interrupts and processing of interrupt requests follows the priority order set in these registers by the user.

10.1 Features

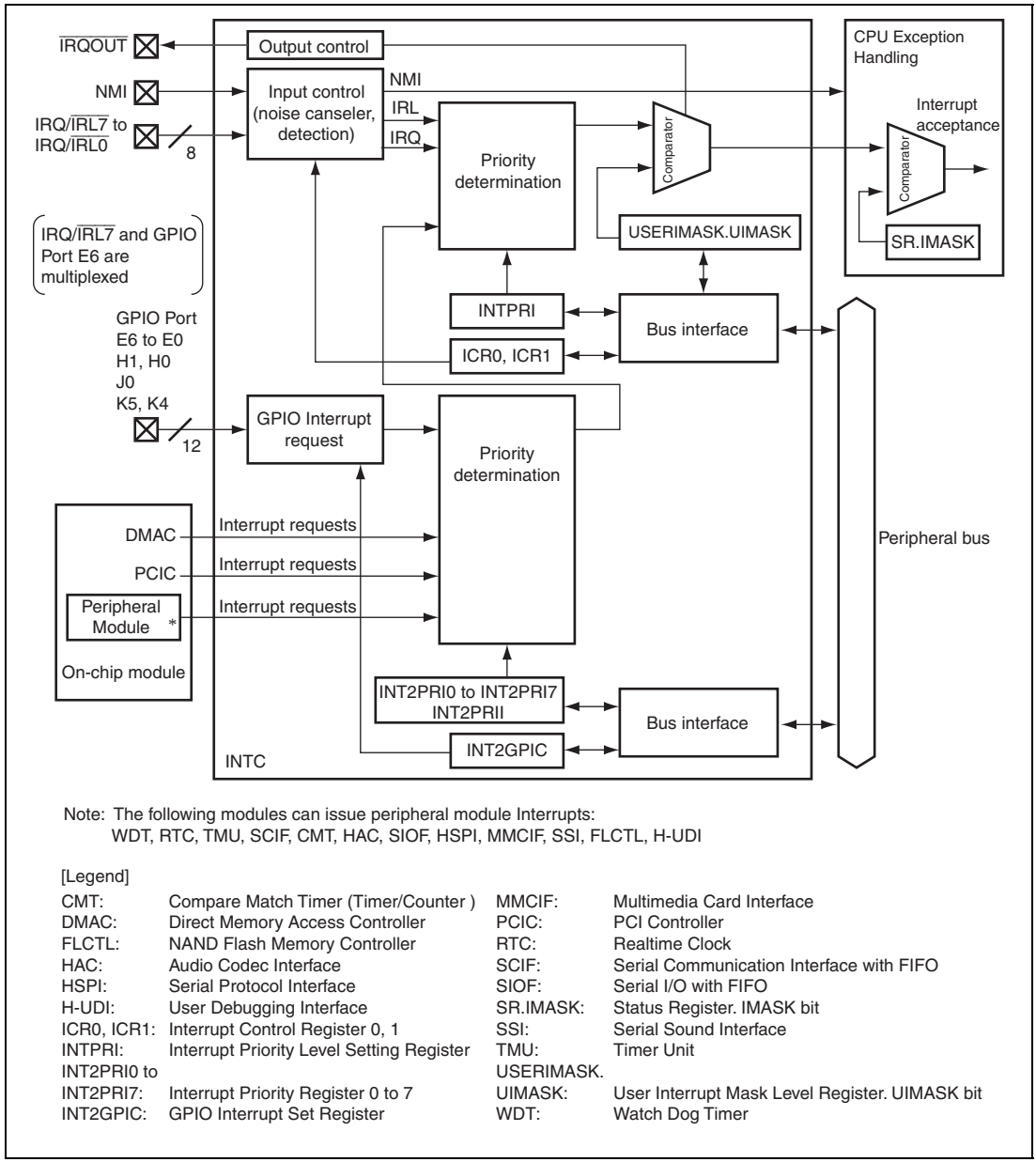
SH-4 compatible specifications

- Fifteen levels of external interrupt priority can be set
  By setting the interrupt priority registers, the priorities of external interrupts can be selected 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 interrupt 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 selected.

Extended functions for the SH-4A

- Automatically updates the IMASK bit in SR according to the accepted interrupt level
- Thirty priority levels for interrupts from on-chip modules
  By setting the interrupt priority registers (INT2PRI0 to INT2PRI7) for the on-chip module interrupts, any of 30 priority levels can be assigned to the individual requesting sources.
- User-mode interrupt disabling function
  An interrupt mask level in the user interrupt mask level register (USERIMASK) can be specified to disable interrupts which do not have higher priority than the specified mask level. 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 FIFO
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. UIMASK bit
INT2GPIC:	GPIO Interrupt Set Register	WDT:	Watch Dog Timer

Figure 10.1 Block Diagram of INTC

10.1.1 Interrupt Method

The basic flow of exception handling for interrupts is as follows.

In interrupt exception handling, the contents of the program counter (PC), status register (SR), and R15 are saved in the saved program counter (SPC), saved status register (SSR), and saved general register15 (SGR), and the CPU starts execution of the interrupt exception handling routine at the corresponding vector address. An interrupt exception handling routine is a program written by the user to handle a specific exception. The interrupt exception handling routine is terminated and control returned to the original program by executing a return-from-exception instruction (RTE). This instruction restores the contents of PC and SR and returns control to the normal processing routine at the point at which the exception occurred. The contents of SGR are not written back to 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 routine, vector base register (VBR) + H'600.
8. The flow of processing branches to the address corresponding to the interrupt within the exception handler and processing to handle the interrupt starts up.

10.1.2 Interrupt Types in INTC

Table 10.1 shows an example of the interrupt types. The INTC supports both external interrupts and on-chip module interrupts.

External interrupts refer to the interrupts input through the external NMI, IRL, and IRQ pins.

The IRQ and IRL interrupts are assigned to the same pins in the SH7780. The pin functions are selected to suit the system configuration.

Ether level-sense, or the rising or falling edge, can be selected for the detection of IRQ input.

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 $\overline{\text{IRL[7:4]}}$ pin = H'0 means the external pin input levels are: $\overline{\text{IRL[7]}}$ pin = Low $\overline{\text{IRL[6]}}$ pin = Low $\overline{\text{IRL[5]}}$ pin = Low $\overline{\text{IRL[4]}}$ pin = Low so the priority level is 15 (H'F) (see table 10.11)	H'200	$\overline{\text{IRL[7:4]}}$ pin = H'0	High
					$\overline{\text{IRL[3:0]}}$ pin = H'0	
				H'220	$\overline{\text{IRL[7:4]}}$ pin = H'1	
					$\overline{\text{IRL[3:0]}}$ pin = H'1	
				H'240	$\overline{\text{IRL[7:4]}}$ pin = H'2	
					$\overline{\text{IRL[3:0]}}$ pin = H'2	
				H'260	$\overline{\text{IRL[7:4]}}$ pin = H'3	
					$\overline{\text{IRL[3:0]}}$ pin = H'3	
				H'280	$\overline{\text{IRL[7:4]}}$ pin = H'4	
					$\overline{\text{IRL[3:0]}}$ pin = H'4	
				H'2A0	$\overline{\text{IRL[7:4]}}$ pin = H'5	
					$\overline{\text{IRL[3:0]}}$ pin = H'5	
				H'2C0	$\overline{\text{IRL[7:4]}}$ pin = H'6	
					$\overline{\text{IRL[3:0]}}$ pin = H'6	
				H'2E0	$\overline{\text{IRL[7:4]}}$ pin = H'7	
					$\overline{\text{IRL[3:0]}}$ pin = H'7	
				H'300	$\overline{\text{IRL[7:4]}}$ pin = H'8	
					$\overline{\text{IRL[3:0]}}$ pin = H'8	Low

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks	
External interrupts	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'320	$\overline{IRL[7:4]}$ pin = H'9	High
					$\overline{IRL[3:0]}$ pin = H'9	
				H'340	$\overline{IRL[7:4]}$ pin = H'A	
					$\overline{IRL[3:0]}$ pin = H'A	
				H'360	$\overline{IRL[7:4]}$ pin = H'B	
					$\overline{IRL[3:0]}$ pin = H'B	
				H'380	$\overline{IRL[7:4]}$ pin = H'C	
					$\overline{IRL[3:0]}$ 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	Low
	IRQ interrupt	8	Values set in INTPRI	H'240	IRQ[0]	High
				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]	Low
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	

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts	DMAC(0)	7(5/7)	Values set in INT2PRI0 to INT2PRI7	H'640	DMINT0*[2]
				H'660	DMINT1*[2]
				H'680	DMINT2*[2]
				H'6A0	DMINT3*[2]
				H'6C0	DMAE*[2]
	SCIF-ch0	4		H'700	ERI0*[2]
				H'720	RXI0*[2]
				H'740	BRI0*[2]
				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

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts	MMCIF	4	Values set in INT2PRI0 to INT2PRI7	H'D00	FSTAT
				H'D20	TRAN
				H'D40	ERR
				H'D60	FRDY
	DMAC(1)	6 (4/6)		H'D80	DMINT8*2
				H'DA0	DMINT9*2
				H'DC0	DMINT10*2
				H'DE0	DMINT11*2
	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 E2)
				H'FA0	GPIOI1 (Port E3 to E5)
				H'FC0	GPIOI2 (Port H0, 1, Port J0, Port K4)
				H'FE0	GPIOI3 (Port E6, Port K5)

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

FLTRQ0: FLCTL data FIFO transfer request interrupt
FLTRQ1: FLCTL control code FIFO transfer request interrupt

10.2 Input/Output Pins

Table 10.2 shows the pin configuration.

Table 10.2 INTC Pin Configuration

Pin Name	Function	I/O	Description
NMI	Nonmaskable interrupt input pin	Input	Nonmaskable interrupt request signal input
IRQ/IRL3 to IRQ/IRL0	External interrupt input pin	Input	Interrupt request signal input IRL [3:0] 4-bit level-encoded interrupt input when ICR0.IRLM0 = 0; IRQ3 to IRQ0 individual pin interrupt input 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 interrupt input when ICR0.IRLM1 = 0; IRQ7 to IRQ4 individual pin interrupt input when ICR0.IRLM1 = 1
$\overline{\text{IRQOUT}}$*[2]	Interrupt request output	Output	Indicates that an interrupt request has been generated This pin is asserted even if the CPU does not accept the interrupt request, except if the interrupt is masked, 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.

10.3 Register Descriptions

Table 10.3 shows the INTC register configuration. Table 10.4 shows the register states in each operating mode.

Table 10.3 INTC Register Configuration

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt control register 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32	Pck
Interrupt control register 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32	Pck
Interrupt priority register	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32	Pck
Interrupt source register	INTREQ	R/(W)	H'FFD0 0024	H'1FD0 0024	32	Pck
Interrupt mask register 0	INTMSK0	R/W	H'FFD0 0044	H'1FD0 0044	32	Pck
Interrupt mask register 1	INTMSK1	R/W	H'FFD0 0048	H'1FD0 0048	32	Pck
Interrupt mask register 2	INTMSK2	R/W	H'FFD4 0080	H'1FD4 0080	32	Pck
Interrupt mask clear register 0	INTMSKCLR0	R/W	H'FFD0 0064	H'1FD0 0064	32	Pck
Interrupt mask clear register 1	INTMSKCLR1	R/W	H'FFD0 0068	H'1FD0 0068	32	Pck
Interrupt mask clear register 2	INTMSKCLR2	R/W	H'FFD4 0084	H'1FD4 0084	32	Pck
NMI flag control register	NMIFCR	R/(W)	H'FFD0 00C0	H'1FD0 00C0	32	Pck
User interrupt mask level register	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32	Pck
On-chip module interrupt priority registers	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32	Pck
	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32	Pck
	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32	Pck
	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32	Pck
	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32	Pck
	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32	Pck
	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32	Pck
	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32	Pck
Interrupt source register (not affected by the mask state)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32	Pck

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync. Clock
Interrupt source register (affected by the mask state)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32	Pck
Interrupt mask register	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32	Pck
Interrupt mask clear register	INT2MSKCR	R/W	H'FFD4 003C	H'1FD4 003C	32	Pck
On-chip module interrupt source registers	INT2B0	R	H'FFD4 0040	H'1FD4 0040	32	Pck
	INT2B1	R	H'FFD4 0044	H'1FD4 0044	32	Pck
	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32	Pck
	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32	Pck
	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32	Pck
	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32	Pck
	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32	Pck
	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32	Pck
GPIO interrupt set register	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32	Pck

Notes: Pck is the peripheral clock.

(W) : To clear the flag, 0 can only be written to the corresponding bit.

Table 10.4 Register States in Each Operating Mode

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep by SLEEP Instruction
Interrupt control register 0	ICR0	H'x000 0000*	H'x000 0000*	Retained
Interrupt control register 1	ICR1	H'0000 0000	H'0000 0000	Retained
Interrupt priority register	INTPRI	H'0000 0000	H'0000 0000	Retained
Interrupt source register	INTREQ	H'0000 0000	H'0000 0000	Retained
Interrupt mask register 0	INTMSK0	H'FF00 0000	H'FF00 0000	Retained
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

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep by SLEEP Instruction
On-chip module interrupt source registers	INT2B0	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B1	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B2	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B3	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B4	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B5	H'xxxx xxxx	H'xxxx xxxx	Retained
	INT2B6	H'xxxx xxxx	H'xxxx xxxx	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 NMI pin.

10.3.1 Interrupt Control Register 0 (ICR0)

ICR0 is a 32-bit readable and partially writable register that sets the input signal detection mode for the external interrupt input pins (IRQ/IRL [7:0]) and NMI pin, and indicates the level being input on the NMI pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMIL	MAI	—	—	—	—	NMIB	NMIE	IRLM0	IRLM1	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	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 NMI pin. Reading this bit allows the user to know the NMI pin 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 the level initially being input on the NMI pin.
30	MAI	0	R/W	MAI (mask all interrupts) Interrupt Mask Specifies whether all interrupts are masked while the NMI pin is at the low level regardless of the setting of the BL bit in SR of the CPU. 0: Interrupts remain enabled even when the NMI pin goes low 1: Interrupts are disabled when the NMI pin goes low
29 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Name	Initial Value	R/W	Description
25	NMIB	0	R/W	NMI Block Mode Selects whether an NMI interrupt is held until the BL bit in SR is cleared to 0 or detected immediately when the BL bit in SR of the CPU is set to 1. 0: An NMI interrupt is held when the BL bit in SR is set to 1 (initial value) 1: An NMI interrupt is not held when the BL bit in SR is set to 1 Note: If interrupts are accepted with the BL bit in SR set to 1, information saved for any previous exception (SSR, SPC, SGR, and INTEVT) is lost.
24	NMIE	0	R/W	NMI Edge Select Selects whether an interrupt request signal to the NMI pin is detected at the rising edge or the falling edge. 0: An interrupt request is detected at the falling edge of NMI input (initial value) 1: An interrupt request is detected at the rising edge of NMI input Note: NMI interrupt is not detected for at least six bus 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 used as 4-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 level-encoded interrupt requests (IRL [3:0] interrupt; initial value) 1: IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as four independent interrupt requests (IRQ [n] interrupt; n = 3 to 0) Note: The level-encoded IRL interrupt is not detected unless the same pin levels are sampled in four consecutive bus clock cycles.

Bit	Name	Initial Value	R/W	Description
22	IRLM1	0	R/W	IRL Pin Mode 1 Selects whether IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as 4-bit level-encoded interrupt requests or as four independent interrupts. 0: IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as the 4-bit level-encoded interrupt requests (IRL [7:4] interrupt; initial value) 1: IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ are used as four independent interrupt requests (IRQ [n] interrupt; n = 7 to 4) Note: The level-encoded IRL interrupt is not detected unless the same pin levels are sampled in four 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 function for level-encoded IRL and level-sense IRQ interrupts. 0: IRQ level-sense and IRL interrupt requests are held (initial value) 1: IRQ level-sense and IRL interrupt requests are not held (compatible with current SH-4 behavior for IRQ in level-sense mode and IRL level-encoded interrupts) Note: This setting is only valid for IRQ/$\overline{\text{IRL}}$ pins used as a 4-bit level-encoded IRL interrupt or as level-sense IRQ interrupts. When using this function, also refer to sections 10.4.2 IRQ Interrupts, 10.4.3 IRL Interrupts, and 10.7. Usage Notes.
20 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.3.2 Interrupt Control Register 1 (ICR1)

ICR1 is a 32-bit readable/writable register that specifies the individual input signal detection modes of external interrupt input pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$. These settings are only valid for pins configured as individual IRQ interrupts; that is, for pins for which the IRLM0 or IRLM1 bit in ICR0 is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0S		IRQ1S		IRQ2S		IRQ3S		IRQ4S		IRQ5S		IRQ6S		IRQ7S	
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/W

Bit:	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
R/W:	R	R	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 pin interrupt signal on the IRQ/IRL7 to IRQ/IRL0 pins is detected on rising or falling edges, or at the high or low level. 00: The interrupt request is detected on falling edges of the IRQn input. 01: The interrupt request is detected on rising edges of the IRQn input. 10: The interrupt request is detected when the IRQn input is at the low level. 11: The interrupt request is detected when the IRQn input is at the low level. Note: When either level is selected, the IRQ level interrupt request is not detected unless the same level is sampled in three consecutive bus-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 value should always be 0.

Note: When an IRQ pin is set for level input (IRQnS1 = 1), the interrupt source is held until the CPU accepts the interrupt (this is also true for other interrupts). Therefore, even if an interrupt source is disabled before this LSI returns from sleep mode, branching of processing to the interrupt handler when this LSI returns from sleep mode is guaranteed. A held interrupt can be cleared by setting the corresponding interrupt mask bit (the IM bit in the interrupt mask register) to 1.

10.3.3 Interrupt Priority Register (INTPRI)

INTPRI is a 32-bit readable/writable register used to set the priorities of IRQ[7:0] (as levels from 15 to 0). These settings are only valid for IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ when 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	17	16
	IP0				IP1				IP2				IP3			
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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IP4				IP5				IP6				IP7			
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/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 request.
27 to 24	IP1	H'0	R/W	Set the priority of IRQ1 as an individual pin interrupt request.
23 to 20	IP2	H'0	R/W	Set the priority of IRQ2 as an individual pin interrupt request.
19 to 16	IP3	H'0	R/W	Set the priority of IRQ3 as an individual pin interrupt request.
15 to 12	IP4	H'0	R/W	Set the priority of IRQ4 as an individual pin interrupt request.
11 to 8	IP5	H'0	R/W	Set the priority of IRQ5 as an individual pin interrupt request.
7 to 4	IP6	H'0	R/W	Set the priority of IRQ6 as an individual pin interrupt request.
3 to 0	IP7	H'0	R/W	Set the priority of IRQ7 as an individual pin interrupt request.

Interrupt priorities should be established by setting values from H'F to H'1 in each of the 4-bit fields. A larger value corresponds to a higher priority. When the value H'0 is set in a field, the corresponding interrupt is masked (initial value).

10.3.4 Interrupt Source Register (INTREQ)

INTREQ is a 32-bit readable and conditionally writable register that indicates which of the IRQ [n] (n = 0 to 7) interrupts is currently asserting a request for the INTC.

Even if an interrupt is masked by the setting in INTPRI or INTMSK0, operation of the corresponding INTREQ bit is not affected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7	—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit:	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
R/W:	R	R	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 IRQ interrupt pin is not asserted.
29	IR2	0	R/(W)		1: The corresponding IRQ interrupt pin is asserted, but the CPU has not accepted the interrupt yet.
28	IR3	0	R/(W)	1: The corresponding IRQ interrupt request has been detected.	
27	IR4	0	R/(W)		
26	IR5	0	R/(W)		
25	IR6	0	R/(W)	[When writing]*	Values written have no effect.
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 should always be 0.	

10.3.5 Interrupt Mask Registers (INTMSK0 to INTMSK2)

INTMSK0 to INTMSK2 are 32-bit readable and conditionally writable registers that control mask settings for the interrupt requests. To clear a mask setting for interrupts, write 1 to the corresponding bit in INTMSKCLR0 to INTMSKCLR2. Writing 0 to a bit in INTMSK0 to INTMSK2 has no effect.

- Interrupt mask register 0 (INTMSK0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM00	IM01	IM02	IM03	IM04	IM05	IM06	IM07	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	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	R	R

Bit:	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
R/W:	R	R	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.	[When reading] 0: No masking 1: Masking [When writing] 0: No effect 1: Masks the interrupt
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.	

Bit	Name	Initial Value	R/W	Description
23 to 0		All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When the 4-bit encoded interrupt inputs are to be used, write B'1111 to the IM [03:00] or IM [07:04] bits to mask single-pin interrupts in the ranges IRQ/IRL [3:0] or IRQ/IRL [7:4], respectively.

- Interrupt mask register 1 (INTMSK1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM10	IM11	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	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 are accepted. 1: The interrupts are masked. [When writing] 0: No effect 1: Masks the interrupts
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 value should always be 1.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- Interrupt mask register 2 (INTMSK2)

INTMSK2 settings are valid for particular IRL interrupt codes generated by the pattern of input signals on pins $\overline{IRL7}$ to $\overline{IRL4}$ or $\overline{IRL3}$ to $\overline{IRL0}$ and when all IRL interrupts from the corresponding set of pins are not masked by the setting in INTMSK1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM015	IM014	IM013	IM012	IM011	IM010	IM009	IM008	IM007	IM006	IM005	IM004	IM003	IM002	IM001	—
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IM115	IM114	IM113	IM112	IM111	IM110	IM109	IM108	IM107	IM106	IM105	IM104	IM103	IM102	IM101	—
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	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).
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).

[When reading]

0: The interrupt is acceptable.

1: The interrupt is masked.

[When writing]

0: No effect

1: Masks the interrupt

Bit	Name	Initial Value	R/W	Description	
23	IM007	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLLL (H'8).	[When reading] 0: The interrupt is acceptable. 1: The interrupt is masked. [When writing] 0: No effect 1: Masks the interrupt Initial value: 0
22	IM006	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLLH (H'9).	
21	IM005	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLHL (H'A).	
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 should 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 interrupt 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).	

Bit	Name	Initial Value	R/W	Description
11	IM111	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHLL (H'4).
10	IM110	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHLH (H'5).
9	IM109	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHHL (H'6).
8	IM108	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHHH (H'7).
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 should always be 0.

[When reading]
0: The interrupt is acceptable.
1: The interrupt is masked.
[When wswriting]
0: No effect
1: Masks the interrupt
Initial value: 0

Note: 'H' and 'L' indicate high- and low-level input on the corresponding IRQ/$\overline{IRL}$ pin. For the relationship between the input signal level and the priority level, refer to table 10.11.

10.3.6 Interrupt Mask Clear Registers (INTMSKCLR0 to INTMSKCLR2)

INTMSKCLR0 to INTMSKCLR2 are 32-bit write-only registers that clear the mask settings for each interrupt request. Values read are undefined.

- Interrupt mask clear register 0 (INTMSKCLR0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC00	IC01	IC02	IC03	IC04	IC05	IC06	IC07	—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit:	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
R/W:	R	R	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 value should always be 0.

[When reading]
Values read are undefined.

[When writing]
0: No effect
1: Clears the corresponding interrupt mask (enables the interrupt)

- Interrupt mask clear register 1 (INTMSKCLR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC10	IC11	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

- Interrupt mask clear register 2 (INTMSKCLR2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC015	IC014	IC013	IC012	IC011	IC010	IC009	IC008	IC007	IC006	IC005	IC004	IC003	IC002	IC001	—
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC115	IC114	IC113	IC112	IC111	IC110	IC109	IC108	IC107	IC106	IC105	IC104	IC103	IC102	IC101	—
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	Name	Initial Value	R/W	Description
31	IC015	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLLL (H'0).
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).

[When reading]

Values read are undefined.

[When writing]

0: No effect

1: Clears the corresponding interrupt mask (enables the interrupt)

Bit	Name	Initial Value	R/W	Description
22	IC006	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLLH (H'9).
21	IC005	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLHL (H'A).
20	IC004	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLHH (H'B).
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 should always be 0.
15	IC115	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLLL (H'0).
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).

[When reading]
Values read are undefined.

[When writing]
0: No effect
1: Clears the corresponding interrupt mask (enables the interrupt)

Bit	Name	Initial Value	R/W	Description
10	IC110	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHLH (H'5).
9	IC109	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHHL (H'6).
8	IC108	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHHH (H'7).
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 should always be 0.

[When reading]
Values read are undefined.

[When writing]
0: No effect
1: Clears the corresponding interrupt mask (enables the Interrupt)

Note: 'H' and 'L' indicate high- and low-level input on the corresponding IRQ/$\overline{IRL}$ pin. For the relationship between the input signal level and the priority level, refer to table 10.11.

10.3.7 NMI Flag Control Register (NMIFCR)

NMIFCR is a 32-bit readable and conditionally writable register that has an NMI flag (NMIFL bit) which can be read or cleared by software. The NMIFL bit is automatically set to 1 by hardware when an NMI interrupt is detected by the INTC. Writing 0 to the NMIFL bit clears it.

The value of the NMIFL bit does not affect acceptance of the NMI by the CPU. Although an NMI request detected by the INTC is cleared when the CPU accepts the NMI, the NMIFL bit is not cleared automatically. Even if 0 is written to the NMIFL bit before the NMI request is accepted by the CPU, the NMI request is not canceled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMIL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NMIFL
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	R/(W)

Bit:	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
R/W:	R	R	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 NMI pin; that is, this bit is read to determine the level on the NMI 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 value should always be 0.

Bit	Name	Initial Value	R/W	Description
16	NMIFL	0	R/(W)	NMI Interrupt Request Signal Detection Indicates whether an NMI interrupt request signal has been detected. This bit is automatically set to 1 when the INTC detects an NMI interrupt request. Write 0 to clear the bit. Writing 1 to this bit has no effect. [When reading] 1: NMI has been detected 0: NMI has not been detected [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 value should always be 0.

10.3.8 User Interrupt Mask Level Register (USERIMASK)

USERIMASK is a 32-bit readable and conditionally writable register that sets the acceptable interrupt level. When addresses in area 7 are accessed by using the MMU's address translation function, USERIMASK can be accessed in user mode. Since only USERIMASK is allocated to the 64-Kbyte page (other INTC registers are allocated to a different area), it can be set to be accessible in user mode.

Interrupts with priority levels lower than the level set in the UIMASK bits are masked. When the value H'F is set in the UIMASK bit, all interrupts other than the NMI are masked.

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

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

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 power-on reset or manual reset.

To prevent incorrect writing, the value written to bits 31 to 24 must always be set to H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WKEY (H'A5)								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

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

Bit	Name	Initial Value	R/W	Description
31 to 24	WKEY	H'00	R/W	When writing a value to bits 7 to 4, always write H'A5 here. These bits are always read as 0.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	UIMASK	H'0	R/W	Interrupt Mask Level Mask interrupts with priority levels lower than the level set in the UIMASK bits.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value 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 disabled. This function can be used to disable less urgent interrupts during the execution of urgent tasks that run in user mode, e.g. device drivers, and thus reduce times until completion for such tasks.

USERIMASK is allocated to a different 64-Kbyte page than that to which the other INTC registers are allocated. When accessing this register in user mode, translate the address through the MMU. In a system with a multitasking OS, the memory-protection functions of the MMU must be used to control which processes have access to USERIMASK. When terminating a task or switching to another task, be sure to clear USERIMASK to 0 beforehand. If the UIMASK bits are erroneously left set at a value other than zero, interrupts which are not higher in priority than the UIMASK level remain disabled, and operation may be incorrect (for example, the OS might be unable to 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 interrupts higher than that of the B-priority interrupts.
   A. Interrupts to be accepted by device drivers (interrupts for use by the operating system: a 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 only be accessed by the device driver for which interrupts should be disabled.
3. Branch to the device driver.

4. Set the UIMASK bits so that B-type interrupts are masked during execution of the device driver that is operating in user mode.
5. Process interrupts with a high priority in the device driver.
6. Clear the UIMASK bits to 0 to return from processing in the device driver.

10.3.9 On-chip Module Interrupt Priority Registers (INT2PRI0 to INT2PRI7)

INT2PRI0 to INT2PRI7 are 32-bit readable/writable registers used to set priorities (levels 31 to 0) for the on-chip module interrupts. INT2PRI0 to INT2PRI7 are initialized to H'0000 0000 by a reset.

INT2PRI0 to INT2PRI7 contain five-bit fields that are used to set up to 30 priority levels for the individual interrupt sources (interrupt requests are masked by settings of H'00 and H'01).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—						—	—	—					
Initial value:	0	0	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/W	R/W

Bit:	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
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/W	R/W

Table 10.5 shows the correspondence between interrupt request sources and bits in INT2PRI0 to 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 channel 2 input capture
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 when the bits are set to H'00 or H'01. For details, see the description above.

10.3.10 Interrupt Source Register (INT2A0: Not affected by Mask States)

INT2A0 is a 32-bit read-only register that indicates interrupt states of interrupt source modules regardless of the corresponding mask states. Even if interrupt masking is set in the interrupt mask register, corresponding bits in INT2A0 indicate source modules for which interrupt conditions have been satisfied (the corresponding interrupt is not generated). When sources that are masked should not be indicated, use INT2A1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—										
Initial value:	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	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	1	0
					—	—				—						
Initial value:	—	—	—	—	0	0	—	—	—	0	—	—	—	—	—	—
R/W:	R	R	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 sources for the individual peripheral modules (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
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	

Bit	Initial Value	R/W	Source	Function	Description
19	—	R	PCIC (5)	Indicates PCIERR and PCIPWD3 to PCIPWD0 interrupt sources	Indicates interrupt sources for the individual peripheral modules (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
18	—	R	PCIC (4)	Indicates PCIINTD interrupt source	
17	—	R	PCIC (3)	Indicates PCIINTC interrupt source	
16	—	R	PCIC (2)	Indicates PCIINTB 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	

Bit	Initial Value	R/W	Source	Function	Description
2	—	R	RTC	Indicates RTC interrupt source	Indicates interrupt sources for the individual peripheral modules (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
1	—	R	TMU channels 3 to 5	Indicates the TMU channel 3 to 5 interrupt sources	
0	—	R	TMU channels 0 to 2	Indicates the TMU channel 0 to 2 interrupt sources	

10.3.11 Interrupt Source Register (INT2A1: Affected by Mask States)

INT2A is a 32-bit read-only register that indicates interrupt states of interrupt source modules for which the interrupts are not masked. Note that if an interrupt mask is set in the interrupt mask register, INT2A1 does not indicate the interrupt state of the source module in the corresponding bit. To check whether interrupts have been generated, regardless of the state of the interrupt mask register, use INT2A0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—										
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	R	R	R

Bit:	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
R/W:	R	R	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 sources for the individual peripheral modules (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
25	0	R	GPIO	Indicates GPIO interrupt source	
24	0	R	FLCTL	Indicates FLCTL interrupt source	
23	0	R	SSI	Indicates SSI interrupt source	
22	0	R	MMCIF	Indicates MMC interrupt source	
21	0	R	HSPI	Indicates HSPI interrupt source	
20	0	R	SIOF	Indicates SIOF interrupt source	

Bit	Initial Value	R/W	Source	Function	Description
19	0	R	PCIC (5)	Indicates PCIERR and PCIPWD3 to PCIPWD0 interrupt sources	Indicates interrupt sources for the individual peripheral modules (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
18	0	R	PCIC (4)	Indicates PCIINTD interrupt source	
17	0	R	PCIC (3)	Indicates PCIINTC interrupt source	
16	0	R	PCIC (2)	Indicates PCIINTB 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	

Bit	Initial Value	R/W	Source	Function	Description
2	0	R	RTC	Indicates the RTC interrupt source	Indicates interrupt sources for the individual peripheral modules (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupt 1: An interrupt has been generated Note: Interrupt sources can also be identified by directly reading the INTEVT code that is sent to the CPU. In this case, reading INT2A0 is not necessary.
1	0	R	TMU channels 3 to 5	Indicates the TMU channel 3 to 5 interrupt source	
0	0	R	TMU channels 0 to 2	Indicates the TMU channel 0 to 2 interrupt source	

10.3.12 Interrupt Mask Register (INT2MSKR)

INT2MSKR is a 32-bit readable/writable register that sets interrupt masking for each of the sources indicated in the interrupt source register. The CPU is not notified of interrupts for which 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	17	16
	—	—	—	—	—	—										
Initial value:	1	1	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/W	R/W

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

Table 10.8 shows the correspondence between bits in INT2MSKR and interrupt masking.

Table 10.8 Correspondence between Bits in INT2MSKR and Interrupt Masking

Bit	Initial Value	R/W	Target	Function	Description
31 to 26	All 1	R	(Reserved)	These bits are always read as 1. The write value should always be 1.	Masks interrupts for individual modules. [When reading] 0: No masking 1: Masking [When writing] 0: No effect 1: Masks the interrupt
25	1	R/W	GPIO	Masks the GPIO interrupt	
24	1	R/W	FLCTL	Masks the FLCTL interrupt	
23	1	R/W	SSI	Masks the SSI interrupt	
22	1	R/W	MMCIF	Masks the MMC interrupt	
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	

Bit	Initial Value	R/W	Target	Function	Description
2	1	R/W	RTC	Masks the RTC interrupt	Masks interrupts for individual modules. [When reading] 0: No masking 1: Masking [When writing] 0: No effect 1: Masks the interrupt
1	1	R/W	TMU channels 3 to 5	Masks TMU channels 3 to 5 interrupts	
0	1	R/W	TMU channels 0 to 2	Masks TMU channels 0 to 2 interrupts	

10.3.13 Interrupt Mask Clear Register (INT2MSKCR)

INT2MSKCR is a 32-bit write-only register used to clear mask settings in the interrupt mask register. Setting a bit in this register to 1 clears the masking of the corresponding interrupt source. The bits of this register are always read as 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—										
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/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	1	0
					—	—				—						
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	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.9 shows the correspondence between bits in INT2MSKCR and interrupt mask clearing.

Table 10.9 Correspondence between Bits in INT2MSKCR and Interrupt Mask Clearing

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 interrupt masking for individual modules. [When reading] Always 0 [When writing] 0: Invalid 1: Interrupt mask is cleared
25	0	R/W	GPIO	Clears the GPIO interrupt masking	
24	0	R/W	FLCTL	Clears the FLCTL interrupt masking	
23	0	R/W	SSI	Clears the SSI interrupt masking	
22	0	R/W	MMCIF	Clears the MMC interrupt masking	
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	

Bit	Initial Value	R/W	Target	Function	Description
16	0	R/W	PCIC (2)	Clears the PCIINTB interrupt masking	Clears interrupt masking for each peripheral module. [When reading] Always 0 [When writing] 0: Invalid 1: Interrupt mask is cleared
15	0	R/W	PCIC (1)	Clears the PCIINTA interrupt masking	
14	0	R/W	PCIC (0)	Clears the PCISERR interrupt masking	
13	0	R/W	HAC	Clears the HAC interrupt masking	
12	0	R/W	CMT	Clears the CMT 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	

10.3.14 On-chip Module Interrupt Source Registers (INT2B0 to INT2B7)

INT2B0 to INT2B7 are 32-bit read-only registers that indicate more details on sources within interrupt source modules for which the interrupt state is indicated in the interrupt source register. INT2B0 to INT2B7 are not affected by the state of masking in the interrupt mask register. Bits for modules in the interrupt mask and interrupt enable registers enable and disable the operation of the corresponding detailed interrupt source bits.

The initial values of these registers are undefined (reserved bits are always read as 0).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	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	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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 sources. This register indicates the TMU interrupt sources even if the mask setting for TMU interrupts has been made in the interrupt mask 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	

INT2B1: Indicates detailed interrupt sources for the RTC.

Module	Bit	Name	Detailed Source	Description
RTC	31 to 3	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates RTC interrupt sources. This register indicates the RTC interrupt sources even if the mask setting for RTC interrupts has been made in the interrupt mask register.
	2	CUI	RTC carry interrupt	
	1	PRI	RTC period interrupt	
	0	ATI	RTC alarm 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 sources. This register indicates the SCIF interrupt sources even if the mask setting for SCIF interrupts has been made in the interrupt mask 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	

INT2B3: Indicates detailed interrupt sources for the DMAC.

Module	Bit	Name	Detailed Source	Description
DMAC	31 to 14	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates DMAC interrupt sources. This register indicates DMAC interrupt sources even if a mask setting for DMAC interrupts has been made in the interrupt mask register.
	13	DMAE1	DMA channels 6 to 11 address error interrupt	
	12	DMAE0	DMA channels 0 to 5 address 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 half finished with the condition of specified to the corresponding TCR.

INT2B4: Indicates detailed interrupt sources for the PCIC.

Module	Bit	Name	Detailed Source	Description
PCIC	31 to 10	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates PCIC interrupt sources. This register indicates the PCIC interrupt sources even if a mask setting for PCIC interrupts has been made in the interrupt mask register.
	9	PWD0	PCIC power state D0 state interrupt	
	8	PWD1	PCIC power state D1 state interrupt	
	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	

INT2B5: Indicates detailed interrupt sources for the MMC.

Module	Bit	Name	Detailed Source	Description
MMCIF	31 to 4	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates MMC interrupt sources. This register indicates MMC interrupt sources even if the mask setting for MMC interrupts has been made in the interrupt mask register.
	3	FRDY	FIFO ready interrupt	
	2	ERR	CRC error interrupt, data timeout error interrupt, or command 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	

INT2B6: Indicates detailed interrupt sources for the FLCTL.

Module	Bit	Name	Detailed Source	Description
FLCTL	31 to 4	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates FLCTL interrupt sources. This register indicates FLCTL interrupt sources even if the mask setting for FLCTL interrupts has been made in the interrupt mask register.
	3	FLTRQ1	FLCTL FLECFIFO transfer request interrupt	
	2	FLTRQ0	FLCTL TLDTFIFO transfer request interrupt	
	1	FLTEND	FLCTL transfer end interrupt	
	0	FLSTE	FLCTL status error interrupt or ready/busy timeout error interrupt	

INT2B7: Indicates detailed interrupt sources for the GPIO.

Module	Bit	Name	Detailed Source	Description
GPIO	31 to 26	—	(Reserved)	Indicates GPIO interrupt sources. This register indicates the states of GPIO interrupt sources even if GPIO interrupts have been masked by the setting in the interrupt mask register.
	25	PORTE6I	GPIO interrupt from port E pin 6.	
	24	PORTK5I	GPIO interrupt from port K pin 5.	
	23 to 20	—	(Reserved) These bits are always read as 0. Writing to these bits 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.	

10.3.15 GPIO Interrupt Set Register (INT2GPIC)

INT2GPIC enables interrupt requests input from the following pins: pins 0 to 6 of port E, pins 0 and 1 of port H, pin 0 of port J, and pins 4 and 5 of port J.

A GPIO interrupt is an active low level-sensed signal. Within the register, bits for the pins are arranged in four groups. Pins 0 to 2 of port E are allocated to group 0, pins 3 to 5 of port E are allocated to group 1, pins 0 and 1 of port H, pin 0 of port J, and pin 4 of port K are allocated to group 2, and pin 5 of port K and pin 6 of port E are allocated to group 3. Before enabling any of these interrupt requests, set the corresponding pin as an input in the corresponding port-control register (PECR, PHCR, PJCR, PKCR). For the port-control registers, see section 28, General Purpose I/O (GPIO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—			—	—	—	—				
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/W	R/W	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
	—	—	—	—	—				—	—	—	—	—			
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/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

When a GPIO port pin is configured as an interrupt, the INTC is notified when the interrupt condition is satisfied on that pin. However, the interrupt is indicated as a one-bit source in the INT2A0 or INT2A1 register of the INTC. The port and pin on which the interrupt was received can be identified by referring to the on-chip module interrupt source register INT2B7. The port group where the interrupt was generated can also be identified by referring to the INTEVT code in the CPU.

Table 10.10 shows the correspondence between the interrupt input pins and bits in INT2GPIC.

Table 10.10 Correspondence between Interrupt Input Pins and Bits in INT2GPIC

Bit	Initial Value	R/W	Name	Function	Description
31 to 26	All 0	R/W	(Reserved)	These bits are always read as 0. The write value should always be 0.	Enables a GPIO interrupt request for each pin. 0: Disables the corresponding interrupt request 1: Enables the corresponding interrupt request
25	0	R/W	PORTE6E	Enables interrupt request from pin 6 of port E.	
24	0	R/W	PORTK5E	Enables interrupt request 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.	

10.4 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip modules. Each interrupt has a priority level (16 to 0), with level 16 as the highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and interrupt requests are ignored.

10.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. It is always accepted unless the BL bit in SR of the CPU is set to 1. In sleep mode, the interrupt is accepted even if the BL bit is set to 1.

A setting can also be made to have the NMI interrupt accepted even if the BL bit is set to 1. Input from the NMI pin is edge-detected. The NMI edge selection bit (NMIE) in ICR0 is used to select either the rising or falling edge for detection. After modification of the NMIE bit in ICR0, the NMI interrupt is not detected for at least six bus clock cycles after the modification. When the INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to level 15 on the acceptance of an NMI interrupt. When the INTMU bit in CPUOPM is cleared to 0, the IMASK value in SR is not affected by the acceptance of an NMI interrupt.

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 interrupts 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, falling-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 interrupt pin input level should be held until the CPU has accepted the interrupt and started interrupt exception handling.

When high- or low-level detection has been selected, usage or non-usage of the holding function for interrupt requests can be selected by setting or clearing the LSH bit in ICR0. When usage of the holding function has been selected (ICR0.LSH = 0), interrupt requests are held in the detection circuit and the interrupt request must be cleared in the exception handling routine after acceptance of the interrupt. For details, refer to section 10.7 Usage Notes. To select non-usage of the holding function, set the LSH bit in ICR0 to 1. In this case, the operation of IRQ level detection provides

upward compatibility with the “level-sense IRQ mode” of current SH-4 products (here, too, the detection of high or low levels is selectable).

Note: When high-or low-level detection is selected, once the interrupt request has been detected, the INTC holds the interrupt request as an interrupt source in INTREQ even if the level on the IRQ interrupt pin has been changed and canceled. The interrupt source is held until the CPU accepts any interrupt request (IRQ or not) or the corresponding interrupt mask bit is set to 1. Moreover, when the holding function is selected by clearing the LSH bit in ICR0 to 0, the interrupt request is held in the detection circuit. In this case, clearing of the interrupt request in the exception handling routine must be followed by clearing of the interrupt source setting being held in INTREQ. For details, see section 10.7 Usage Notes.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of an accepted interrupt. When the INTMU bit is cleared to 0, 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 IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$. The priority level is the value indicated by the levels (active low) on pins IRQ/$\overline{\text{IRL7}}$ to 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/$\overline{\text{IRL4}}$ or IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ corresponds to the highest-level interrupt request (interrupt priority level 15), and the high level on all pins corresponds to no interrupt request (interrupt priority level 0). 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.

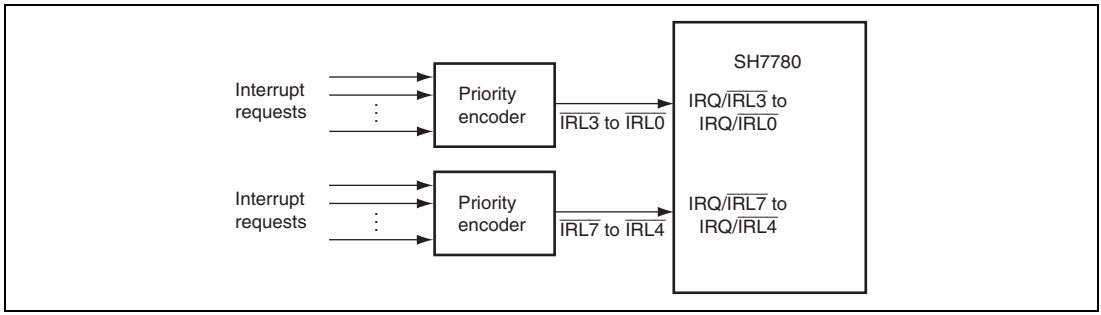


Figure 10.2 Example of IRL Interrupt Connection

Table 10.11 IRL[3:0], IRL[7:4] Pins and Interrupt Levels

$\overline{\text{IRL3}}$ or $\overline{\text{IRL7}}$	$\overline{\text{IRL2}}$ or $\overline{\text{IRL6}}$	$\overline{\text{IRL1}}$ or $\overline{\text{IRL5}}$	$\overline{\text{IRL0}}$ or $\overline{\text{IRL4}}$	Interrupt Priority Level	Interrupt Request
Low	Low	Low	Low	15	Level 15 interrupt request
Low	Low	Low	High	14	Level 14 interrupt request
Low	Low	High	Low	13	Level 13 interrupt request
Low	Low	High	High	12	Level 12 interrupt request
Low	High	Low	Low	11	Level 11 interrupt request
Low	High	Low	High	10	Level 10 interrupt request
Low	High	High	Low	9	Level 9 interrupt request
Low	High	High	High	8	Level 8 interrupt request
High	Low	Low	Low	7	Level 7 interrupt request
High	Low	Low	High	6	Level 6 interrupt request
High	Low	High	Low	5	Level 5 interrupt request
High	Low	High	High	4	Level 4 interrupt request
High	High	Low	Low	3	Level 3 interrupt request
High	High	Low	High	2	Level 2 interrupt request
High	High	High	Low	1	Level 1 interrupt request
High	High	High	High	0	No interrupt request

IRL interrupt detection requires a built-in noise-cancellation feature; that is, a mechanism to ensure that transient level changes on the IRL pins are not detected as interrupts. For this purpose, an IRL interrupt is not detected unless the levels sampled per bus-clock cycle remain unchanged for four consecutive cycles.

The IRL interrupt priority level should be maintained until the CPU has accepted the interrupt and started interrupt exception handling. It is possible to change the priority level to a higher priority.

When IRL level-encoded interrupts have been selected, usage or non-usage of the holding function for interrupt requests can be selected by clearing or setting the LSH bit in ICR0. When usage of the holding function has been selected (ICR0.LSH = 0), interrupt requests are held in the detection circuit and the interrupt request must be cleared in the exception handling routine after acceptance of the interrupt. For details, refer to section 10.7 Usage Notes. To select non-usage of the holding function, set the LSH bit in ICR0 to 1. In this case, the operation of IRL level detection provides upward compatibility with the level-encoded IRL interrupts on current SH-4 products.

Note: There is no interrupt source register for IRL interrupt requests. When the holding function is in use, however, operation is as follows. If, after detection of an IRL interrupt, the levels on the IRL pins are changed or withdraw the interrupt before it has been accepted by the CPU, the detection circuit retains the highest detected priority level for IRL interrupts until the CPU accepts any interrupt request (IRL or not) or the corresponding mask bit has been set to 1. The interrupt exception handling routine must then clear the IRL interrupt request held in the detection circuit. For details, see section 10.7 Usage Notes.

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the acceptance of an interrupt.

10.4.4 On-chip Module Interrupts

On-chip module interrupts are interrupts generated by on-chip modules. The interrupt sources are not assigned unique interrupt vectors; however, the sources are reflected in the interrupt event register (INTEVT), so using the INTEVT value as a branch offset in the exception handling routine provides a convenient and useful way to identify the sources and handle the individual interrupts.

A priority level from 31 to 0 can be set for each module by means of INT2PRI0 to INT2PRI7. The INTC rounds off the lowest order bit and sends a 4-bit code to the CPU. For details, see section 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-chip module interrupt.

Interrupt source flags and interrupt enable flags for on-chip modules should only be updated when the BL bit in SR is set to 1 or while the corresponding interrupt will not occur because its mask bit has been set. To prevent the erroneous acceptance of interrupts from sources that should have been updated, start by reading the on-chip module register that contains the corresponding flag, wait for the priority determination time shown in table 10.13 (i.e. the period required to read a register in INTC; this operation is driven by the peripheral clock), and then clear the BL bit to 0 or clear the corresponding interrupt mask. This will secure the necessary time internally. When a number of flags have to be updated, reading only the register containing the last flag to have been updated causes no problems.

If flag updating is performed while the BL bit is cleared to 0, the program may jump to the interrupt handling routine when the INTEVT value is 0. In this case, interrupt processing is initiated due to the timing relationship between the updating of the flag and recognition of the

interrupt request within this LSI. Processing can be continued without any problem after the execution of an RTE instruction.

10.4.5 Interrupt Priority Levels of On-chip Module Interrupts

When any interrupt is generated, the INTC outputs the corresponding interrupt exception code (INTEVT code) to the CPU. The code identifies the individual interrupt source. When the CPU accepts an interrupt, the corresponding INTEVT code is indicated in INTEVT. Even without reading the interrupt source register of the INTC, the interrupt source can be identified by reading INTEVT of the CPU from the interrupt handler. Table 10.12 lists the sources of interrupts and the corresponding interrupt exception codes.

An on-chip module interrupt source can be assigned any of 30 (5-bit) priority levels (see figure 10.3). The interrupt level-reception interface is four bits wide and thus handles 15 priority levels (with H'0 as the interrupt-request mask setting). The value in the INTC consists of five bits, one bit of which is an extension that allows the assignment of an individual priority level to each of the on-chip modules. When the CPU is notified of the priority, the lowest-order bit is rounded off to leave four bits of data. For example, two interrupt sources with priority levels set to H'1A and H'1B will both be output to the CPU as the 4-bit priority level H'D. That is, the two interrupt sources have the same priority value. However, although the rounded codes are the same for both interrupt sources, the interrupt with priority level H'1B clearly has priority when we consider the 5-bit data in the priority setting. That is, the 5-bit values in the fields shown in table 10.5 give INTC a way to differentiate between interrupts with the same four-bit priority level.

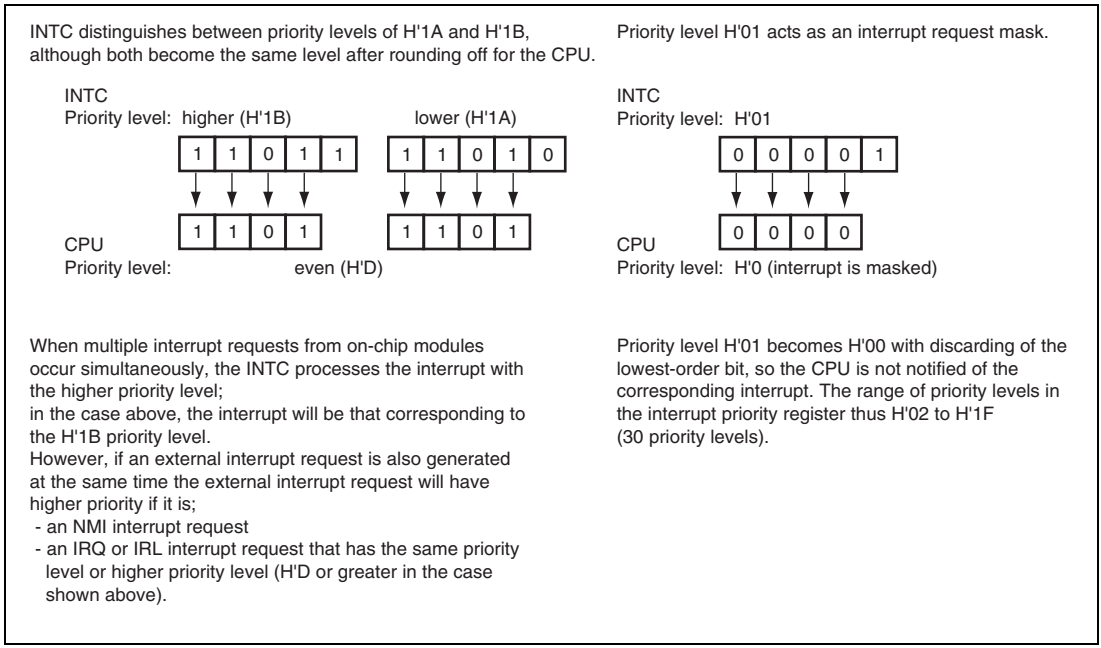


Figure 10.3 On-chip Module Interrupt Priority

10.4.6 Interrupt Exception Handling and Priority

Table 10.12 lists the codes for the interrupt event register (INTEVT) and the order of interrupt priority.

Each interrupt source is assigned a unique INTEVT code. The start address of the exception handling routine is the same for all of the interrupt sources. Therefore, the INTEVT value is used to control branching at the start of the exception handling routine. For instance, the INTEVT values are suitable for use as branch offsets.

The priority order of the on-chip modules is specified as desired by setting values from 31 to 2 in INT2PRI0 to INT2PRI7. Values 0 and 1 mask the corresponding interrupt. The priority values for the on-chip modules are returned to 0 by a reset.

When interrupt sources share the same priority level and are generated simultaneously, they are handled according to the default priority order given in table 10.12.

Values of INTPRI, INT2PRI0 to INT2PRI7, INTMSK0 to INTMSK2, and INT2MSKR should only be updated while the BL bit in SR is set to 1, or the corresponding interrupt is masking. To prevent erroneous interrupt acceptance, clear the BL bit to 0 after having read one of the interrupt

priority level-setting registers, or clear the corresponding interrupt mask. This will secure the necessary timing internally.

Table 10.12 Interrupt Exception Handling and Priority

Interrupt Source		INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
NMI	—	H'1C0	16	—	—	—		High
IRL L: Low level input H: High level input (See table 10.11)	$\overline{IRL[7:4]}$ = LLLL (H'0)	H'200	15	INTMSK2[15] INTMSKCLR2[15]	—	—		▲
	$\overline{IRL[3:0]}$ = LLLL (H'0)			INTMSK2[31] INTMSKCLR2[31]	—	—		
	$\overline{IRL[7:4]}$ = 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]	—	—		Low

	Interrupt Source	INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
IRL L: Low level input H: High level input (See table 10.11)	$\overline{IRL[7:4]}$ = LHHH (H'7)	H'2E0	8	INTMSK2[8] INTMSKCLR2[8]	—	—		High
	$\overline{IRL[3:0]}$ = LHHH (H'7)			INTMSK2[24] INTMSKCLR2[24]	—	—		
	$\overline{IRL[7:4]}$ = HLLL (H'8)	H'300	7	INTMSK2[7] INTMSKCLR2[7]	—	—		
	$\overline{IRL[3:0]}$ = HLLL (H'8)			INTMSK2[23] 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]	—	—		Low

	Interrupt Source	INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
IRQ	IRQ[0]	H'240	INTPRI [31:28]	INTMSK0[31] INTMSKCLR0 [31]	INTREQ [31]	—	High	High
	IRQ[1]	H'280	INTPRI [27:24]	INTMSK0[30] INTMSKCLR0 [30]	INTREQ [30]	—		
	IRQ[2]	H'2C0	INTPRI [23:20]	INTMSK0[29] INTMSKCLR0 [29]	INTREQ [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]		Low

Interrupt Source		INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
H-UDI	H-UDII	H'600	INT2PRI3 [28:24]	INT2MSKR[7] INT2MSKCR[7]	INT2A0[7] INT2A1[7]	—		High
DMAC(0)	DMINT0*	H'640	INT2PRI3 [20:16]	INT2MSKR[8] INT2MSKCR[8]	INT2A0[8] INT2A1[8]	INT2B3[0]	High	
	DMINT1*	H'660				INT2B3[1]		
	DMINT2*	H'680				INT2B3[2]		
	DMINT3*	H'6A0				INT2B3[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]		Low

Interrupt Source		INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
PCIC(5)	PCIERR	H'AA0	INT2PRI5 [4:0]	INT2MSKR[19] INT2MSKCR[19]	INT2A0[19] INT2A1[19]	INT2B4[5]	High	High
	PCIPWD3	H'AC0				INT2B4[6]		
	PCIPWD2	H'AE0				INT2B4[7]		
	PCIPWD1	H'B00				INT2B4[8]		
	PCIPWD0	H'B20				INT2B4[9]	Low	
SCIF-ch1	ERI1*	H'B80	INT2PRI2 [20:16]	INT2MSKR[4] INT2MSKCR[4]	INT2A0[4] INT2A1[4]	INT2B2[4]	High	
	RXI1*	H'BA0				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]	—		Low

	Interrupt Source	INTEVT Code	Interrupt Priority	Mask/Clear Register & Bit	Interrupt Source Register	Detail Source Register	Priority within Sets of Sources	Default Priority
FLCTL	FLSTE*	H'F00	INT2PRI7 [28:24]	INT2MSKR[24] INT2MSKCR[24]	INT2A0[24] INT2A1[24]	INT2B6[0]	High	High
	FLTEND*	H'F20				INT2B6[1]		
	FLTRQ0*	H'F40				INT2B6[2]		
	FLTRQ1*	H'F60				INT2B6[3]	Low	
GPIO	GPIOI0 (Port E0)	H'F80	INT2PRI7 [20:16]	INT2MSKR[25] INT2MSKCR[25]	INT2A0[25] INT2A1[25]	INT2B7[0]	High	
	GPIOI0 (Port E1)					INT2B7[1]		
	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	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 0 to 11
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

10.5 Operation

10.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 10.4 is the flowchart of the operations.

1. Interrupt request sources send interrupt request signals to the INTC.
2. The INTC selects the interrupt with the highest-priority among the interrupts that have been sent, according to the priority levels set in INTPRI and INT2PRI0 to INT2PRI7. Lower-priority interrupts are held as pending interrupts. If two of the interrupts have the same priority level or multiple interrupts are generated by a single module, the interrupt with the highest priority is selected according to table 10.12.
3. The priority level of the interrupt selected by the INTC is compared with the interrupt mask level (IMASK) set in SR of the CPU. If the priority level is higher than the mask level, the 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 same time, 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 sum of the 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 provides a convenient way to differentiate between the interrupt sources. Execution thus branches to the handling routines for the individual interrupt sources.

Notes: 1. When the INTMU bit in the CPU operating mode register (CPUOPM) is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.
2. The interrupt source flag should be cleared in the interrupt handling routine. To ensure that an interrupt source which should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, wait for the time shown in 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 interrupts, IRL interrupts, and interrupts for the on-chip modules. Thus, INTMSKCLR must be used to clear the interrupt mask setting (INTMSK) for any required IRQ, IRL, and on-chip module interrupts .

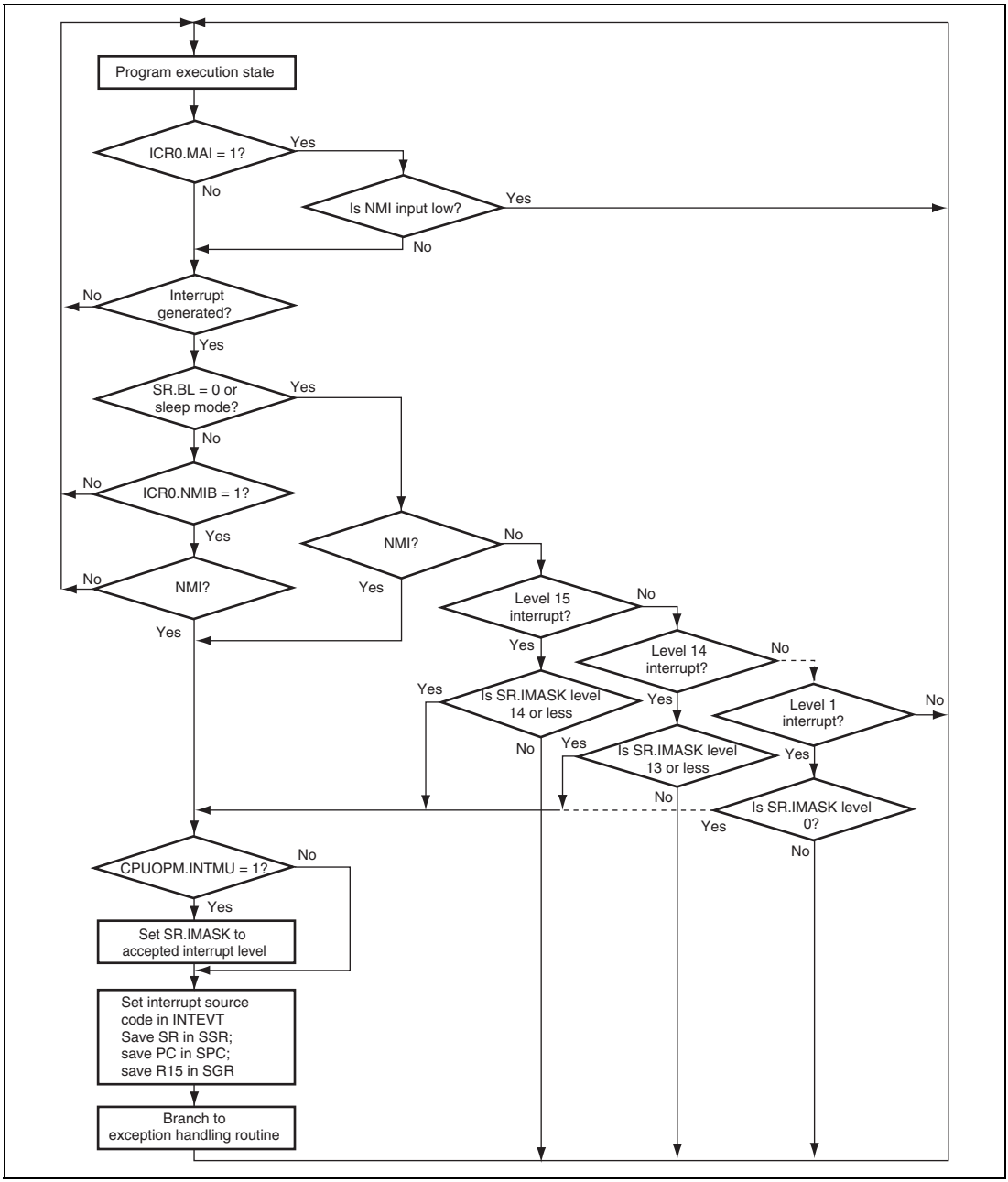


Figure 10.4 Interrupt Operation Flowchart

10.5.2 Multiple Interrupts

When multiple interrupts must be handled, the interrupt handling routine should include the following procedure:

1. Identify the interrupt source by using the INTEVT code as an offset in branching to the corresponding interrupt handling routine.
2. Clear the interrupt source in the corresponding interrupt handling routine.
3. Save SSR and SPC on the stack.
4. Clear the BL bit in SR. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the priority level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, use software to set the IMASK bit in SR to the 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 generated, an interrupt with higher priority than the one currently being handled can be accepted after step 4. 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 low 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 interrupts due to NMI signal input are generated.

10.6 Interrupt Response Time

Table 10.13 shows the components of the interrupt response time for the five classes of interrupt in terms of response time. The response time is the interval from generation of an interrupt request until the start of interrupt exception handling; i.e. until fetching of the first instruction of the exception handling routine.

Table 10.13 Interrupt Response Time

		Formulae for Response Time					
					On-chip Modules		
Item		NMI	IRL	IRQ	Other than GPIO/PCIC/ RTC	GPIO/PCIC/ RTC	Remarks
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	When Icyc:Scyc: Bcyc:Pcyc = 4:4:2:1

[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

10.7 Usage Notes

10.7.1 To Clear Interrupt Request When Holding Function Selected

When an IRQ level-sense interrupt request or IRL level-encoded interrupt request (IRQ/IRL level interrupt request) is generated and the holding function is in use, the interrupt request must be cleared in the interrupt handling routine after it has been accepted. Figure 10.5 shows an example of an interrupt-handling routine to clear interrupt request holding in the detection circuit.

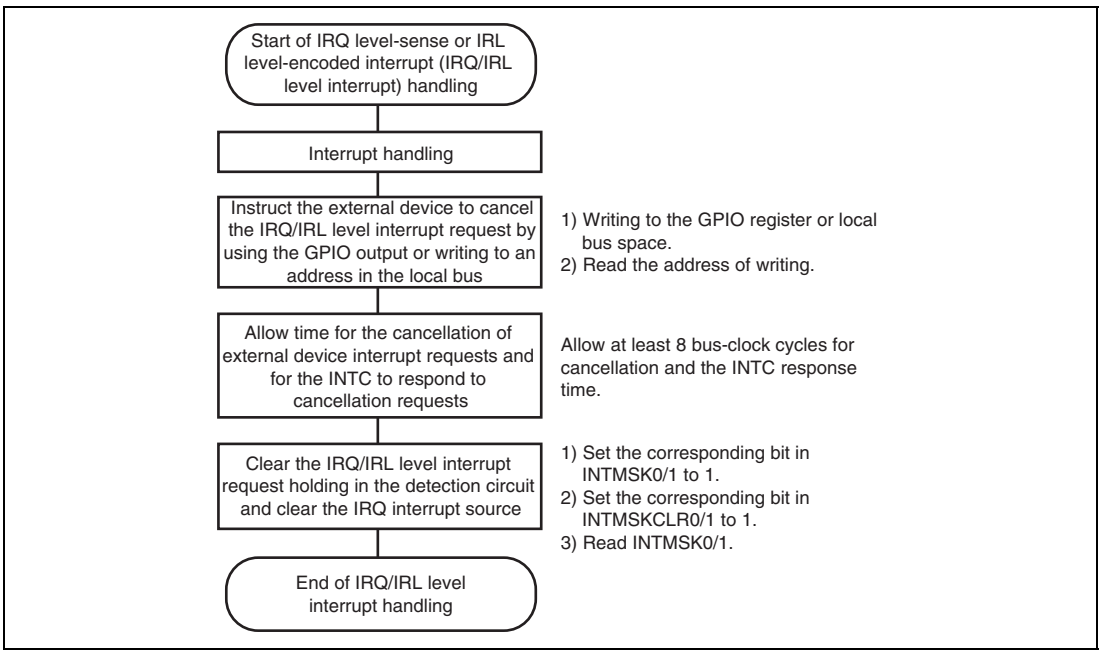


Figure 10.5 Example of Interrupt Handling Routine

To cancel an interrupt request after its acceptance by the CPU, the external device that generated the request must be notified of its acceptance. The method of notification might take the form of using the GPIO to output the acceptance level or interrupt pin information, or writing to a special address in the local bus space. It is necessary to consecutively execute writing to and reading from 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 required for the CPU to detect the interrupt has elapsed. To ensure this time, consecutively execute writing to INTMSK0/1 and INTMSKCLR0/1 and reading of INTMSK0/1.

10.7.2 Notes on Setting IRQ/$\overline{\text{IRL}}$[7:0] Pin Function

When switching between individual interrupt and level-encoded interrupt functions on the IRQ/$\overline{\text{IRL}}$[7:0] pins, the INTC may wind up holding an interrupt that was generated by mistake. Therefore, to prevent the detection of such unintentional interrupts, mask all IRQ and IRL interrupts before switching between IRQ/$\overline{\text{IRL}}$[7:0] pin functions.

Table 10.14 Switching Sequence of IRQ/$\overline{\text{IRL}}$[7:0] Pin Function

Sequence	item	Procedure
1	IRL interrupt request and IRQ interrupt request masking	Write 1 to all bits in INTMSK0 and INTMSK1
2	Setting IRL/$\overline{\text{IRQ}}$[7:4] pins to operate as interrupt-request pins	Write 0 to the OMSEL12 bit in OMSELR Write 0 to the PE6MD[1:0] bits in PECR
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 LSH bit in ICR0
4	Start of IRL and IRQ interrupt detection	Write 1 to the corresponding bit in INTMSKCLR0 and INTMSKCLR1

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 from the IRQ/IRL[3:0] pins by writing a 1 to the IM10 bit in INTMSK1, and clear an IRL interrupt request from the IRQ/IRL[7:4] pins by writing a 1 to the IM11 bit in the same register. IRL interrupt requests held in the detection circuit are not cleared even if each of the corresponding interrupt levels is masked by the setting in INTMSK2.

  When the holding function is not in use (ICR0.LSH = 1), interrupt requests are simply not held.

- **To clear IRQ level-sense interrupt requests**

  When the holding function is in use (ICR0.LSH = 0), clear an IRQ level-sense interrupt request from the IRQ/$\overline{\text{IRL}}$[7:0] pins by writing a 1 to the corresponding mask bit (IM07 to IM00) of INTMSK0.

IRQ interrupt requests held in the detection circuit are not cleared even if a 0 is written to the corresponding bit in INTPRI. The IRQ interrupt sources detected by the INTC (which will be cleared when they are accepted by the CPU) can be confirmed by reading INTREQ.

When not using holding function (ICR0.LSH = 1), the interrupt request is not held but the interrupt source is set to the corresponding bit in INTREQ that is to be cleared when the CPU accepts it.

- **To clear IRQ edge-detection interrupt requests**

  To clear an IRQ edge-detection interrupt request from the IRQ/$\overline{\text{IRL}}$[7:0] pins, read the value 1 from the corresponding IRn (n = 0 to 7) bit in INTREQ and then write a 0 to the same bit.

  An IRQ interrupt request detected by the INTC is not cleared even if a 1 is written to the corresponding bit in INTMSK0.

Section 11 Local Bus State Controller (LBSC)

The local bus state controller (LBSC) divides the external memory space and outputs control signals corresponding to the specifications of various types of memory and bus interfaces. The LBSC enables the connection of SRAM or ROM, etc., to this LSI. It also supports the PCMCIA interface protocol, which is used to implement simplified system design and high-speed data transfers in a compact system.

11.1 Features

The LBSC has the following features.

- Controls six areas, areas 0 to 2 and 4 to 6, of an external memory space divided into seven 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, which is 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 memory accesses
  - Insertion of cycles to ensure the setup time and hold time to the write strobe on a write 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

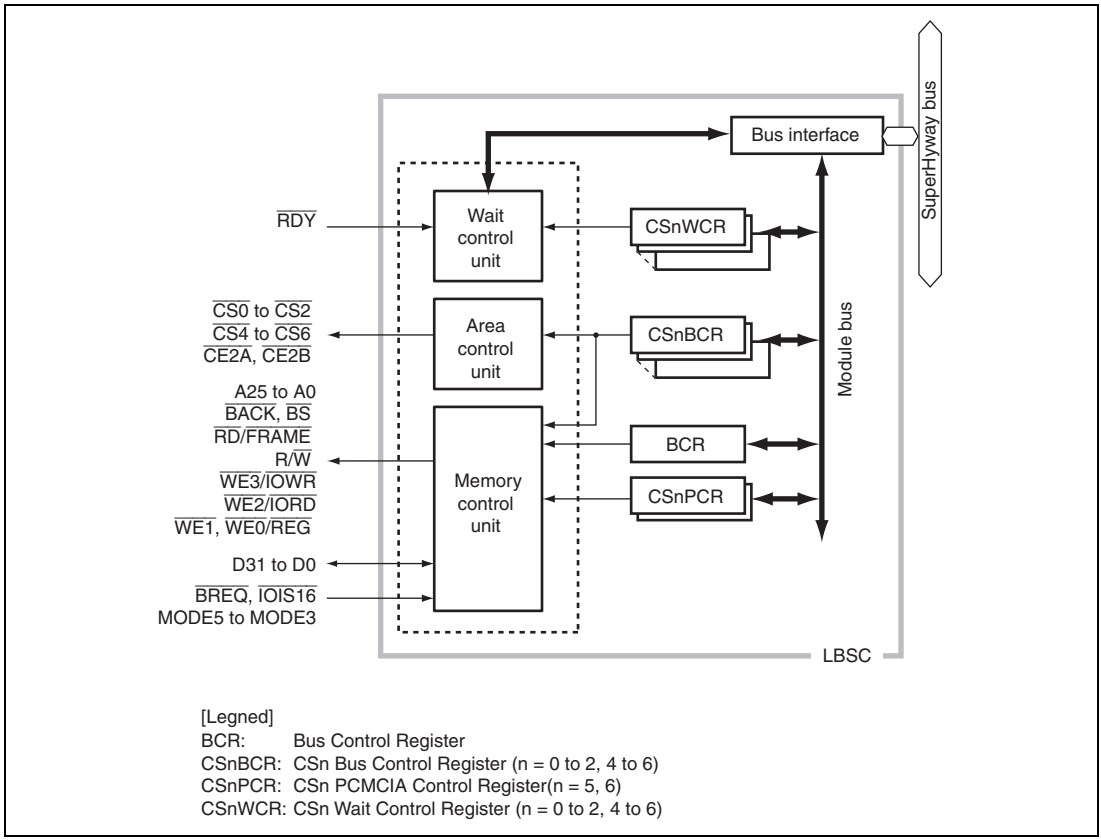
    Settable bus widths: 32, 16, and 8 bits
- Burst ROM interface
  - Wait-cycle insertion can be controlled by a program
  - Burst length specified by the register

    Connectable areas: 0 to 2 and 4 to 6

    Settable bus widths: 32, 16, and 8 bits

- MPX interface
  - Address/data multiplexing
    Connectable areas: 0 to 2 and 4 to 6
    Settable bus width: 32 bits
- Byte control SRAM interface
  - SRAM interface with byte control
    Connectable areas: 1 and 4
    Settable bus widths: 32 and 16 bits
- PCMCIA interface
  - 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

11.2 Input/Output Pins

Table 11.1 shows the LBSC pin configuration.

Table 11.1 Pin Configuration

Pin Name	Function	I/O	Description
A25 to A0	Address Bus	Output	Address output
D31 to D0*[1]	Data Bus	I/O	Data input/output
$\overline{BS}$	Bus Cycle Start	Output	Signal that indicates the start of a bus cycle. Asserted once for a burst transfer when setting MPX interface. Asserted each data cycle for a burst transfer 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 area being accessed. $\overline{CS5}$ and $\overline{CS6}$ can also be used as $\overline{CE1A}$ to $\overline{CE1B}$ of PCMCIA.
R/$\overline{W}$	Read/Write	Output	Data bus input/output direction designation signal. Also used as PCMCIA interface write designation signal.
$\overline{RD}$/$\overline{FRAME}$	Read/Cycle Frame	Output	Strobe signal indicating a read cycle. $\overline{FRAME}$ signal when setting MPX interface.
$\overline{WE0}$/$\overline{REG}$	Data Enable 0	Output	When setting SRAM interface: write strobe signal for D7 to D0 When setting PCMCIA interface: $\overline{REG}$ signal
$\overline{WE1}$	Data Enable 1	Output	When setting SRAM interface: write strobe signal for D15 to D8 When setting PCMCIA interface: Write strobe signal
$\overline{WE2}$/$\overline{IORD}$	Data Enable 2	Output	When setting SRAM interface: write strobe signal for D23 to D16 When setting PCMCIA interface: $\overline{IORD}$ signal
$\overline{WE3}$/$\overline{IOWR}$	Data Enable 3	Output	When setting SRAM interface: write strobe signal for D31 to D24 When setting PCMCIA interface: $\overline{IOWR}$ signal
$\overline{RDY}$	Ready	Input	Wait cycle request signal

Pin Name	Function	I/O	Description
$\overline{\text{IOIS16}}$*2	16-Bit I/O	Input	16-bit I/O signal when setting PCMCIA interface. Valid only in little endian mode
$\overline{\text{BREQ}}$*3	Bus Release Request	Input	Bus release request signal
$\overline{\text{BACK}}$	Bus Request Acknowledge	Output	Bus release acknowledge signal
$\overline{\text{CE2A}}$*4, $\overline{\text{CE2B}}$*4	PCMCIA Card Select	Output	When setting PCMCIA, $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$
MODE3*5, MODE4*5	Area 0 Bus Width	Input	Signal setting area 0 bus width and MPX interface at power-on reset
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 external device which has output $\overline{\text{DREQ0}}$*11, regarding DMA transfer request
$\overline{\text{DACK1}}$*7, 10	DMA channel 1 transfer end notification	Output	Strobe output from channel 1 to external device which has output $\overline{\text{DREQ1}}$*11, regarding DMA transfer request
$\overline{\text{DACK2}}$*8, 10	DMA channel 2 transfer end notification	Output	Strobe output from channel 2 to external device which has output $\overline{\text{DREQ2}}$*11, regarding DMA transfer request
$\overline{\text{DACK3}}$*9, 10	DMA channel 3 transfer end notification	Output	Strobe output from channel 3 to external device which has output $\overline{\text{DREQ3}}$*11, regarding DMA 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 set to b'100, $\overline{\text{CE2A}}$ act as PCMCIA output pin, and bits TYPE2 to TYPE0 in the CS6 bus control 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 section 14, Direct Memory Access Controller (DMAC).
11. Can be selectable the polarity and detection edge (initial state is low active). For details, see section 14, Direct Memory Access Controller (DMAC).

11.3 Area Overview

11.3.1 Space Divisions

The architecture of this LSI provides a 32-bit address space. The virtual address space is divided into five areas (P0 to P4 areas) according to the upper address value.

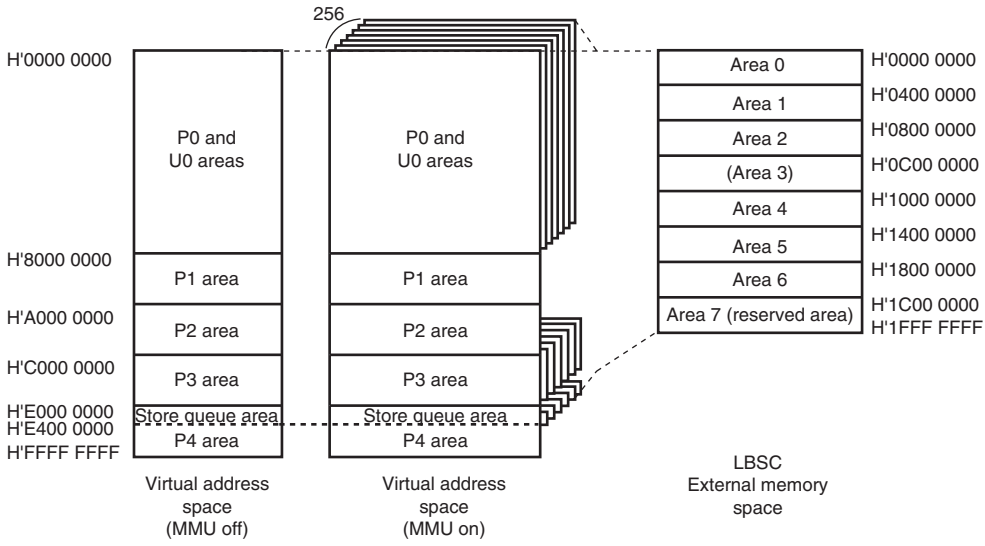
This LSI supports both a 29-bit and a 32-bit physical address space, and the LBSC supports a 29-bit physical address space. The 29-bit physical address space is divided into eight areas (areas 0 to 7) according to the upper three bits [28:26] of an address and the LBSC can control areas 0 to 2 and 4 to 6 as an external memory space. The maximum capacity of each area used as an external memory space is 64 Mbytes; the LBSC can control a total of 6 areas with a maximum capacity of 384 Mbytes as the external memory spaces.

A virtual address can be allocated to any physical address through the address translation function 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 areas as shown in table 11.2, and accordingly output the chip select signals ($\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{CE2A}$ and $\overline{CE2B}$). $\overline{CS0}$ to $\overline{CS2}$ are asserted when accessing areas 0 to 2 individually, and $\overline{CS4}$ to $\overline{CS6}$ are asserted when accessing areas 4 to 6 individually. When the PCMCIA interface is selected for area 5 or 6, $\overline{CE2A}$ or $\overline{CE2B}$ is asserted along with $\overline{CS5}$ and $\overline{CS6}$ for the bytes to be accessed.

Area 3 is for DDR-SDRAM memory space and controlled by the DDR-SDRAM Interface (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 also be used for the PCI memory space by setting the Memory Address Map Select Register (MMSELR). Area 7 is a reserved area. For the PCI memory space, see section 13, PCI Controller (PCIC). Both DDRIF and PCIC support a 32-bit physical address space in addition to a 29-bit address. For a 32-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 Memory Space of LBSC

Table 11.2 LBSC External Memory Space Map

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width (bits)	Access Size*[8]
0	H'0000 0000 to H'03FF FFFF	64 Mbytes	SRAM	8, 16, 32*[1]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[1]	
			MPX	32*[1]	
1	H'0400 0000 to H'07FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			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 bits and 32 bytes
			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 bits and 32 bytes
4*[4, 5]	H'1000 0000 to H'13FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			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 bits and 32 bytes
			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 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			PCMCIA	8, 16*[2, 6]	

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width (bits)	Access Size*[8]
7*[7]	H'1C00 0000 to H'1FFF FFFF	64 Mbytes	—	—	—

Notes: 1. The memory bus width is specified by external pins (MODE3 and MODE4).
2. The memory bus width is specified by the register.
3. Area 3 is used specifically for the DDR-SDRAM. For details, see section 12, DDR-SDRAM Interface (DDRIF).
4. These areas can be used for the DDR-SDRAM by setting MMSELR. For details, see section 12, DDR-SDRAM Interface (DDRIF).
5. This area can be used for the PCI memory by setting MMSELR. For details, see section 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 cannot be guaranteed.
8. If 8 or 16 bytes access transfer by another LSI internal bus master module is being executed, the LBSC is executing two or four times 32-bit access individually.

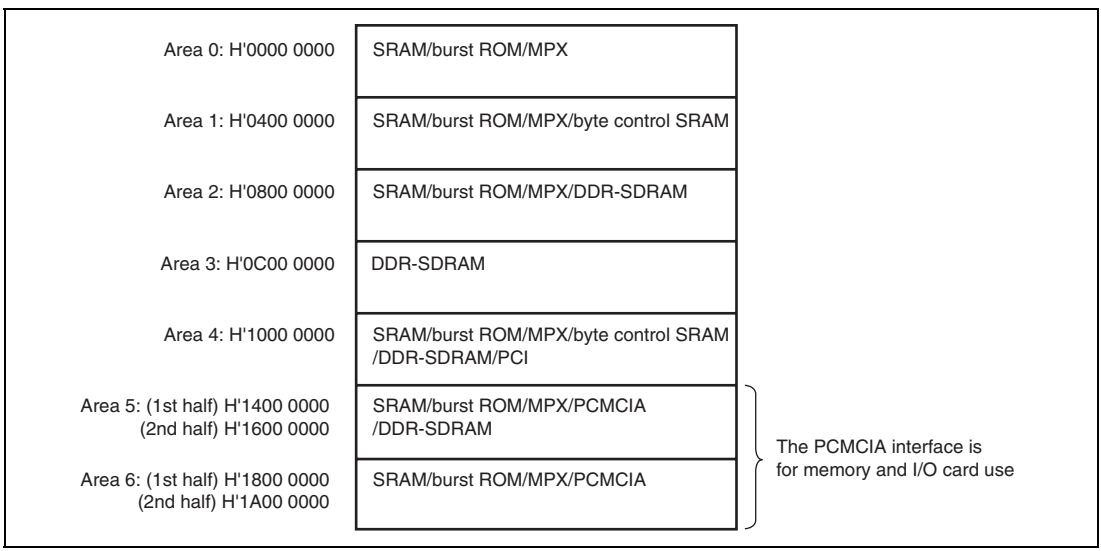


Figure 11.3 External Memory Space Allocation (29-bit address mode)

11.3.2 Memory Bus Width

The memory bus width of the LBSC can be set independently for each area. For area 0, a bus width of 8, 16, or 32 bits is set according to the external pin settings at a power-on reset by the $\overline{\text{PRESET}}$ pin. The correspondence between the external pins (MODE 4 and MODE 3) and the bus width at a power-on reset is shown below.

Table 11.3 Correspondence Between External Pins (MODE4 and MODE3)

Mode 4	Mode 3	Bus Width
Low	Low	32 bits (For MPX interface)
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 of 8, 16, or 32 bits can be selected through the CSn bus control register (CSnBCR). When the burst ROM interface is used, a bus width of 8, 16, or 32 bits can be selected. When the byte-control SRAM interface is used, a bus width of 16 or 32 bits can be selected. When the MPX interface is used, a 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 details, see 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 section 12, 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 used.

11.3.3 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data alignment method is specified using the external pin (MODE5) at a power-on reset.

Table 11.4 Correspondence Between External Pin (MODE5) and Endian

Mode 5	Endian
Low	Big endian
High	Little endian

11.3.4 PCMCIA Support

This LSI supports the PCMCIA interface specifications for areas 5 and 6 in the external memory space.

The IC memory card interface and I/O card interface prescribed in JEIDA specifications version 4.2 (PCMCIA2.1) are supported.

Both the IC memory card interface and the I/O card interface are supported in areas 5 and 6 in the 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, ATA 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 devices control register

Table 11.6 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name*[1]	I/O*[1]	Function	Signal Name*[1]	I/O*[1]	Function	Corresponding Pin of this LSI
1	GND		Ground	GND		Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CS5}$ or $\overline{CS6}$
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 on port
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

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name*[1]	I/O*[1]	Function	Signal Name*[1]	I/O*[1]	Function	Corresponding Pin of this LSI
30	D0	I/O	Data	D0	I/O	Data	D0
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP*[2]	O	Write protect	$\overline{\text{IOIS16}}$	O	16-bit I/O port	$\overline{\text{IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	Sensed on port
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	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	$\overline{\text{RFSH}}$ ($\overline{\text{VS1}}$)	I	Refresh request	$\overline{\text{RFSH}}$ ($\overline{\text{VS1}}$)	I	Refresh request	Output from port
44	RSRVD		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{IORD}}$
45	RSRVD		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{IOWR}}$
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	—

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name*[1]	I/O*[1]	Function	Signal Name*[1]	I/O*[1]	Function	Corresponding Pin of this LSI
58	RESET	I	Reset	RESET	I	Reset	Output from port
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{RDY}}$*[3]
60	RSRVD		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital voice signal	Sensed on port
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed on port
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 on port
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, while the 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.

11.4 Register Descriptions

Table 11.7 shows the LBSC register configuration. Table 11.8 shows the register state in each processing mode.

Table 11.7 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*
Memory Address Map Select Register	MMSELR	R/W	H'FF40 0020	H'1F40 0020	32
Bus Control Register	BCR	R/W	H'FF80 1000	H'1F80 1000	32
CS0 Bus Control Register	CS0BCR	R/W	H'FF80 2000	H'1F80 2000	32
CS1 Bus Control Register	CS1BCR	R/W	H'FF80 2010	H'1F80 2010	32
CS2 Bus Control Register	CS2BCR	R/W	H'FF80 2020	H'1F80 2020	32
CS4 Bus Control Register	CS4BCR	R/W	H'FF80 2040	H'1F80 2040	32
CS5 Bus Control Register	CS5BCR	R/W	H'FF80 2050	H'1F80 2050	32
CS6 Bus Control Register	CS6BCR	R/W	H'FF80 2060	H'1F80 2060	32
CS0 Wait Control Register	CS0WCR	R/W	H'FF80 2008	H'1F80 2008	32
CS1 Wait Control Register	CS1WCR	R/W	H'FF80 2018	H'1F80 2018	32
CS2 Wait Control Register	CS2WCR	R/W	H'FF80 2028	H'1F80 2028	32
CS4 Wait Control Register	CS4WCR	R/W	H'FF80 2048	H'1F80 2048	32
CS5 Wait Control Register	CS5WCR	R/W	H'FF80 2058	H'1F80 2058	32
CS6 Wait Control Register	CS6WCR	R/W	H'FF80 2068	H'1F80 2068	32
CS5 PCMCIA Control Register	CS5PCR	R/W	H'FF80 2070	H'1F80 2070	32
CS6 PCMCIA Control Register	CS6PCR	R/W	H'FF80 2080	H'1F80 2080	32

Note: * Do not access registers with other than the designated access size.

Table 11.8 Register State in Each Processing Mode

Register Name	Abbrev.	Power-On Reset	Manual Reset	Sleep
Memory Address Map Select Register	MMSELR	H'0000 0000	H'0000 0000	Retained
Bus Control Register	BCR	H'x000 0000	Retained	Retained
CS0 Bus Control Register	CS0BCR	H'7777 7770	Retained	Retained
CS1 Bus Control Register	CS1BCR	H'7777 7770	Retained	Retained
CS2 Bus Control Register	CS2BCR	H'7777 7770	Retained	Retained
CS4 Bus Control Register	CS4BCR	H'7777 7770	Retained	Retained
CS5 Bus Control Register	CS5BCR	H'7777 7770	Retained	Retained
CS6 Bus Control Register	CS6BCR	H'7777 7770	Retained	Retained
CS0 Wait Control Register	CS0WCR	H'7777 770F	Retained	Retained
CS1 Wait Control Register	CS1WCR	H'7777 770F	Retained	Retained
CS2 Wait Control Register	CS2WCR	H'7777 770F	Retained	Retained
CS4 Wait Control Register	CS4WCR	H'7777 770F	Retained	Retained
CS5 Wait Control Register	CS5WCR	H'7777 770F	Retained	Retained
CS6 Wait Control Register	CS6WCR	H'7777 770F	Retained	Retained
CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	Retained	Retained
CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	Retained	Retained

11.4.1 Memory Address Map Select Register (MMSELR)

MMSELR is a 32-bit register that selects memory address maps for areas 2 to 5. This register should be accessed at the address H'FF40 0020 in longword. Writing is accepted only when the upper 16-bit data is H'A5A5 to prevent unintentional writing. The upper 29 bits are always read as 0. This register is initialized by a power-on reset or a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AREASEL		
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	R/W	R/W	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 AREASEL 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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	AREASEL	000	R/W	DDRIF/PCIC Memory Space Select 000: Sets area 3 (H'0C00 0000 to H'0FFF FFFF) as the DDRIF space and other areas as the LBSC space 001: Sets area 3 (H'0C00 0000 to H'0FFF FFFF) as the DDRIF space, area 4 (H'1000 0000 to H'13FF FFFF) as the PCI memory space, and other areas as the LBSC space 010: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDRIF space and other areas as the LBSC space 011: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDRIF space, area 4 (H'1000 0000 to H'13FF FFFF) as the PCI memory space, and other areas as the LBSC space 100: Sets areas 2 to 5 (H'0800 0000 to H'17FF FFFF) as the 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 module must be set not to access to any resources, and all processing should be finished (for example, the SYNCO instruction preceding the MOV instruction should be executed) before MMSELR is modified.

In addition, execute the MOV instruction to read out MMSELR (a dummy read) twice and the SYNCO instruction in succession immediately after a MOV instruction of write to MMSELR.

Example:

```
------------------------------------------------------------------------
MOV.L   #H'FF400020, R0         ;
MOV.L   #MMSELR_DATA, R1        ; MMSELR_DATA=Writing value of MMSELR
SYNCO                           ;                   (upper word=H'A5A5)
MOV.L   R1, @R0                 ; Writing to MMSELR
MOV.L    @R0, R2
MOV.L    @R0, R2
SYNCO
------------------------------------------------------------------------
```

The instruction to modify the value of the MMSELR should be allocated non-cacheable P2 area and an address that will not be affected by an address map change.

Write to MMSELR before enabling the Instruction cache, Operand cache, and MMU address translation, and then do not write to it again until after power-on reset or manual reset.

11.4.2 Bus Control Register (BCR)

BCR is a 32-bit readable/writable register that specifies the function and bus cycle status for each area. BCR is initialized to H'0000 0000 in big endian or H'8000 0000 in little endian by a power-on reset, but is not initialized by a manual reset mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ENDIAN	—	—	—	—	DPUP	—	OPUP	DACKBST[3:0]				—	—	BREQEN	DMABST
Initial value:	0/1*	0	0	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	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	HIZCNT	—	—	—	—	—	—	—	ASYNC[6:0]						
Initial value:	0	0	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	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) designating the endian mode is sampled at a power-on reset by the $\overline{\text{PRESET}}$ pin. This bit determines the endian mode of all spaces. 0: Indicates that the external pin (MODE5) designating the endian mode is low at a power-on reset and big-endian mode is specified for this LSI. 1: Indicates that the external pin (MODE5) designating the endian mode is high at a power-on reset and little-endian mode is specified for this LSI.
30 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
26	DPUP	0	R/W	Data Pin Pull-Up Resistor Control Specifies the pull-up resistor state of the data pins (D31 to D0). This bit is initialized by a power-on reset. The pins are not pulled up when access is performed or when the bus is released, even if the pull-up resistor is on. 0: Cycles in which the pull-up resistors of the data pins (D31 to D0) are turned on are inserted before and after a memory access.* 1: Pull-up resistor is off for data pins (D31 to D0). Note: * We recommend that a pull-up resistor be externally connected to the data pins if it is required.
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
24	OPUP	0	R/W	Control Output Pin Pull-Up Resistor Control Specifies the pull-up resistor state (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAME}$, $\overline{WE}$, R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) when the control output pins are high-impedance. This bit is initialized by a power-on reset. 0: Pull-up resistors are on for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAME}$, $\overline{WE}$, R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) 1: Pull-up resistors are off for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAME}$, $\overline{WE}$, 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 $\overline{DACK3}$ signals. 0: $\overline{DACK}$ signals asserted in synchronization with the bus cycle. 1: $\overline{DACK}$ signals remain asserted from burst start to end in DMA burst transfer mode Only set to 1 when the area of a $\overline{DACK}$ assertion in DMA transfer is the PCMCIA interface memory area, otherwise this bit should be cleared to 0. DACKBST[3]: $\overline{DACK3}$ DACKBST[2]: $\overline{DACK2}$ DACKBST[1]: $\overline{DACK1}$ DACKBST[0]: $\overline{DACK0}$

Bit	Bit Name	Initial Value	R/W	Description
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BREQEN	0	R/W	$\overline{\text{BREQ}}$ Enable Indicates whether or not an external bus request can be accepted. This bit is initialized to the state where an external bus request is not accepted at a power-on reset. 0: An external bus request is not accepted 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 by the DMAC. When this bit is cleared to 0, the priority is as follows: bus release, DMAC (burst mode), CPU, DMAC, PCIC. When this bit is set to 1, the bus release is not performed until completion of the DMAC burst transfer. This bit is initialized at a power-on reset. 0: DMAC burst mode transfer priority setting off 1: DMAC burst mode transfer priority setting on
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	HIZCNT	0	R/W	High Impedance (Hi-Z) Control Specifies the state of signals $\overline{\text{WE}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ during the bus-released state. 0: Signals of $\overline{\text{WE}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ are high-impedance during the bus-released state 1: Signals of $\overline{\text{WE}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ are output during the bus-released state
13 to 7	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	ASYNC[6:0]	All 0	R/W	Asynchronous Input Enable asynchronous input to the corresponding pins. 0: Input signals to the corresponding pins must be synchronized with CLKOUT 1: Input signals to the corresponding pins can be asynchronous to CLKOUT ASYNC[6]: $\overline{\text{DREQ3}}$ ASYNC[5]: $\overline{\text{DREQ2}}$ ASYNC[4]: $\overline{\text{DREQ1}}$ ASYNC[3]: $\overline{\text{DREQ0}}$ ASYNC[2]: $\overline{\text{IOIS16}}$ ASYNC[1]: $\overline{\text{BREQ}}$ ASYNC[0]: $\overline{\text{RDY}}$

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 to 2 and 4 to 6), numbers of wait, setup, and hold cycles to be inserted, burst length, and memory types.

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 access to different areas or writing to a memory immediately after reading. This LSI automatically inserts 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 manual reset. 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	17	16
	—	IWW			—	IWRWD			—	IWRWS			—	IWRRD		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IWRRS			BST		SZ		RDSPL	BW			MPX	TYPE		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	0	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/W	R/W

Note: * Bits SZ and MPX in CS0BCR are read-only.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW	111	R/W	Idle Cycles between Write-Read/Write-Write Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are write-read cycles and write-write cycles. For details, see section 11.5.8, Wait 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
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	IWRWD	111	R/W	Idle Cycles between Read-Write to Different Spaces Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-write cycles to different spaces. For details, see section 11.5.8, Wait 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

Bit	Bit Name	Initial Value	R/W	Description
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	IWRWS	111	R/W	Idle Cycles between Read-Write to Same Space Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-write cycles to the same space. For details, see section 11.5.8, Wait 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
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	IWRRD	111	R/W	Idle Cycles between Read-Read to Different Spaces Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-read cycles to different spaces. For details, see section 11.5.8, Wait 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

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	IWRRS	111	R/W	Idle Cycles between Read-Read to Same Space Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-read cycles to the same space. For details, see section 11.5.8, Wait 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
11, 10	BST	01	R/W	Burst Length When a burst ROM interface is used, these bits specify the number of accesses in a burst. The MPX interface is not affected. 00: 4 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 01: 8 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 10: 16 consecutive accesses (Can be used with 8-, or 16-bit bus width) 11: 32 consecutive accesses (Can be used with 8-bit bus width)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	SZ	11	R/W*	Bus Width Specify the bus width. Set to 11 for the MPX interface, and set to 10 or 11 for the byte control SRAM interface. In CS0BCR, the external pins (MODE3 and MODE4) are sampled at a power-on reset. 00: Reserved 01: 8 bits 10: 16 bits 11: 32 bits Note: * Bits SZ in CS0BCR are read-only. The SZ bits in CS0BCR are set to 11 when area 0 is set to the MPX interface by the MODE3 and MODE4 pins.
7	RDSPL	0	R/W	$\overline{\text{RD}}$ Hold Cycle Specifies the number of cycles to be inserted into the $\overline{\text{RD}}$ assertion period to elongate the data hold time for the read data sample timing. When setting this bit to 1, specify the number of $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycles as 1 or more by setting the RDH bit in CSnWCR. Also the $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycle is reduced by 1 cycle when this bit is set to 1 (Available only when the SRAM interface or byte control SRAM 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 bits specify the number of wait cycles to be inserted after the second data access in a burst transfer. 000: No idle cycle inserted, $\overline{\text{RDY}}$ signal disabled 001: 1 idle cycle inserted, $\overline{\text{RDY}}$ signal enabled 010: 2 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 011: 3 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 100: 4 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 101: 5 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 110: 6 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 111: 7 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled

Bit	Bit Name	Initial Value	R/W	Description
3	MPX	0	R/W*	MPX Interface Setting Selects the type of MPX interface 0: Interface that is specified by TYPE bits 1: MPX interface selected Note: * The MPX bit in CS0BCR is read-only.
2 to 0	TYPE	000	R/W	Memory Type Setting Specify the type of memory connected to the space. 000: SRAM (Initial value) 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 and CS4BCR 2. Setting possible only in CS5BCR and CS6BCR

11.4.4 CSn Wait Control Register (CSnWCR)

CSnWCR (n = 0 to 2, 4 to 6) are 32-bit readable/writable registers that specify the number of wait cycles to be inserted, the pitch of data access for burst memory accesses, and the number of cycles to be inserted for the address setup time to the read/write strobe assertion or for the data hold time from the write strobe negation.

CSnBCR is initialized to H'7777 770F by a power-on reset, but is not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ADS			—	ADH			—	RDS			—	RDH		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WTS			—	WTH			—	BSH			IW[3:0]			
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	1	1	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/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	ADS	111	R/W	Address Setup Cycle Specify the number of cycles to be inserted to ensure the address setup time to the $\overline{\text{CSn}}$ assertion. (Available only when the SRAM interface, byte control SRAM 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

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	ADH	111	R/W	Address Hold Cycle Specify the number of cycles to be inserted to ensure the address hold time to the $\overline{CSn}$ negation. (Available only when the SRAM interface, byte control SRAM interface, or 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
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	RDS	111	R/W	$\overline{RD}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{RD}$ Assertion Delay Cycle) Specify the number of cycles to be inserted form $\overline{CSn}$ assertion to $\overline{RD}$ assertion (Available only when the SRAM interface, byte control SRAM interface, or 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

Bit	Bit Name	Initial Value	R/W	Description
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	RDH	111	R/W	$\overline{RD}$ Hold Cycle ($\overline{RD}$ Negation–$\overline{CSn}$ Negation Delay Cycle) Specify the number of cycles to be inserted from $\overline{RD}$ negation to $\overline{CSn}$ negation. (Available only when the SRAM interface, byte control an SRAM interface, or 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
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	WTS	111	R/W	$\overline{WE}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{WE}$ Assertion Delay Cycle) Specify the number of cycles to be inserted from $\overline{CSn}$ assertion to $\overline{WE}$ assertion. (Available only when the SRAM interface, byte control an SRAM interface, or 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

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	WTH	111	R/W	$\overline{WE}$ Hold Cycle ($\overline{WE}$ Negation–$\overline{CSn}$ Negation Delay Cycle) Specify the number of cycles to be inserted from $\overline{WE}$ negation to $\overline{CSn}$ negation. (Available only when the SRAM interface, byte control SRAM interface, or 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
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	BSH	000	R/W	$\overline{BS}$ Hold Cycle Specify the number of cycles for the $\overline{BS}$ assertion. Total access cycle number is not change by setting these bits. This setting is valid when $\overline{CSn}$ assertion-$\overline{RD}$ or -$\overline{WE}$ assertion delay cycle (RDS or WTS setting) is set 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

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	IW[3:0]	1111	R/W	Insert Wait Cycle Specify the number of wait cycles to be inserted. • When the SRAM interface, byte control SRAM interface, burst ROM interface (first data cycle only), or PCMCIA interface is selected, the following cycles are inserted. The external wait cycle insertion by using the $\overline{RDY}$ pin monitor cannot be used when no cycle inserted is selected. 0000: No cycle inserted 1000: 8 cycles inserted 0001: 1 cycle inserted 1001: 9 cycles inserted 0010: 2 cycles inserted 1010: 11 cycles inserted 0011: 3 cycles inserted 1011: 13 cycles inserted 0100: 4 cycles inserted 1100: 15 cycles inserted 0101: 5 cycles inserted 1101: 17 cycles inserted 0110: 6 cycles inserted 1110: 21 cycles inserted 0111: 7 cycles inserted 1111: 25 cycles inserted • When the MPX interface is selected, the following cycles are inserted by the setting value of IW [2:0] and then IW3 setting is invalid. And the external wait cycle insertion by using the $\overline{RDY}$ pin monitor can be used in all the following settings. IW2 specifies the number of wait cycle to be inserted into second data or after. 0: No cycle inserted 1: 1 cycle inserted IW[1:0] specify the number of wait cycles to be inserted into first data. 00: 1 cycle inserted into read cycle and no cycle inserted into write cycle 01: 1 cycle inserted into read cycle and 1 cycle inserted into write cycle 10: 2 cycles inserted into read cycle and 2 cycles inserted into write cycle 11: 3 cycles inserted into read cycle and 3 cycles inserted into write cycle

11.4.5 CSn PCMCIA Control Register (CSnPCR)

CSnPCR is a 32-bit readable/writable register that specifies the timing for the PCMCIA interface connected to area n (n = 5 or 6), the space property, and the assert/negate timing for the $\overline{OE}$ ($\overline{RD}$) and $\overline{WE}$ signals. In addition, the wait timing for area 5 and 6 access can be set in CSnPCR for the first half and second half individually. The first half of area 5 is allocated from H'1400 0000 to H'15FF FFFF, and the second half of area 5 is allocated from H'1600 0000 to H'17FF FFFF. The first half of area 6 is allocated from H'1800 0000 to H'19FF FFFF, and the second half of area 6 is allocated from H'1A00 0000 to H'1BFF FFFF (each address is an external address).

The pulse widths of $\overline{OE}$ and $\overline{WE}$ assertion for the first half of area 5 and 6 are set using the IW bits in CSnWCR. CSnPCR is initialized to H'7700 0000 by a power-on reset, but is not initialized by a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SAA			—	SAB			PCWA		PCWB		PCIW			
Initial value:	0	1	1	1	0	1	1	1	0	0	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	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	SAA	111	R/W	Space Property A Specify the space property of PCMCIA connected to 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

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	SAB	111	R/W	Space Property B Specify the space property of PCMCIA connected to second 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
23, 22	PCWA	00	R/W	PCMCIA Wait A Wait cycle for low-speed PCMCIA. The number of wait cycles specified by these bits is added to the number designated by the IW bits in CSnWCR. These bits are valid, when the access area of PCMCIA 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 number of wait cycles specified by these bits is added to the number designated by PCIW. These bits are valid, when the access area of PCMCIA 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

Bit	Bit Name	Initial Value	R/W	Description
19 to 16	PCIW	0000	R/W	PCMCIA Insert Wait Cycle B Specify the number of wait cycles to be inserted. These bits are valid, when the access area of PCMCIA interface is second half of area n (n = 5 and 6). 0000: No cycle inserted 0001: 1 cycle inserted 0010: 2 cycles inserted 0011: 3 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 designated by CSnWCR when the access area of PCMCIA interface is first half of area 5 or 6.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	TEDA	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay A These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion for the access of first half area of 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
11	—	0	R	Reserved This bit is always read as 0. The write value should 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 output to $\overline{OE}$/$\overline{WE}$ assertion for the access of second half area of 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 should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	TEHA	000	R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay A These bits set the delay time from $\overline{OE}$/$\overline{WE}$ negation to address hold for the access of first half area of 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
3	—	0	R	Reserved This bit is always read as 0. The write value should 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 $\overline{OE}$/$\overline{WE}$ negation to address hold for the access of second half area of 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

11.5 Operation

11.5.1 Endian/Access Size and Data Alignment

This LSI supports both big-endian mode, in which the upper byte in a string of byte data is at address 0, and little-endian mode, in which the lower byte in a string of byte data is at address 0. The mode is specified by the external pin (MODE5 pin) at a power-on reset through the RESET pin. At a power-on reset by $\overline{\text{PRESET}}$, big-endian mode is specified when the MODE5 pin is low, and little-endian mode is specified when the MODE5 pin is high.

A data bus width of 8, 16, or 32 bits can be selected for the normal memory interface, and one of 8 or 16 bits can be selected for the PCMCIA interface. Data alignment is carried out according to the data bus width and endian mode of each device. Accordingly, when the data bus width is smaller than the access size, multiple bus cycles are automatically generated to reach the access size. In this case, access is performed by incrementing the addresses corresponding to the bus width. For example, when a longword access is performed at the area with an 8-bit width in the SRAM interface, each address is incremented one by one, and then access is performed four times. In the 32-byte transfer, a total of 32-byte data is continuously transferred according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during these transfers. In this LSI, data alignment and data length conversion between different interfaces is performed automatically.

When an 8- or 16-byte transfer is requested, the LBSC executes the transfer in two or four separate 4-byte accesses.

The relationship between the endian mode, device data length, and access unit are shown in tables 11.9 to 11.14.

Table 11.9 32-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	Data 7 to 0	—	—	—	Assert			
	4n + 1	1	—	Data 7 to 0	—	—		Assert		
	4n + 2	1	—	—	Data 7 to 0	—			Assert	
	4n + 3	1	—	—	—	Data 7 to 0				Assert
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	Assert
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.10 16-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	Data 7 to 0	—			Assert	
	2n + 1	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert
	4n + 2	2	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert

Table 11.11 8-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	—	Data 15 to 8				Assert
	2n + 1	2	—	—	—	Data 7 to 0				Assert
Longword	4n	1	—	—		Data 31 to 24				Assert
	4n + 1	2	—	—	—	Data 23 to 16				Assert
	4n + 2	3	—	—	—	Data 15 to 8				Assert
	4n + 3	4	—	—	—	Data 7 to 0				Assert

Table 11.12 32-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	—	—	—	Data 7 to 0				Assert
	4n + 1	1	—	—	Data 7 to 0	—			Assert	
	4n + 2	1	—	Data 7 to 0	—	—		Assert		
	4n + 3	1	Data 7 to 0	—	—	—	Assert			
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	Assert		
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.13 16-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	—	Data 7 to 0				Assert
	2n + 1	1	—	—	Data 7 to 0	—			Assert	
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	4n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert

Table 11.14 8-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	—	Data 7 to 0				Assert
	2n + 1	2	—	—	—	Data 15 to 8				Assert
Longword	4n	1	—	—	—	Data 7 to 0				Assert
	4n + 1	2	—	—	—	Data 15 to 8				Assert
	4n + 2	3	—	—	—	Data 23 to 16				Assert
	4n + 3	4	—	—	—	Data 31 to 24				Assert

11.5.2 Areas

(1) Area 0

For area 0, physical address bits 28 to 26 are 000.

The interfaces that can be set for this area are the SRAM, burst ROM and MPX interfaces.

A bus width of 8, 16, or 32 bits is selectable with external pins MODE4 and MODE3 at a power-on reset. For details, see section 11.3.2, Memory Bus Width.

When area 0 is accessed, the $\overline{CS0}$ signal is asserted.

In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as OE, and write 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 selectable with bits BW2 to BW0 in CS0BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (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 selected 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 and write strobe signals can be set within a range of 0 to 7 cycles by CS0WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

(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-control SRAM interfaces.

A bus width of 8, 16, or 32 bits is selectable with bits SZ in CS1BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ in CS1BCR. When using the byte-control SRAM interface, select a bus width of 16 or 32 bits.

When area 1 is accessed, the $\overline{CS1}$ signal is asserted.

In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS1WCR can be selected.

When the burst ROM interface is used, a burst pitch number in the range of 0 to 7 is selectable with bits BW2 to BW0 in CS1BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{CS1}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS1WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

(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-SDRAM interfaces.

When the SRAM interface is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ in CS2BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits 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 write 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 ($\overline{RDY}$). (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 and write strobe signals can be set within a range of 0 to 7 cycles by CS2WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

When using area 2 for the DDR-SDRAM interface, set the AREASEL bit in MMSELR. Then the $\overline{CS2}$ signal is not asserted. When the DDR-SDRAM is used, see section 12, DDR-SDRAM Interface (DDRIF).

(4) Area 3

For area 3, physical address bits 28 to 26 are 011.

This area is used only for the DDR-SDRAM interface. For details, see section 12, DDR-SDRAM Interface (DDRIF).

(5) Area 4

For area 4, physical address bits 28 to 26 are 100.

The interfaces that can be set for this area are the SRAM, burst ROM, MPX , byte control SRAM, 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 interface is used, a bus width of 32 bits should be selected through bits SZ1 and SZ0 in CS4BCR. When the byte control SRAM interface is used, select a bus width of 16 or 32 bits. For details, see section 11.3.2, Memory Bus Width.

When area 4 is accessed, the $\overline{\text{CS4}}$ signal is asserted (except for DDR-SDRAM and PCI areas).

In the case where the SRAM interface is set, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS4WCR can be selected. Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (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 $\overline{\text{CS4}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS4WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

When using area 4 as the DDR-SDRAM or PCI local bus interface, set the AREASEL bit in MMSELR. Then the $\overline{\text{CS4}}$ signal is not asserted. When the DDR-SDRAM or PCI is used, see section 12, DDR-SDRAM Interface (DDRIF) or section 13, PCI Controller (PCIC), respectively.

(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, and DDR-SDRAM interfaces.

When the SRAM or burst ROM interface is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ in CS5BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ in CS5BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ in CS5BCR. For details, see section 11.3.2, Memory Bus Width.

When area 5 is accessed, the $\overline{CS5}$ signal is asserted.

In addition, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted. While the PCMCIA interface is used, the $\overline{CE1A}$ and $\overline{CE2A}$ signals, the $\overline{RD}$ signal, (which can be used as $\overline{OE}$), the $\overline{WE0}$, $\overline{WE1}$, $\overline{WE2}$, and $\overline{WE3}$ signals, (which can be used as, $\overline{REG}$, $\overline{WE}$, $\overline{IORD}$, and $\overline{IOWR}$, respectively) are asserted.

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 pin ($\overline{RDY}$). (When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{CS5}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS5WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{CE1A}$ and $\overline{CE2A}$) can be specified within a range from 0 to 15 cycles through bits TEDA/B2 to TEDA/B0 and TEDA/B to TEHA/B in CS5PCR. In addition, the number of wait cycles can be specified within a range from 0 to 50 cycles through bits PCWA/B1 and PCWA/B0. The number of wait cycles specified by CS5PCR is added to the value specified by IW3 to IW0 in CS5WCR or PCIW3 to PCIW0 in CS5PCR.

When using area 5 for the DDR-SDRAM interface, set the AREASEL bit in MMSELR. Then the $\overline{CS5}$ signal is not asserted. When the DDR-SDRAM is used, see section 12, DDR-SDRAM 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 PCMCIA interfaces.

When the SRAM or burst ROM is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ in CS6BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ in CS6BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ in CS6BCR. For details, see section 11.3.2, Memory Bus Width.

When area 6 is accessed, the $\overline{CS6}$ signal is asserted.

In addition, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted. While the PCMCIA interface is used, the $\overline{CE1B}$ and $\overline{CE2B}$ signals, the $\overline{RD}$ signal (which can be used as $\overline{OE}$), and the $\overline{WE0}$, $\overline{WE1}$, $\overline{WE2}$, and $\overline{WE3}$ signals (which can be used as $\overline{REG}$, $\overline{WE}$, $\overline{IORD}$, and $\overline{IOWR}$, respectively) are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS6WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{CS6}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS6WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1 or more.

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{CE1B}$ and $\overline{CE2B}$) can be specified within a range from 0 to 15 cycles by bits TEDA/B2 to TEDA/B0 and TEHA/B2 to TEHA/B0 in CS6PCR. In addition, the number of wait cycles can be specified within a range from 0 to 50 cycles by bits PCWA/B1 and PCWA/B0. The number of wait cycles 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 SRAM connection. Figure 11.4 shows the basic timing of the SRAM interface. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{CSn}$ signal is asserted at the rising edge of the clock in the T1 state, and negated at the next rising edge of the clock in the T2 state. Therefore, there is no negation period in the case of access at minimum pitch.

During reading, specification of an access size is not needed. The output of an access address on the address pins (A25 to A0) is correct, however, since the access size is not specified, 32-bit data is always output when a 32-bit device is in use, and 16-bit data is output when a 16-bit device is in use. During writing, only the $\overline{WE}$ signal corresponding to the byte to be written is asserted. For 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 width set. The first access is performed on the data for which there was an access request, and the remaining

accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.

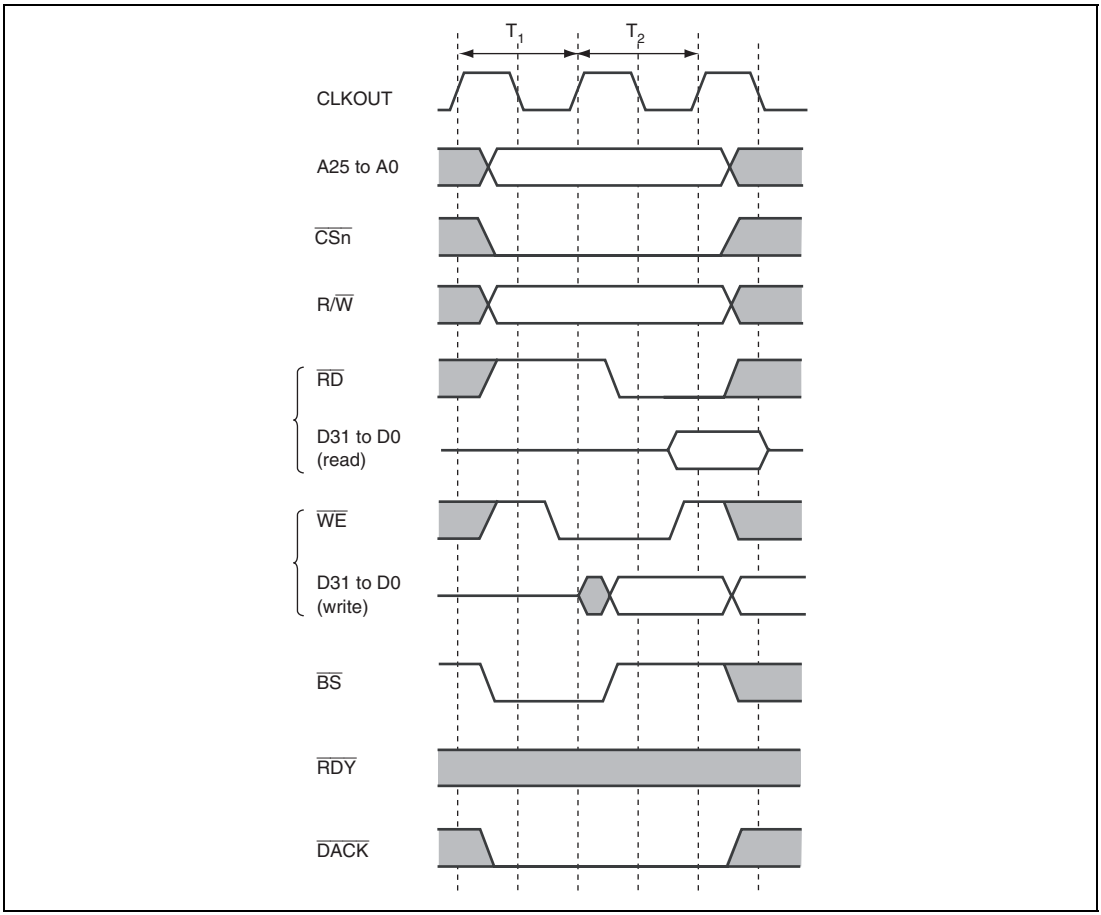


Figure 11.4 Basic Timing of SRAM Interface

Figures 11.5 to 11.7 show examples of the connection to SRAM with data width of 32, 16, and 8 bits.

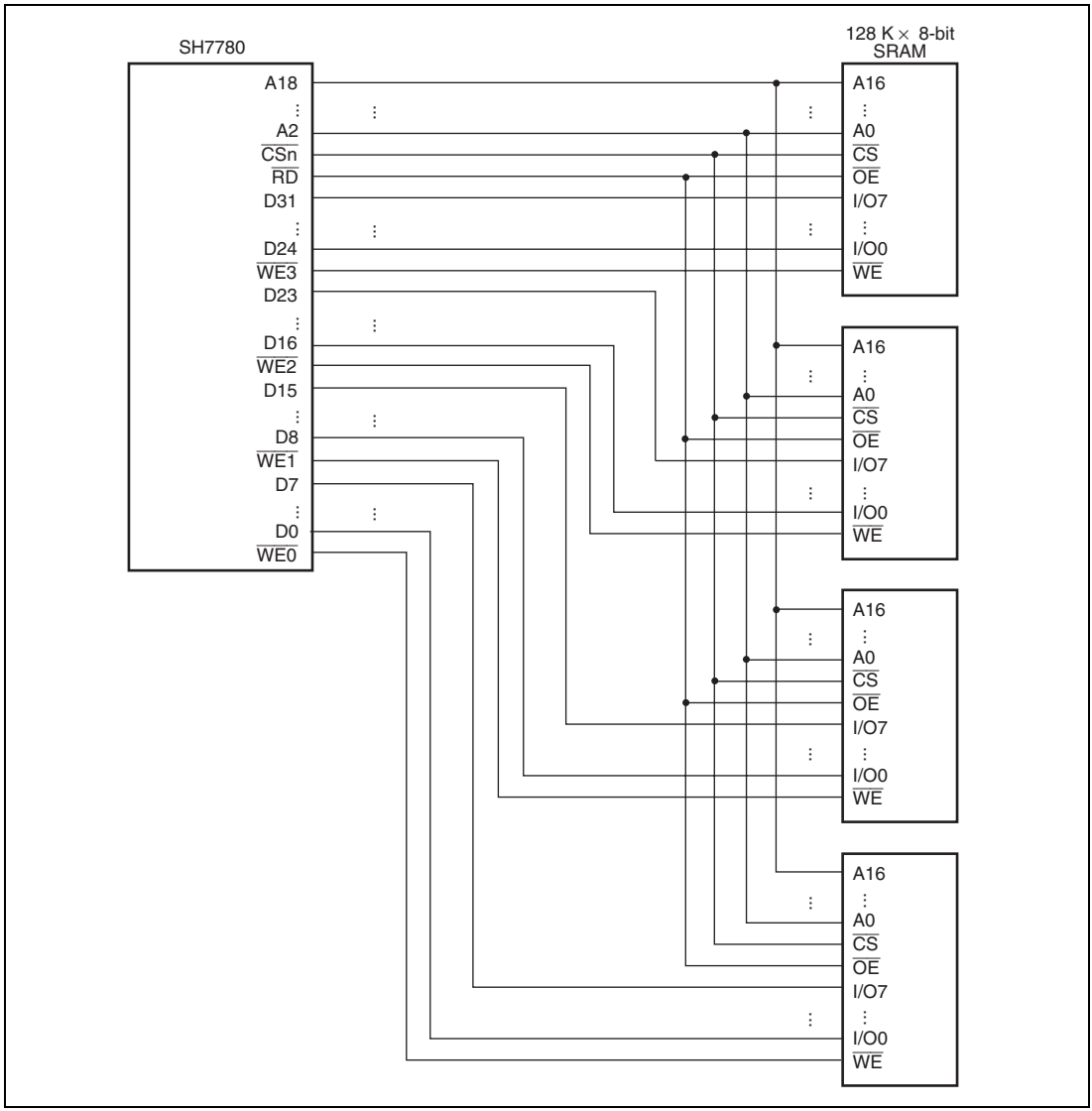


Figure 11.5 Example of 32-Bit Data-Width SRAM Connection

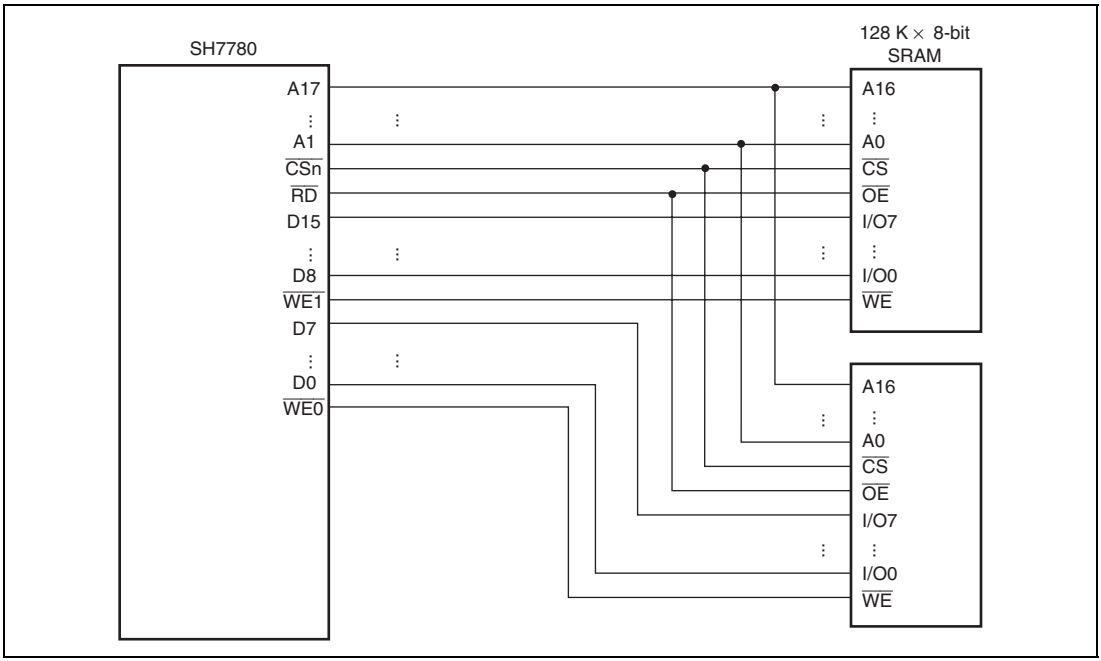


Figure 11.6 Example of 16-Bit Data-Width SRAM Connection

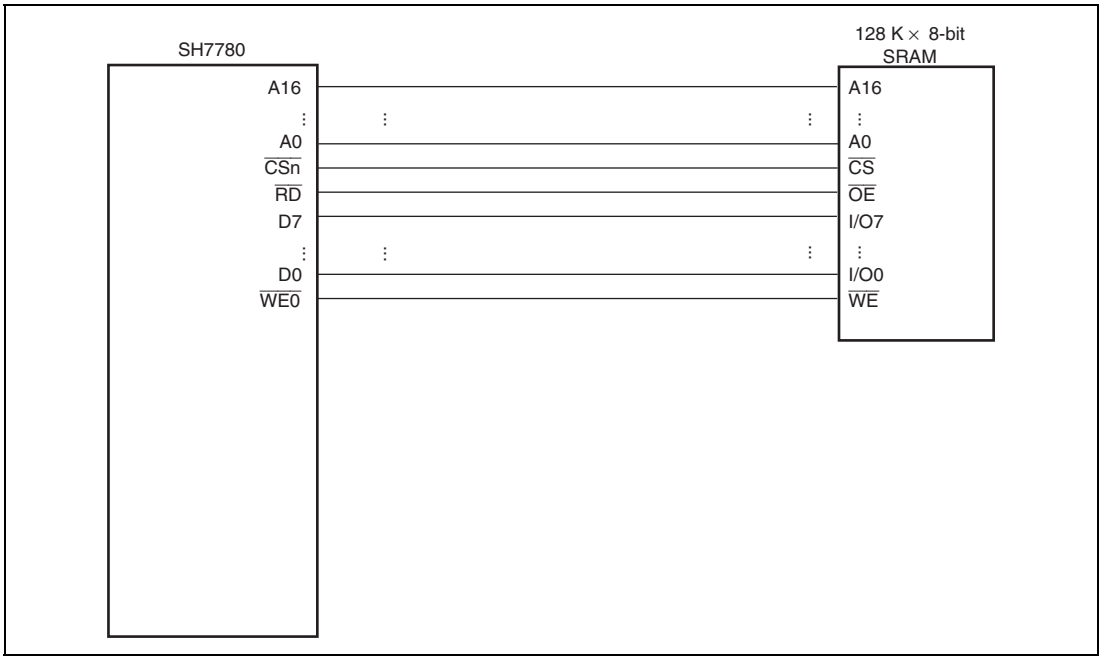


Figure 11.7 Example of 8-Bit Data-Width SRAM Connection

(2) Wait Cycle Control

Wait cycle insertion for the SRAM interface can be controlled by CSnWCR. If the IW bits for each area in CSnWCR is not 0, a software wait is inserted in accordance with the wait-control bits. For details, see section 11.4.4, CSn Wait Control Register (CSnWCR).

A specified number of Tw cycles is inserted as wait cycles in accordance with the CSnWCR setting. The insertion timing of the wait cycle is shown in figure 11.8.

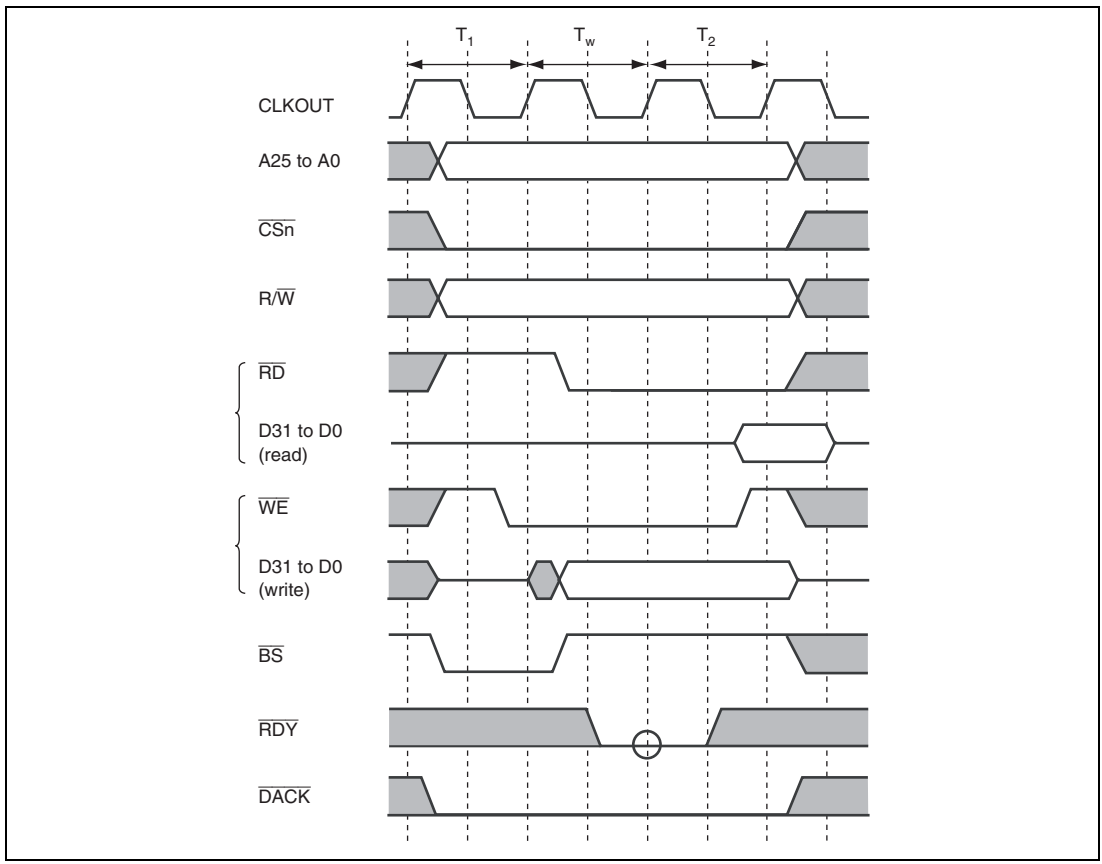


Figure 11.8 SRAM Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by CSnWCR, the external wait input signal ($\overline{\text{RDY}}$) is also sampled. The $\overline{\text{RDY}}$ signal sampling timing is shown in figure 11.9, where a single wait cycle is specified as a software wait. The $\overline{\text{RDY}}$ signal is sampled at the transition from the Tw state to the T2 state. Therefore, the assertion of the $\overline{\text{RDY}}$ signal has no effect in the T1 cycle or in the first Tw cycle. The $\overline{\text{RDY}}$ signal is sampled on the rising edge of the clock.

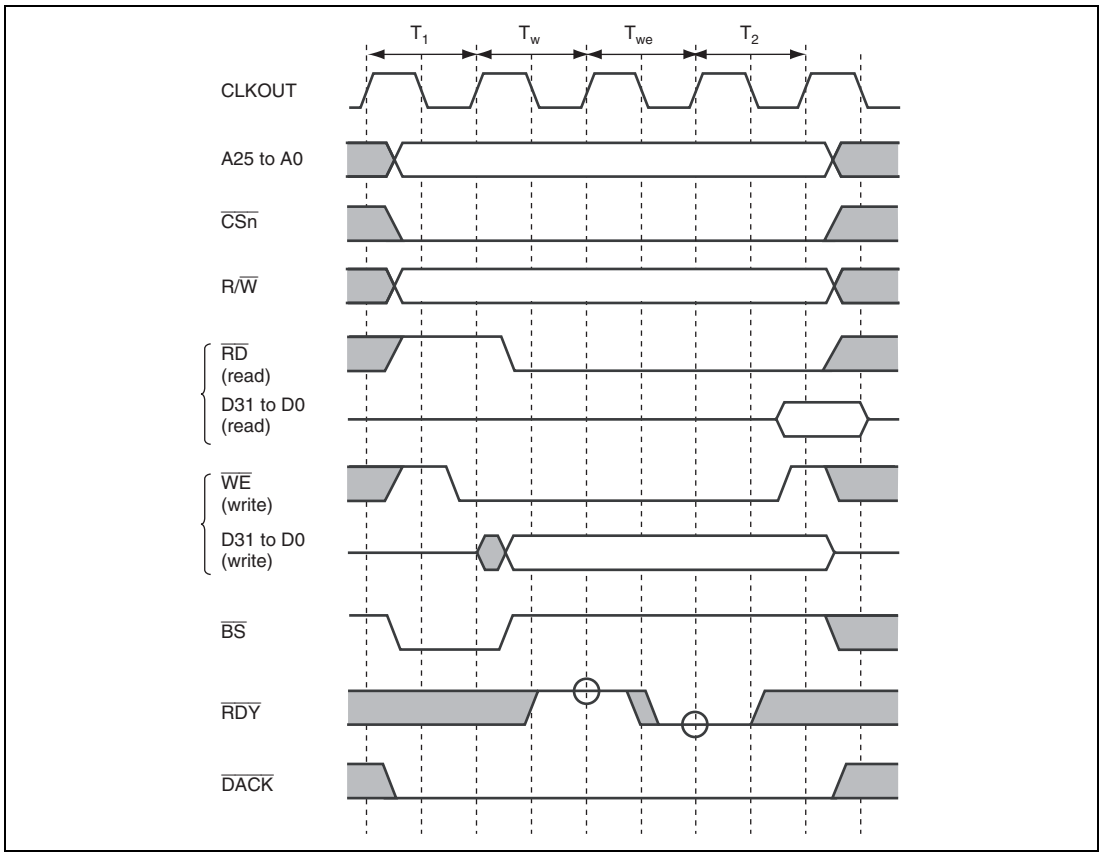
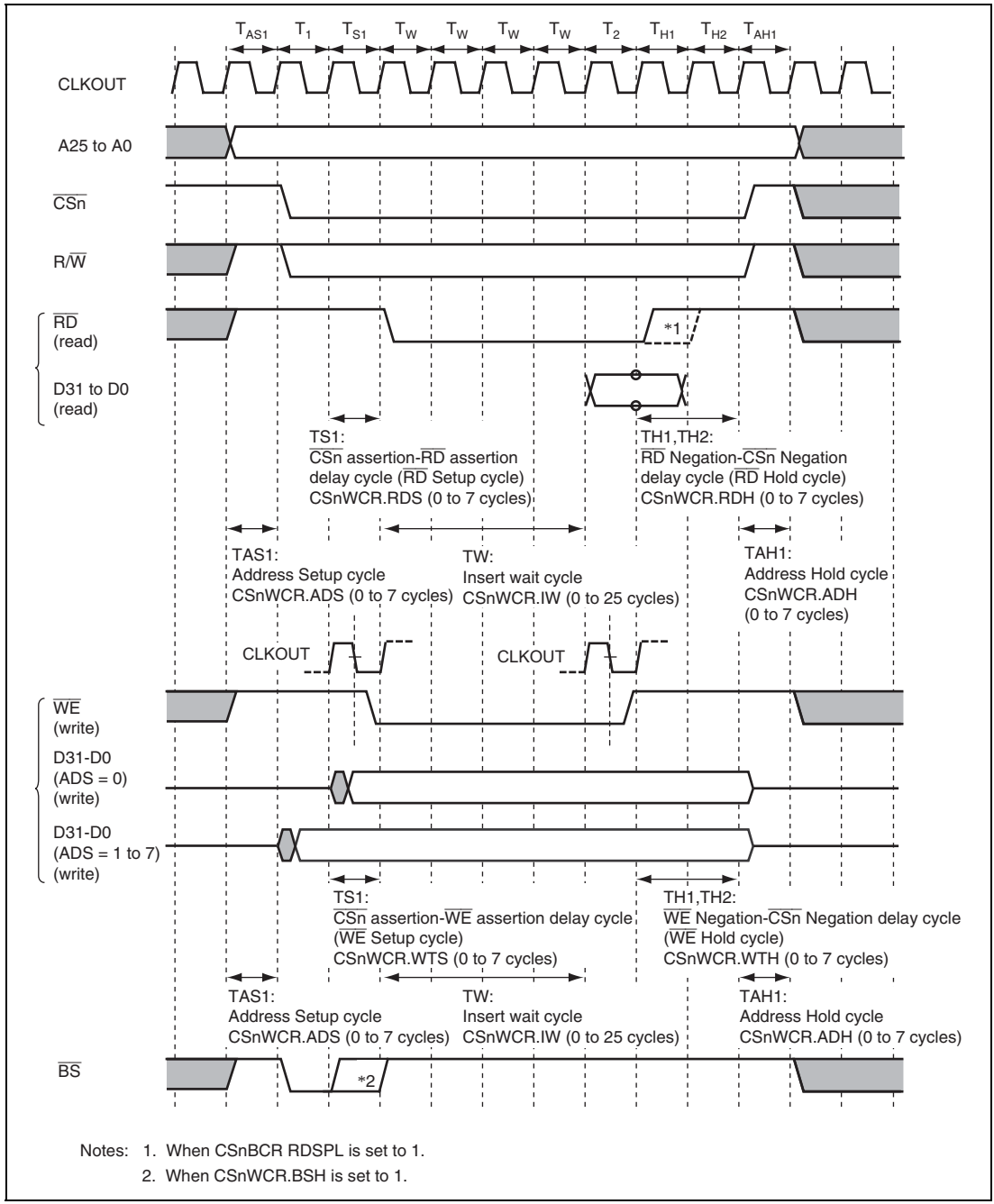


Figure 11.9 SRAM Interface Wait Timing
(Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal, $\overline{\text{RDY}}$ Signal is synchronous input)

(3) Read-Strobe Negate Timing

When the SRAM interface is used, the negation timing of the strobe signal during a read operation can be specified through the RDSPL bit in CSnBCR. For details of settings, see section 11.4.3, CSn Bus Control Register (CSnBCR). Clear the RDSPL bit to 0, when using a byte control SRAM.



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 Setting)

11.5.4 Burst ROM (Clock Asynchronous) Interface

Setting the TYPE bit in CSnBCR (n = 0 to 2 and 4 to 6) to 010 allows a burst ROM (clock asynchronous memory) to be connected to areas 0 to 2 and 4 to 6. The burst ROM interface provides high-speed access to ROM that has a burst access function. The burst access timing of burst ROM is shown in figure 11.11. The wait cycle is set to 0 cycle. Although the access is similar to that of the SRAM interface, only the address is changed when the first cycle ends and then the next access is started. When 8-bit ROM is used, the number of consecutive accesses can be set as 4, 8, 16, or 32 through bits BST2 to BST0 in CSnBCR (n = 0 to 2 and 4 to 6). Similarly, when 16-bit ROM is used, 4, 8 or 16 accesses can be set; when 32-bit ROM is used, 4 or 8 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 in the 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 interface.

In a 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The burst 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 the 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 specified by CSnWCR.

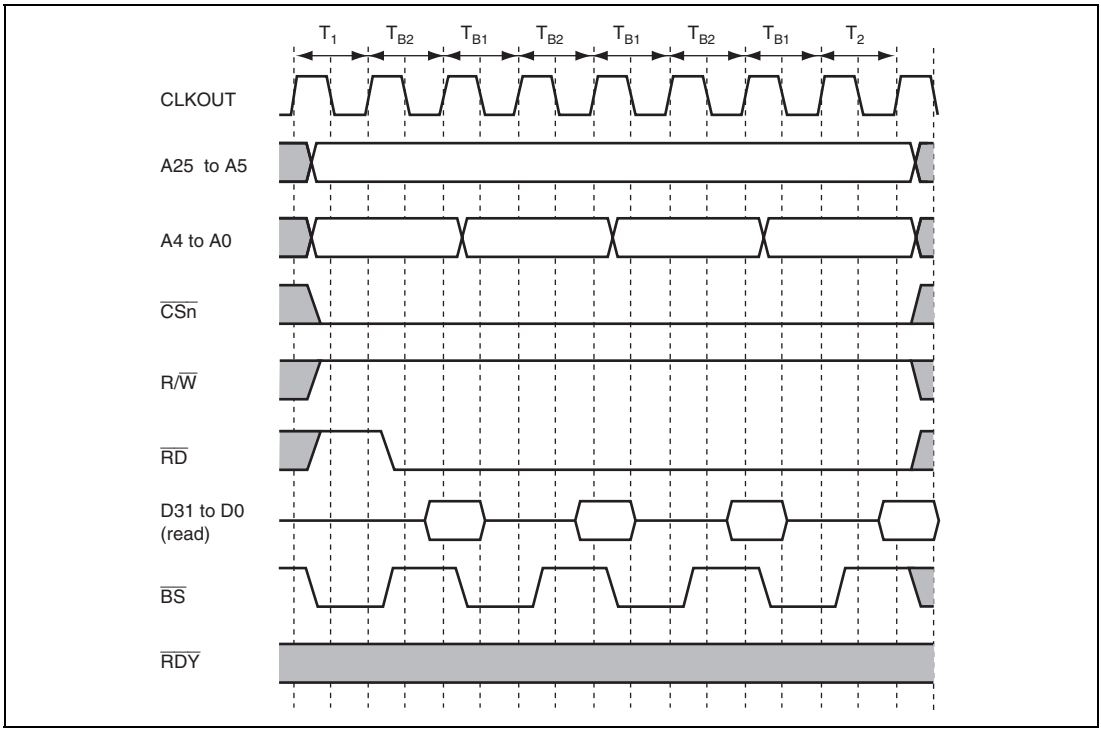


Figure 11.11 Burst ROM Basic Timing

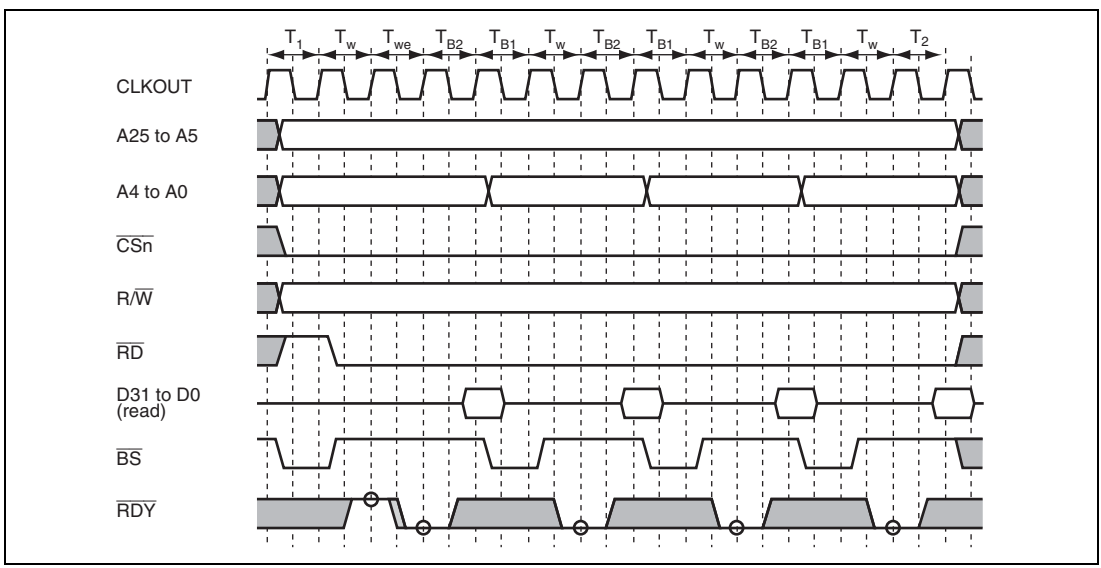


Figure 11.12 Burst ROM Wait Timing

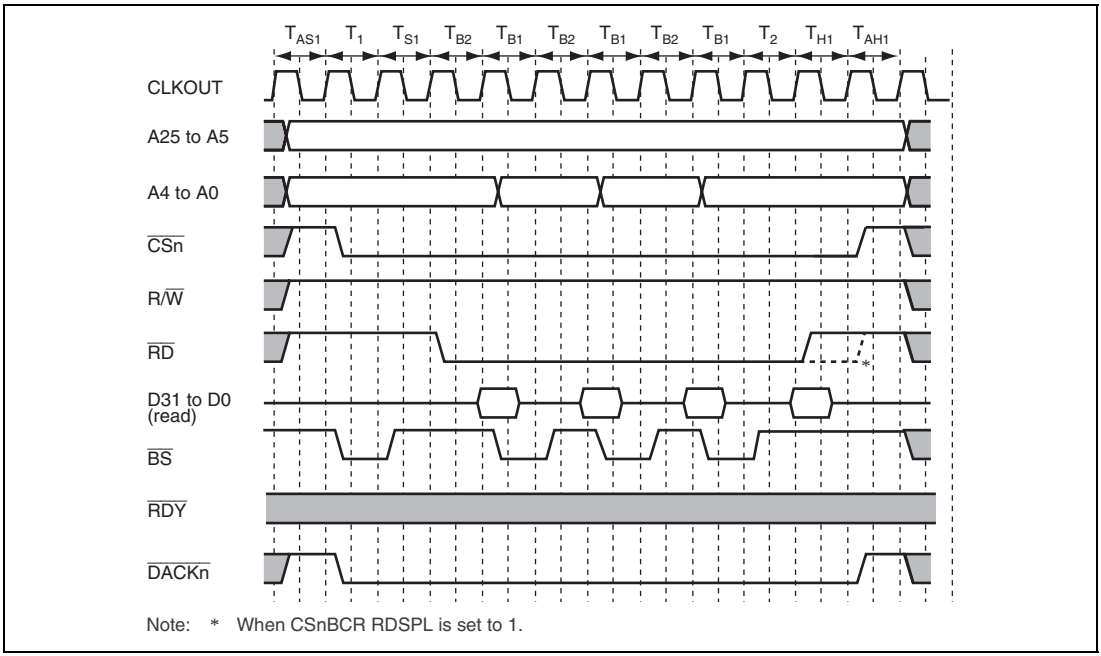


Figure 11.13 Burst ROM Wait Timing

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 stipulated in JEIDA specification version 4.2 (PCMCIA 2.1), by setting the TYPE bits in CS5BCR and CS6BCR.

Since operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA standard, this LSI supports the PCMCIA interface only in little-endian mode through little-endian mode setting.

The PCMCIA interface can select the space property from among 8-bit common memory, 16-bit common memory, 8-bit attribute memory, 16-bit attribute memory, 8-bit I/O space, 16-bit I/O space, dynamic I/O bus sizing, and ATA complement mode by depending on the setting of 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, TEDA, and TEHA in CSnPCR (n = 5 or 6) are selected. When the second half area is accessed, bit IW in CSnWCR (n = 5 or 6) and bits PCWB, TEDB, and TEHB in CSnPCR (n = 5 or 6) are selected.

Bits PCWA/B1 and PCWA/B0 can be used to set the number of wait cycles to be inserted in a low-speed bus cycle as 0, 15, 30, or 50. This value is added to the number of inserted wait cycles specified by IW bit in CSnWCR or PCIW bit in CSnPCR. Bit TEDA/B (with a setting range from 0 to 15) can be used to ensure the setup times of the address, $\overline{CE1A}$ ($\overline{CS5}$), $\overline{CE1B}$ ($\overline{CS6}$), $\overline{CE2A}$, $\overline{CE2B}$ and $\overline{REG}$ to the $\overline{RD}$ and $\overline{WE1}$ signals. Bits TEHA/B (with a setting range from 0 to15) can be used to ensure the hold times of the address, $\overline{CE1A}$ ($\overline{CS5}$), $\overline{CE1B}$ ($\overline{CS6}$), $\overline{CE2A}$, $\overline{CE2B}$, and $\overline{REG}$ to the $\overline{RD}$ and $\overline{WE1}$ signals.

Bits IWW, IWRWD, IWRWS, IWRRD, and IWRRS in the CS5 bus control register (CS5BCR) or CS6 bus control register (CS6BCR) are used to set the number of idle cycles between cycles. The selected number of wait cycles between cycles depends only on the area to be accessed (area 5 or 6). When area 5 is accessed, bits IWW, IWRWD, IWRWS, IWRRD, and IWRRS in CS5BCR are selected, and when area 6 is accessed, bits IWW, IWRWD, IWRWS, IWRRD, and IWRRS in CS6BCR are selected.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.

ATA complement mode is to access the ATA device register connected to this LSI. The Device Control Register, Alternate Status Register, Data Register, and Data Port can be accessed in ATA complement mode.

To access the Device Control Register and Alternate Status Register, use a CPU byte access (do not use a DMA transfer), and to access the Data Register, use the CPU word access (do not use a DMA transfer). When a CPU byte access is executed, $\overline{CE1x}$ is negated and $\overline{CE2x}$ is asserted (x = A, B). When a CPU word access is executed, $\overline{CE1x}$ is asserted and $\overline{CE2x}$ is negated.

To access the Data Port use a DMA transfer. The setting example of the DMAC is external request, burst mode, level detection, overrun 0, $\overline{DACK}$ output to the correspondent PCMCIA connected area. When DMA transfer of an ATA complement mode area is executed, neither $\overline{CE1x}$ nor $\overline{CE2x}$ is asserted. Set the DACKBST bit in BCR of the corresponding DMA transfer channel to 1, so that the corresponding $\overline{DACK}$ signal is asserted from the beginning to the end of the DMA transfer cycle.

Specify the number of wait cycles between accesses as 0 for the $\overline{DACK}$ assertion area when setting the DMA transfer size to 16-byte. After the DMA burst transfer that DACKBST was enabled has finished, set the DACKBST bit to 1 again before starting the next DMA burst transfer.

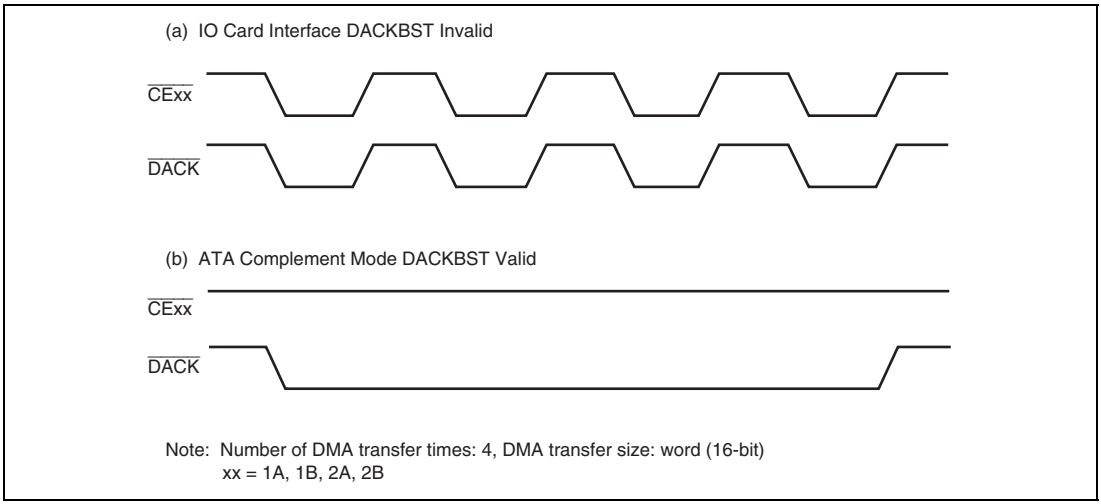


Figure 11.14 $\overline{\text{CExx}}$ and $\overline{\text{DACK}}$ Output of ATA Complement Mode in DMA Transfer

Figure 11.15 shows an example of PCMCIA card connection to this LSI. To enable hot insertion of PCMCIA cards (i.e., insertion or removal while system power is being supplied), a three-state buffer must be connected between this LSI bus interface and the PCMCIA cards.

Table 11.15 Relationship between Address and CE When Using PCMCIA Interface

Bus (Bits)	Read/ Write	Access (bits)*[1]	Odd/ Even	$\overline{IOIS16}$	Access	$\overline{CE2}$	$\overline{CE1}$	A0	D15 to D8	D7 to D0
8	Read	8	Even	×	—	H	L	L	Invalid	Read data
			Odd	×	—	H	L	H	Invalid	Read data
		16	Even	×	First	H	L	L	Invalid	Lower read data
			Even	×	Second	H	L	H	Invalid	Upper read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	H	L	L	Invalid	Write data
			Odd	×	—	H	L	H	Invalid	Write data
		16	Even	×	First	H	L	L	Invalid	Lower write data
			Even	×	Second	H	L	H	Invalid	Upper write data
			Odd	×	—	—	—	—	—	—
16	Read	8	Even	×	—	H	L	L	Invalid	Read data
			Odd	×	—	L	H	H	Read data	Invalid
		16	Even	×	—	L	L	L	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	H	L	L	Invalid	Write data
			Odd	×	—	L	H	H	Write data	Invalid
		16	Even	×	—	L	L	L	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
Dynamic Bus Sizing*[2]	Read	8	Even	L	—	H	L	L	Invalid	Read data
			Odd	L	—	L	H	H	Read data	Invalid
		16	Even	L	—	L	L	L	Upper read data	Lower read data
			Odd	L	—	—	—	—	—	—
	Write	8	Even	L	—	H	L	L	Invalid	Write data
			Odd	L	—	L	H	H	Write data	Invalid
		16	Even	L	—	L	L	L	Upper write data	Lower write data
			Odd	L	—	—	—	—	—	—
	Read	8	Even	H	—	H	L	L	Invalid	Read data
			Odd	H	First	L	H	H	Invalid	Invalid
			Odd	H	Second	H	L	L	Invalid	Read data
		16	Even	H	First	L	L	L	Invalid	Lower read data
			Even	H	Second	H	L	H	Invalid	Upper read data
			Odd	H	—	—	—	—	—	—

Bus (Bits)	Read/ Write	Access (bits)*1	Odd/ Even	$\overline{\text{IOIS16}}$	Access	$\overline{\text{CE2}}$	$\overline{\text{CE1}}$	A0	D15 to D8	D7 to D0
Dynamic Bus Sizing*2	Write	8	Even	H	—	H	L	L	Invalid	Write data
			Odd	H	First	L	H	H	Invalid	Write data
			Odd	H	Second	H	L	H	Invalid	Write data
		16	Even	H	First	L	L	L	Upper write data	Lower write data
			Even	H	Second	H	L	H	Invalid	Upper write data
			Odd	H	—	—	—	—	—	—
ATA comple-ment mode	read (does not output DACK)	8	Even	×	—	L	H	L	Invalid	Read data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	write (does not output DACK)	8	Even	×	—	L	H	L	Invalid	Write data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
	read (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Read data
			Odd	×	—	H	H	L	Read data	Invalid
		16	Even	×	—	H	H	H	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	write (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Write data
			Odd	×	—	H	H	L	Write data	Invalid
		16	Even	×	—	H	H	H	Upper write data	Lower write data
			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 incremented by the bus width, and then above accesses are repeated until the transfer data size is reached.

2. PCMCIA I/O card interface only.

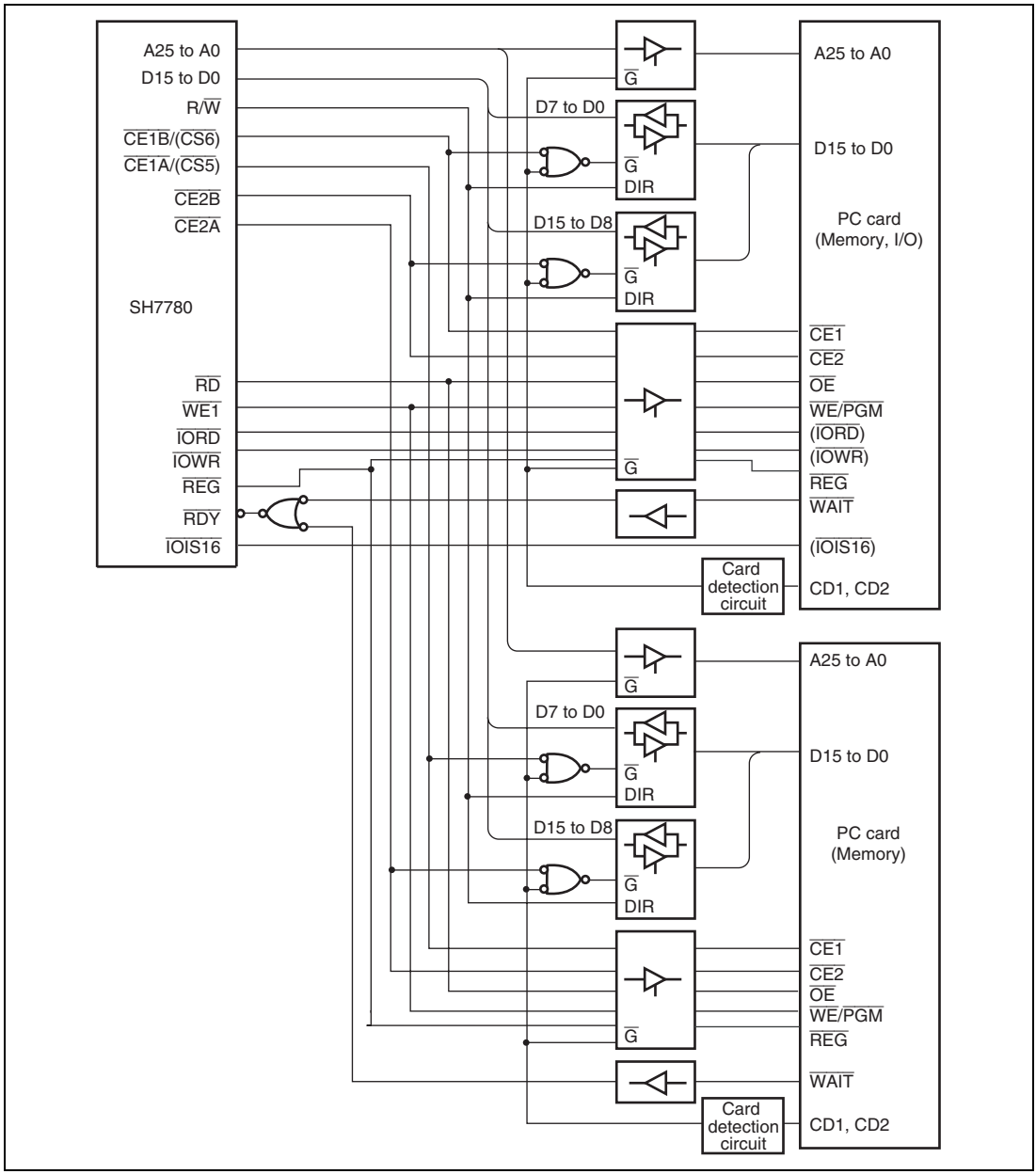


Figure 11.15 Example of PCMCIA Interface

(1) Memory Card Interface Basic Timing

Figure 11.16 shows the basic timing for the PCMCIA memory card interface, and figure 11.17 shows the wait timing for the PCMCIA memory card interface.

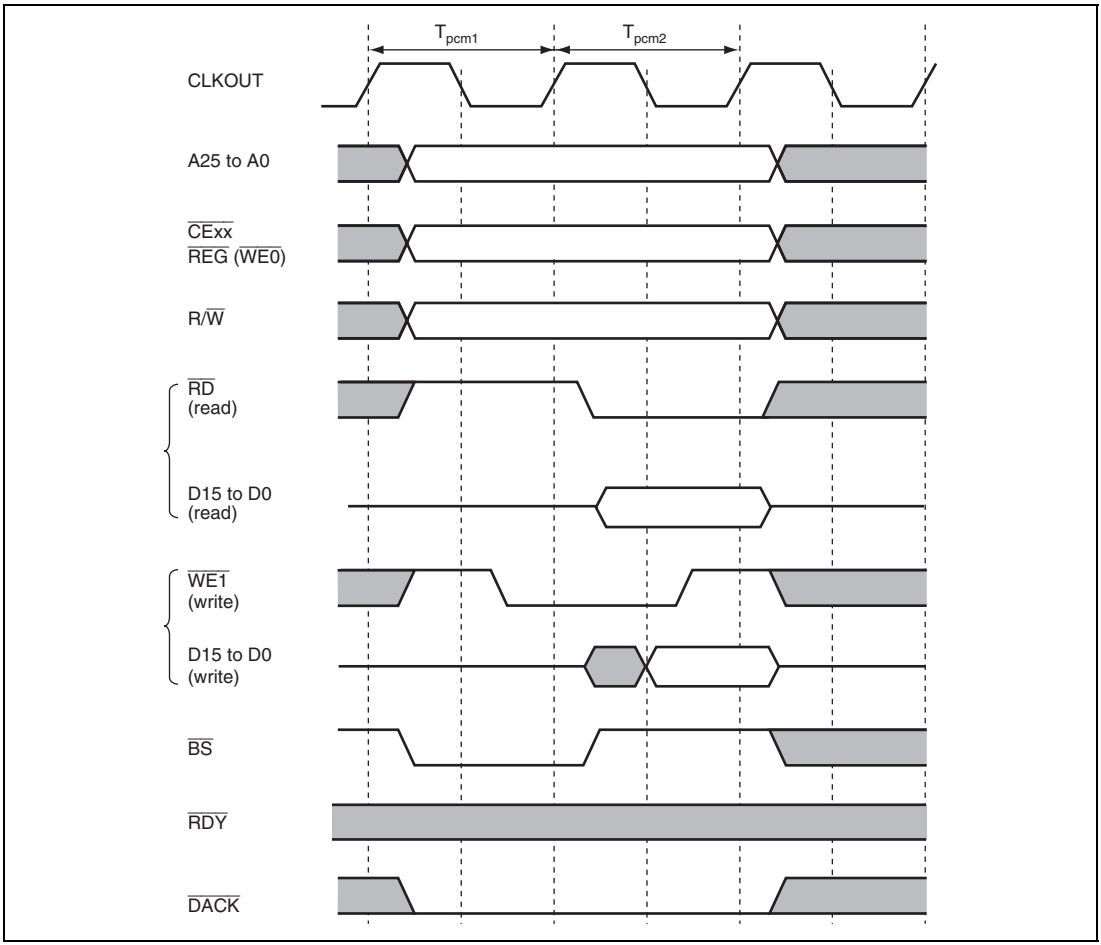


Figure 11.16 Basic Timing for PCMCIA Memory Card Interface

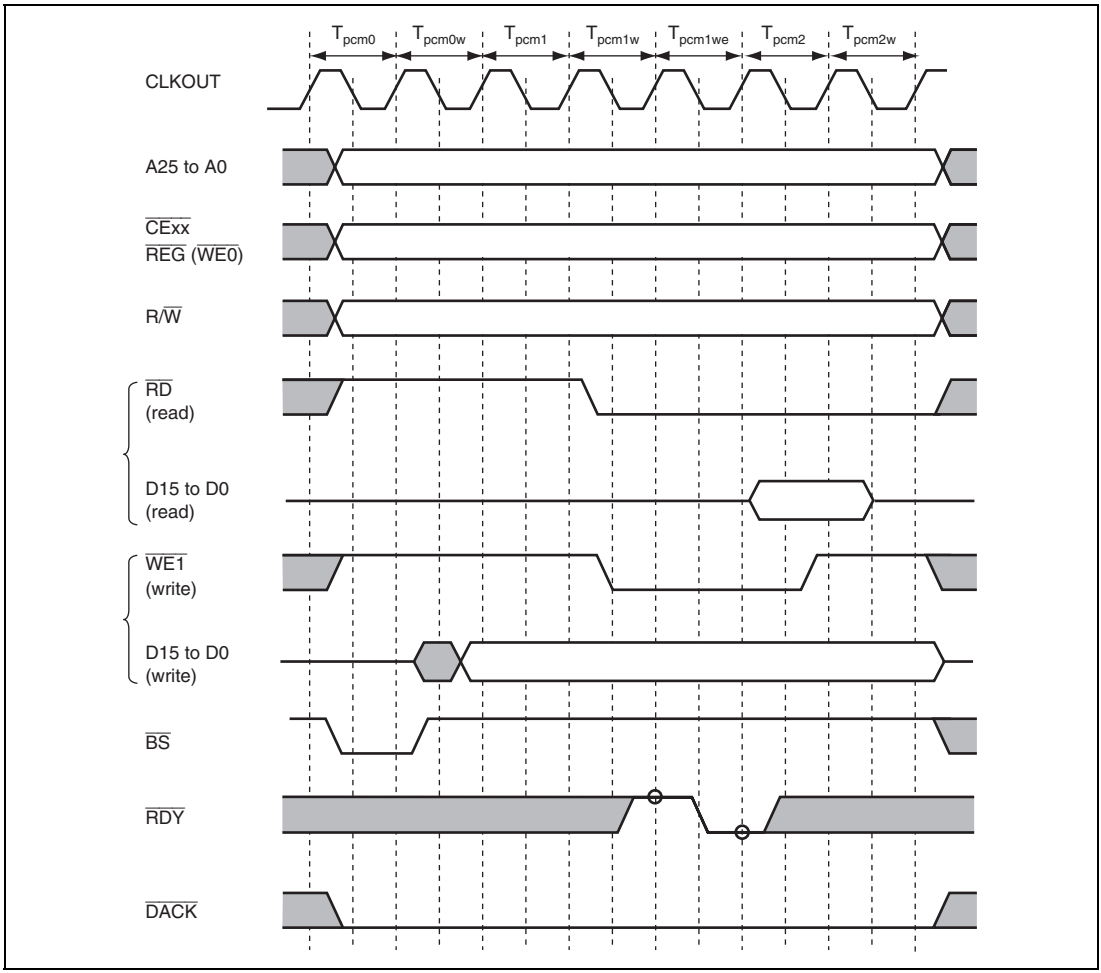


Figure 11.17 Wait Timing for PCMCIA Memory Card Interface

(2) I/O Card Interface Timing

Figures 11.18 and 11.19 show the timing for the PCMCIA I/O card interface.

When accessing a PCMCIA card via the I/O card interface, it is possible to perform dynamic sizing of the I/O bus width using the $\overline{\text{IOIS16}}$ pin. With the 16-bit bus width selected, if the $\overline{\text{IOIS16}}$ signal is high during the word-size I/O bus cycle, the I/O port is recognized as eight bits in bus width. In this case, a data access for only eight bits is performed in the I/O bus cycle being executed, and this is automatically followed by a data access for the remaining eight bits. Dynamic bus sizing is also performed for byte-size access to address 2n + 1.

Figure 11.20 shows the basic timing for dynamic bus sizing.

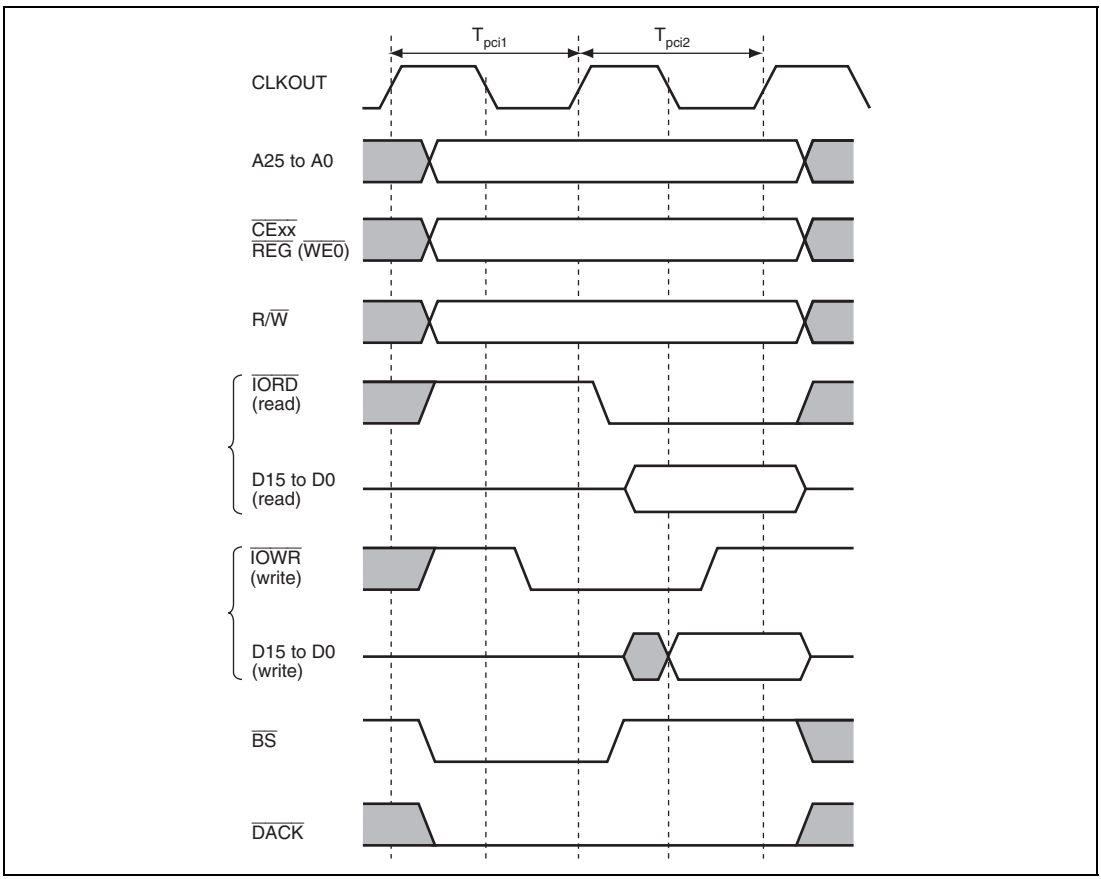


Figure 11.18 Basic Timing for PCMCIA I/O Card Interface

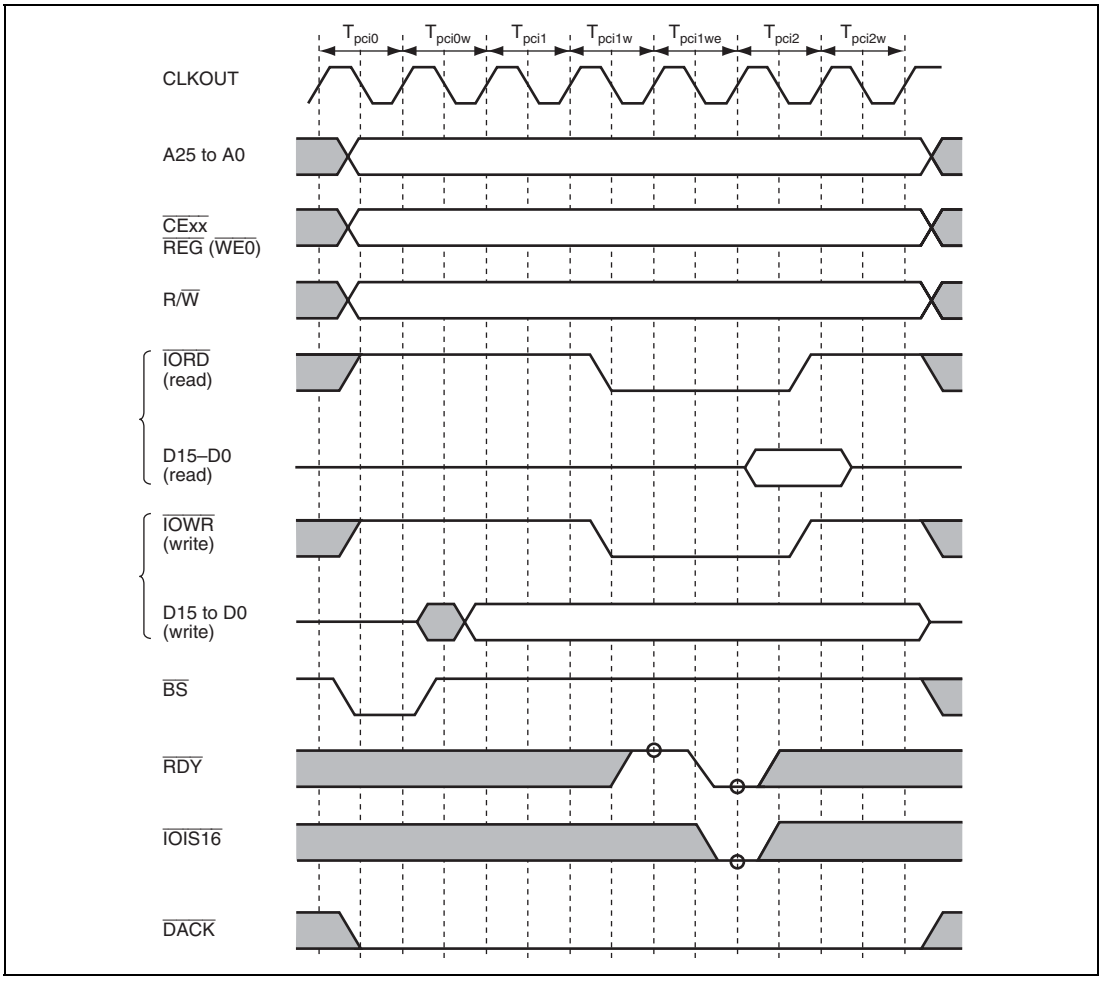


Figure 11.19 Wait Timing for PCMCIA I/O Card Interface

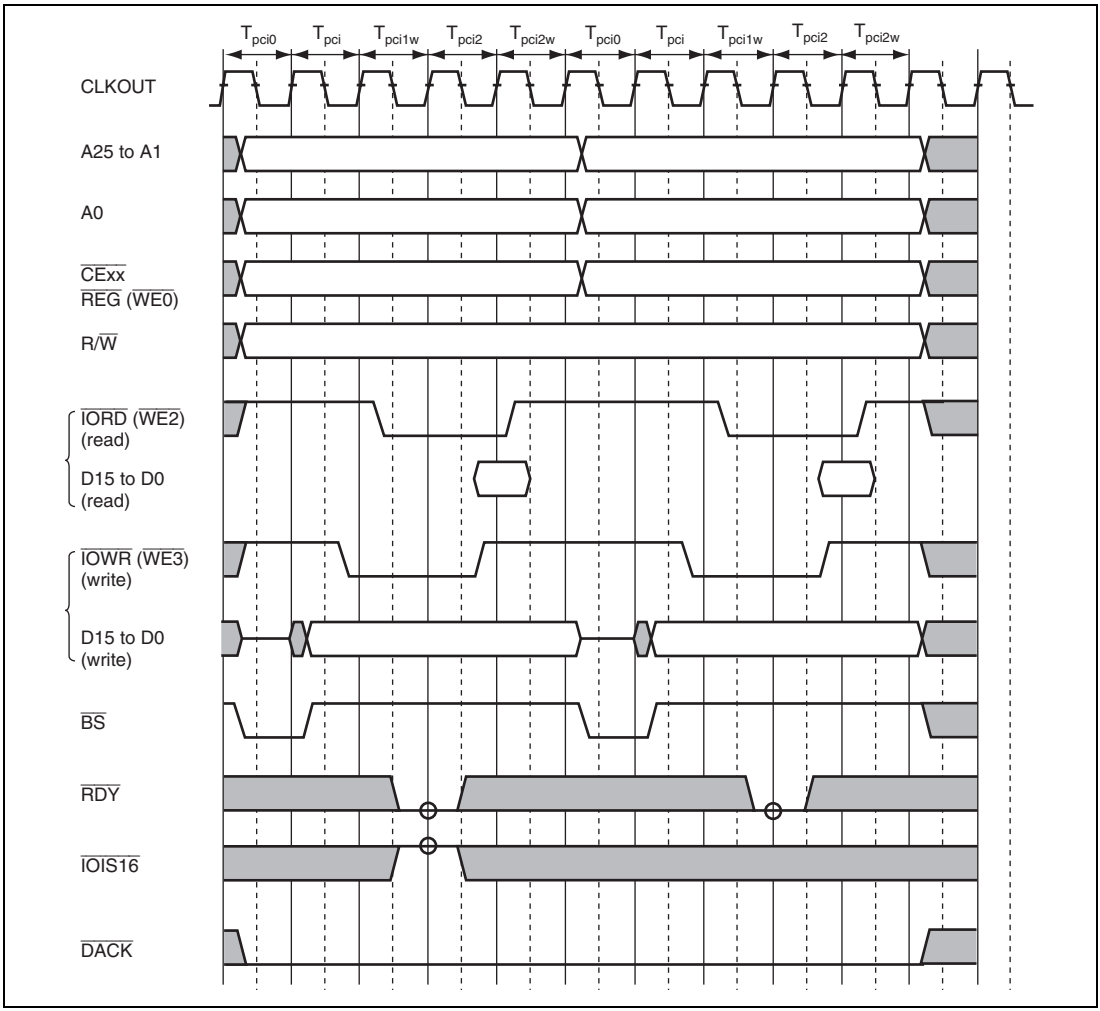


Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

11.5.6 MPX Interface

When both the MODE4 and MODE3 pins are set to 0 at a power-on reset by the $\overline{\text{PRESET}}$ pin, the MPX interface is selected for area 0. The MPX interface is selected for areas 1, 2, and 4 to 6 by the MPX bit in CS1BCR, CS2BCR, and CS4BCR to CS6BCR. The MPX interface provides an address/data multiplex-type bus protocol and facilitates connection with external memory controller chips using an address/data multiplex-type 32-bit single bus. A bus cycle consists of an address phase and a data phase. Address information is output on D25 to D0 and the access size is output on D31 to D29 in the address phase. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the address phase. The $\overline{\text{CSn}}$ signal is asserted at the rising edge in Tm1 and is negated after the end of the last data transfer in the data phase. Therefore, a negation cycle does not occur in the case of minimum pitch access. The $\overline{\text{FRAME}}$ signal is asserted at the rising edge in Tm1 and negated at the start of the last data transfer cycle in the data phase. Therefore, an external device for the MPX interface must internally store the address information and access size output in the address phase and perform data input/output for the data phase. For details, see section 11.5.1, Endian/Access 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 bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed according to the set bus width. If the access size is larger than the bus width in this case, a burst access with continuing multiple data cycle occurs after one address 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

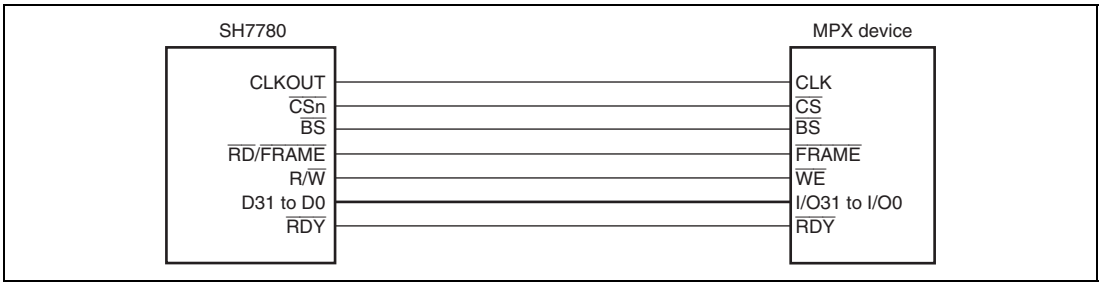


Figure 11.21 Example of 32-Bit Data Width MPX Connection

The MPX interface timing is shown below.

When the MPX interface is used for areas 1, 2, and 4 to 6, a bus size of 32 bits should be specified by CSnBCR.

In wait control, either waits by CSnWCR or waits by the RDY pin can be inserted.

In a read, one wait cycle is automatically inserted after address output, even if CSnWCR is cleared to 0.

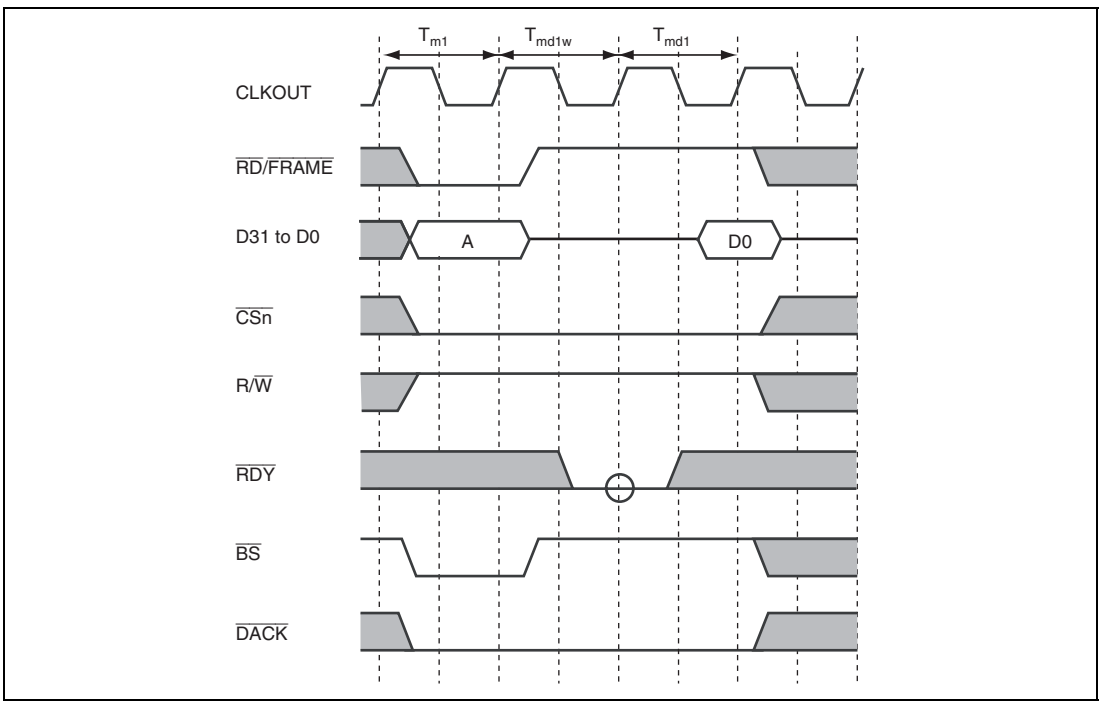


Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External Wait)

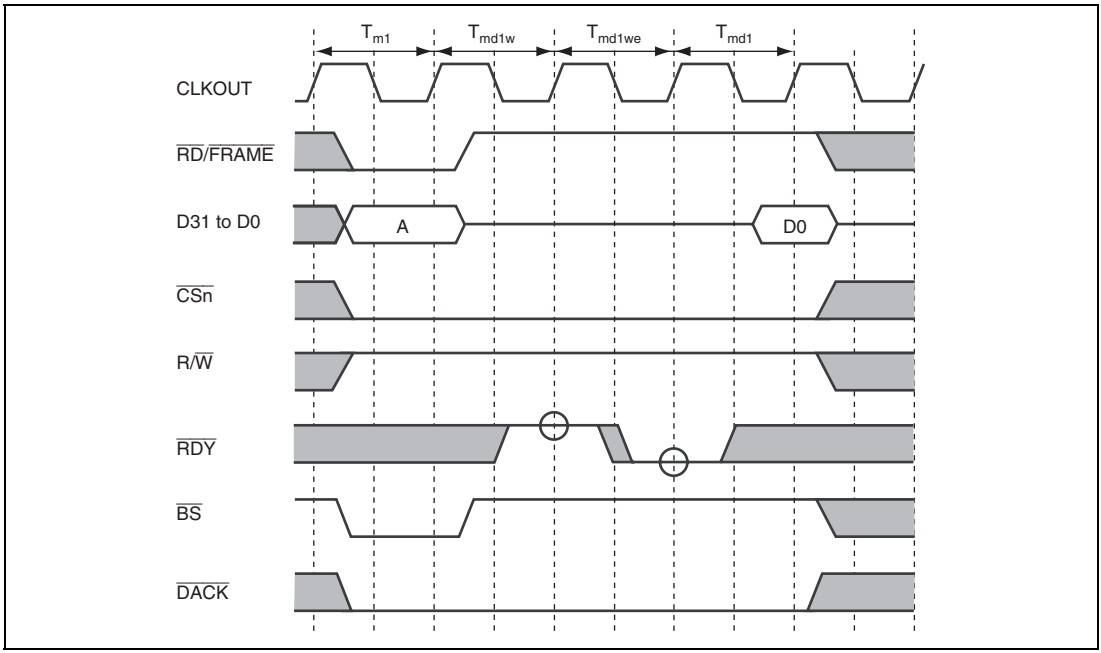


Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait Inserted)

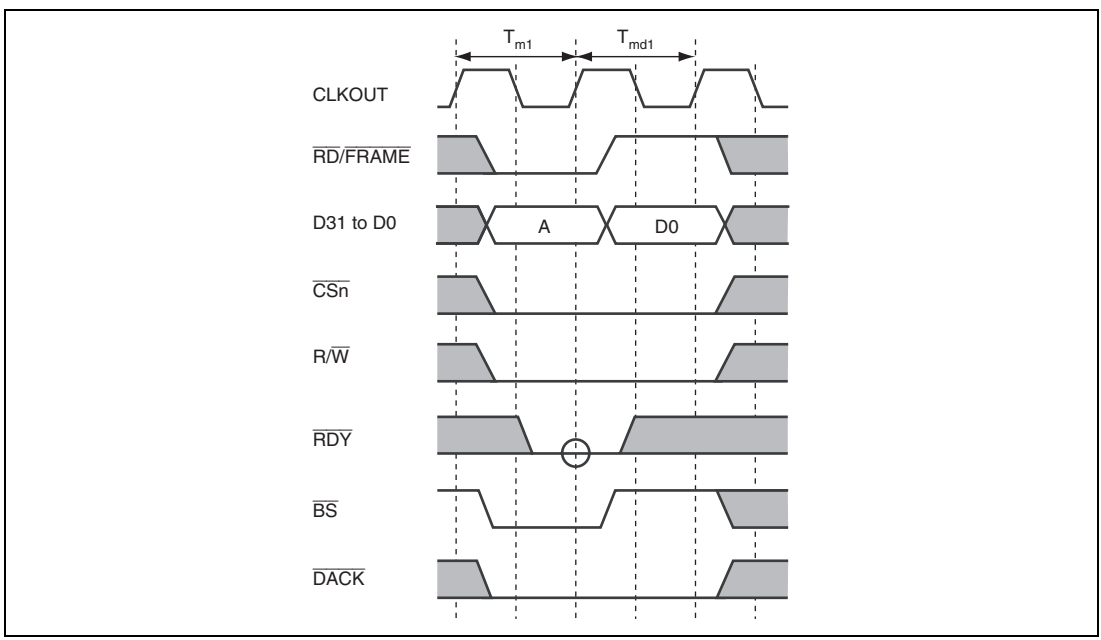


Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No External Wait)

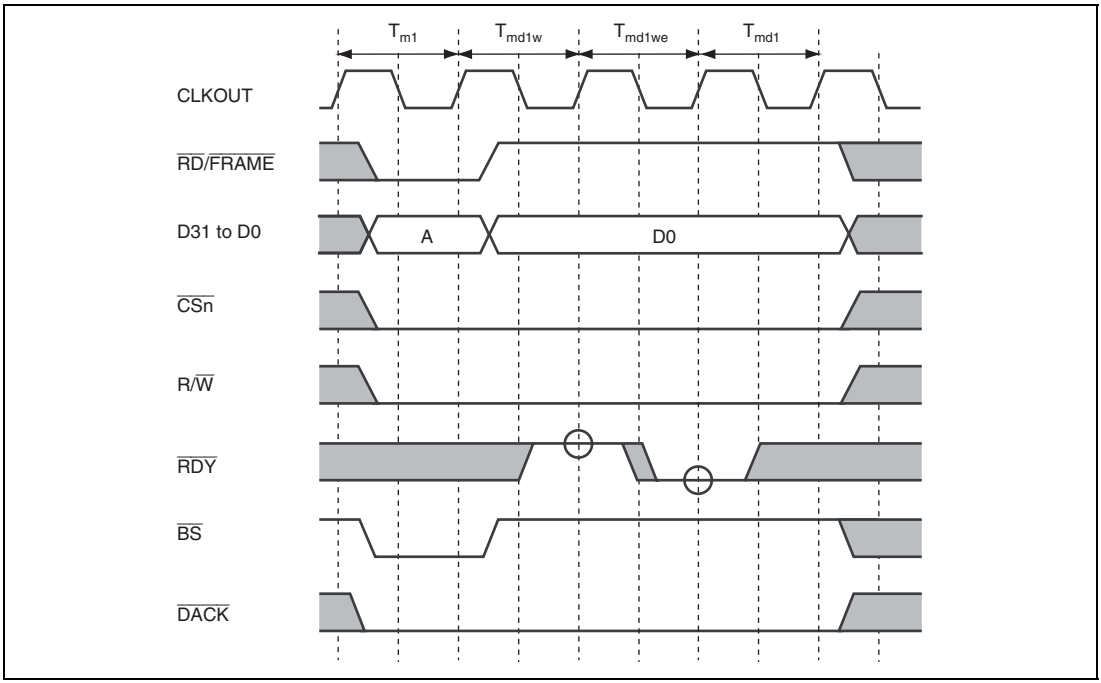


Figure 11.25 MPX Interface Timing 4 (Single Write Cycle, IW = 1, One External Wait Inserted)

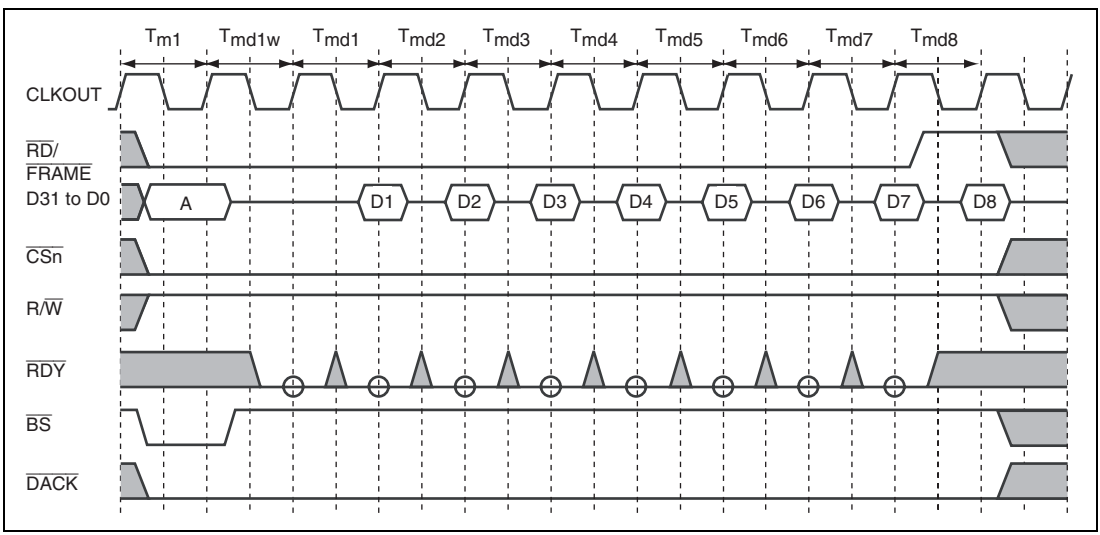


Figure 11.26 MPX Interface Timing 5 (Burst Read Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)

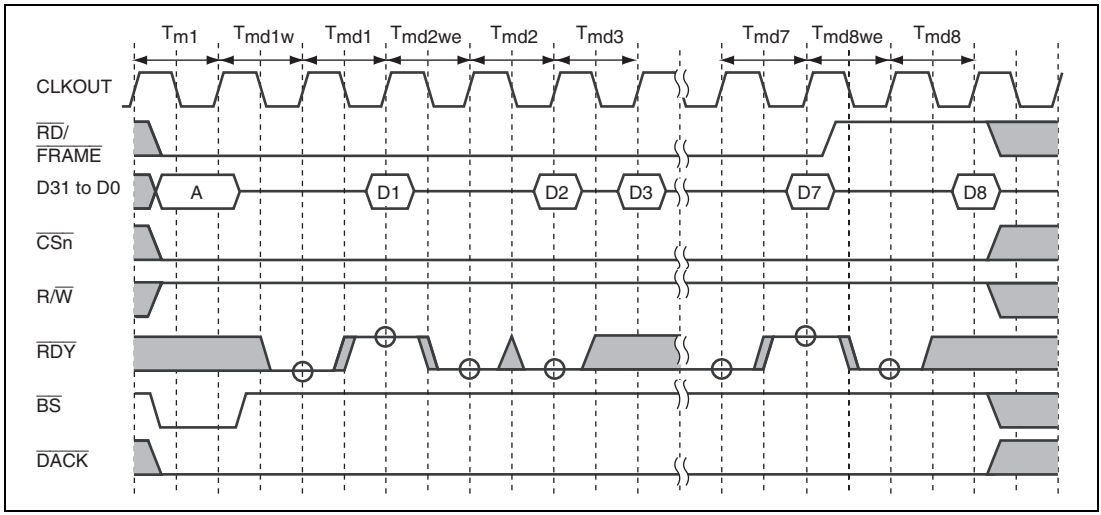


Figure 11.27 MPX Interface Timing 6
(Burst Read Cycle, IW = 0, External Wait Control, 32-Byte Data Transfer)

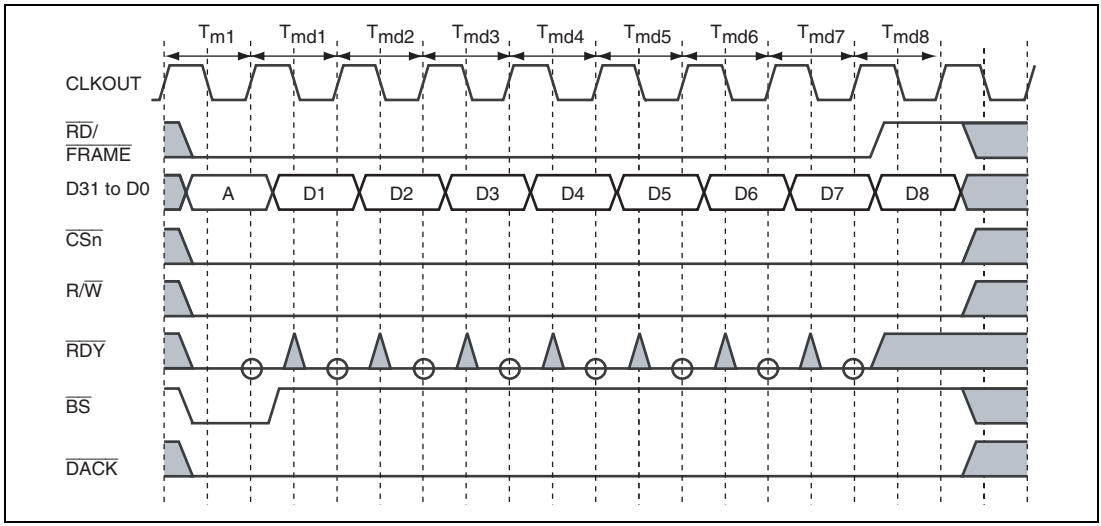


Figure 11.28 MPX Interface Timing 7
(Burst Write Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)

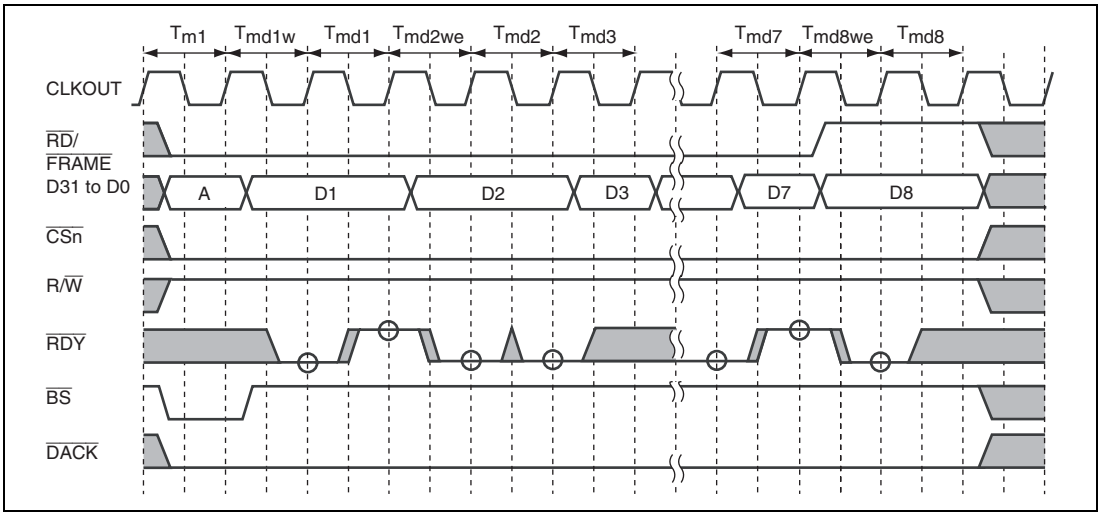


Figure 11.29 MPX Interface Timing 8
(Burst Write Cycle, IW = 1, External Wait Control, 32-Byte Data Transfer)

11.5.7 Byte Control SRAM Interface

The byte control SRAM interface is a memory interface that outputs a byte-select strobe ($\overline{WE}$) in both read and write bus cycles. This interface has 16-bit data pins and can be connected to SRAM having an upper byte select strobe and lower select strobe functions, such as UB and LB.

Areas 1 and 4 can be specified as a byte control SRAM interface. However, when these areas are set to the MPX interface, the MPX interface has priority.

The write timing for the byte control SRAM interface is identical to that of a normal SRAM interface.

In read operations, on the other hand, the $\overline{WE}$ pin timing is different. In a read access, only the $\overline{WE}$ signal for the byte being read is asserted. Assertion is synchronized with the falling edge of the CLKOUT clock in the same way as for the $\overline{WE}$ signal, while negation is synchronized with the 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 bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound method according to the set bus width. The bus is not released during this transfer.

Figure 11.30 shows an example of a byte control SRAM connection, and figures 11.31 to 11.33 show examples of byte-control SRAM read cycles.

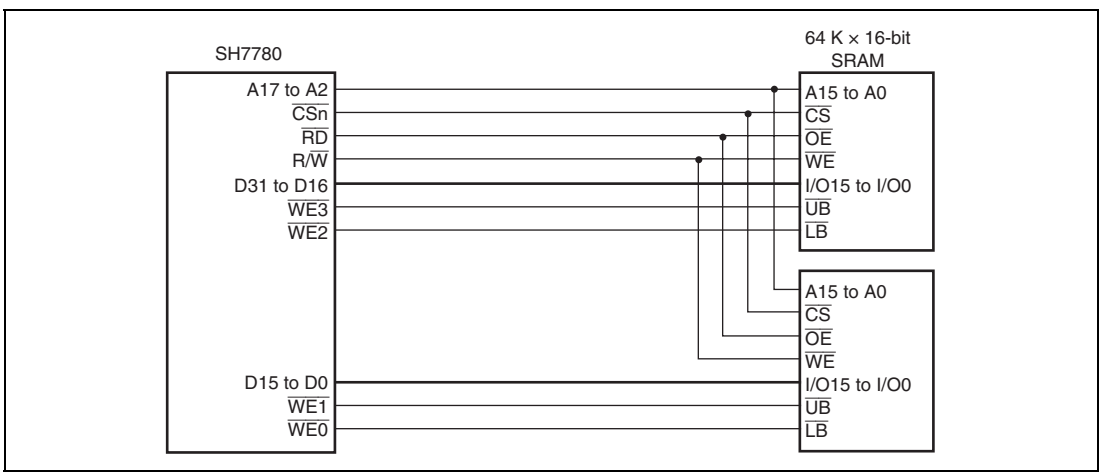


Figure 11.30 Example of 32-Bit Data-Width Byte-Control SRAM

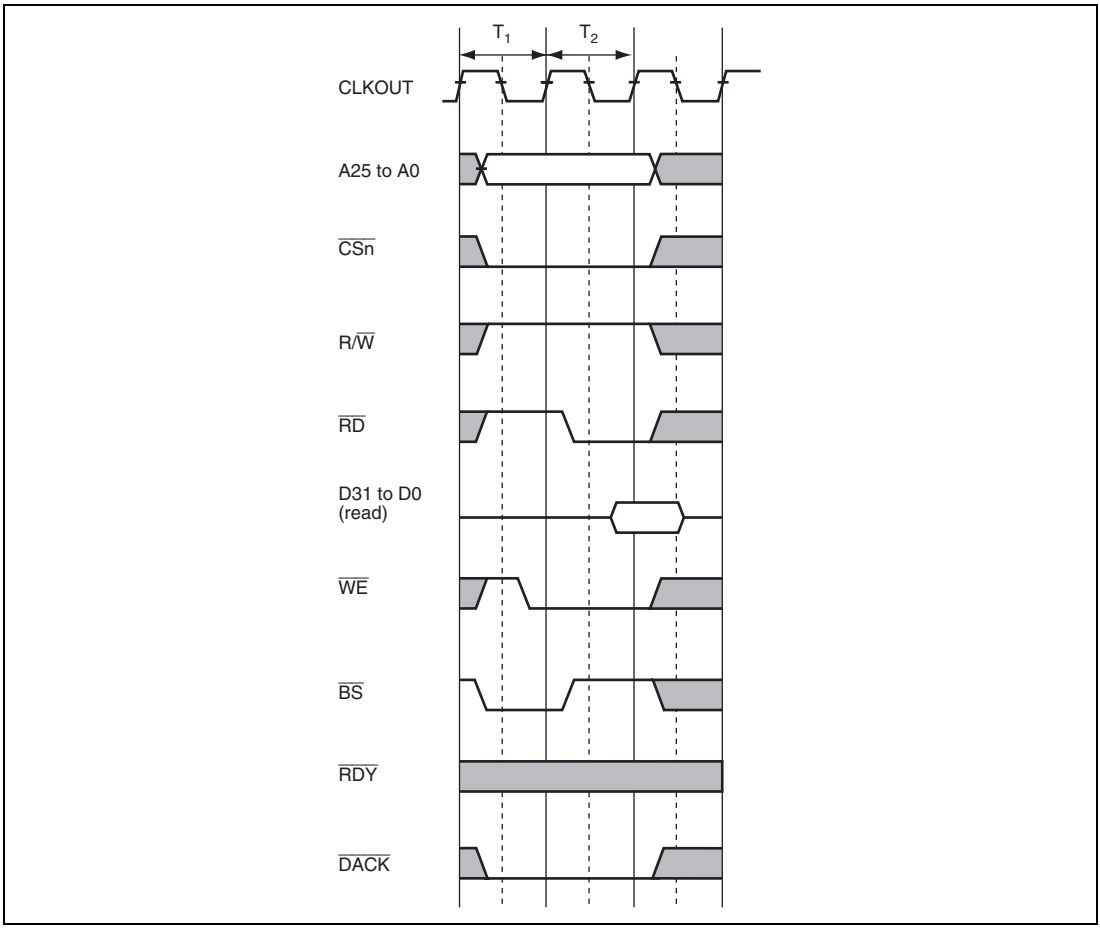


Figure 11.31 Byte-Control SRAM Basic Read Cycle (No Wait)

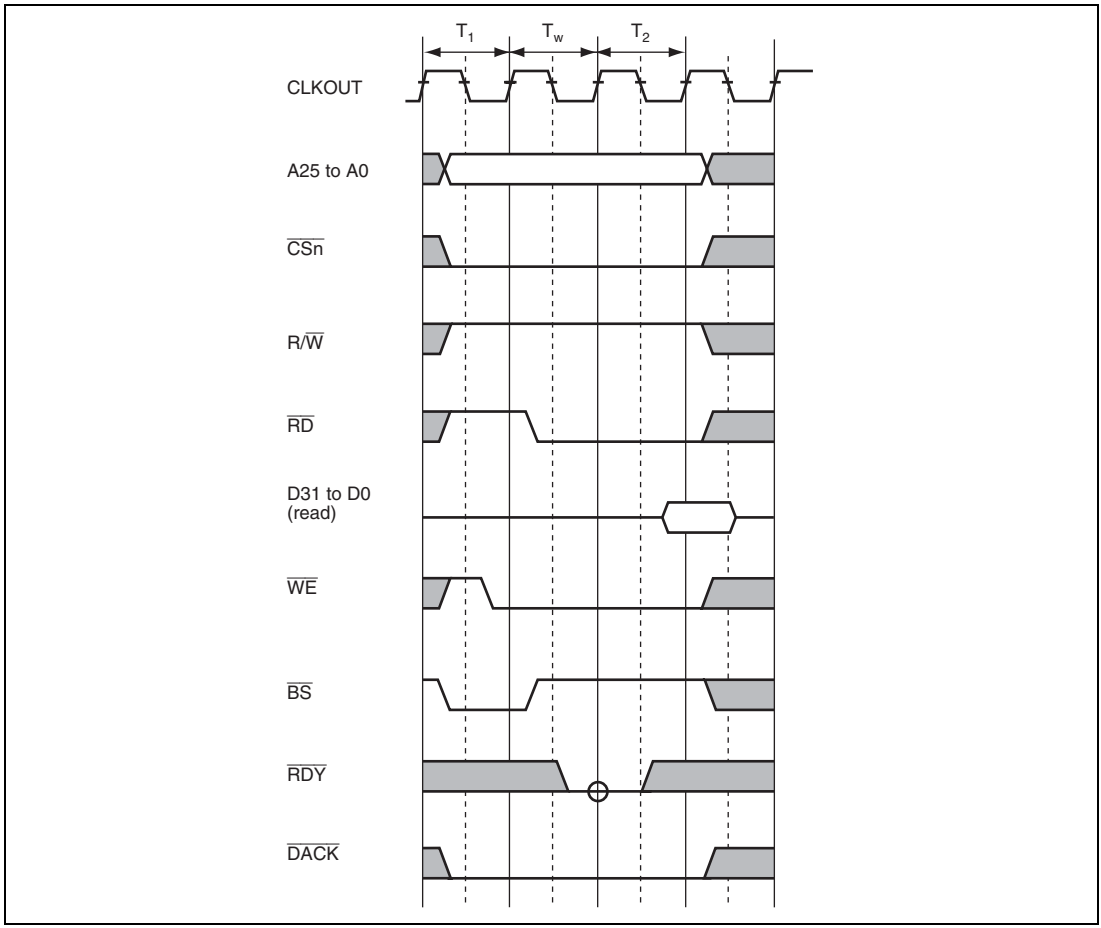


Figure 11.32 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cycle)

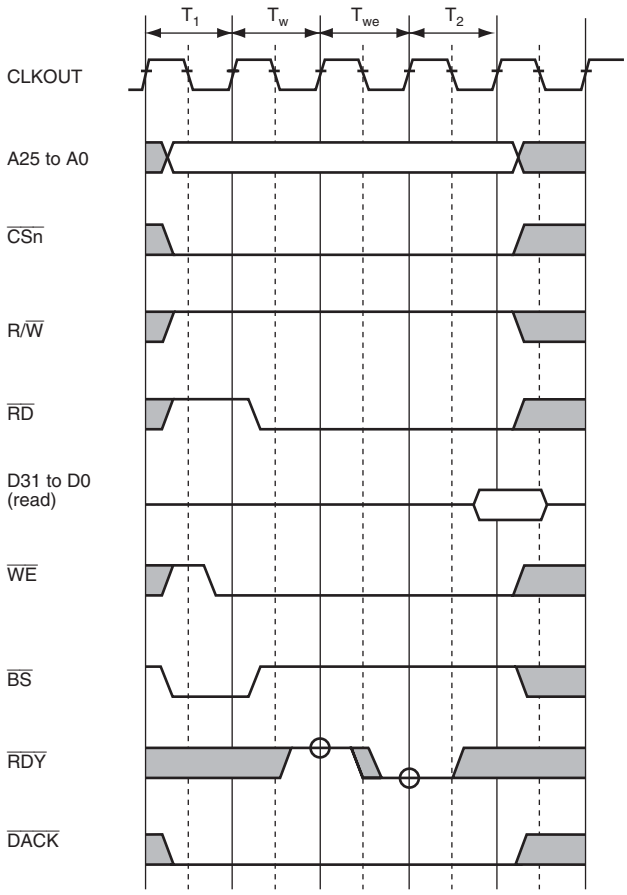


Figure 11.33 Byte-Control SRAM Basic Read Cycle (One Internal Wait + One External Wait)

11.5.8 Wait Cycles between Accesses

A problem associated with higher operating frequencies for external memory buses is that the data buffer turn-off after completion of a read from a low-speed device may be too slow, causing a collision with the data in the next access, and resulting in lower reliability or malfunctions. To prevent this problem, this module provides a data collision prevention function. It stores the preceding access area and the type of read/write and inserts a wait cycle before the access cycle if there is a possibility of a bus collision when the next access is started. The process for wait cycle insertion consists of inserting idle cycles between the access cycles as shown in section 11.4.3, CSn Bus Control Register (CSnBCR). If bits IWW, IWRWD, IWRWS, IWRRD and IWRRS in CSnBCR (n = 0 to 2 and 4 to 6) are used to set the number of idle cycles between accesses, the number of inserted idle cycles is only the specified number of idle cycles minus the number of idle cycles specified by the bits.

When bus arbitration is performed, the bus is released after wait cycles are inserted between the cycles.

When a DMA transfer is performed, wait cycles are inserted as set in CSnBCR idle cycle bits.

When access the MPX interface area continuously after read access, 1 wait cycle is inserted even 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.

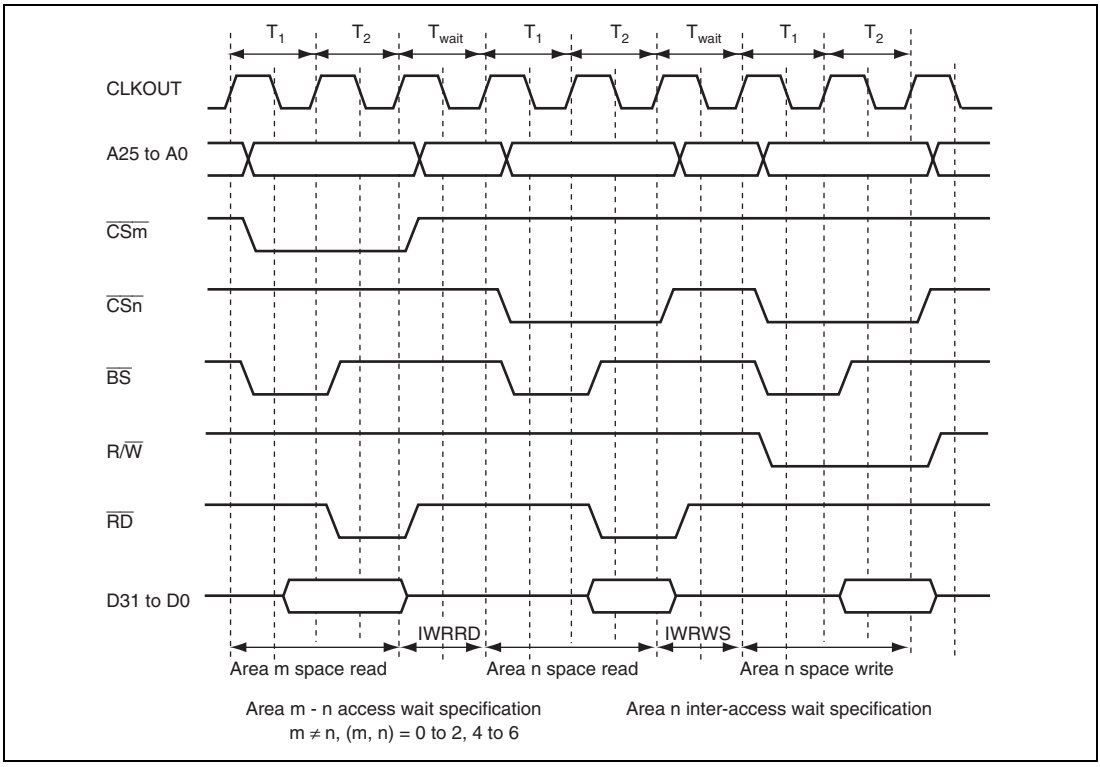


Figure 11.34 Wait Cycles between Access Cycles

11.5.9 Bus Arbitration

The LBSC is provided with a bus arbitration function that grants the bus to an external device when it makes a bus request.

In normal operation, the bus is held by the LBSC (bus master), and is released to another device in response to a bus request. It is possible to connect an external device that issues bus requests. In the following description, an external device that issues bus requests is also referred to as a slave.

The SH7780 has three internal bus masters: the CPU, DMAC, and PCIC. In addition to these are bus requests from external devices (highest priority). If requests occur simultaneously, the LRU method is used to decide the request priority. The initial priority order is : CPU > DMAC > PCIC.

To prevent incorrect operation of connected devices when the bus is transferred between master and slave, all bus control signals are negated before the bus is released. When mastership of the bus is received, also, bus control signals begin driving the bus from the negated state. Since signals are driven to the same value by the master and slave exchanging the bus, output buffer collisions can be avoided. By turning off the output buffer on the side releasing the bus, and turning on the output buffer on the side receiving the bus, simultaneously with respect to the bus control signals, it is possible to eliminate the signal high-impedance period. It is not necessary to provide the pull-up resistors usually inserted in these control signal lines to prevent incorrect 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 soon as the currently executing bus cycle ends, and outputs the bus use permission signal ($\overline{\text{BACK}}$). However, bus release is not performed during multiple bus cycles generated because the data bus width is smaller than the access size (for example, when performing longword access to 8-bit bus width memory) or during a 32-byte transfer such as a cache fill or write-back. In addition, bus release is not performed between read and write cycles during execution of a TAS instruction, or between read and write cycles in DMA dual address mode of the bus locked. When $\overline{\text{BREQ}}$ is negated, $\overline{\text{BACK}}$ is negated and use of the bus is resumed.

As the CPU is connected to cache memory by a dedicated internal bus, reading from cache memory can still be carried out when the bus is being used by another bus master inside or outside the SH7780. When writing from the CPU, an external write cycle is generated when write-through has been set for the cache in the SH7780, or when an access is made to a cache-off area. There is consequently a delay until the bus is returned.

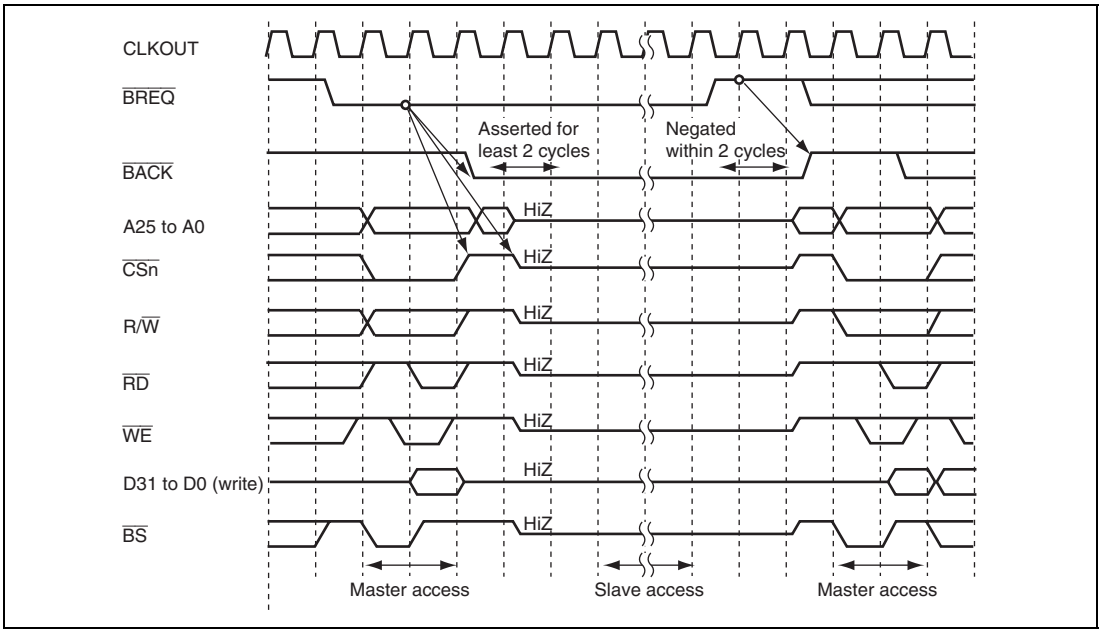


Figure 11.35 Arbitration Sequence

11.5.10 Bus Release and Acquire Sequence

The LBSC holds the bus itself unless it receives a bus request.

On receiving an assertion (low level) of the bus request signal ($\overline{\text{BREQ}}$) from off-chip, the LBSC releases the bus and asserts (drives low) the bus use permission signal ($\overline{\text{BACK}}$) as soon as the currently executing bus cycle ends. On receiving the $\overline{\text{BREQ}}$ negation (high level) indicating that the slave has released the bus, the LBSC negates (drives high) the $\overline{\text{BACK}}$ signal and resumes use of the bus.

The actual bus release sequence is as follows.

First, the bus use permission signal is asserted in synchronization with the rising edge of the clock. The address bus and data bus go to the high-impedance state in synchronization from next rising edge of the clock after this $\overline{\text{BACK}}$ assertion. At the same time, the bus control signals ($\overline{\text{BS}}$, $\overline{\text{CS}}$n, $\overline{\text{WE}}$, $\overline{\text{RD}}$, R/$\overline{\text{W}}$, $\overline{\text{CE2A}}$, and $\overline{\text{CE2B}}$) go to the high-impedance state. These bus control signals are negated no later than one cycle before going to high-impedance. Bus request signal sampling is 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 and bus control signal driving is started. Driving of the address bus and data bus starts at the next rising edge of an in-phase clock. The bus control signals are asserted and the bus cycle is actually started, at the earliest, at the clock rising edge at which the address and data signals are driven.

In order to reacquire the bus and start execution of bus access, the $\overline{\text{BREQ}}$ signal must be negated for at least two cycles.

Using the LCKN bit in CHCR of the DMAC, it is possible to restrain the bus release in the cycle between read and write access.

If a DMA transfer is executed for the space that the source and destination address are in the LBSC space and the LCKN bit in CHCR is cleared to 0, the bus is not released in the cycle between read and write accesses even if the bus release signal ($\overline{\text{BREQ}}$) is asserted.

If a DMA transfer is executed for the space that the source address is in the LBSC space and the destination address is out of the LBSC space and the LCKN bit in CHCR is cleared to 0, the bus is not released after the DMA write access is ended even if the bus release signal ($\overline{\text{BREQ}}$) is asserted. And then execute read or write access to any address of the LBSC space from the CPU, the bus is released after the access. This procedure does not need when the LCKN bit is set to 0.

If a DMA transfer is executed for the space that the source address is out of the LBSC space and the destination address is in the LBSC space and the LCKN bit in CHCR is cleared to 0, the bus is released in the cycle between read access and write access.

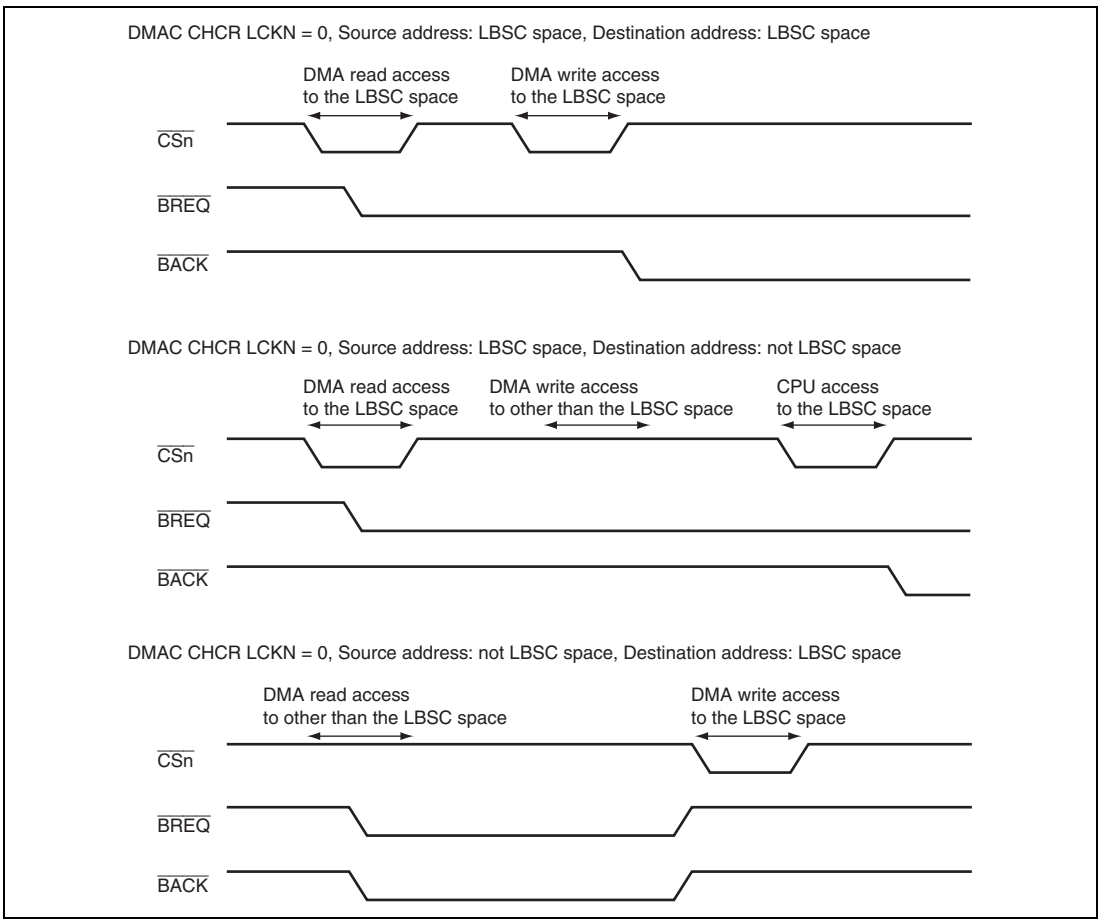


Figure 11.36 Example of the Bus Release Restraint by the DMAC CHCR LCKN bit

11.5.11 Cooperation between Master and Slave

To enable system resources to be controlled in a harmonious fashion by master and slave, their respective roles must be clearly defined.

When designing an application system that includes the SH7780, all control, including initialization, and low power consumption control, are supposed to be carried out by the SH7780.

In a power-on reset, the SH7780 will not accept bus requests from the slave until the $\overline{\text{BREQ}}$ enable bit (BCR.BREQEN) is set to 1.

To ensure that the slave processor does not access memory requiring initialization before use, write 1 to the $\overline{\text{BREQ}}$ enable bit only after the SH7780 has performed the initialization.

Section 12 DDR-SDRAM Interface (DDRIF)

The DDR-SDRAM interface (DDRIF) is an interface for the control of DDR-SDRAM. The DDRIF supports DDR320- and DDR266-SDRAM.

12.1 Features

- The data bus width of the DDRIF is 32 bits
- Supports DDR-SDRAM self-refreshing
- Supports DDR320 (160 MHz) or DDR266 (133 MHz)
- 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 Mbits)
  — Parallel connection of two 256-Mbit DDR-SDRAMs (× 16) (Total Size 512 Mbits)
  — 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 setting at the time of a power-on reset

Note: DDR320 indicates the DDR-SDRAM bus interface which operates at a frequency of 160 MHz in this manual.

Figure 12.1 shows a block diagram of the DDRIF.

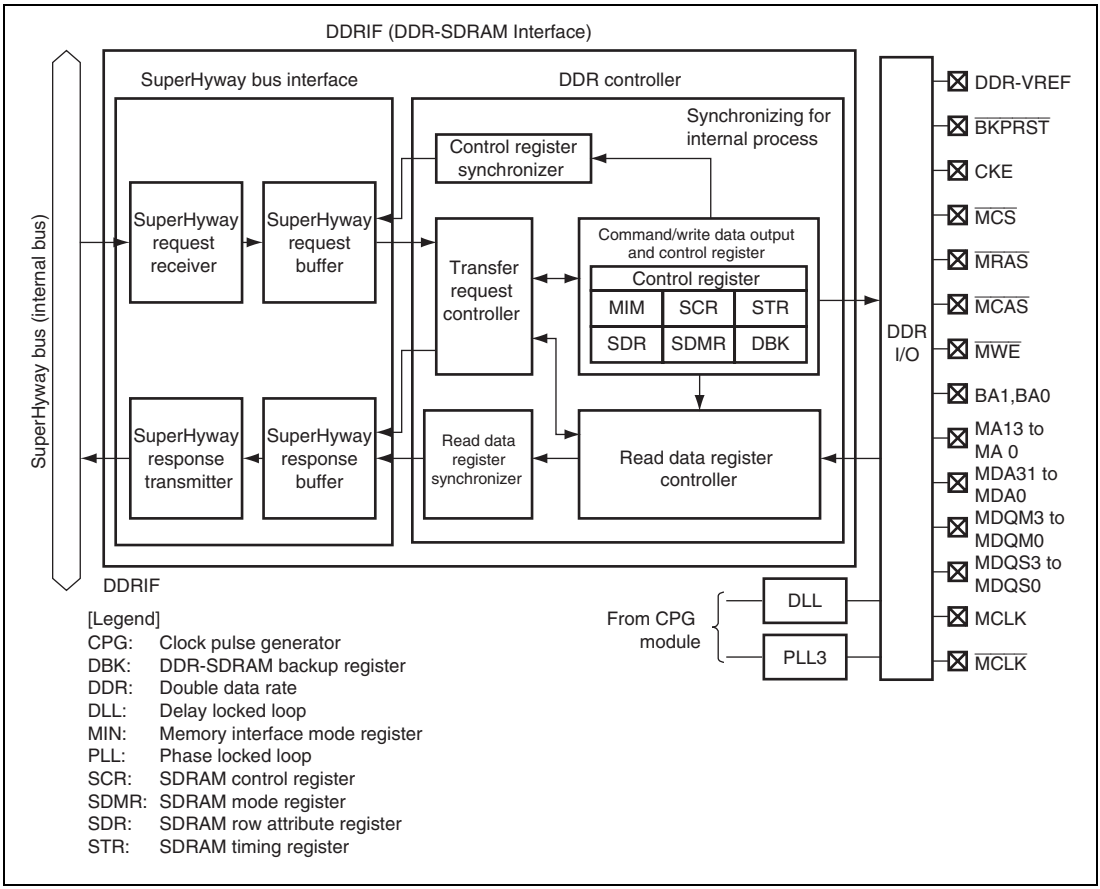


Figure 12.1 DDRIF Block Diagram

12.2 Input/Output Pins

Table 12.1 shows the DDRIF pin configuration. For details on connection with the DDR-SDRAM, see the DDR-SDRAM pin information.

Table 12.1 Pin Configuration

Pin Name	Function	I/O	Description
MCLK	DDR-SDRAM clock	Output	Clock output for DDR-SDRAM
$\overline{\text{MCLK}}$	DDR-SDRAM clock	Output	Clock output for DDR-SDRAM Inverse of the MCLK
CKE	Clock enable	Output	When this pin is set high, the clock signal is active. When this pin is set low, the clock signal is inactive.
$\overline{\text{MCS}}$	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 signal
$\overline{\text{BKPRST}}$	Power back-up reset	Input	When this pin goes low, the CKE pin also goes low
DDR-VREF	Reference voltage input	Input	Input reference voltage

12.3 Address Space, Bus Width, and Data Alignment

12.3.1 Address Space of the DDRIF

This LSI supports both 29-bit and 32-bit physical address spaces (29-bit address mode and 32-bit address extended mode), and the address space is selectable from among five kinds of map by setting Memory Address Map Select Register (MMSELR) of the LBSC. Figure 12.2 shows the physical address space of this LSI.

The DDRIF supports both 29-bit and 32-bit physical address spaces and can control an externally 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 area 3, but also areas 2, 4, and 5, which are also within the 29-bit address range. The DDRIF can control a total of 4 areas with a maximum capacity of 256 Mbytes as the external DDR-SDRAM memory space.

In the case of the 32-bit address extended mode, the DDRIF controls not only area 3 (and, with some settings, area 2, 4 and 5) within the 29-bit address range but also DDR-SDRAM areas in the physical address range from H'4000 0000 to H'7FFF FFFF. However, this 1-Gbyte range includes areas where the areas actually allocated to the DDRIF by the MMSELR are shadowed. The actual area controlled by the DDRIF as the external DDR-SDRAM memory space is still a total of 256 Mbytes.

For further information on the 32-bit address extended mode, see section 7.7, 32-Bit Address Extended Mode.

	MMSELR. AREASEL[2:0]*	B'000	B'001	B'010	B'011	B'100
H'0000 0000	Area 0 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'0400 0000	Area 1 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'0800 0000	Area 2 (LBSC/DDRIF)	LBSC	LBSC	DDRIF-0	DDRIF-0	DDRIF-0
H'0C00 0000	Area 3 (DDRIF)	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'1000 0000	Area 4 (LBSC/DDRIF/PCIC)	LBSC	PCIC	LBSC	PCIC	DDRIF-2
H'1400 0000	Area 5 (LBSC/DDRIF)	LBSC	LBSC	LBSC	LBSC	DDRIF-3
H'1800 0000	Area 6 (LBSC)	LBSC	LBSC	LBSC	LBSC	LBSC
H'1C00 0000	Area 7 (reserved area)					
H'2000 0000	(Undefined)					
H'4000 0000		DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
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	(shaded box) : Shadow	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
H'6400 0000		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 H'FFFF FFFF	(Internal resources)					

H'0000 0000 – H'1C00 0000: 29-bit physical address space (Normal mode)

H'0000 0000 – H'FFFF FFFF: 32-bit physical address space (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.

12.3.3 Data Alignment

The DDRIF supports both big endian mode, where the address of the highest order byte is 0, and little endian mode, where the address of the lowest order byte is 0. These modes can be switched by changing the level on an external pin (MODE5) and then generating a power-on reset. Note that wraparound in memory data access is on 32-byte boundaries.

Table 12.2 Access and Data Alignment in Little Endian Mode

	MD31 to MD24	MD23 to MD16	MD15 to MD8	MD7 to MD0
Byte access at address 0				Bit 7 to 0
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 0
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 0
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 0
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 0
Longword access at address 4	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to 0
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 0
Quadword access at address 0 (second round: from address 4)	Bit 63 to 56	Bit 55 to 48	Bit 47 to 40	Bit 39 to 32
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 0
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 0
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 0
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 0

	MD31 to MD24	MD23 to MD16	MD15 to MD8	MD7 to MD0
32-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 0
32-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 0
32-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 0
32-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 0
32-byte access at address 0 (fifth round: from address 20 (H'14))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to 0
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 to 0
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 to 0
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 to 0

Table 12.3 Access and Data Alignment in Big Endian Mode

	MD31 to MD24	MD23 to MD16	MD15 to MD8	MD7 to MD0
Byte access at address 0	Bit 7 to 0			
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 0			
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 0		
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 0		
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 0
Longword access at address 4	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to 0
Quadword access at address 0 (first round: from address 0)	Bit 63 to 56	Bit 55 to 48	Bit 47 to 40	Bit 39 to 32
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 0
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 0
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 0
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 0
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 0

	MD31 to MD24	MD23 to MD16	MD15 to MD8	MD7 to MD0
32-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 0
32-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 0
32-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 0
32-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 0
32-byte access at address 0 (fifth round: from address 16 (H'10))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to 0
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 to 0
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 to 0
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 to 0

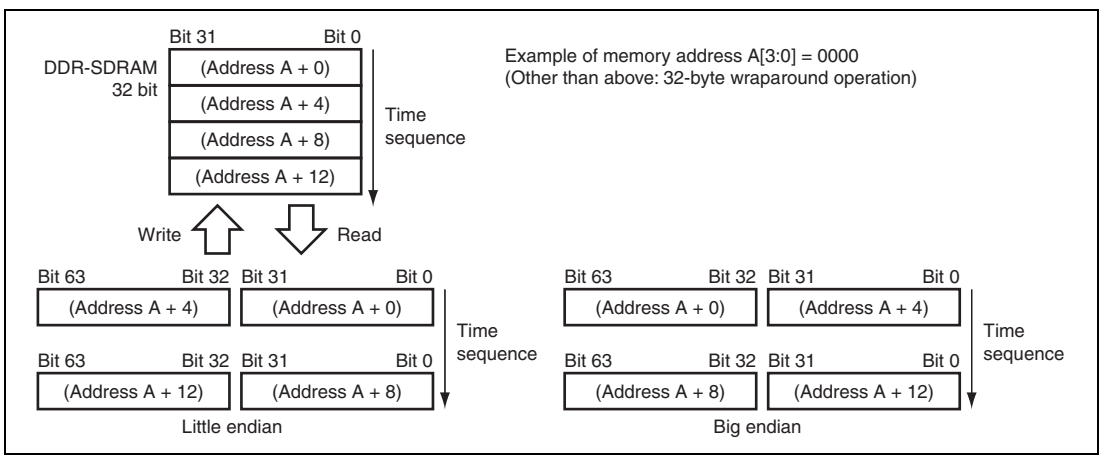


Figure 12.3 Data Alignment in DDR-SDRAM and DDRIF

12.4 Register Descriptions

Table 12.4 shows the DDRIF register configuration. Table 12.5 shows the register states in each processing mode.

These registers should only be set while access to the DDR-SDRAM from a module is not in progress. Furthermore, access to registers other than the memory interface mode register (MIM) should only proceed when the MIM's DCE bit (DDR-SDRAM control enable) is cleared to 0 or the MIM's SELFS bit (self-refresh status) is set to 1.

Although the registers are 64 bits wide, they should be accessed in longword (32-bit) units. The value of a longword written to the register will be reflected correctly. A longword read from the register will contain the value in the corresponding half of the register at the time of reading. Whether the current endian is big or little, specify the address listed below to access bits 63 to 32. 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	Access Size
Memory interface mode register	MIM	R/W	H'FE80 0008	H'1E80 0008	32
DDR-SDRAM control register	SCR	R/W	H'FE80 0010	H'1E80 0010	32
DDR-SDRAM timing register	STR	R/W	H'FE80 0018	H'1E80 0018	32
DDR-SDRAM row attribute register	SDR	R/W	H'FE80 0030	H'1E80 0030	32
DDR-SDRAM mode register	SDMR	W	H'FECx xxxx*	H'1ECx xxxx*	32
DDR-SDRAM back-up register	DBK	R	H'FE80 0400	H'1E80 0400	32

Note: * For details, see section 12.4.5, SDRAM Mode Register (SDMR).

Table 12.5 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep
Memory interface mode register	MIM	H'0000 0000 0C34 xx00*[1]	H'0000 0000 0C34 xx00*[1]	Retained
DDR-SDRAM control register	SCR	H'0000 0000 0000 0000	H'0000 0000 0000 0000	Retained
DDR-SDRAM timing register	STR	H'0000 0000 0000 0000	H'0000 0000 0000 0000	Retained
DDR-SDRAM row attribute register	SDR	H'0000 0000 0000 0100	H'0000 0000 0000 0100	Retained
DDR-SDRAM mode register	SDMR	—	—	—
DDR-SDRAM back-up register	DBK	H'0000 0000 0000 000x*[2]	H'0000 0000 0000 000x*[2]	Retained

Notes: 1. The initial value of bit 8 (ENDIAN bit) depends on the setting of external pins (MODE5).
2. The initial value of bit 0 (SDBUP bit) depends on the setting of external pin ($\overline{\text{BKPRST}}$).

12.4.1 Memory Interface Mode Register (MIM)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	BOMODE		—	PCKE	—	—	—	—	—	—	—	—	—	SELFS	RMODE	—
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	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DRI												
Initial value:	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0
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	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK				—	—	DRE	ENDIAN	BW	—	—	—	DLLEN	—	—	DCE
Initial value:	—	—	—	—	0	0	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	R	R/W

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 value 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 modes. For details on operation in each of the modes, see section 12.5.4, SDRAM Access Mode. 00: Bank open mode 01: Bank closed mode Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
45	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
44	PCKE	0	R/W	Power Down This bit controls a low power consumption mode in which the CKE pin is set low to place the DDR-SDRAM in “power-down mode” whenever the DDR-SDRAM is not being accessed (whether it is in the idle state or bank active state). When the PCKE bit is set to 1, the DDR-SDRAM enters this power down mode. For details, see section 12.5.5 (2), Power-Down Mode (when CKE Goes Low). Note that the SMS bits in SCR should be set so that the CKE pin is enabled when the SDRAM is in its initial state.
43 to 35	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
34	SELFS	0	R	Self-Refresh Decision Indicates whether the DDR-SDRAM is or is not in the 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 auto-refresh mode or to self-refresh mode. This bit is only 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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
28 to 16	DRI	H'0C34	R/W	DRAM Refresh Interval When refreshing is valid (the DRE bit in MIM is set to 1), these bits specify the maximum refresh interval (auto-refresh). The unit for counting is the cycle of the MCLK. In 160-MHz operation, the unit corresponds to 6.3 ns. The smallest possible setting is H'0020 units. If a lower setting is made, H'020 is added to the value for counting. The DDRIF has a 13-bit internal counter. When the DCE or DRE bit is cleared to 0, or the RMODE bit is set to 1, this counter is cleared to 0. Otherwise, the counter is incremented by the external MCLK. The value in the counter is compared with the DRI bits. If the values match, an auto-refresh request is generated in the controller and auto-refreshing is performed. Note that the counter is cleared to 0 on the match, after which incrementation begins again. A single instance of the internally generated request for auto-refresh is recorded; if the DCE and DRE bits are set to 1 and the RMODE bit is cleared to 0, the auto-refresh request is not cleared until auto-refreshing has been performed. When setting these bits, start by making the setting and writing a 0 to the DRE bit at the same time. Make the setting again, but this time write a 1 to the DRE bit at the same time. This is required for timing consistency.
15 to 12	LOCK	Undefined	R	DLL Lock Status These bits indicate the state of locking by the DLL that generates the read timing for the DDR-SDRAM When these bits are all set to 1 and the DLLEN bit is 1, access to memory is possible.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	DRE	0	R/W	DRAM Refresh Enable This bit enables or disables the use of refresh modes. 0: Disable 1: Enable

Bit	Bit Name	Initial Value	R/W	Description
8	ENDIAN	Undefined*	R	Endian Identifier Indicates whether big endian or little endian mode is selected for the external data bus. 0: Little endian mode 1: Big endian mode
7	BW	0	R/W	Bus Width Specifies the DDR-SDRAM bus width. This bit should always be cleared to 0.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	DLLEN	0	R/W	DLL Enable Sets whether the DLL for generating the read timing for the DDR-SDRAM is valid or invalid. When this bit is set to 1, the DLL is enabled and read access to memory is possible.
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DCE	0	R/W	DDR Controller Enable Enables or disables SDRAM control by the DDRIF. 0: Disables SDRAM control 1: Enables SDRAM control

Note: * Depends on the setting of external pin (MODE5).

12.4.2 SDRAM Control Register (SCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	SMS	000	R/W	SDRAM Mode Select These bits initialize the DDR-SDRAM when power is supplied and after release of the reset signal. Software can be used to set these bits as listed below so that the corresponding command is issued.For details on the initialization procedure, see section 12.5.2, DDR-SDRAM Initialization Sequence. After the DDR-SDRAM has been initialized, normal operation (000) is specified. 000: Normal operation 001: A NOP command is issued (only valid if the DCE bit in MIM is set to 1). 010: A PREALL command is issued (only valid if the DCE bit in MIM is set to 1). 011: The CKE pin is enabled. At that time, the DESELECT command is issued (only valid if the DCE bit in MIM is set to 1). 100: The REFA (auto-refresh) command is issued (only valid if the DCE bit in MIM is set to 1). Settings other than the above are prohibited. If such settings are made, correct operation is not guaranteed. Note that the PCKE bit in MIM is used to set the CKE pin low for reduced power consumption of the DDR-SDRAM.

12.4.3 SDRAM Timing Register (STR)

STR specifies various timing parameters for the DDR-SDRAM.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	WR		RW	
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
	SRFC			SWR	SRRD	SRAS			SRC			SCL			SRCD	SRP
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/W

Bit	Bit Name	Initial Value	R/W	Description
63 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19, 18	WR	00	R/W	Minimum Number of Cycles from Write command to Read Commands These bits specify the minimum number of cycles required by the SDRAM from the issuing of a WRITE command to the issuing of a subsequent READ command. 00: 3 cycles 01: 4 cycles 10: 5 cycles 11: 6 cycles

Bit	Bit Name	Initial Value	R/W	Description
17, 16	RW	00	R/W	Minimum Number of Cycles from Read Command to Write Command These bits specify the minimum number of cycles required by the SRAM from the issuing of a READ command to the issuing of a subsequent WRITE command. 00: 3 cycles 01: 4 cycles 10: 5 cycles 11: 6 cycles
15 to 13	SRFC	000	R/W	Number of Cycles within a Single Individual Bank These bits specify the number of cycles between the following access operations in a given bank (the corresponding time is tRFC). (1) From auto-refresh to issuing the ACT command (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 cycles from the last postamble to the issuing of a PRE/PREALL command (the corresponding time is tWR). 0: 2 cycles 1: 3 cycles
11	SRRD	0	R/W	Inter-bank Number of Cycles between ACT Commands Specifies the minimum number of cycles between the issuing of ACT commands (the corresponding time is tRRD) for any two banks. 0: 2 cycles 1: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	SRAS	000	R/W	Minimum Number of Cycles between ACT and PRE Commands These bits specify the minimum number of cycles until a PRE command is issued after an ACT command has been issued (the corresponding time is tRAS) for the same bank. 000: 6 cycles 001: 7 cycles 010: 8 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 between the following access operations in a given bank (the corresponding time is tRC). (1) From issuing the ACT command to auto-refresh (2) From issuing one ACT command to issuing the next 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 read operation. 000: 2.5 cycles Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
1	SRCD	0	R/W	Numbers of Cycles between RAS and CAS Commands Specifies the number of cycles from an RAS (ACT) command to a subsequent CAS (READ/READA, WRITE/WRITEA) command (the corresponding time is tRCD). 0: 3 cycles 1: 4 cycles
0	SRP	0	R/W	Number of Cycles between PRE and ACT Commands Specifies the number of cycles from a PRE command to a subsequent ACT command (the corresponding 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	49	47
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	SPLIT	0001	R/W	DDR-SDRAM Memory Configuration These bits specify the row/column configuration of the DDR-SDRAM. 0001: 12 × 9 (= product with 8 M × 16 bits) 0011: 13 × 9 (= product with 16 M × 16 bits) 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 numbers 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 value 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. Since the SDMR is physically within the SDRAM rather than the DDRIF, reading the registers is invalid. Only the address bits have any meaning for the DDR-SDRAM and any data included in the write operation is ignored.

Writing to the SDMR proceeds when the signal output on pins connected to the DDR-SDRAM is as shown in the table below.

Address bits 12 to 3 correspond to external pins MA9 to MA0, address bits 14 and 13 to external pins BA1 and BA0, and address bits 18 to 15 to external pins MA13 to MA10. These bits contain the values for the mode registers.

CKE						Address Bit Correspondence		
n-1	n	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	BA1 and BA0	MA13 to MA10	MA9 to MA0
H	H	L	L	L	L	Bits 14 and 13	Bits 18 to 15	Bits 12 to 3

Figure 12.4 shows the relationship between write values in SDMR and output signals to the memory pins.

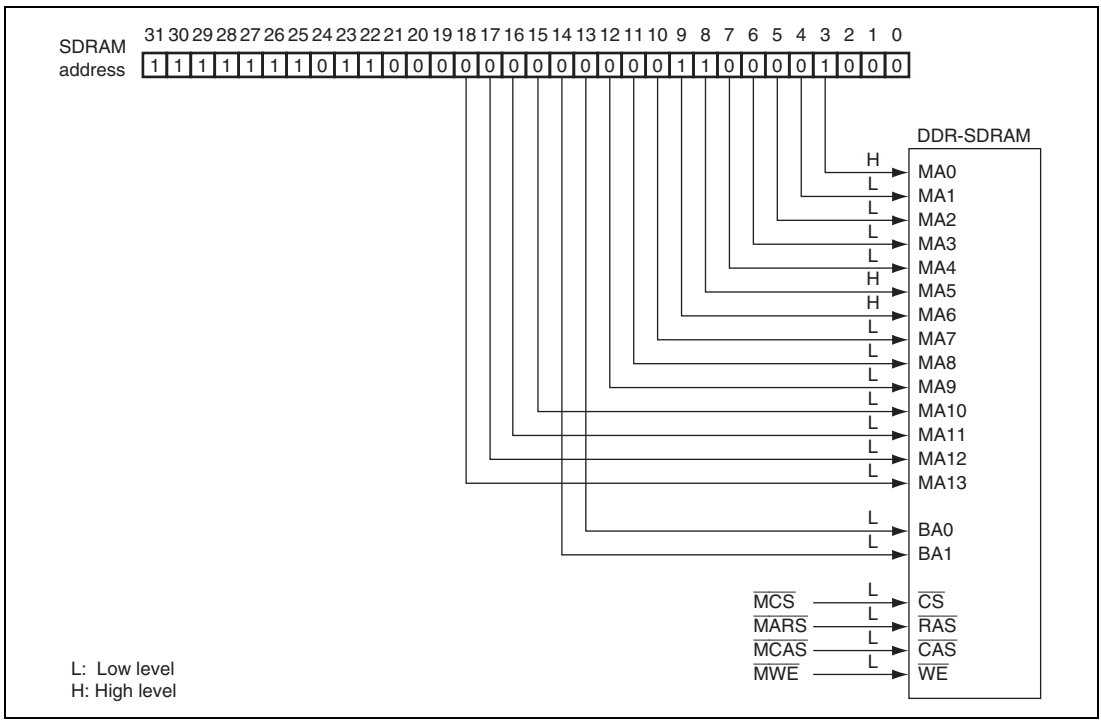


Figure 12.4 Relationship between Write Values in SDMR and Output Signals to Memory Pins

For example, to release the DLL from the reset state, set a CAS latency of 2.5 cycles, sequential burst sequence, and burst length of 2 in the mode register of the SDRAM, the following signals 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 = high, MA4 = low, MA3 = low, MA2 = low, MA1 = low, and MA0 = high

To output the above control signals, write access to address H'FEC0 0308 in SDMR is made in longwords. Then the above control signals are output to the SDRAM pins. Write data to SDMR is Don't care.

12.4.6 DDR-SDRAM Back-up Register (DBK)

This register indicates the DDR-SDRAM back-up status. For details, see section 17, Power-Down Mode.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDBUP
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	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 value should always be 0.
0	SDBUP	Undefined*	R	Back-up Status Determine whether DDR-SDRAM is or is not being battery back-up status. 0: Battery back-up 1: Not back-up

Note: * Depends on the setting of external pin ($\overline{\text{BKPRST}}$).

12.5 Operation

12.5.1 DDR-SDRAM Access

The DDR-SDRAM is accessed with a burst length of 2. Read or write commands for the same page can be issued consecutively and the data is read or written continuously.

12.5.2 DDR-SDRAM Initialization Sequence

Since the internal state of the SDRAM is undefined immediately after power is initially supplied, initialize the SDRAM according to the following sequence. The device may be damaged if you don't follow this sequence.

The below description is only an example of the initialization sequence for the DDR-SDRAM. For 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, VREF, and VTT.
2. After stabilization of the power supply, reference voltage, and clock signals, maintain the 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 endian) or the area 7 address H'1E80 0604 (big endian)/H'1E80 0600 (little endian) with 32-bit access.

Note: The initial value of this address field is H'A500 0002 and the writing value is retained in sleep mode and initialized after a power-on reset or a manual reset. When accessing the 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 endian, and so 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, CAS latency, and so on.
11. After the PREALL command has been issued, use the SMS field in SCR to issue the REFA command twice.

12. Use SDMR to issue the MRS command, release the DLL reset (MA8 = low), and determine the operating mode. In this case, use the settings for burst length, etc. that were specified in step 10.
13. After the DLL is reset, wait for 200 cycles of the MCLK: normal memory access will then be possible.

Ensure that the above SDMR settings, etc. of the SDRAM match the settings of the DDRIF registers.

12.5.3 Supported SDRAM Commands

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	MA9 to MA0
Device deselect	DESELECT	H	X	H	X	X	X	X	X	X	X
No operation	NOP	H	X	L	H	H	H	X	X	X	X
Read	READ	H	X	L	H	L	H	V	L	V	V
Read with auto precharge	READA	H	X	L	H	L	H	V	H	V	V
Write	WRITE	H	X	L	H	L	L	V	L	V	V
Write with auto precharge	WRITEA	H	X	L	H	L	L	V	H	V	V
Bank activate	ACT	H	X	L	L	H	H	V	V	V	V
Precharge select bank	PRE	H	X	L	L	H	L	X	L	V	X
Precharge all banks	PREALL	H	X	L	L	H	L	X	H	X	X
Auto refresh	REFA	H	H	L	L	L	H	X	X	X	X
Self-refresh entry from IDLE	REFS	H	L	L	L	L	H	X	X	X	X
Exit self refresh	REFSX	L	H	H	X	X	X	X	X	X	X
Enter power down	PWRDN	H	L	H	X	X	X	X	X	X	X
Exit power down	PWRDNX	L	H	H	X	X	X	X	X	X	X
Mode register set	MRS/ EMRS	H	X	L	L	L	L	V	V	V	V

[Legend]
H: High level
L: Low level
X: Don’t care

V: Valid data

The DESELECT command in table 12.6 is automatically issued whenever the SDRAM is not being accessed by any module. The DESELECT command therefore cannot be explicitly issued by the user.

12.5.4 SDRAM Access Mode

The DDRIF supports the following two SDRAM access modes. The BOMODE bits in MIM are used to select the required mode.

Bank Open Mode: The SDRAM is accessed without the PRE command immediately after a memory read or memory write, meaning that the bank is always open. This mode is useful for applications in which a single bank is the target of consecutive memory accesses. When another bank becomes the target, the PRE command is automatically issued.

Bank Closed Mode: Immediately after each round of reading or writing, the PRE command is output and the target bank is closed. This mode is useful for applications in which the same bank 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 timing and refresh addresses. Once the self-refresh mode has been set by setting the DRE and RMODE bits in MIM to 1, the self-refresh state is retained even if the CPU enters the sleep mode. If an 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 of the DDRIF, the sequence given below should be followed.

Note that in the transition from auto-refresh state to self-refresh state, the current auto-refresh state 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 all-banks) command. This closes any SDRAM bank that was open. After that, use the SMS bits in SCR to issue the REFA (auto-refresh) command to ensure that all memory rows are refreshed.

3. The STR settings do not establish a relationship between the timing of the PREALL and REFA commands that are issued by using SCR. A period of waiting that is suitable for the memory unit must be inserted.
4. Make the SDRAM enter the self-refresh state by setting the DRE and RMODE bits in MIM to 1 (in this case, the value of the DCE bit should be left at 1).
5. The DDRIF automatically issues the self-refresh command and sets the CKE pin low. The SDRAM then automatically enters the self-refresh mode.
6. Read the SELFS status bit in MIM to check whether or not the SDRAM has actually entered the self-refresh mode.

[Return from self-refresh state]

1. Clear the RMODE and DRE bits in MIM to 0 to take the DDS-SDRAM out of the self-refresh state.
2. Read the SELFS status bit in MIM to check whether or not the SDRAM has actually returned from the self-refresh mode.
3. After allowing the time required for recovery from the self-refresh state, set registers so that auto-refreshing is performed at an appropriate interval. After the recovery, wait for the time 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 command other 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-refresh) 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 make the delay adjustment unit in the DDRIF operate.
8. Set MIM so that the counter for the auto-refresh function starts counting and thus drives auto-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 enters or 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 access and has to leave the power-down mode before each round of memory access, an overhead of one cycle of the MCLK is incurred in each case.

12.5.6 Address Multiplexing

Address multiplexing is performed in line with the settings of the SPLIT bits in SDR so that connecting the SDRAM does not require an external address-multiplexing circuit. Table 12.7 shows the relationship between the settings of SPLIT bits and address multiplexing. The number of ROW or COL line is the addresses (bit) that are output to the address pins according to the setting of the SPLIT bits. If a setting not specified in table 12.7 is used, correct operation is not guaranteed.

Table 12.7 Relationship between SPLIT Bits and Address Multiplexing

	SPLIT[3:0]	ROW × COL		BA1	BA0	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
128 Mbit × 2	0001	12 × 9	ROW	13	12	—	—	11	24	23	22	21	20	19	18	17	16	15	14
(8 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	4	3	2
256 Mbit × 2	0011	13 × 9	ROW	13	12	—	11	25	24	23	22	21	20	19	18	17	16	15	14
(16 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	4	3	2
512 Mbit × 2	0100	13 × 10	ROW	13	12	—	26	25	24	23	22	21	20	19	18	17	16	15	14
(32 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	4	3	2
1 Gbit × 2	0110	14 × 10	ROW	13	12	27	26	25	24	23	22	21	20	19	18	17	16	15	14
(64 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	4	3	2

Note: * Auto-precharge

12.6 DDR-SDRAM Basic Timing

Figures 12.5 to 12.14 show basic timing of the DDRIF.

In each timing chart, the DDR-SRAM has been idle at T0.

The settings in the SDRAM timing register (STR) must set up timing that is within the specifications of the DDR-SDRAM.

Note that the only CAS latency supported by the DDRIF is 2.5.

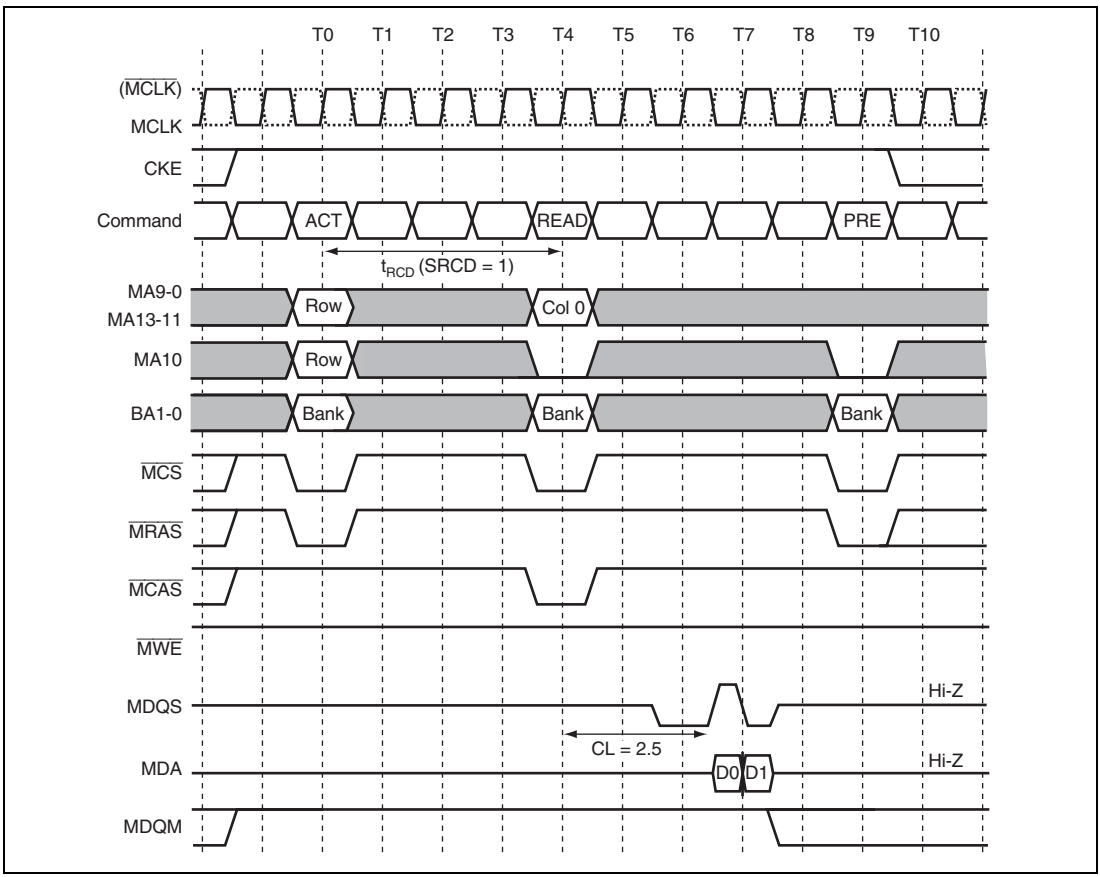


Figure 12.5 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Read without Auto Precharge)

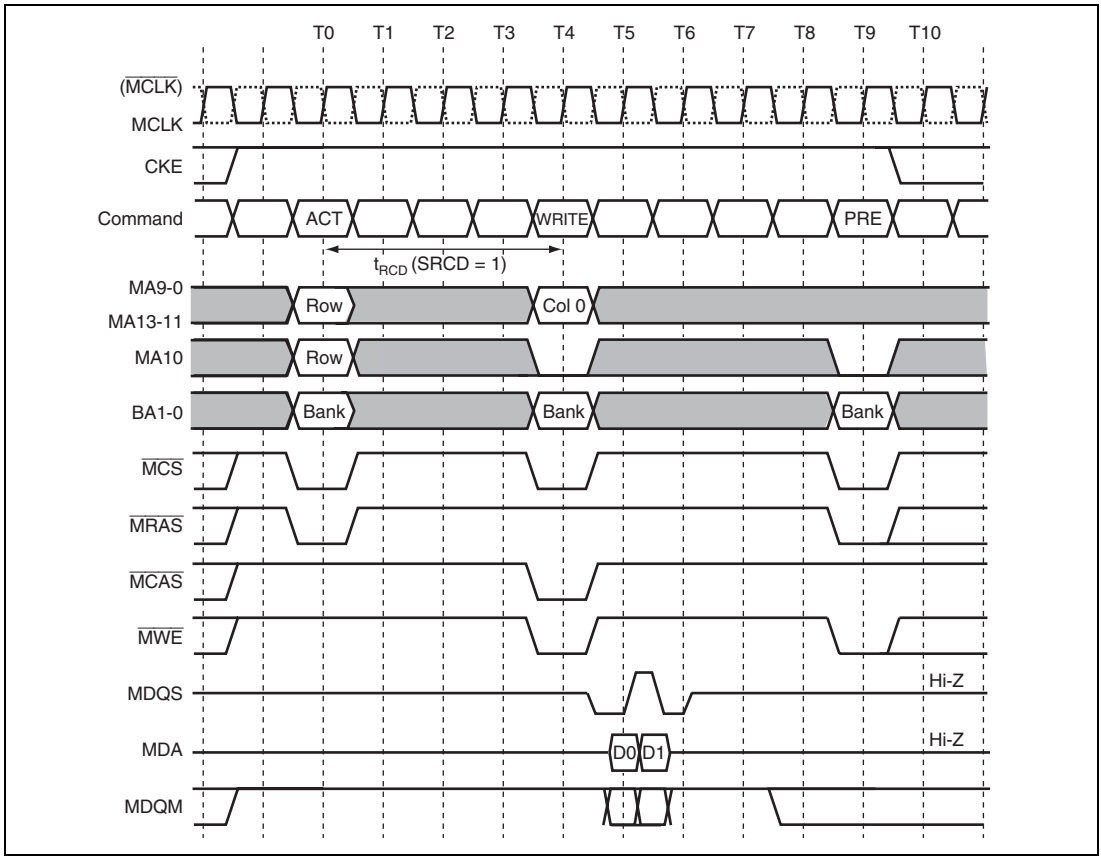


Figure 12.6 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write without Auto Precharge)

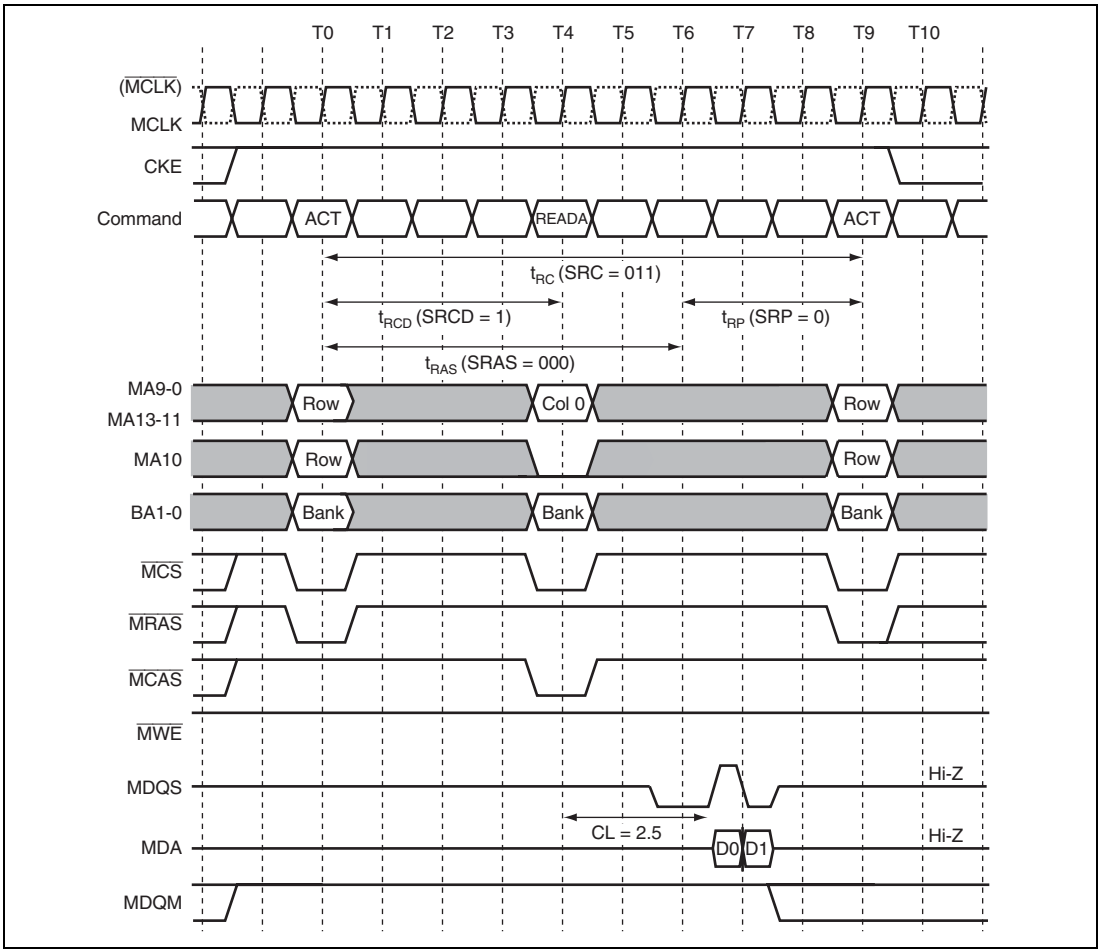


Figure 12.7 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Read with Auto Precharge)

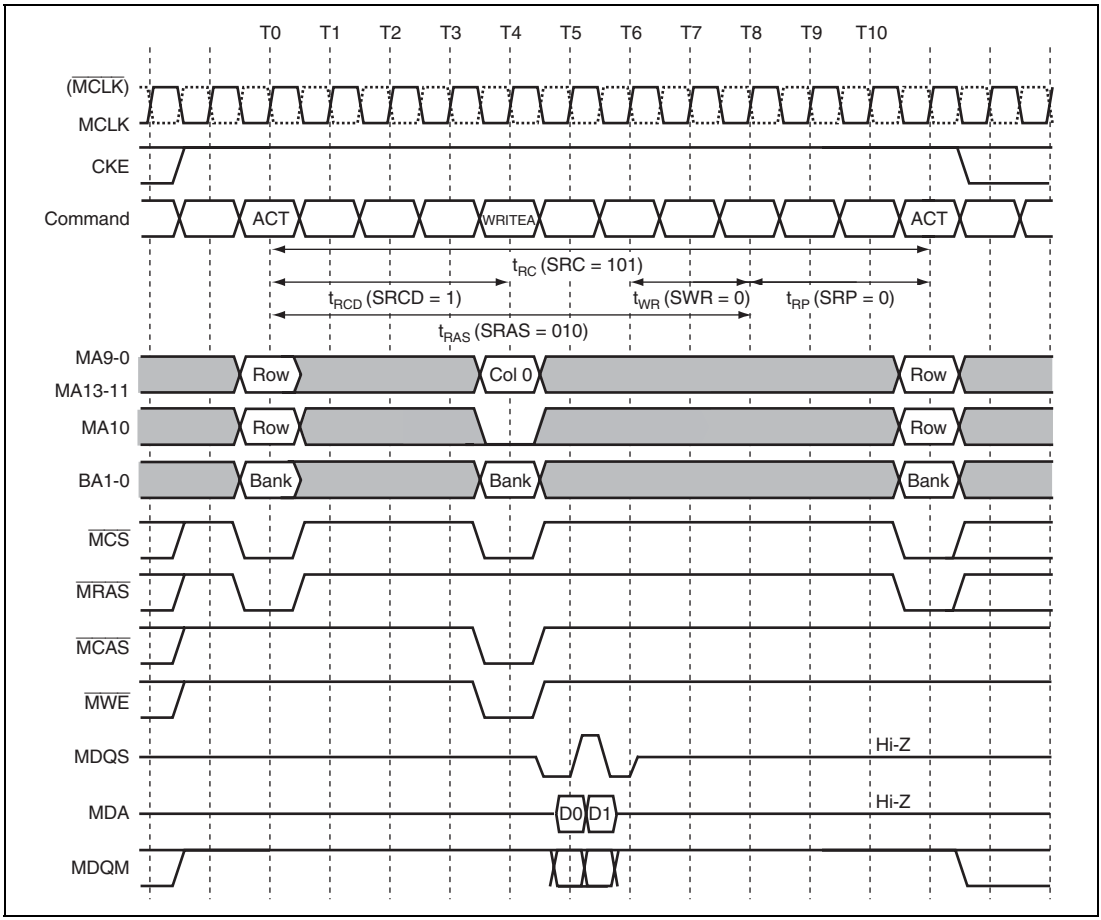


Figure 12.8 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write with Auto Precharge)

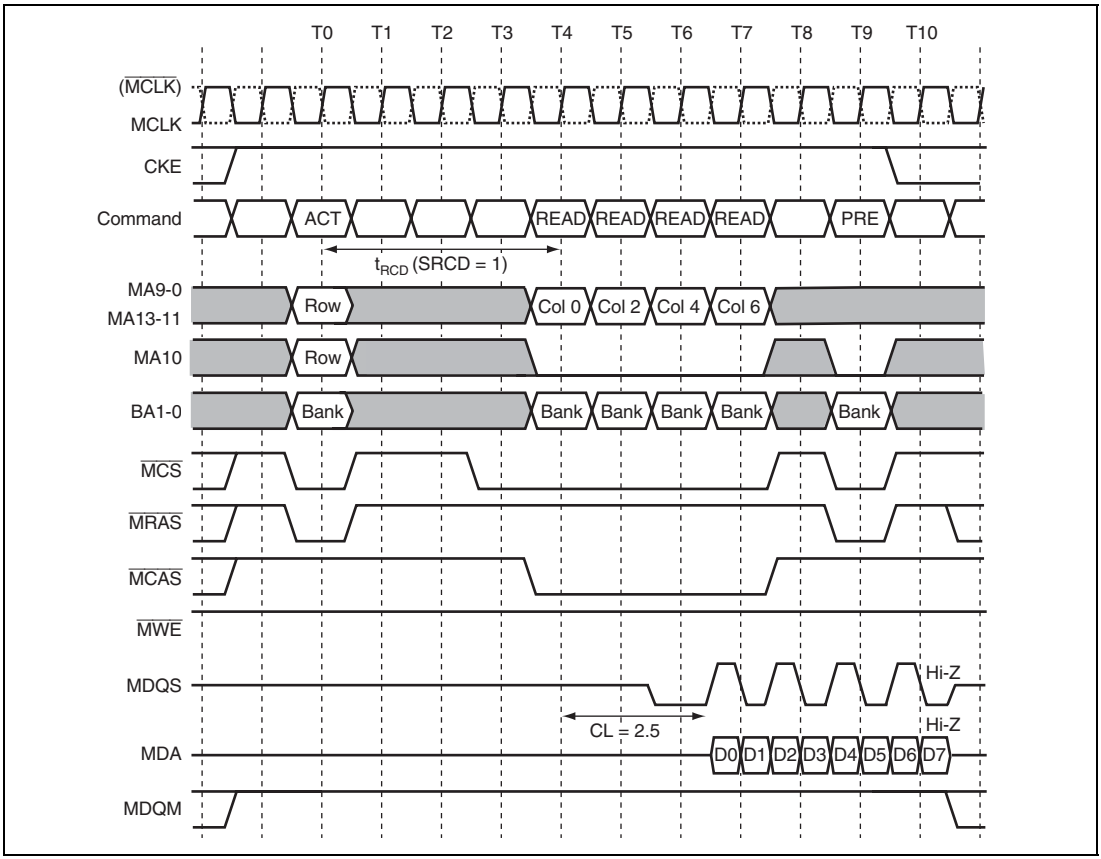


Figure 12.9 DDRIF Basic Timing
(4 Burst Read: 32-byte without Auto Precharge)

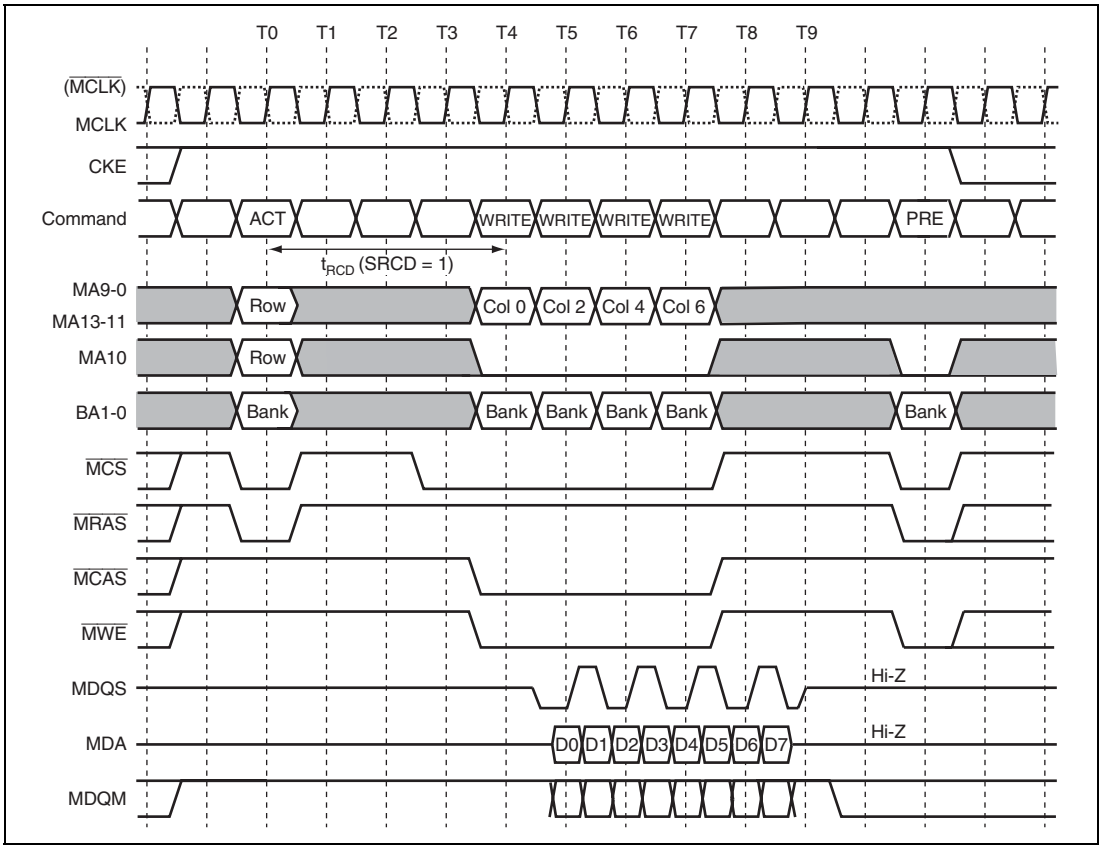


Figure 12.10 DDRIF Basic Timing
(4 Burst Write: 32-byte without Auto Precharge)

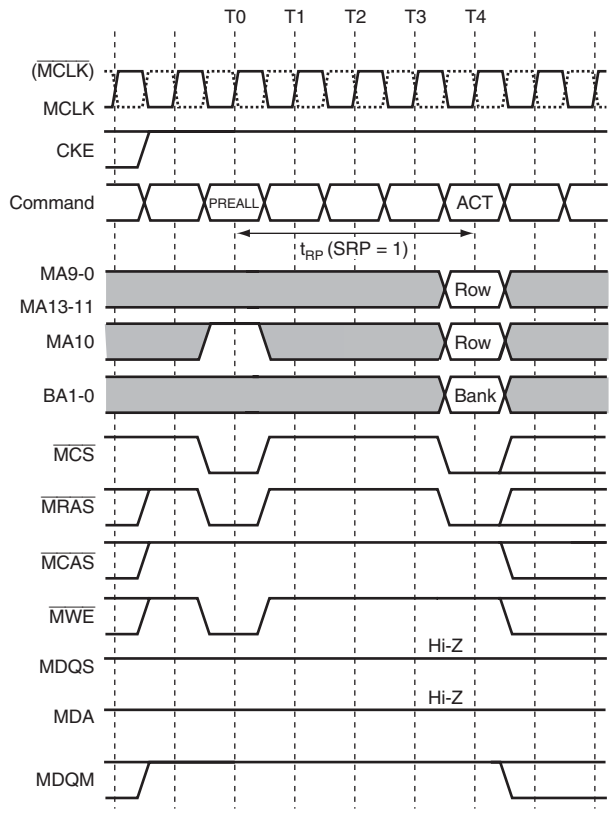
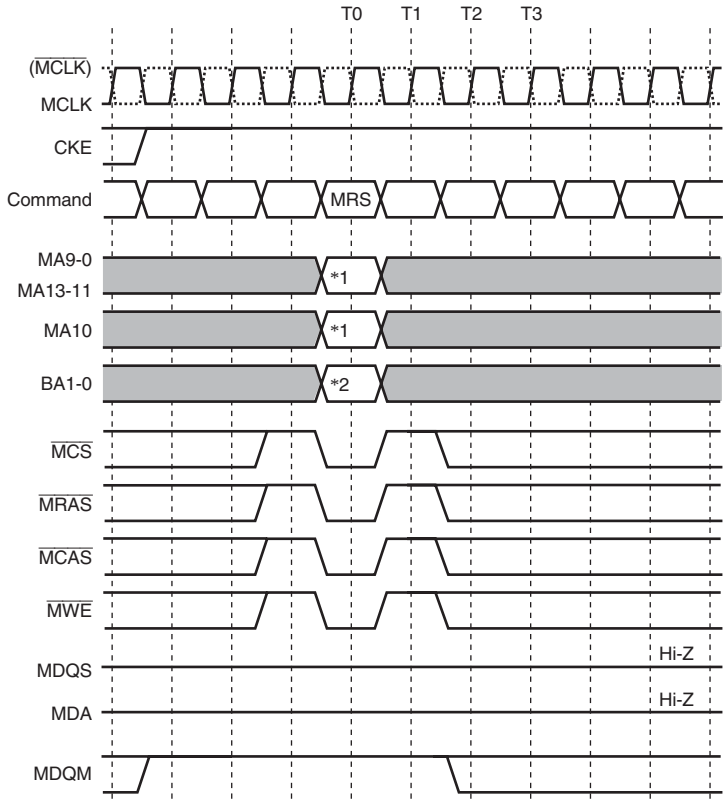


Figure 12.11 DDRIF Basic Timing (from Precharging All Banks to Bank Activation)



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)

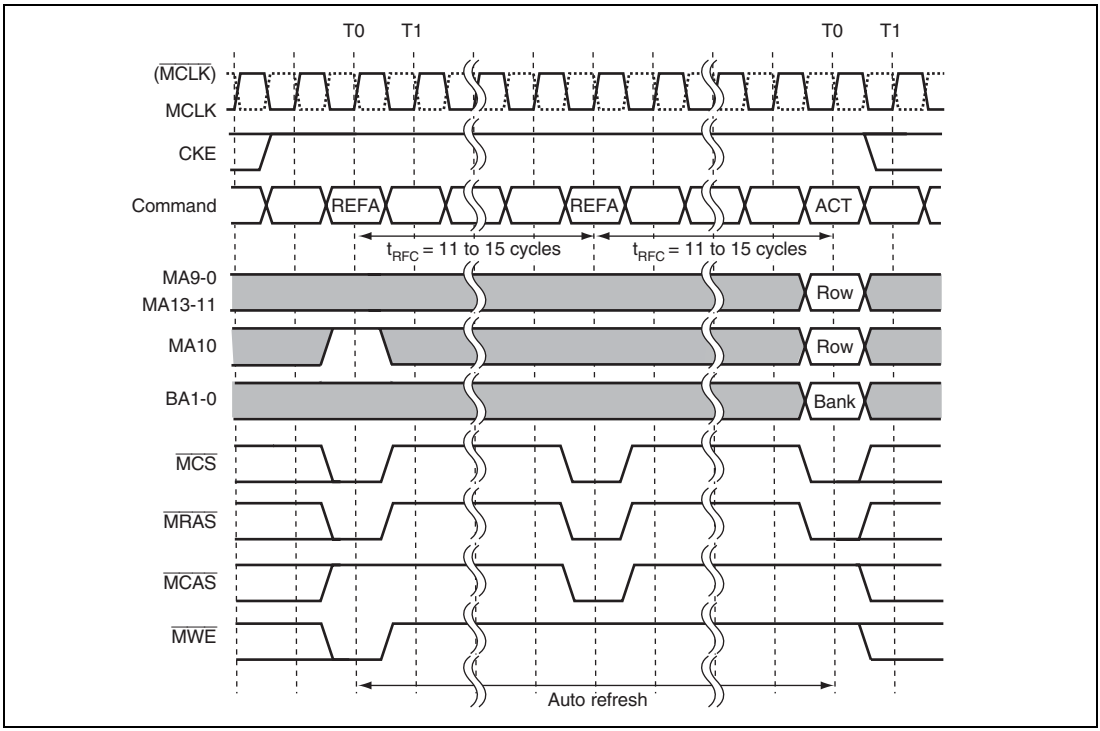
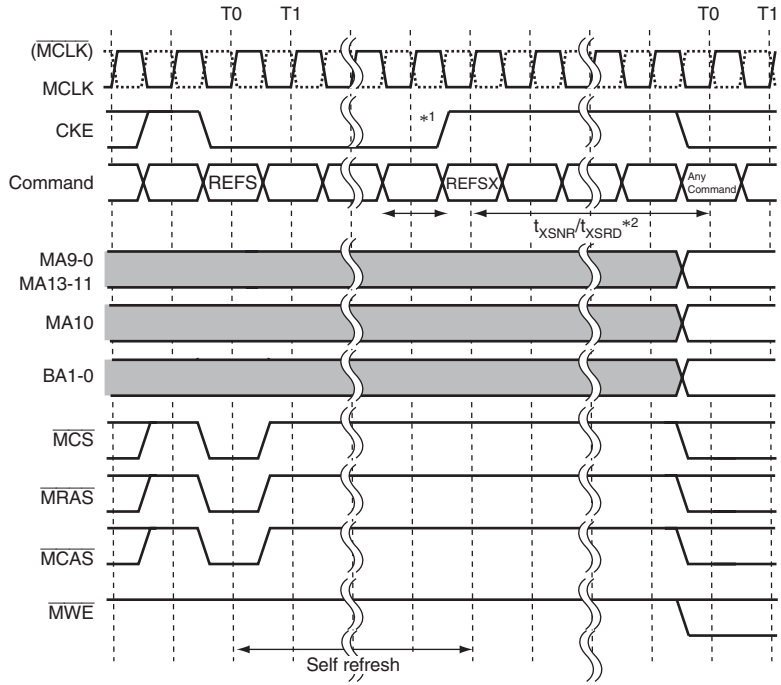


Figure 12.13 DDRIF Basic Timing (Enter Auto-Refresh/Exit to Bank Activation)



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 Issuing)

12.7 Usage Notes

12.7.1 Operating Frequency

The DDRIF supports ratios of 5:4 (DDR320) and 1:1 (DDR266) between the frequencies of the SuperHyway clock (SHck) and DDR clock (DDRck). For details, see section 15, Clock Pulse Generator (CPG). The maximum operating frequency of the SuperHyway clock is 200 MHz. The minimum operating frequency depends on the frequency of the DDR-SDRAM clock. Therefore, see the datasheet for the DDR-SDRAM.

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 the frequency change register (FRQCR) of the CPG.

Since the clock signal is not being supplied in the above situations, auto-refreshing does not proceed. Since the refresh cycle is not being maintained, data in the SDRAM will be lost. To prevent this, software should place the SDRAM in the self-refresh state before supply of the clock signal is stopped. For details on making the SDRAM enter and leave the self-refresh mode, see 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 written to). When the REFA (auto-refresh) command is issued by using the SMS bits in SCR, be sure to close the bank by using the SMS bits in SCR to issue the PREALL command. The same operation is necessary when the SCR register setting is used to issue a REFA command for refreshing all rows 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 issuing an ACT command to issuing a read with auto-precharge or write with auto-precharge command and the number of cycles (tRCD) required from issuing an ACT command to issuing a read or write command are the same. If the two numbers differ, the SDRAM should be accessed in bank open mode.

12.7.5 Setting Auto-Refresh Interval

The auto-refresh interval is specified by the DRI bits in MIM. If the DRE bit is set to 1 at the same time as the DRI bits are set, the time until the first auto-refresh is that selected by the value of the DRI bits before the new setting was made. However, the second and subsequent auto-refresh intervals take on the value corresponding to the new setting for the DRI bits. To avoid this situation, clear the DRE bit to 0 whenever you change the settings of the DRI bits. When the DRE bit is subsequently set to 1 and the same DRI setting is repeated, auto-refreshing proceeds with the specified interval from the first round. In this case, take care to ensure that the DRI bits have the same value.

Section 13 PCI Controller (PCIC)

The PCI controller (PCIC) controls the PCI bus for data transfers between memory connected to an external bus and a PCI device connected to the PCI bus. The ability to connect PCI devices facilitates the design of systems using the PCI bus and enables more compact systems capable of faster data transfer.

The PCIC functions as a bus bridge which connects an external PCI bus to the internal SuperHyway bus. It provides a device connected to the external PCI bus with a channel for access to the on-chip modules connected to the SuperHyway bus. The PCIC supports both the host bus bridge mode and normal mode (non-host mode). In host busbridge mode, PCI bus arbitration control is available and in normal mode, arbitration is executed by the external PCI bus arbiter.

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 settings)
- 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

- Exclusive access (target only)
  - Once locked, only accessible from the device that accessed the $\overline{\text{LOCK}}$ signal
  - The SuperHyway bus in not locked during lock transfer
- Can support cache coherency between a device connected to the PCI bus and system memory (PCI target) although device performance may become suboptimal
- Supports four external interrupt inputs ($\overline{\text{INTD}}$ to $\overline{\text{INTA}}$) in host bus bridge mode
- Supports one external interrupt output ($\overline{\text{INTA}}$) in normal mode
- Supports both big endian and little endian formats for the SuperHyway bus (the PCI bus operates in the little endian format)

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 is a block diagram of the PCIC.

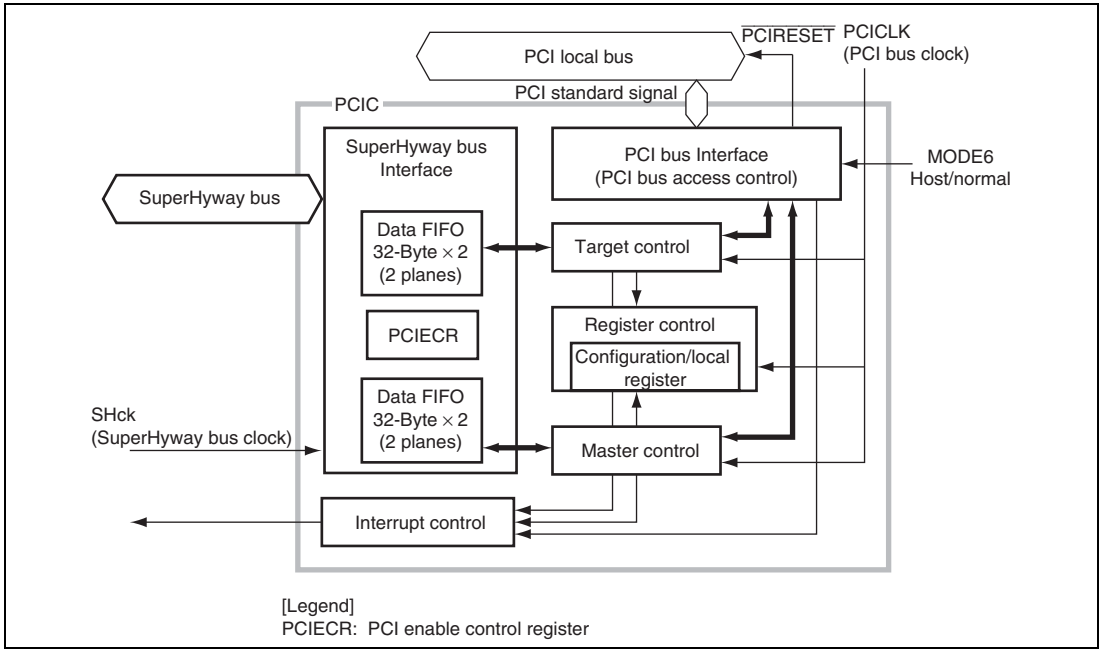


Figure 13.1 PCIC Block Diagram

The PCIC comprises two blocks: the PCI bus interface and SuperHyway bus interface block.

The PCI bus interface block comprises the PCI configuration register, local register, PCI master, 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 data FIFO.

The functions of the SuperHyway bus interface are access translation between the PCI bus interface and the CPU or DMAC via SuperHyway bus.

The interrupt controller requests interrupt request to the INTC of this LSI.

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the PCIC.

Table 13.1 Input/Output Pins

Pin Name	PCI standard signal name	I/O	Description
AD31 to AD0*[1]	AD[31:0]	I/O (TRI)	PCI Address/Data Bus Address and data buses are multiplexed. Each bus transaction consists of an address phase followed by one or more data phases.
CBE3 to CBE0	C/$\overline{BE}$[3:0]	I/O (TRI)	PCI Command/Byte Enable Bus command and byte enables are multiplexed. These signals indicate the type of transaction during the address phase and the byte enables during the data phases.
PAR	PAR	I/O (TRI)	PCI Parity Generates/checks even parity across AD[31:0] and CBE[3:0].
PCICLK	CLK	Input	PCI Clock Provides timing for all transactions on the PCI bus.
$\overline{PCIFRAME}$	$\overline{FRAME}$	I/O (STRI)	PCI Frame Current initiator drives this signal, which indicates the start and duration or end of a transaction.
$\overline{TRDY}$	$\overline{TRDY}$	I/O (STRI)	PCI Target Ready Selected target drives this signal, which indicates the target is ready to execute a transaction. During a write, this signal indicates that the target is ready to accept data. During a read, this signal indicates that valid data is present on the AD [31:0] lines.
$\overline{IRDY}$	$\overline{IRDY}$	I/O (STRI)	PCI Initiator Ready The current bus master drives this signal. During a write, this signal indicates that valid data is present on the AD [31:0] lines. During a read, this signal indicates that the master is ready to accept data.
$\overline{STOP}$	$\overline{STOP}$	I/O (STRI)	PCI Stop Selected target drives this signal to stop the current transaction.
$\overline{LOCK}$	$\overline{LOCK}$	I/O (STRI)	PCI Lock

Pin Name	PCI standard signal name	I/O	Description
IDSEL	IDSEL	Input	PCI Configuration Device Select This signal is input to the PCI device to select configuration cycles (only for normal mode).
$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	I/O (STRI)	PCI Device Select Indicates the device driving this signal has decoded its address as the target. As an input, this signal indicates that a device has been selected.
$\overline{\text{INTD}}$*[2] $\overline{\text{INTC}}$*[3] $\overline{\text{INTB}}$*[3]	$\overline{\text{INTD}}$ $\overline{\text{INTC}}$ $\overline{\text{INTB}}$	Input	Interrupts D, C, and B Indicate that a PCI device is requesting an interrupt. Only these signals are available in host bus bridge mode.
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	I/O (output: O/D)	Interrupt A Indicates that a PCI device is requesting an interrupt (input) in host bus bridge mode. This signal is used to request an interrupt (output: O/D) in normal mode.
$\overline{\text{REQ3}}$ 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 bridge 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 bridge 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 mode)
MODE6*[5]	—	Input	PCI Operating Mode Select Low: PCI normal mode in which the PCIC operates as a PCI bridge on the PCICLK High: PCI host bus bridge mode in which the PCIC operates as a PCI bridge on the PCICLK

[Legend]
TRI: Tri-state
STRI: Sustained tri-state
O/D: Open Drain
Notes: 1. These pins are multiplexed with the GPIO pins (port A to D).
2. This pin is multiplexed with the SCIF channel 0 and GPIO pins.
3. These pins are multiplexed with the DMAC, H-UDI and GPIO pins.
4. These pins are multiplexed with the GPIO pins.
5. This pin is multiplexed with the INTC and FLCTL pins.

13.3 Register Descriptions

Table 13.2 shows the PCIC register configuration. Table 13.3 shows the register states in each operating mode. The PCI configuration register address and its offset are used for little endian operation.

Table 13.2 List of PCIC Registers

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
Control register space						
PCIC enable control register	PCIECR	R/W	—	H'FE00 0008	H'1E00 00008	32
PCI configuration register space						
PCI vendor ID register	PCIVID	R	R	H'FE04 0000	H'1E04 0000	16
PCI device ID register	PCIDID	R	R	H'FE04 0002	H'1E04 0002	16
PCI command register	PCICMD	R/W	R/W	H'FE04 0004	H'1E04 0004	16
PCI status register	PCISTATUS	R/WC	R/WC	H'FE04 0006	H'1E04 0006	16
PCI revision ID register	PCIRID	R	R	H'FE04 0008	H'1E04 0008	8
PCI program interface register	PCIPIF	R/W	R	H'FE04 0009	H'1E04 0009	8
PCI sub class code register	PCISUB	R/W	R	H'FE04 000A	H'1E04 000A	8
PCI base class code register	PCIBCC	R/W	R	H'FE04 000B	H'1E04 000B	8
PCI cacheline size register	PCICLS	R	R	H'FE04 000C	H'1E04 000C	8
PCI latency timer register	PCILTM	R/W	R/W	H'FE04 000D	H'1E04 000D	8
PCI header type register	PCIHDR	R	R	H'FE04 000E	H'1E04 000E	8
PCI BIST register	PCIBIST	R	R	H'FE04 000F	H'1E04 000F	8
PCI I/O base address register	PCIIBAR	R/W	R/W	H'FE04 0010	H'1E04 0010	32
PCI Memory base address register 0	PCIMBAR0	R/W	R/W	H'FE04 0014	H'1E04 0014	32
PCI Memory base address register 1	PCIMBAR1	R/W	R/W	H'FE04 0018	H'1E04 0018	32
PCI subsystem vendor ID register	PCISVID	R/W	R	H'FE04 002C	H'1E04 002C	16
PCI subsystem ID register	PCISID	R/W	R	H'FE04 002E	H'1E04 002E	16
PCI capabilities pointer register	PCICP	R	R	H'FE04 0034	H'1E04 0034	8
PCI interrupt line register	PCIINTLINE	R/W	R/W	H'FE04 003C	H'1E04 003C	8
PCI interrupt pin register	PCIINTPIN	R/W	R	H'FE04 003D	H'1E04 003D	8

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
PCI minimum grant register	PCIMINGNT	R	R	H'FE04 003E	H'1E04 003E	8
PCI maximum latency register	PCIMAXLAT	R	R	H'FE04 003F	H'1E04 003F	8
PCI capability ID register	PCICID	R	R	H'FE04 0040	H'1E04 0040	8
PCI next item pointer register	PCINIP	R	R	H'FE04 0041	H'1E04 0041	8
PCI power management capability register	PCIPMC	R/W	R/W	H'FE04 0042	H'1E04 0042	16
PCI power management control/status register	PCIPMCSR	R/W	R/W	H'FE04 0044	H'1E04 0044	16
PCI PMCSR bridge support extension register	PCIPMCSR BSE	R	R	H'FE04 0046	H'1E04 0046	8
PCI power consumption/dissipation data register	PCIPCDD	R/W	R	H'FE04 0047	H'1E04 0047	8
PCI local register space						
PCI control register	PCICR	R/W	R	H'FE04 0100	H'1E04 0100	32
PCI local space register 0	PCILSR0	R/W	R	H'FE04 0104	H'1E04 0104	32
PCI local space register 1	PCILSR1	R/W	R	H'FE04 0108	H'1E04 0108	32
PCI local address register 0	PCILAR0	R/W	R	H'FE04 010C	H'1E04 010C	32
PCI local address register 1	PCILAR1	R/W	R	H'FE04 0110	H'1E04 0110	32
PCI interrupt register	PCIIR	R/WC	R	H'FE04 0114	H'1E04 0114	32
PCI interrupt mask register	PCIIMR	R/W	R	H'FE04 0118	H'1E04 0118	32
PCI error address information register	PCIAIR	R	R	H'FE04 011C	H'1E04 011C	32
PCI error command information register	PCICIR	R	R	H'FE04 0120	H'1E04 0120	32
PCI arbiter interrupt register	PCIAINT	R/WC	R	H'FE04 0130	H'1E04 0130	32
PCI arbiter interrupt mask register	PCIAINTM	R/WC	R	H'FE04 0134	H'1E04 0134	32
PCI arbiter bus master error information register	PCIBMIR	R	R	H'FE04 0138	H'1E04 0138	32
PCI PIO*[3] address register	PCIPAR	R/W	—	H'FE04 01C0	H'1E04 01C0	32
PCI power management interrupt register	PCIPINT	R/WC	—	H'FE04 01CC	H'1E04 01CC	32
PCI power management interrupt mask register	PCIPINTM	R/W	—	H'FE04 01D0	H'1E04 01D0	32

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
PCI memory bank register 0	PCIMBR0	R/W	—	H'FE04 01E0	H'1E04 01E0	32
PCI memory bank mask register 0	PCIMBMR0	R/W	—	H'FE04 01E4	H'1E04 01E4	32
PCI memory bank register 1	PCIMBR1	R/W	—	H'FE04 01E8	H'1E04 01E8	32
PCI memory bank mask register 1	PCIMBMR1	R/W	—	H'FE04 01EC	H'1E04 01EC	32
PCI memory bank register 2	PCIMBR2	R/W	—	H'FE04 01F0	H'1E04 01F0	32
PCI memory bank mask register 2	PCIMBMR2	R/W	—	H'FE04 01F4	H'1E04 01F4	32
PCI I/O bank register	PCIIOBR	R/W	—	H'FE04 01F8	H'1E04 01F8	32
PCI I/O bank master register	PCIIOBMR	R/W	—	H'FE04 01FC	H'1E04 01FC	32
PCI cache snoop control register 0	PCICSCR0	R/W	—	H'FE04 0210	H'1E04 0210	32
PCI cache snoop control register 1	PCICSCR1	R/W	—	H'FE04 0214	H'1E04 0214	32
PCI cache snoop address register 0	PCICSAR0	R/W	—	H'FE04 0218	H'1E04 0218	32
PCI cache snoop address register 1	PCICSAR1	R/W	—	H'FE04 021C	H'1E04 021C	32
PCI PIO*[3] data register	PCIPDR	R/W	—	H'FE04 0220	H'1E04 0220	32

Notes: 1. SH: SuperHyway bus (internal bus). PCI: PCI local bus. WC: Cleared by writing 1 (Writing of 0 is no effect). —: Accessing is prohibited.
2. When accessing a register, do not use a size smaller than the register's access size.
3. PIO: Programmed I/O.

Table 13.3 Register States in Each Operating Mode

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode
Control register space				
PCIC enable control register	PCIECR	H'0000 0000	Retained	Retained
PCI configuration register space				
PCI vendor ID register	PCIVID	H'1912	Retained	Retained
PCI device ID register	PCIDID	H'0002	Retained	Retained
PCI command register	PCICMD	H'0080	Retained	Retained
PCI status register	PCISTATUS	H'0290	Retained	Retained
PCI revision ID register	PCIRID	H'00	Retained	Retained
PCI program interface register	PCIPIF	H'00	Retained	Retained
PCI sub class code register	PCISUB	H'00	Retained	Retained
PCI base class code register	PCIBCC	H'00	Retained	Retained
PCI cache line size register	PCICLS	H'20	Retained	Retained
PCI latency timer register	PCILTM	H'00	Retained	Retained
PCI header type register	PCIHDR	H'00	Retained	Retained
PCI BIST register	PCIBIST	H'00	Retained	Retained
PCI I/O base address register	PCIIBAR	H'0000 0001	Retained	Retained
PCI Memory base address register 0	PCIMBAR0	H'0000 0000	Retained	Retained
PCI Memory base address register 1	PCIMBAR1	H'0000 0000	Retained	Retained
PCI subsystem vendor ID register	PCISVID	H'0000	Retained	Retained
PCI subsystem ID register	PCISID	H'0000	Retained	Retained
PCI capabilities pointer register	PCICP	H'40	Retained	Retained
PCI interrupt line register	PCIINTLINE	H'00	Retained	Retained
PCI interrupt pin register	PCIINTPIN	H'01	Retained	Retained
PCI minimum grant register	PCIMINGNT	H'00	Retained	Retained
PCI maximum latency register	PCIMAXLAT	H'00	Retained	Retained
PCI capability ID register	PCICID	H'01	Retained	Retained
PCI next item pointer register	PCINIP	H'00	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode
PCI power management capability register	PCIPMC	H'000A	Retained	Retained
PCI power management control/status register	PCIPMCSR	H'0000	Retained	Retained
PCI PMCSR bridge support extension register	PCIPMCSR BSE	H'00	Retained	Retained
PCI power consumption/dissipation data register	PCIPCDD	H'00	Retained	Retained
PCI local register space				
PCI control register	PCICR	H'0000 00xx	Retained	Retained
PCI local space register 0	PCILSR0	H'0000 0000	Retained	Retained
PCI local space register 1	PCILSR1	H'0000 0000	Retained	Retained
PCI local address register 0	PCILAR0	H'0000 0000	Retained	Retained
PCI local address register 1	PCILAR1	H'0000 0000	Retained	Retained
PCI interrupt register	PCIIR	H'0000 0000	Retained	Retained
PCI interrupt mask register	PCIIMR	H'0000 0000	Retained	Retained
PCI error address information register	PCIAIR	H'xxxx xxxx	Retained	Retained
PCI error command information register	PCICIR	H'xx00 000x	Retained	Retained
PCI arbiter interrupt register	PCIAINT	H'0000 0000	Retained	Retained
PCI arbiter interrupt mask register	PCIAINTM	H'0000 0000	Retained	Retained
PCI arbiter bus master error information register	PCIBMIR	H'0000 00xx	Retained	Retained
PCI PIO address register	PCIPAR	H'80xx xxxx	Retained	Retained
PCI power management interrupt register	PCIPINT	H'0000 0000	Retained	Retained
PCI power management interrupt mask register	PCIPINTM	H'0000 0000	Retained	Retained
PCI memory bank register 0	PCIMBR0	H'0000 0000	Retained	Retained
PCI memory bank mask register 0	PCIMBMR0	H'0000 0000	Retained	Retained
PCI memory bank register 1	PCIMBR1	H'0000 0000	Retained	Retained
PCI memory bank mask register 1	PCIMBMR1	H'0000 0000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode
PCI memory bank register 2	PCIMBR2	H'0000 0000	Retained	Retained
PCI memory bank mask register 2	PCIMBMR2	H'0000 0000	Retained	Retained
PCI I/O bank register	PCIIOBR	H'0000 0000	Retained	Retained
PCI I/O bank master register	PCIIOBMR	H'0000 0000	Retained	Retained
PCI cache snoop control register 0	PCICSCR0	H'0000 0000	Retained	Retained
PCI cache snoop control register 1	PCICSCR1	H'0000 0000	Retained	Retained
PCI cache snoop address register 0	PCICSAR0	H'0000 0000	Retained	Retained
PCI cache snoop address register 1	PCICSAR1	H'0000 0000	Retained	Retained
PCI PIO data register	PCIPDR	H'xxxx xxxx	Retained	Retained

[Legend] x: Undefined

13.3.1 PCIC Enable Control Register (PCIECR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ENBL	0	R/W	PCI Enable Bit. Enable the PCIC 0: PCIC disable The access from both the CPU and external PCI devices to the PCIC is invalid (including the configuration and local register), except PCIECR. 1: PCIC enable

13.3.2 Configuration Registers

The configuration registers defines the programming model and usages for the configuration register space in a PCI compliant device. For details, refer to “PCI Local Bus Specification Revision 2.2 Chapter 6 Configuration Space ”.

(1) PCI Vender ID Register (PCIVID)

This register identifies the manufacturer of device.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VID															
Initial value:	0	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0
SH R/W:	R	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	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 (vender ID) that is allocated by PCI-SIG. Renesas Technology’s vendor ID is H'1912.

(2) PCI Device ID Register (PCIDID)

This register uniquely identifies this LSI amongst PCI devices manufactured by the vendor.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SH R/W:	R	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	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 PCI devices manufactured by the vendor indicated by the PCI vender field.
				The SH7780’s device ID is H'0002.

(3) PCI Command Register (PCICMD)

The PCI command register provides coarse control over a device's ability to generate and respond to PCI cycles. When 0 is written to this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FBBE	SERRE	WCC	PER	VGAPS	MWIE	SC	BM	MS	IOS
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W

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 value 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-to-back transactions to different device.
				0: Fast back-to-back transactions are only allowed to the same target
				1: Master is allowed to generate fast back-to-back 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 for a master write, only an address for a master read, and only data for a target read are output for at least two clock cycles.
				0: Address/data stepping control disabled
				1: Address/data stepping control enabled

Bit	Bit Name	Initial Value	R/W	Description
6	PER	0	SH: R/W	Parity Error
			PCI: R/W	Controls the device's response when the PCIC detects a parity error or receives a parity error. When this bit is set to 1, the $\overline{\text{PERR}}$ signal is asserted.
				0: No response parity error
				1: Response parity error
5	VGAPS	0	SH: R	VGA Palette Snoop Control
			PCI: R	0: VGA compatible device
				1: Palette register write is not supported (not supported)
4	MWIE	0	SH: R	PCI Memory Write and Invalidate Control
			PCI: R	Controls issuance of a memory write and invalidate command in a master access.
				0: Memory write is used
				1: Memory write and invalidate command is executable (not supported)
3	SC	0	SH: R	PCI Special Cycles
			PCI: R	Indicates whether or not to support the special cycle 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 LSI. When this bit is cleared to 0, a memory transfer to the PCIC is terminated with a master abort.
				0: Does not respond to memory space accesses
				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. When this 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

(4) PCI Status Register (PCISTATUS)

This status register is used to record status information for PCI bus related events. The definition of each of the bits is given in the table below. A device may not need to implement all the bits, depending on device functionality. For instance, since a device that acts as a target does not inform a target abort, bit 11 does not need to be implemented. Reserved bits should be read-only and return zero when the bits are read.

Reads from this register operates normally. Writes are slightly different in that bits can be cleared, but not set. A one bit is cleared whenever the register is written to, and the write data in the corresponding bit location is a 1. For instance, to clear bit 14 and not affect any other bits, write 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	1	0
	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	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	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	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 detected in read data when the PCIC is a master or in write data when the PCIC is a target.
				This bit must be set by the device whenever it detects a parity error, even if parity error handling is disabled.
				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 $\overline{\text{SERR}}$ signal.
				0: $\overline{\text{SERR}}$ has not been asserted
				1: $\overline{\text{SERR}}$ has been asserted (the value retained until cleared)

Bit	Bit Name	Initial Value	R/W	Description
13	RMA	0	SH: R/WC	Master Abort Receive Status
			PCI: R/WC	Indicates that the PCIC has terminated a transaction with a master abort when the PCIC is a master.
				0: PCIC has not terminated a transaction with a master abort
				1: PCIC has terminated a transaction with a master abort
12	RTA	0	SH: R/WC	Target Abort Receive Status
			PCI: R/WC	Indicates that a transaction is terminated by a target device with a target abort when the PCIC functions as a master.
				0: Transaction has not been terminated with a target abort
				1: Transaction has been terminated with a target abort
11	STA	0	SH: R/WC	Target Abort Execution Status
			PCI: R/WC	Indicates that the PCIC has terminated a transaction with a target-abort when the PCIC functions as a target.
				0: PCIC has not terminated a transaction with a target-abort
				1: PCIC has terminated a transaction with target-abort
10, 9	DEVSEL	01	SH: R	DEVSEL Timing Status
			PCI: R	Indicate the response timing status of the $\overline{\text{DEVSEL}}$ 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 $\overline{\text{PERR}}$ signal or detected the assertion of the $\overline{\text{PERR}}$ signal if the PCIC functions as a master. Only when the parity response bit has been set to 1, this bit is set to 1.
				0: Data parity error has not been generated
				1: Data parity error has been generated

Bit	Bit Name	Initial Value	R/W	Description
7	FBBC	1	SH: R	Fast Back-to-Back Status
			PCI: R	Indicates whether or not the PCIC is capable of accepting fast back-to-back transactions when the transactions are not to the same agent if the PCIC functions as a target.
				0: Fast back-to-back transactions to different agents not supported
				1: Fast back-to-back transactions to different agents supported
6	—	0	SH: R/W	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
5	66C	0	SH: R/W	66MHz-Operation Capable Status
			PCI: R	Indicates whether or not the PCIC is capable of 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 management 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 value should always be 0.

(5) PCI Revision ID Register (PCIRID)

This register specifies a device specific revision identifier.

Bit:	7	6	5	4	3	2	1	0
	RID							
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	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 version of 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 of the 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 writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3	PIS	0	SH: R/W	PCI Programmable Indicator (Secondary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.
2	OMS	0	SH: R/W	PCI Operating Mode (Secondary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. 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 writable. When the CFINIT bit in PCICR is 1, writing is ignored. 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 writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.

(7) PCI Sub Class Code Register (PCISUB)

This register identifies the sub class code. For details of the class code, refer to “PCI Local Bus Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	SUB							
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	SUB	H'00	SH: R/W	Sub Class Code
			PCI: R	Indicate the sub class code. The initial value is H'00.

(8) PCI Base Class Code Register (PCIBCC)

This register identifies the base class code. For details of the class code, refer to “PCI Local Bus 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 value is H'00.

(9) PCI Cacheline Size Register (PCICLS)

Bit:	7	6	5	4	3	2	1	0
	CLS							
Initial value:	0	0	1	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	CLS	H'20	SH: R	Cache Line Size: Not supported
			PCI: R	A memory target does not support a cache. SDON and $\overline{\text{SBO}}$ 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 bus 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 acquisition clocks of PCI bus when the PCIC is operating as the master.

(11) PCI Header Type Register (PCIHDR)

Bit:	7	6	5	4	3	2	1	0
	MFE	HDR						
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	MFE	0	SH: R	Multiple Function Enable
			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 registers.
				H'00: Type "00h" layout supported
				H'01: Type "01h" layout supported (not supported)

(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 and status.
			PCI: R	
				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 value should always be 0.

(13) PCI I/O Base Address Register (PCIIBAR)

This register packages the I/O space base address register of the PCI configuration register 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	17	16
	IOB (upper)															
Initial value:	0	0	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/W	R/W
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOB (upper)								IOB (lower)						—	ASI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
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	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	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 address that corresponds the PCIC local register space (PCIC 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 value should always be 0.
0	ASI	1	SH: R	Address Space Indicator
			PCI: R	Indicates whether the base address in this register indicates the I/O or memory space.
				0: Memory space
				1: I/O space

(14) PCI Memory Base Address Register 0 (PCIMBAR0)

This register packages the memory space base address register of the PCI configuration register 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	17	16
	MBA (upper)												MBA (lower)			
Initial value:	0	0	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	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	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBA (lower)												LAP	LAT		ASI
Initial value:	0	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	R	R	R
PCI R/W:	R	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 20	MBA (upper)	H'000	SH: R/W	Memory Space 0 Base Address (upper 12 bits)
			PCI: R/W	Specifies the upper 12 bits of memory base address that corresponds the local address space 0 (SuperHyway bus address space of this LSI).
				Update value
				PCILSR [28:20] Address space Effective bit of MBA (upper)
				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	Memory Space 0 Base Address (lower 16 bits)
			PCI: R	These bits are fixed H'0000 by hardware.
3	LAP	0	SH: R	Prefetch Control
			PCI: R	Indicates whether or not local address space 0 is prefetchable.
				0: Not prefetchable
				1: Prefetchable (not supported)

Bit	Bit Name	Initial Value	R/W	Description
2, 1	LAT	00	SH: R	Memory Type
			PCI: R	Indicates the memory type of local address space 0.
				00: 32-bit base address and 32-bit space
				01: 32-bit base address and 1-Mbyte space (Not supported)
				10: 64-bit base address (Not supported)
				11: Reserved
0	ASI	0	SH: R	Address Space Indicator
			PCI: R	Indicates whether the base address in this register 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 register 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	17	16
	MBA (upper)												MBA (lower)			
Initial value:	0	0	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	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	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBA (lower)												LAP	LAT		ASI
Initial value:	0	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	R	R	R
PCI R/W:	R	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 20	MBA (upper)	H'000	SH: R/W PCI: R/W	PCI Memory Space 1 Base Address (upper 12 bits) Specifies the upper 12 bits of PCI memory base address that corresponds the base address of local address space 1 (SuperHyway bus address space of this LSI). PCILSE0 [28:20] — Address space — Effective bit of MBA (upper) 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 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 1 is 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 space 1. 00: 32-bit base address and 32-bit space 01: 32-bit base address and 1-Mbyte space (Not 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 register indicates the I/O or memory space. 0: Memory space 1: I/O space

(16) PCI Subsystem vender ID Register (PCISVID)

Refer to miscellaneous registers section of PCI local bus specification Revision 2.2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVID															
Initial value:	0	0	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/W	R/W
PCI R/W:	R	R	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	SVID	H'0000	SH: R/W	Subsystem Vendor ID
			PCI: R	Specifies the subsystem vendor ID of the PCIC. The initial value is H'0000.
				This field can be modified during initializing PCIC registers (PCICR.CFINIT = 0), but cannot be modified after initialized PCIC register (PCICR.CFINIT = 1) even if writing this field.

(17) PCI Subsystem ID Register (PCISID)

Refer to section about miscellaneous registers of PCI local bus specification Revision 2.2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSID															
Initial value:	0	0	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/W	R/W
PCI R/W:	R	R	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 initial value is H'0000.
				This field can be modified during initializing PCIC registers (PCICR.CFINIT = 0), but cannot be modified after initialized PCIC register (PCICR.CFINIT = 1) even if writing this field.

(18) PCI Capability Pointer Register (PCICP)

This register is the expansion function pointer register of the PCI configuration register that is prescribed in the PCI power management specification.

Bit:	7	6	5	4	3	2	1	0
	CP							
Initial value:	0	1	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	CP	H'40	SH: R	Capabilities pointer
			PCI: R	The offset address of the expansion function register.

(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 interrupt of this LSI. Specify these bits by system software during initialization. The initial value is H'00. The setting value of this field does not affect the operation of this LSI.

(20) PCI Interrupt Pin Register (PCIINTPIN)

Bit:	7	6	5	4	3	2	1	0
	INTPIN							
Initial value:	0	0	0	0	0	0	0	1
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	INTPIN	H'01	SH: R/W	Interrupt Pin Select
			PCI: R	Specifies which interrupt pin is used for connection 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 master device (not supported).

(22) PCI Maximum Latency Register (PCIMAXLAT)

This register is not programmable.

Bit:	7	6	5	4	3	2	1	0
	MAXLAT							
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	MAXLAT	H'00	SH: R	Maximum Latency
			PCI: R	Specify the worst time from the bus request by the 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 pointed to is the PCI power management data structure. Each function of a PCI device may have only one 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 management.

(24) PCI Next Item Pointer Register (PCINIP)

PCINIP gives the location of the next item in the function's capability list.

Bit:	7	6	5	4	3	2	1	0
	NIP							
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	NIP	H'00	SH: R	Next Item Pointer
			PCI: R	Specifies the offset to the next expansion function.
				H'00: Power management function is listed as the last item.

(25) PCI Power Management Capability Register (PCIPMC)

PCIPMCS is a 16-bit register that provides information on the capabilities of the power management related functions. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”. This register must be set during initializing the PCIC registers (PCICR.CFINIT = 0).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCS					D2S	D1S	—	—	—	DSI	—	PMEC	PMV		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
SH R/W:	R	R	R	R	R	R/W	R/W	R	R	R	R	R	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	R	R

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 which the function may assert $\overline{\text{PME}}$. A value of 0b for any bit indicates that the function is not capable of asserting 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 hot Bit15: 1xxxx - $\overline{\text{PME}}$ can be asserted from D3 cold 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 D2 power management state. When the D2 power management state is not supported, this bit is read as 0.
9	D1S	0	SH: R/W PCI: R	When this bit is 1, This function supports the D1 power management state. When the D1 power management state is not supported, this bit is read as 0.

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
5	DSI	0	SH: R	DSI
			PCI: R	Specifies whether or not the device requires the specific initialization.
				0: Does not require the specific initialization
4	—	0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3	PMEC	1	SH: R/W	PCI PME clock
			PCI: R	Specifies whether or not the device requires the clock to support $\overline{\text{PME}}$ generation.
				1: Requires the clock to support $\overline{\text{PME}}$ generation
				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 management specifications.
				010: This LSI's power management specification is conformed to revision 1.1

(26) PCI Power Management Control/Status Register (PCIPMCSR)

This 16-bit register is used to manage the PCI function's power management status as well as to enable/monitor PMEs. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMES	DSC		DSL				PME EN	—	—	—	—	—	—	PS	
Initial value:	0	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	R	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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 supported)
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 value 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, optional state to these bits, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.
				00: D0 state
				01: D1 state
				10: D2 state
				11: D3 hot state (power-down mode)

(27) PCIPMCSR Bridge Support Extension Register (PCIPMCSRBSE)

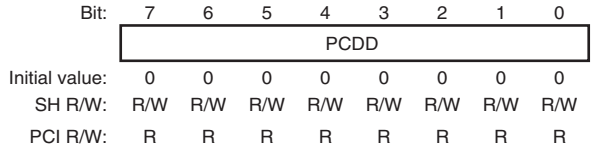
This register supports PCI bridge specific functionality and is required for all PCI-to-PCI bridges.

Bit:	7	6	5	4	3	2	1	0
	BPCCEN	B2B3N	—	—	—	—	—	—
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	BPCCEN	0	SH: R PCI: R	When the bus power/clock control mechanism is disabled, the power state bits in bridge's PCIPMCSR cannot be used by the system software to control the power or clock of the bridge's secondary bus.
6	B2B3N	0	SH: R PCI: R	The state of this bit determines the action that is to occur as a direct result of programming the function to the D3 hot state. 0: Indicates that when the bridge function is set to the D3 hot state, its secondary bus will have its power removed (B3). 1: Indicates that when the bridge function is set to the D3 hot state, its secondary bus's PCI clock will be stopped (B2). This bit is only valid if bit 7 (BPCCEN) is set to 1.
5 to 0	—	All 0	SH: R PCI: R	Reserved These bits are always read as 0. The write value should always be 0.

(28) PCI Power Consumption/Radiation Register (PCIPCDD)

The data register is an 8-bit register that provides a mechanism for the function to report state dependent operating data such as power consumed or heat dissipation. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”.



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 dependent data requested by the PCIPMCSR.DSL bits. The value of this register is scaled by the value reported by the PCIPMCSR.DSC bits.

13.3.3 Local Register

(1) PCI Control Register (PCICR)

PCICR is a 32-bit register which specifies the operation of the PCIC.

The register is write protected; only writes in which the upper eight bits (that is, bits 31 to 24) have the value H'A5 are performed. All other writes are ignored.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	—	—	—	PFCS	FTO	PFE	TBS	—	BMAM	—	—	SERR	IOCS	RST CTL	CFI NIT
Initial value:	0	0	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	R/W	R/W
PCI R/W:	R	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 24	—	H'00	SH: R/W	Reserved
			PCI: R	Set these bits to H'A5 only when writing to bits 11 to 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 value 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 5 cycles before disconnection.
				0: Disabled
				1: Enabled
9	PFE	0	SH: R/W	PCI Pre-Fetch Enable
			PCI: R	0: Disabled
				1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
8	TBS	0	SH: R/W	Byte Swap
			PCI: R	Specifies whether or not byte data is swapped when accessing to the PCI local bus. 0: No swap 1: Byte data is swapped For details, see section 13.4.3 (5), Endian or section 13.4.4 (6), Endian.
7	—	0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
6	BMAM	0	SH: R/W	Bus Master Arbitration
			PCI: R	Controls the PCI bus arbitration mode when the PCIC operates in host bus bridge mode. This bit is ignored when the PCIC operates in normal mode. 0: Fixed mode (PCIC > device0 > device1 > device2 > device3) 1: Pseudo round robin (the most recently granted device is assigned the lowest priority)
5, 4	—	Undefined	SH: R	Reserved
			PCI: R	These bits are always read as an undefined value. 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 bit is valid only in normal mode (do not use in host bus bridge mode). This bit is valid only when the SERRE bit in PCICMD is 1. 0: Makes $\overline{\text{SERR}}$ output high-impedance state (driven high by pull-up register) 1: Asserts $\overline{\text{SERR}}$ output during one PCICLK clock 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 bit is valid only in normal mode. 0: Makes $\overline{\text{INTA}}$ output high-impedance state (driven high by pull-up registor) 1: Asserts $\overline{\text{INTA}}$ output (low level output)

Bit	Bit Name	Initial Value	R/W	Description
1	RSTCTL	0	SH: R/W	$\overline{\text{PCIRESET}}$ Output
			PCI: R	Controls the $\overline{\text{PCIRESET}}$ output by software. This bit is valid when the PCIC operates in host bus bridge mode. 0: Negates $\overline{\text{PCIRESET}}$ output (high level output) 1: Asserts $\overline{\text{PCIRESET}}$ output (low level output) Note: The $\overline{\text{PCIRESET}}$ is also asserted during power-on reset.
0	CFINIT	0	SH: R/W	PCI Internal Register Initialize Control
			PCI: R	Set this bit to 1 after the initialization of the PCIC internal registers are completed. Setting this bit enables accesses from the PCI bus. During initialization in host bus bridge mode, the bus is not given to the device on the PCI bus. In normal mode, the PCIC returns RETRY when it is accessed from the 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	17	16
	—	—	—	LSR									—	—	—	—
Initial value:	0	0	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	R	R
PCI R/W:	R	R	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MBARE
Initial value:	0	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	R	R	R/W
PCI R/W:	R	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 value should always be 0.
28 to 20	LSR	0 0000 0000	SH: R/W	Size of Local Address Space 0 (9 bits)
			PCI: R	Specify the size of local address space 0 (SuperHyway bus address space of this LSI) in units of Mbyte. The value set in these bits must be the size minus 1 Mbytes. Setting all the bits to 0 ensures 1-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 value should always be 0.
0	MBARE	0	SH: R/W	PCI Memory Base Address Register 0 Enable
			PCI: R	The local address space 0 can be accessed by setting this bit to 1. 0: PCIMBAR0 disabled 1: PCIMBAR0 enabled

(3) PCI Local Space Register 1 (PCILSR1)

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	17	16
	—	—	—	LSR									—	—	—	—
Initial value:	0	0	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	R	R
PCI R/W:	R	R	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MBARE
Initial value:	0	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	R	R	R/W
PCI R/W:	R	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 value 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) in units of Mbyte. The value set in these bits must be the size minus 1 Mbytes. Setting all the bits to 0 ensures 1-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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	MBARE	0	SH: R/W	PCI Memory Base Address Register 1 Enable
			PCI: R	The local address space 1 can be accessed by setting this bit to 1.
				0: PCIMBAR1 disabled
				1: PCIMBAR1 enabled

(4) PCI Local Address Register 0 (PCILAR0)

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	17	16
	LAR												—	—	—	—
Initial value:	0	0	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	R	R
PCI R/W:	R	R	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	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	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	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 address space 0.
				The effective bits of LAR depend on the capacity of local 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 should always be 0.

(5) PCI Local Address Register 1 (PCILAR1)

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	17	16
	LAR												—	—	—	—
Initial value:	0	0	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	R	R
PCI R/W:	R	R	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	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	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	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 address space 1.
				The effective bits of LAR depend on the capacity of local address space 1 as specified in PCILSR1.
				The effective bits are as follows:
				PCILSR1.LS1([28:20]) = 0 0000 0000: Effective bits are [31:20]
				PCILSR1.LS1([28:20]) = 0 0000 0001: Effective bits are [31:21]
				PCILSR1.LS1([28:20]) = 0 0000 0011: Effective bits are [31:22]
				| |
				PCILSR0.LS1([28:20]) = 0 1111 1111: Effective bits are [31:28]
				PCILSR1.LS1([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 should always be 0.

(6) PCI Interrupt Register (PCIIR)

PCIIR records the source of an interrupt.

When multiple interrupts occur, only the first source is registered.

When an interrupt is disabled, the source is registered in corresponding bit (set to 1) in this register, however, no interrupt occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	TTADI	—	—	—	—	TMTOI	MDEI	APEDI	SEDI	DPEITW	DPEITR	TADIM	MADIM	MWPDI	MRDPEI
Initial value:	0	0	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	R/WC	R/WC
PCI R/W:	R	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 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
14	TTADI	0	SH: R/WC	Target Target-Abort Interrupt
			PCI: R	Indicates that the PCIC has terminated a transaction with a target-abort when the PCIC functions as a target. A target-abort is detected as an illegal byte enable when the lower two bits (bits 1 and 0) of the address and the byte enable do not match during an I/O 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.

Bit	Bit Name	Initial Value	R/W	Description
13 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
9	TMTOI	0	SH: R/WC	Target Memory Read Retry Timeout Interrupt
			PCI: R	When the PCIC functions as a target, the master did not attempt a retry within the prescribed number of PCICLK clocks (2^{15}) (detected only in the case of memory read operations). 0: Target memory read retry timeout interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target memory read retry timeout interrupt occurs [Set condition] When a target memory read retry timeout interrupt occurs.
8	MDEI	0	SH: R/WC	Master Function Disable Error Interrupt
			PCI: R	The PCIC attempted a master access when such accesses are disabled, that is, when PCICMD.BM is cleared to 0. 0: Master function disable error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master function disable error interrupt occurs [Set condition] When a master function disable error interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
7	APEDI	0	SH: R/WC	Address Parity Error Detection Interrupt
			PCI: R	Indicates an address parity error has been detected.
				When both the PER and SERRE bits in the PCI command register are set to 1, an address parity error is detected.
				0: Address parity error detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Address parity error detection interrupt occurs
				[Set condition]
				When an address parity error detection interrupt occurs.
6	SEDI	0	SH: R/WC	$\overline{\text{SERR}}$ Detection Interrupt
			PCI: R	Indicates that the assertion of the $\overline{\text{SERR}}$ signal has been detected when the PCIC operates in host bus 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	Data Parity Error Interrupt for Target Write
			PCI: R	Indicates that a data parity error has been detected during a target write access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a target.
				0: Data parity error detection interrupt does not occur
				[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 occurs.

Bit	Bit Name	Initial Value	R/W	Description
4	PEDITR	0	SH: R/WC	Data Parity Error Interrupt for Target $\overline{\text{PERR}}$
			PCI: R	Indicates that the $\overline{\text{PERR}}$ signal has been received during a target read access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a target.
				0: $\overline{\text{PERR}}$ detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: $\overline{\text{PERR}}$ 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 has detected a target-abort, that is, the transaction is 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 transaction with a master-abort when the PCIC functions as a 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.

Bit	Bit Name	Initial Value	R/W	Description
1	MWPDI	0	SH: R/WC	Master Write $\overline{\text{PERR}}$ Detection Interrupt
			PCI: R	Indicates that the $\overline{\text{PERR}}$ signal has been received during a master write access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a master. 0: Master write $\overline{\text{PERR}}$ interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master write $\overline{\text{PERR}}$ 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 detected during a master read access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a master. 0: Master read data perity error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master read data perity error interrupt occurs [Set condition] When a master read data perity error interrupt occurs.

(7) PCI Interrupt Mask Register (PCIIMR)

This register is the mask register for PCIIR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	TTA DIM	—	—	—	—	TMT OIM	MDE IM	APE DIM	SE DIM	DPEI TWM	DPEI TRM	TAD IMM	MAD IMM	MW PDIM	MRD PEIM
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	R/W	R/W
PCI R/W:	R	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 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value 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 value 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 Mask
			PCI: R	0: PCIIR.MDEI disabled (masked)
				1: PCIIR.MDEI enabled (not masked)
7	APEDIM	0	SH: R/W	Address Parity Error Detection Interrupt Mask
			PCI: R	0: PCIIR.APEDI disabled (masked)
				1: PCIIR.APEDI enabled (not masked)

Bit	Bit Name	Initial Value	R/W	Description
6	SEDIM	0	SH: R/W	$\overline{\text{SERR}}$ Detection Interrupt Mask
			PCI: R	0: PCIIR.SEDI disabled (masked)
				1: PCIIR.SEDI enabled (not masked)
5	DPEITWM	0	SH: R/W	Data Parity Error Interrupt Mask for Target Write
			PCI: R	0: PCIIR.DPEITW disabled (masked)
				1: PCIIR.DPEITW enabled (not masked)
4	PEDITRM	0	SH: R/W	$\overline{\text{PERR}}$ Detection Interrupt Mask for Target Read
			PCI: R	0: PCIIR.PEDITR disabled (masked)
				1: PCIIR.PEDITR enabled (not masked)
3	TADIMM	0	SH: R/W	Target-Abort Interrupt Mask for Master
			PCI: R	0: PCIIR.TADIM disabled (masked)
				1: PCIIR.TADIM enabled (not masked)
2	MADIMM	0	SH: R/W	Master-Abort Interrupt Mask for Master
			PCI: R	0: PCIIR.MADIM disabled (masked)
				1: PCIIR.MADIM enabled (not masked)
1	MWPDIM	0	SH: R/W	Master Write Data Parity Error Interrupt Mask
			PCI: R	0: PCIIR.MWPDI disabled (masked)
				1: PCIIR.MWPDI enabled (not masked)
0	MRDPEIM	0	SH: R/W	Master Read Data Parity Error Interrupt Mask
			PCI: R	0: PCIIR.MRDPEI disabled (masked)
				1: PCIIR.MRDPEI enabled (not masked)

(8) PCI Error Address Information Register (PCIAIR)

This register records PCI address information when an error is detected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AIL															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R	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	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AIL															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R	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	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 states of the AD signals) when an error occurs.

(9) PCI Error Command Information Register (PCICIR)

This register records the PCI command information when an error is detected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MTEM	—	—	—	—	RWTET	—	—	—	—	—	—	—	—	—	—
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	R	R
PCI R/W:	R	R	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ECL			
Initial value:	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	R	R
PCI R/W:	R	R	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 master 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 value should always be 0.
26	RWTET	Undefined	SH: R	Target Error
			PCI: R	Indicates that an error has occurred during a target 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 value should always be 0.
3 to 0	ECL	Undefined	SH: R	Command Log
			PCI: R	Hold PCI command information (the state of the CBE[3:0] signal) when an error occurs.

(10) PCI Arbiter Interrupt Register (PCIAINT)

In host bus bridge mode, this register records source of an interrupt. When multiple interrupts occur, only the first source is registered. When an interrupt is disabled, source is registered in corresponding bit (set to 1) in this register, however, no interrupt occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	—	MBI	TB TOI	MB TOI	—	—	—	—	—	—	—	TAI	MAI	RD PEI	WD PEI
Initial value:	0	0	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	R/WC	R/WC
PCI R/W:	R	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 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
13	MBI	0	SH: R/WC	Master-Broken Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{PCIFRAME}}$ signal is not asserted within 16 clock cycles, although the 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.

Bit	Bit Name	Initial Value	R/W	Description
12	TBTOI	0	SH: R/WC	Target Bus Time-Out Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{TRDY}}$ or $\overline{\text{STOP}}$ signal is not asserted within 16 clock cycles on the first data transfer. An interrupt is detected when the $\overline{\text{TRDY}}$ or $\overline{\text{STOP}}$ signal is not asserted within eight clock cycles during the data transfer subsequent to the 2nd. 0: Target bus time-out interrupt does not occur [Clear condition] 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	Master Bus Time-Out Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{IRDY}}$ signal is not asserted within 8 clock cycles. 0: Master bus time-out interrupt does not occur [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	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TAI	0	SH: R/WC	Target-Abort Interrupt
			PCI: R	Indicates that a transaction is terminated with a target-abort when a device other than the PCIC functions as a bus master. 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	MAI	0	SH: R/WC	Master-Abort Interrupt
			PCI: R	Indicates that a transaction is terminated with a master-abort when a device other than the PCIC 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 data read when a device other than the PCIC functions as a bus master. 0: Read parity error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Read parity error interrupt occurs [Set condition] When a read parity error interrupt is detected by the $\overline{\text{PERR}}$ assertion.

Bit	Bit Name	Initial Value	R/W	Description
0	WDPEI	0	SH: R/WC	Write Parity Error Interrupt
			PCI: R	The $\overline{\text{PERR}}$ assertion is detected during a data write when a device other than the PCIC functions as a bus master.
				0: Write parity error interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Write parity error interrupt occurs
				[Set condition]
				When a write parity error interrupt is detected by the $\overline{\text{PERR}}$ 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	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	—	MBIM	TBT OIM	MBT OIM	—	—	—	—	—	—	—	TAIM	MAIM	RDP EIM	WDP EIM
Initial value:	0	0	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	R/W	R/W
PCI R/W:	R	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 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13	MBIM	0	SH: R/WC	Master-Broken Interrupt Mask
			PCI: R	0: PCIAINT.MBI disabled (masked)
				1: PCIAINT.MBI enabled (not masked)
12	TBTOIM	0	SH: R/WC	Target Bus Time-Out Interrupt Mask
			PCI: R	0: PCIAINT.TBTOI disabled (masked)
				1: PCIAINT.TBTOI enabled (not masked)
11	MBTOIM	0	SH: R/WC	Master Bus Time-Out Interrupt Mask
			PCI: R	0: PCIAINT.MBTOI disabled (masked)
				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 value 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)

(12) PCI Arbiter Bus Master Information Register (PCIBMIR)

In host bridge mode, this register records when the interrupt is invoked by PCIAINT.

When multiple interrupts occur, only the first source is registered.

When an interrupt is masked, the source is registered in corresponding bit (set to 1), however, an interrupt occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	R	R	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	REQ4 BME	REQ3 BME	REQ2 BME	REQ1 BME	REQ0 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	R	R
PCI R/W:	R	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 5	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
4	REQ4BME	Undefined	SH: R	REQ4 Error
			PCI: R	An error occurs when the PCIC functions as a bus master.
3	REQ3BME	Undefined	SH: R	REQ3 Error
			PCI: R	An error occurs when device 3 ($\overline{\text{REQ3}}$) functions as a bus master
2	REQ2BME	Undefined	SH: R	REQ2 Error
			PCI: R	An error occurs when device 2 ($\overline{\text{REQ2}}$) functions as a bus master
1	REQ1BME	Undefined	SH: R	REQ1 Error
			PCI: R	An error occurs when device 1 ($\overline{\text{REQ1}}$) functions as a bus master
0	REQ0BME	Undefined	SH: R	REQ0 Error
			PCI: R	An error occurs when device 0 ($\overline{\text{REQ0}}$) functions as a bus master

(13) PCI PIO Address Register (PCIPAR)

This register is configuration address register.

Refer to Section 13.4.5 (2), Configuration Space Access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCIE	—	—	—	—	—	—	—	BN							
Initial value:	1	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DN					FN			CRA						—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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	R
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 generation enable
30 to 24	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
23 to 16	BN	Undefined	SH: R/W	PCI Bus Number
			PCI: —	Specify the PCI bus number for a configuration access. The PCIC is connected to bus number 0. Bus numbers ranging from 0 to 255 are represented in 8 bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	DN	Undefined	SH: R/W	Device Number
			PCI: —	Specify the device number for a configuration access. Device numbers ranging from 0 to 31 are represented in five bits. A single bit of bits 31 to 16 of the AD signals is driven to high level instead of the IDSEL assertion. The bit driven to high level corresponds to the device number set in these bits. The correspondence between the device number and IDSEL (AD[31:16]) is shown below. If a device number is equal to H'10 or greater, all bits 31 to 16 of the AD signals are driven to low level. Device No. IDSEL — Device No. IDSEL H'0: AD[16] = high level — H'8: AD[24] = high level H'1: AD[17] = high level — H'9: AD[25] = high level H'2: AD[18] = high level — H'A: AD[26] = high level H'3: AD[19] = high level — H'B: AD[27] = high level H'4: AD[20] = high level — H'C: AD[28] = high level H'5: AD[21] = high level — H'D: AD[29] = high level H'6: AD[22] = high level — H'E: AD[30] = high level H'7: AD[23] = high level — H'F: AD[31] = high level Other than above AD[31:16] lines are driven to high level.
10 to 8	FN	Undefined	SH: R/W	Function Number
			PCI: —	Specify the function number for a configuration access. The function numbers ranging from 0 to 7 are represented in three bits.
7 to 2	CRA	Undefined	SH: R/W	Configuration Register Address
			PCI: —	Specify the register for a configuration access at a longword boundary.
1, 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(14) PCI Power Management Interrupt Register (PCIPINT)

This register controls the power management interrupt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PMD 3H	PMD 2	PMD 1	PMD 0
Initial value:	0	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/WC	R/WC	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 value should always be 0.
3	PMD3H	0	SH: R/WC	PCI Power Management D3 Hot Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D3 is not detected
				1: Interrupt request for a transition to D3 is detected
2	PMD2	0	SH: R/WC	PCI Power Management D2 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D2 is not detected
				1: Interrupt request for a transition to D2 is detected
1	PMD1	0	SH: R/WC	PCI Power Management D1 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D1 is not detected
				1: Interrupt request for a transition to D1 is detected
0	PMD0	0	SH: R/WC	PCI Power Management D0 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D0 is not detected
				1: Interrupt request for a transition to D0 is detected

(15) PCI Power Management Interrupt Mask Register (PCIPINTM)

This is the mask register for PCIPINT.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PMD 3HM	PMD 2M	PMD 1M	PMD 0M
Initial value:	0	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/W	R/W
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 value should always be 0.
3	PMD3HM	0	SH: R/W PCI: —	PCI Power Management D3 Hot Status Transition 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)

(16) PCI Memory Bank Register 0 (PCIMBR0)

This register specifies the upper 14-bit address of the PCI memory space 0 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA0														—	—
Initial value:	0	0	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	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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: —	PCI Memory Space 0 Bank Address Specify the bank address in PCI memory space 0 for a master access.
17 to 0	—	All 0	SH: R PCI: —	Reserved These bits are always read as 0. The write value should always be 0.

(17) PCI Memory Bank Mask Register 0 (PCIMBMR0)

This register specifies the size of the PCI memory space 0.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MSBAM0						—	—
Initial value:	0	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/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 value 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 value should always be 0.

(18) PCI Memory Bank Register 1 (PCIMBR1)

This register specifies the upper 14-bit address of the PCI memory space 1 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA1														—	—
Initial value:	0	0	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	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 space 1 for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(19) PCI Memory Bank Mask Register 1 (PCIMBMR1)

This register specifies the size of the PCI memory space 1.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MSBAM1								—	—
Initial value:	0	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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 value should always be 0.
25 to 18	MSBAM1	All 0	SH: R/W	PCI Memory Space 1 Bank Address Mask (8 bits)
			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 value should always be 0.

(20) PCI Memory Bank Register 2 (PCIMBR2)

This register specifies the upper 14-bit address of the PCI memory space 2 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA2														—	—
Initial value:	0	0	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	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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: —	PCI Memory Space 2 Bank Address Specify the bank address in PCI memory space 2 for a master access.
17 to 0	—	All 0	SH: R PCI: —	Reserved These bits are always read as 0. The write value should always be 0.

(21) PCI Memory Bank Mask Register 2 (PCIMBMR2)

This register specifies the size of the PCI memory space 2.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	MSBAM2											—	—
Initial value:	0	0	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/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 value 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 value should always be 0.

(22) PCI I/O Bank Register (PCIIOBR)

This register specifies the upper 14-bit address of the PCI I/O space (address bits 31 to 18).

Refer to Section 13.4.3 (3), Accessing PCI I/O Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIOSBA														—	—
Initial value:	0	0	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	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(23) PCI I/O Bank Mask Register (PCIIOBMR)

This register specifies the size of the PCI I/O space.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	IOBAMR			—	—
Initial value:	0	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/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	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	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 value 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 value should always be 0.

(24) PCI Cache Snoop Control Register 0 (PCICSCR0)

An external device can access local memory of this LSI via the PCIC. When an external PCI device accesses cacheable areas of this LSI, the PCIC can support cache snoop function to the on-chip caches. The PCICSCR0 specifies this function that uses cache snoop address registers 0.

Refer to section 13.4.4 (7), Cache Coherency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RANGE			SNPMD	
Initial value:	0	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/W	R/W	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 value 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 be compared.
				000: PCICSAR[n].CADR[31:12] compared (4 Kbytes)
				001: PCICSAR[n].CADR[31:16] compared (64 Kbytes)
				010: PCICSAR[n].CADR[31:20] compared (1 Mbyte)
				011: PCICSAR[n].CADR[31:24] compared (16 Mbytes)
				100: PCICSAR[n].CADR[31:25] compared (32 Mbytes)
				101: PCICSAR[n].CADR[31:26] compared (64 Mbytes)
				110: PCICSAR[n].CADR[31:27] compared (128 Mbytes)
				111: PCICSAR[n].CADR[31:28] compared (256 Mbytes)
				Valid only when PCICSCR0.SNPMD = 10 or 11.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SNPMD	All 0	SH: R/W	Snoop Mode for PCICSAR0
			PCI: —	Specify if PCICSAR0 is compared with address requested by an external device. Also, specify how snoop function is executed when PCICSAR0 is compared.
				00: PCICSAR0 not compared
				01: Reserved (setting prohibited)
				10: PCICSAR0 compared. If hit, snoop function is not executed, otherwise executed.
				11: PCICSAR0 compared. If hit, snoop function is executed, otherwise not executed.

(25) PCI Cache Snoop Control Register 1 (PCICSCR1)

An external device can access local memory of this LSI via the PCIC. When an external PCI device accesses cacheable areas of this LSI, the PCIC can support cache snoop function to the on-chip caches. The PCICSCR1 specifies this function that uses cache snoop address registers 1.

Refer to section 13.4.4 (7), Cache Coherency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	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	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RANGE			SNPMD	
Initial value:	0	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/W	R/W	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 value 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 be compared.
				000: PCICSAR[n].CADR[31:12] compared (4 Kbytes)
				001: PCICSAR[n].CADR[31:16] compared (64 Kbytes)
				010: PCICSAR[n].CADR[31:20] compared (1 Mbyte)
				011: PCICSAR[n].CADR[31:24] compared (16 Mbytes)
				100: PCICSAR[n].CADR[31:25] compared (32 Mbytes)
				101: PCICSAR[n].CADR[31:26] compared (64 Mbytes)
				110: PCICSAR[n].CADR[31:27] compared (128 Mbytes)
				111: PCICSAR[n].CADR[31:28] compared (256 Mbytes)
				Valid only when PCICSCR1.SNPMD = 10 or 11.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SNPMD	All 0	SH: R/W	Snoop Mode for PCICSAR1
			PCI: —	Specify if PCICSAR1 is compared with address requested by an external device. Also, specify how snoop function is executed when PCICSAR1 is compared. 00: PCICSAR1 not compared 01: Reserved (setting prohibited) 10: PCICSAR1 compared. If hit, snoop function is not executed, otherwise executed. 11: PCICSAR1 compared. If hit, snoop function is executed, otherwise not executed.

(26) PCI Cache Snoop Address Register 0 (PCICSAR0)

PCICSAR0 specifies the address to be compared with the PCI address requested by an external device.

Refer to section 13.4.4 (7), Cache Coherency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CADR															
Initial value:	0	0	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/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CADR															
Initial value:	0	0	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/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 PCI address requested by external PCI devices

(27) PCI Cache Snoop Address Register 1 (PCICSAR1)

PCICSAR1 specifies the address to be compared with the PCI address requested by an external device.

Refer to section 13.4.4 (7), Cache Coherency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CADR															
Initial value:	0	0	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/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CADR															
Initial value:	0	0	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/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 PCI address requested by external PCI devices

(28) PCI PIO Data Register (PCIPDR)

When accessed, this register will cause the generation of a configuration cycle on the PCI bus.

Refer to section 13.4.5 (2), Configuration Space Access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PDR															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDR															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R	R	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 cause a PCI configuration cycle on the PCI bus.

13.4 Operation

13.4.1 Supported PCI Commands

Table 13.4 Supported Bus Commands

CBE[3:0]	Command Type	PCI Master	PCI Target
0000	Interrupt acknowledge cycle	No	—
0001	Special cycle	Yes*[1]	—
0010	I/O read	Yes	Yes*[2]
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 yes*[3]
1101	Dual address cycle	No	No
1110	Memory read line	No	Partially yes*[3]
1111	Memory write and invalidate	No	Partially yes*[4]

[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.

13.4.2 PCIC Initialization

After a power-on reset, the PCIC enable bit (ENBL) of the PCIC enable control register (PCIECR) and the internal register initialization bit (CFINIT) of the PCI control register (PCICR) is cleared. At this point, if the PCIC is operating as the PCI bus host (host bus bridge mode), the bus privileges are permanently granted to the PCIC, and no device arbitration is performed on the PCI bus. When the PCIC is not operating as host (normal mode), retries are returned without accepting access from PCI external devices connected to the PCI bus. In addition, all accesses to the PCIC from the CPU are invalid except the access to the PCIECR if the PCIECR.ENBL is cleared to 0. A write access is invalid and a read access will read 0, none of the registers can be modified, and any 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 internal configuration registers and local registers must be initialized before setting the CFINIT bit in the PCICR to 1 (while the CFINIT bit is cleared to 0). On completion of initialization, set the CFINIT bit to 1. When operating as host, arbitration is enabled; when operating as non-host, the PCIC can be accessed from the PCI bus.

Regardless of whether the PCIC is operating as the host or normal, external PCI devices cannot be accessed from the PCIC while the CFINIT bit is being cleared. Set the CFINIT bit to 1 before accessing an external PCIC device.

Be sure to initialize the following registers while the CFINIT bit is being cleared (before setting to 1): PCI command (PCICMD), PCI status (PCISTATUS), PCI sub system vender ID (PCISVID), PCI subsystem ID (PCISID), PCI local space register 0/1 (PCILSR 0/1) and PCI local address register 0/1.

13.4.3 Master Access

This section describes how the PCIC is accessed by software in this LSI and the restrictions on usage, such as buffering and synchronization with other devices, when the PCIC is used in both the host bus bridge and normal modes.

(1) Address Space of PCIC

Table 13.5 shows the PCIC address map.

Table 13.5 PCIC Address Map

	Physical Address		
Memory Area	**29-Bit Address Mode**	**32-Bit Address Extended Mode***	**Space Size**
PCI memory space1 (Area 4)	H'1000 0000 to H'13FF FFFF	H'1000 0000 to H'13FF FFFF	64 Mbytes
PCI memory space 2 (Only 32-bit address extended mode)	—	H'C000 0000 to H'DFFF FFFF	512 Mbytes
PCI memory space 0	H'FD00 0000 to H'FDFF FFFF	H'FD00 0000 to H'FDFF FFFF	16 Mbytes
Control register	H'FE00 0000 to H'FE03 FFFF	H'FE00 0000 to H'FE03 FFFF	256 Kbytes
PCIC internal register (configuration and local registers)	H'FE04 0000 to H'FE07 FFFF	H'FE04 0000 to H'FE07 FFFF	256 Kbytes
Reserved	H'FE08 0000 to H'FE1F FFFF	H'FE08 0000 to H'FE1F FFFF	1.5 Mbytes
PCI I/O space	H'FE20 0000 to H'FE3F FFFF	H'FE20 0000 to H'FE3F FFFF	2 Mbytes

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): the control register space (PCIECR), PCI internal control register (PCI configuration and PCI local registers) space, I/O space, and PCI memory (PCI memory space 0, PCI memory space 1, and PCI memory space 2).

(2) Accessing PCI Memory Space

Figure 13.2 shows the method for accessing the PCI bus allocated to the PCI memory space from the SuperHyway bus.

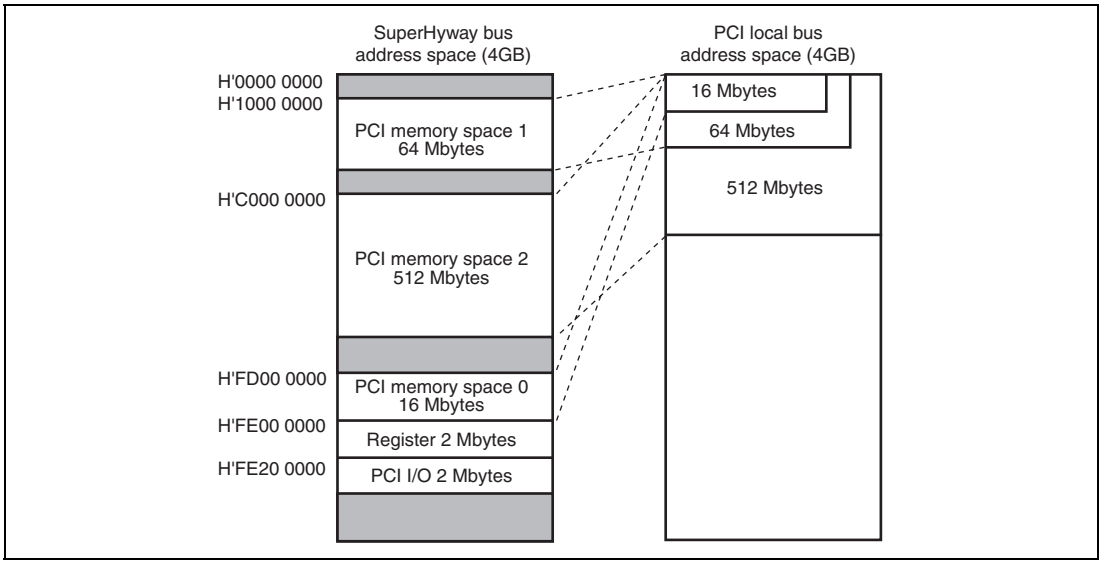


Figure 13.2 SuperHyway Bus to PCI Local Bus Access

To access to the PCI memory address space, use the PCI memory bank register (PCIMBR) and PCI memory bank mask register (PCIMBMR). These registers should have an address space 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 command 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 memory space 0 (16 Mbytes), H'1000 0000 to H'13FF FFFF for PCI memory space 1 (Area 4, 64 Mbytes, selection of the PCIC, DDRIF and LBSC spaces), and H'C000 0000 to H'DFFF FFFF for PCI 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.

For PCI memory space 0 accesses, bits 23 to 18 of a SuperHyway bus address are controlled by PCI memory bank mask register 0 (PCIMBMR0).

Note: In the following items and figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus.

- PCIMBMR0 [23:18] B'1111 11: PCI address [23:18] = SH address [23:18]
- PCIMBMR0 [23:18] B'0111 11: PCI address [23:18] = PCIMBR0 [23], SH address [22:18]

≀

- PCIMBMR0 [23:18] B'0000 01: PCI address [23:18] = PCIMBR0 [23:19], SH address [18]
- 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 24 in PCI memory bank register 0 (PCIMBR0).

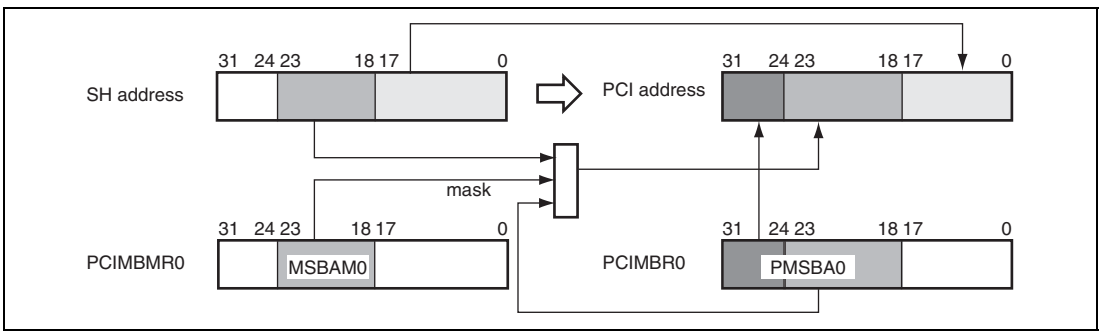


Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 0)

For PCI memory space 1 accesses, bits 25 to 18 of a SuperHyway address are controlled by PCI 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 address [24:18]

≀

- PCIMBMR1 [25:18] B'00 0000 01: PCI address [25:18] = PCIMBR1 [25:19], SH address [18]
- 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 in PCI memory bank register 1 (PCIMBR1).

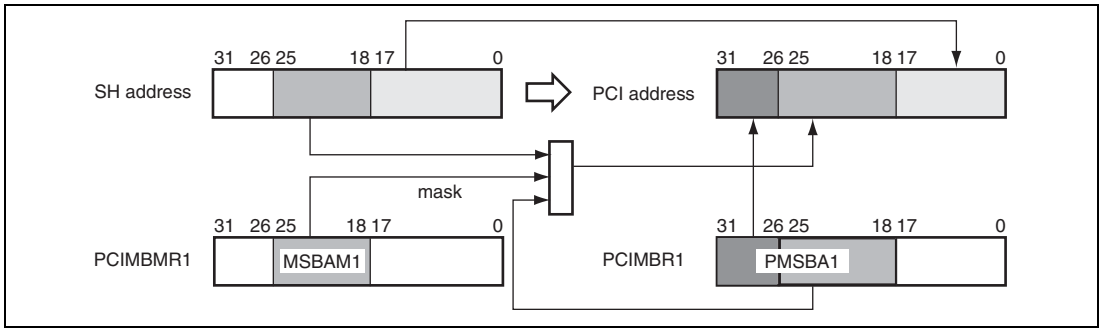


Figure 13.4 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 1)

For PCI memory space 2 accesses, bits 28 to 18 of a SuperHyway address are controlled by the 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 address [27:18]

≀

- PCIMBMR2 [28:18] B'0 0000 0000 01: PCI address [28:18] = PCIMBR2 [28:19], SH address [18]
- 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 29 in PCI memory bank register 2 (PCIMBR2).

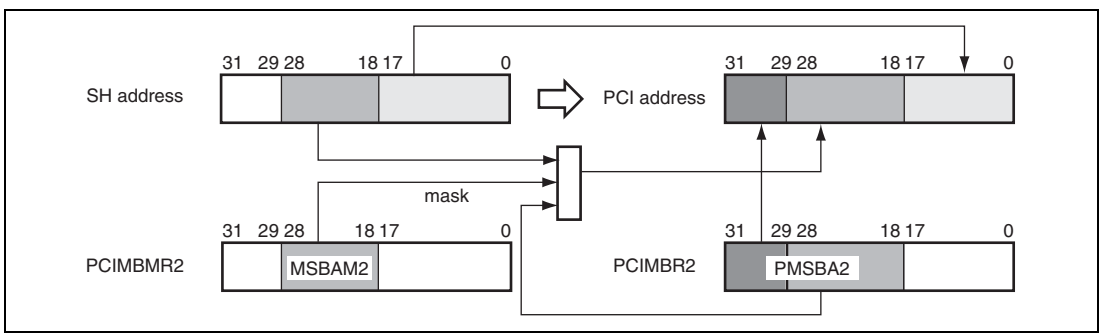


Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 2)

(3) Accessing PCI I/O Space

Access within the size of 4-byte.

Burst I/O transfers are not supported.

The PCI I/O address space is allocated from H'FD20 0000 to H'FE3F FFFF (2 Mbytes).

Address translation from SuperHyway bus to PCI local bus

The lower 15 bits ([17:3]) of a SuperHyway bus address are sent without translation.

Bits 20 to 18 of a SuperHyway bus address are controlled by the PCI I/O bank mask register (PCIIOBMR).

Note: In the following item and figure, “SH” means the SuperHyway bus of this LSI and “PCI” 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 in the PCI I/O bank register (PCIIOBR).

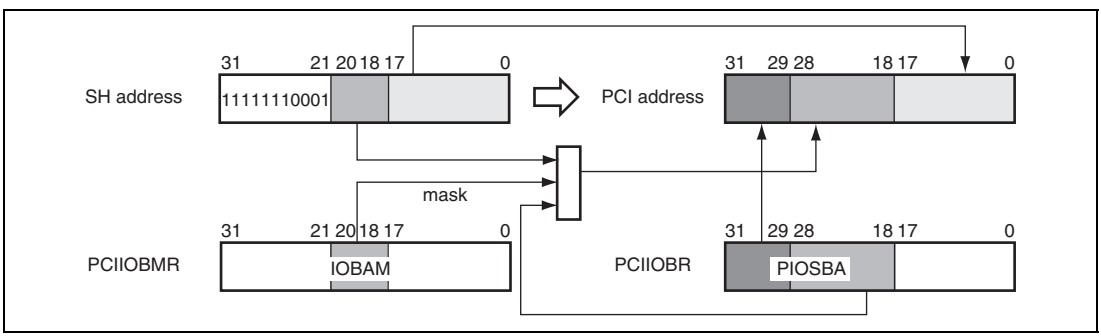


Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/O)

(4) Accessing Internal Registers of this LSI

All internal registers, that is, PCIECR, PCI configuration registers, and PCI local registers are accessible from the CPU.

4-byte, 2-byte, and byte transmission are supported.

(5) Endian

The PCIC of this LSI supports both the big endian and little endian formats. Since PCI local bus is inherently little 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 (PCICR) at a reset.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus. “MSByte” means the most significant byte and “LSByte” means the least significant byte.

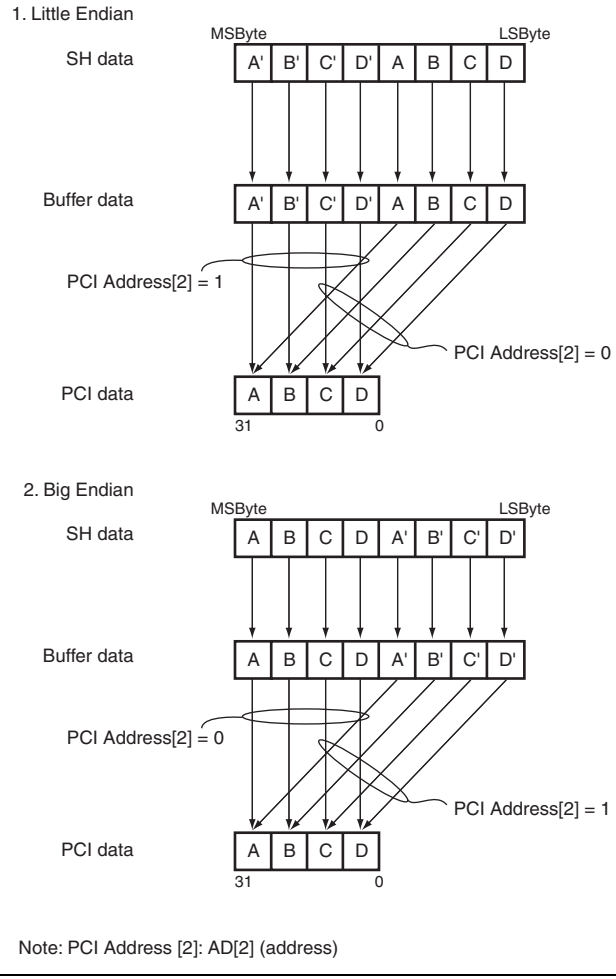


Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus (Non-Byte Swapping: TBS = 0)

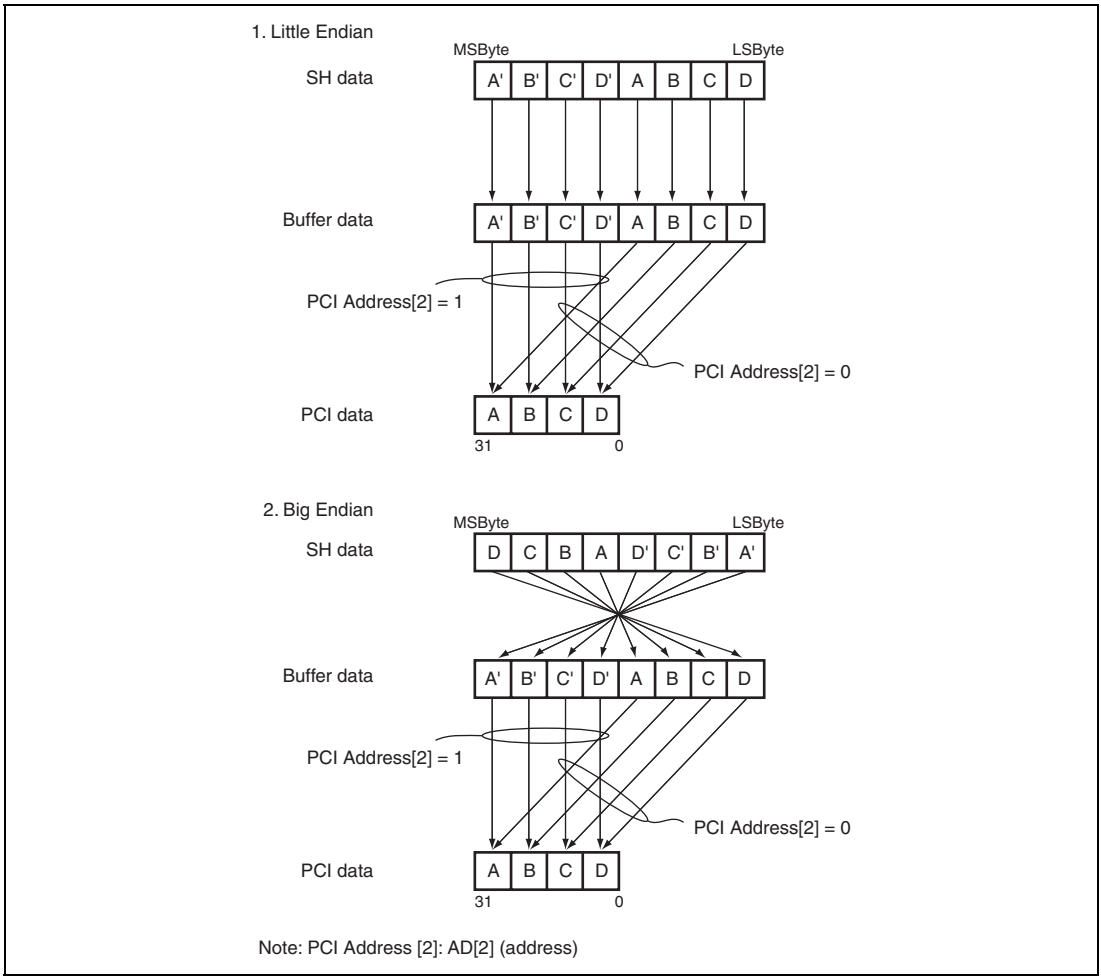


Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus (Byte Swapping: TBS = 1)

13.4.4 Target Access

This section describes how the PCIC of this LSI is accessed by an external PCI local bus master when the PCIC is used in both the host bus bridge and normal modes.

(1) Accessing This LSI Address Space

Accesses to the address space of this LSI by an external PCI bus master are described here.

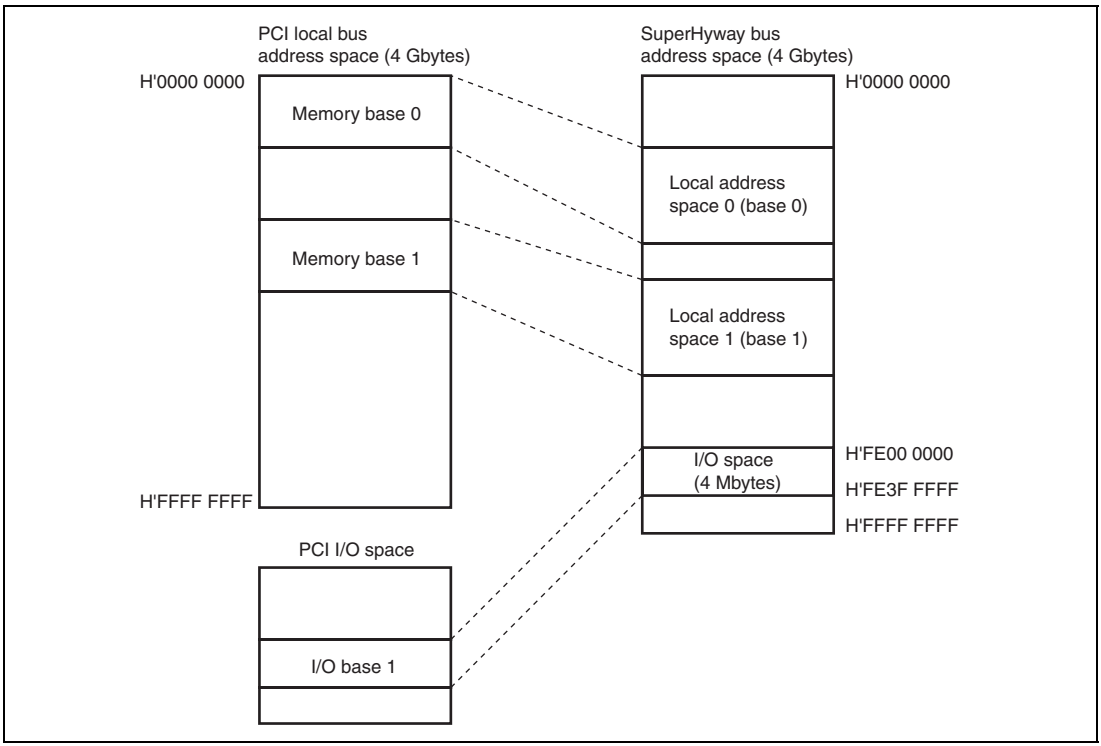


Figure 13.9 PCI local bus to SuperHyway bus Memory Map

To access the address space of this LSI, use the PCI memory base address register (PCIMBAR0/1), PCI local space register (PCILSR0/1), and PCI local address register (PCILAR0/1). The address spaces are mapped by software. The PCIC includes two memory mapping registers.

Setting these two registers enables the use of two spaces.

The size of these address spaces are selectable from 1 Mbyte to 512 Mbytes by setting the PCI local space register (PCILSR0/1).

Single longword and burst transfers are supported for the memory data transfer to a PCI target.

A certain range of the address space on the PCI local bus corresponds to the local address space on the SuperHyway bus. The local address space 0 is controlled by the PCIMBAR0, PCILSR0 and PCILAR0. And the local address space 1 is controlled by the PCIMBAR1, PCILSR1 and 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 device. The PCILAR0/1 specifies the starting address of the local address space 0/1. The PCILSR0/1 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 than the memory size set in the PCILSR0/1 becomes valid. The more significant address bits of the PCIMBAR0/1 and the same field line bits of the PCI local bus address output from an external PCI device are compared for the purpose of determining whether the access is made to the PCIC. When the addresses correspond, the access to the PCIC is recognized, and a local address is generated from the more significant address bits of the PCILAR0/1 and the less significant bits of the PCI local bus address output from the external PCI device. The PCI command is executed for this local address.

If the more significant address bits of the PCI local bus address output from the external PCI device does not correspond with the more significant address bits of the PCIMBAR0/1, the PCIC does not respond to the PCI command.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus.

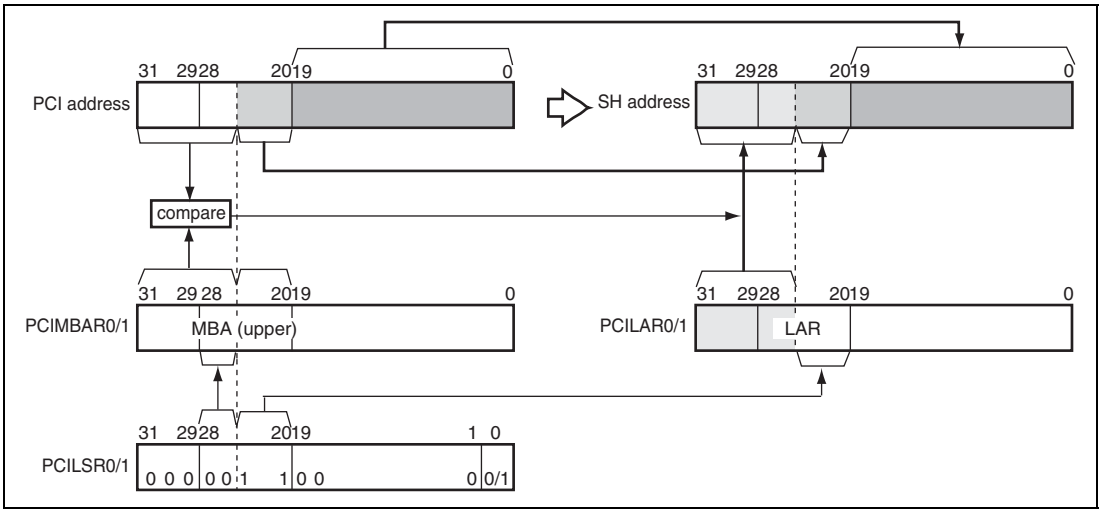


Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation (Local Address Space 0/1)

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 performed, 8 bytes, or 32 bytes of data block is prefetched. (this depends on the settings of the PFE and PFCS 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 register (PCIIBAR), the upper 24 bits of a PCI local bus address are replaced with H'FE04 01.

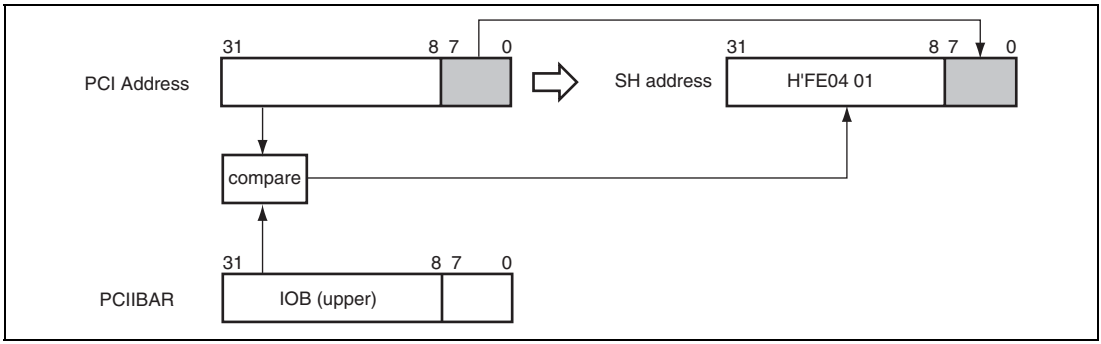


Figure 13.11 PCI Local Bus to SuperHyway Bus Address Translation (PCIC I/O Space)

(3) Accessing PCIC Registers

Configuration Registers: Access the configuration registers using an offset from the PCI configuration register space base address with the configuration read or write command. Only a single access which size should be under longword is performed. If a burst transfer is attempted, it is terminated to end the transaction.

Local Registers: Access the local registers using an offset from a PCI local register space base address with the I/O read or I/O write command. Only a single longword access is performed. If a 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 local bus.

(4) Access to this LSI Address Space

Memory Space: Refer to Section 13.4.4 (1), Accessing This LSI Address Space. Area 0 to area 2 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 read or memory write command via PCI local bus. The operation of this read/write is not guaranteed.

(5) Exclusive Access

The lock access on the PCI bus is supported.

When the PCI local bus is locked, the PCIC is accessible from the device that activates the LOCK signal.

SuperHyway bus resource lock does not occur. (Another on-chip module can access the PCIC during a lock transfer.)

(6) Endian

This LSI supports both the big and little endian formats. Since the PCI local bus is inherently little 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 (PCICR).

Note: In the following figures, “MSByte” means the most significant byte and “LSByte” means the least significant byte.

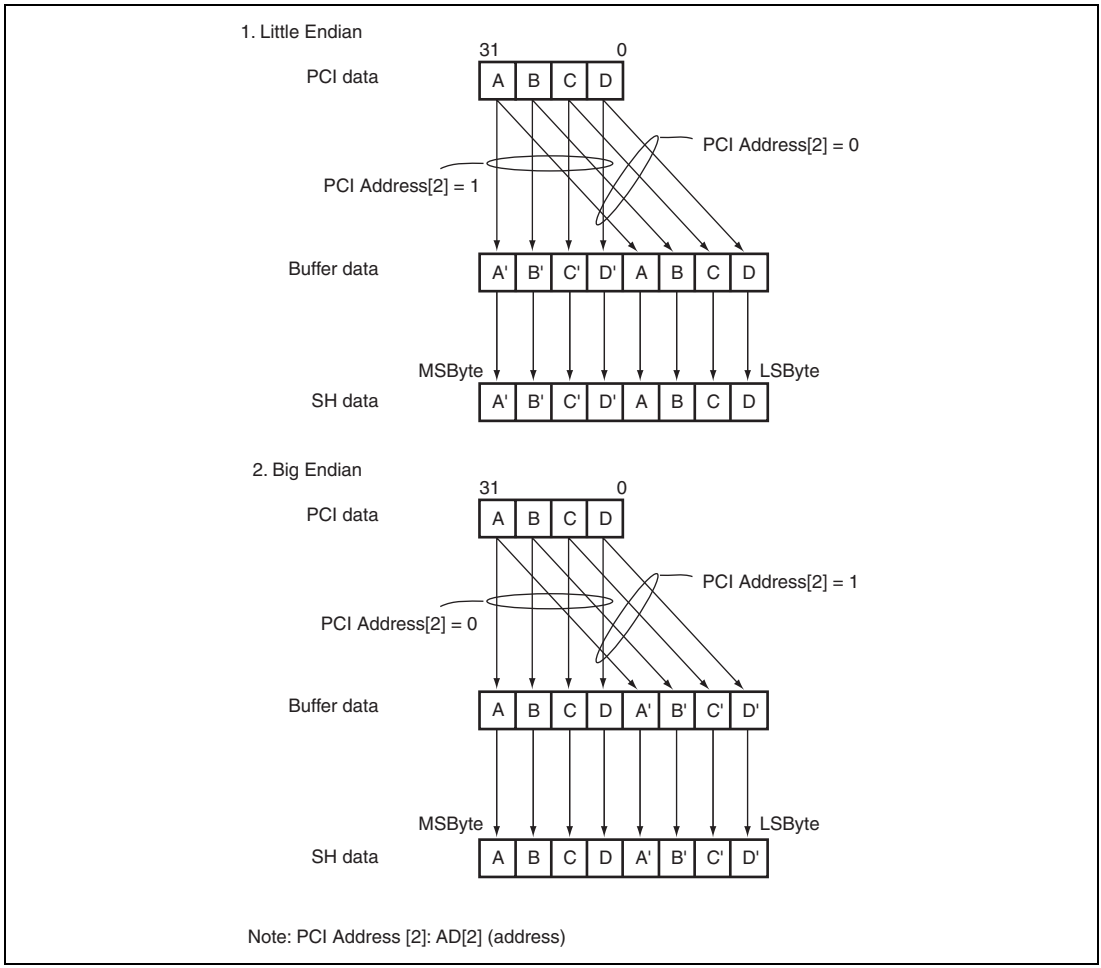


Figure 13.12 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 0)

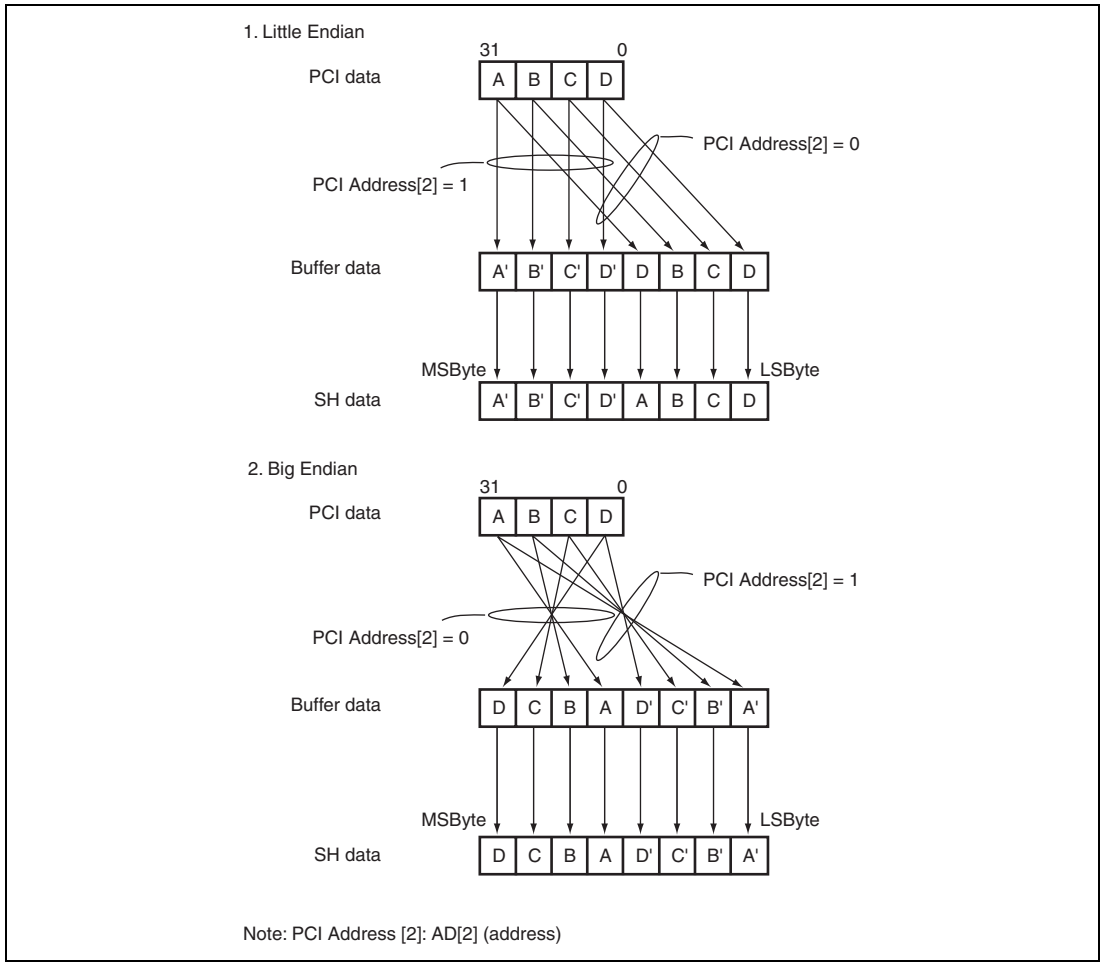


Figure 13.13 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 1)

(7) Cache Coherency

The PCIC supports cache snoop function.

When the PCIC functions as a target device, cache coherency is guaranteed for accesses from a master device connected to a PCI bus in both the host bus bridge mode and normal mode.

When accessing this LSI cacheable area, set the cache snoop registers: the PCI cache snoop control registers (PCICSCR0 and PCICSCR1) and PCI cache snoop address register (PCICSAR0 and PCICSAR1).

Usage Notes

- Up to 2 conditions can be set as snoop address. Address comparison is logical OR of setting 2 conditions.
- When using this function, execute memory read or write after flush/purge request issued to the CPU cache in the access of cache hit. It reduces PCI bus transfer speed and CPU performance.
- 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 state, it becomes an error access on the SuperHyway bus, and memory read or memory write does not execute. Specify the SNPMD bit in the PCICSCR to 00 before the CPU enters sleep mode. To keep the coherency before and after the CPU sleep, cache purge should be executed before sleep instruction executed.
- Do not use ether of the following functions and the cache shoop function simultaneously.
  - Debug function using an emulator (Disable this function when using an emulator).
  - L memory or memory mapped cache access from the DMAC.

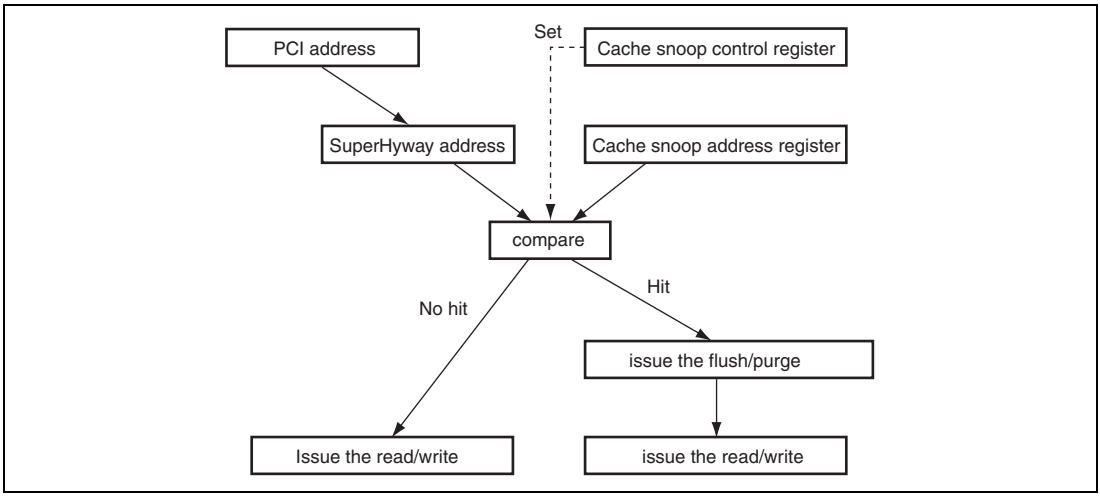


Figure 13.14 Cache Flush/Purge Execution Flow for PCI local Bus to SuperHyway Bus

13.4.5 Host Bus Bridge Mode

(1) PCI Host bus bridge Mode Operation

The PCIC supports a subset of the PCI Local Bus Specification Revision 2.2 and can be connected to a device with a PCI bus interface.

While the PCIC is set in host bus bridge mode, or while set in normal mode, operation differs according to whether or not bus parking is performed, and whether or not the PCI bus arbiter function is enabled or not.

In host bus bridge mode, the AD, CBE, PAR signal lines are driven by the PCIC when transfers are not being performed on the PCI bus. When the PCIC subsequently starts transfers as master, 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. Here, pins $\overline{\text{REQ0}}$/REQOUT, $\overline{\text{REQ1}}$, $\overline{\text{REQ2}}$, and $\overline{\text{REQ3}}$ function as the REQ inputs from the external masters 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 to 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 (PCIPAR) and PCI PIO data register (PCIPDR) correspond to the configuration address register and configuration data register, respectively.

When PCIPDR is read from or written to after PCIPAR has been set, a configuration cycle is issued on a PCI bus.

For a type 0 transfer, bits 10 to 2 of the configuration address register are sent without translation 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 number to 0.

Bit 17 of the AD signal is driven to 1 and the other bits are made 0 by setting the device number to 1. Similarly, setting the device number to 2 drives bit 18 of the AD signal to 1 and setting the 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 number to 16.

For details, refer to "PCI Local Bus Specification Revision 2.2, section 3.2.2.3 Configuration Space Decoding".

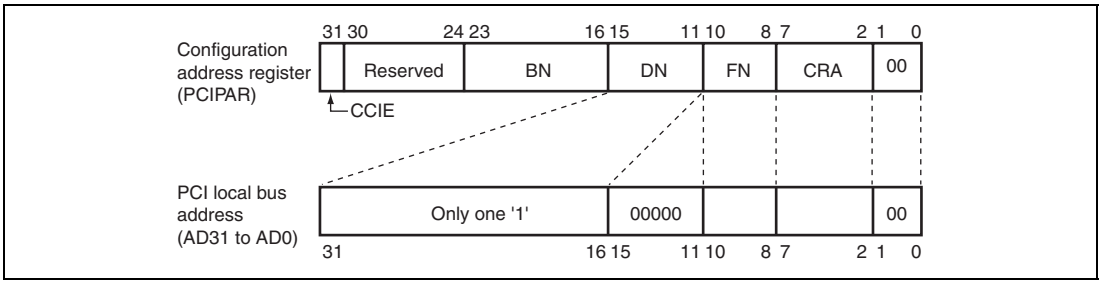


Figure 13.15 Address Generation for Type 0 Configuration Access

In configuration accesses, a PCI master abort (no device connected) will not cause an interrupt.

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 FF00 in 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 to the device with the highest priority.

The PCI bus arbiter supports two modes to determine the priority of devices: fixed priority and 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 are fixed 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 granted device is assigned the lowest priority.

The initial priority is the same as the fixed priority mode.

After device 1 has claimed and granted the bus, and transferred data, the priority is as follows:

PCIC > device 0 > device 2 > device 3 > device 1

Then, after the PCIC has claimed and granted the bus, and transferred data, the priority is changed to:

Device 0 > device 2 > device 3 > device 1 > PCIC

After device 3 has claimed and granted the bus, and transferred data, the priority is changed to:

Device 0 > device 2 > device 1 > PCIC > device 3

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 this LSI
- When the PCIC operates normal mode, $\overline{\text{INTA}}$ output is available to the host device on the PCI 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	Priority
PCISERR	SERR assertion detected in host bus bridge mode	High
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.	Low

The PCIC can store the error information on the PCI bus. If an error occurs, the error address is stored in the PCI error address information register (PCIAIR), the types of transfer and command information are stored in the PCI error command information register. And then if the PCIC operates host bus bridge mode, the bus master information is stored in the PCI error bus master information register.

Error information is stored only one information. This causes only to store the first occurred error information, and not to store after second error information. The error information is initialized by a power-on reset.

13.4.6 Normal mode

When operating in normal mode, the PCI bus arbitration function in the PCIC is disabled and PCI bus arbitration is performed according to the specifications of the externally connected PCI bus arbiter.

In normal mode, the master performs bus parking is decided by the grant signal that asserted from the external bus arbiter. If the master that performing bus parking is different from the next transaction master, the bus will be high-impedance state for minimum one clock cycle before the address phase.

In normal mode, the $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$ pin is used for the grant input signal to the PCIC, and the $\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$ 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 below.

- 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 capabilities pointer register shows the address offset of the configuration registers for power management. In the PCIC, this offset is fixed at CP = H'40. PCI capability ID (PCICID), next item pointer (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 four states: power state D0 (normal) power state D1 (bus idle) power state D2 (clock stop) and power state D3 (power down mode).

Figure 13.16 shows the PCI local bus power down state transition.

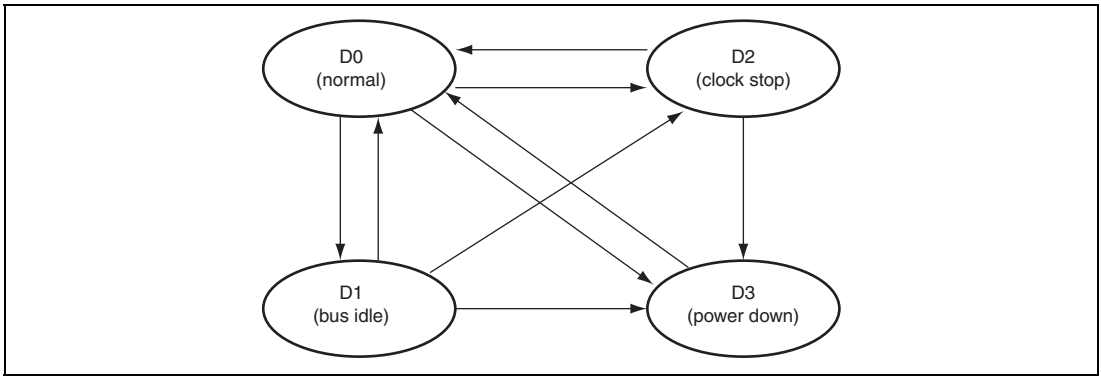


Figure 13.16 PCI Local Bus Power Down State Transition

The PCIC detects when the power state (PS) bit of the PCI power management control/status 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 power management interrupt register (PCIPINT) and PCI power management interrupt mask register (PCIPINTM). Of the power management interrupts, the power state D0 interrupt (PCIPWD0) detects a transition from the power state D1/D2/D3 to D0, while power state D1 interrupt (PCIPWD1) detects a transition from the power state D0 to D1, while power state D2 interrupt (PCIPWD2) detects a transition from the power state D0/D1 to D2, while power state D3 interrupt (PCIPWD3) detects a transition from the power state D0/D1/D2 to D3. Interrupt masks can be set for each interrupt.

No power state D0 interrupt is generated at a power-on reset.

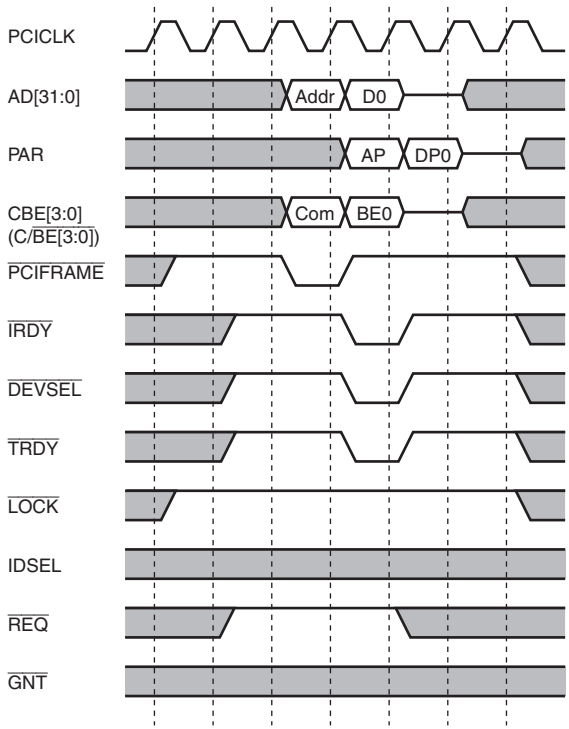
The following cautions should be noted when the PCIC is operating in normal mode and a power down interrupt is received from the host: In PCI power management, the PCI local bus clock stops within a minimum of 16 clocks after the host device has instructed a transition to power state D3. After detecting a power state D3 interrupt, do not, therefore, attempt to read or write to local registers and configuration registers that can be accessed from the SuperHyway bus and PCI local bus access (I/O and memory spaces). Because these accesses operate using the PCI local bus clock, the cycle for these accesses will not be completed if the clock stops and may be hung-up on 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 stipulations and can be connected to a device with a PCI local bus interface. The following figures show the timing for each operation mode.

(1) Master Read/Write Cycle Timing

Figures13.17 is an example of a single-write cycle in host bus bridge mode. Figure 13.18 is an example of a single read cycle in host bus bridge mode. Figure 13.19 is an example of a burst write cycle in normal mode. And Figure 13.20 is an example of a burst read cycle in normal mode. Note that the response speed of $\overline{\text{DEVSEL}}$ and $\overline{\text{TRDY}}$ differs according to the connected target device. In host bus bridge mode, master accesses always use single read/write cycles. The issuing of configuration transfers is only possible in host bus bridge mode.



[Legend]
Addr: PCI space address
AP: Address parity
Com: Command
Dn: nth data
DPn: nth data parity
BEn: nth data byte enable

Figure 13.17 Master Write Cycle in Host Bus Bridge Mode (Single)

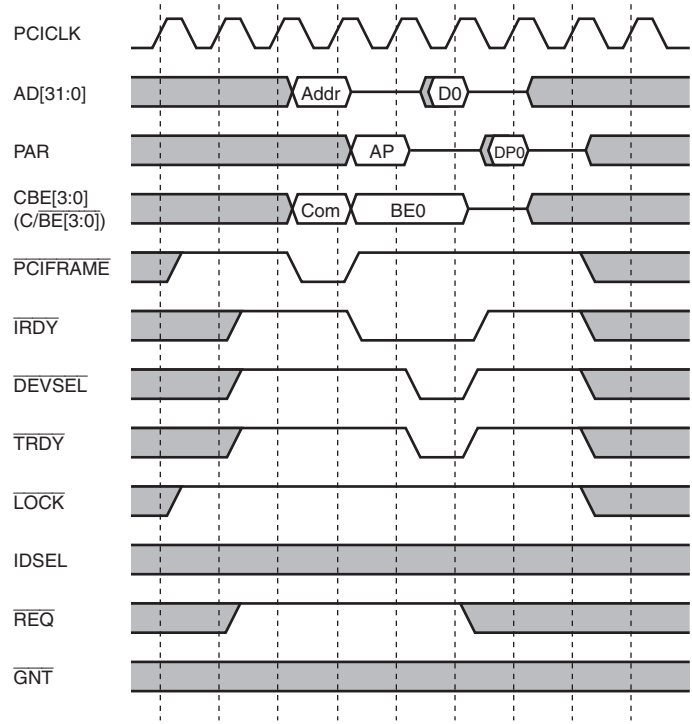


Figure 13.18 Master Read Cycle in Host Bus Bridge Mode (Single)

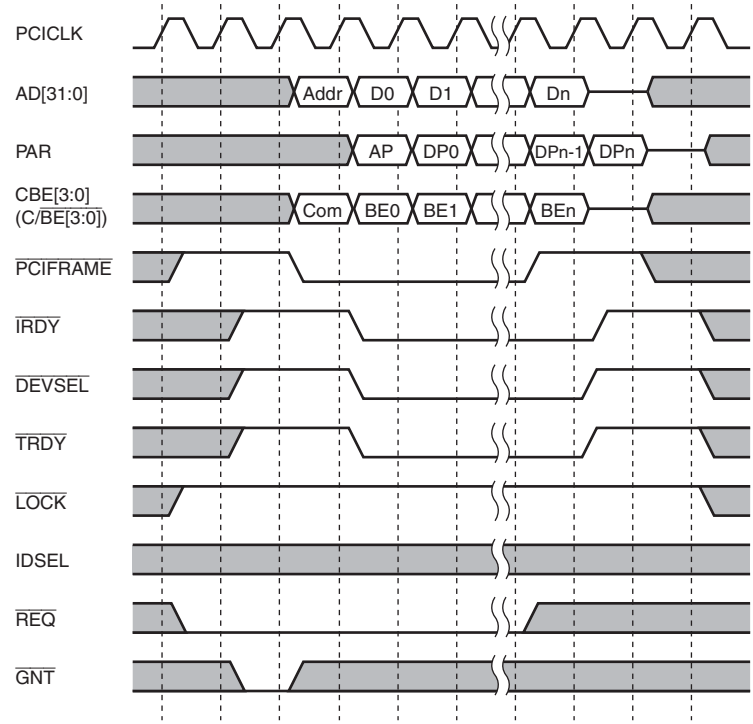


Figure 13.19 Master Write Cycle in Normal Mode (Burst)

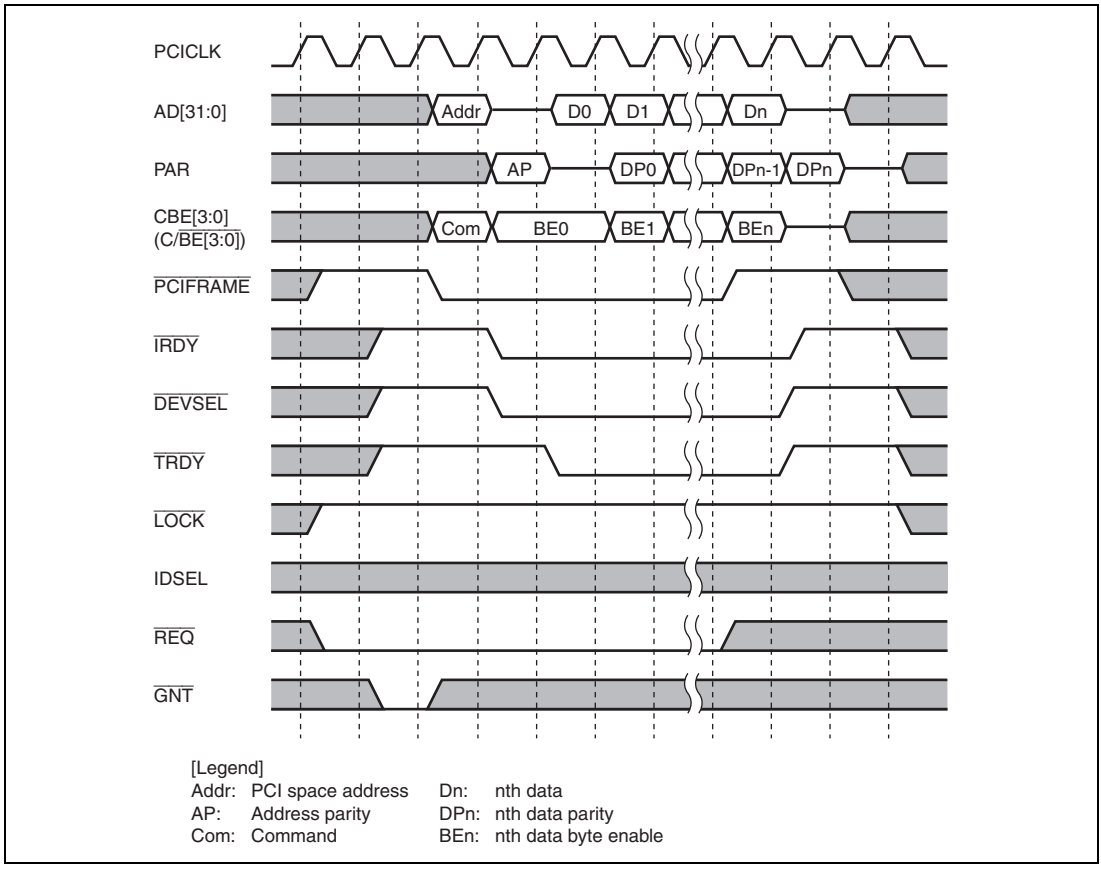


Figure 13.20 Master Read Cycle in Normal Mode (Burst)

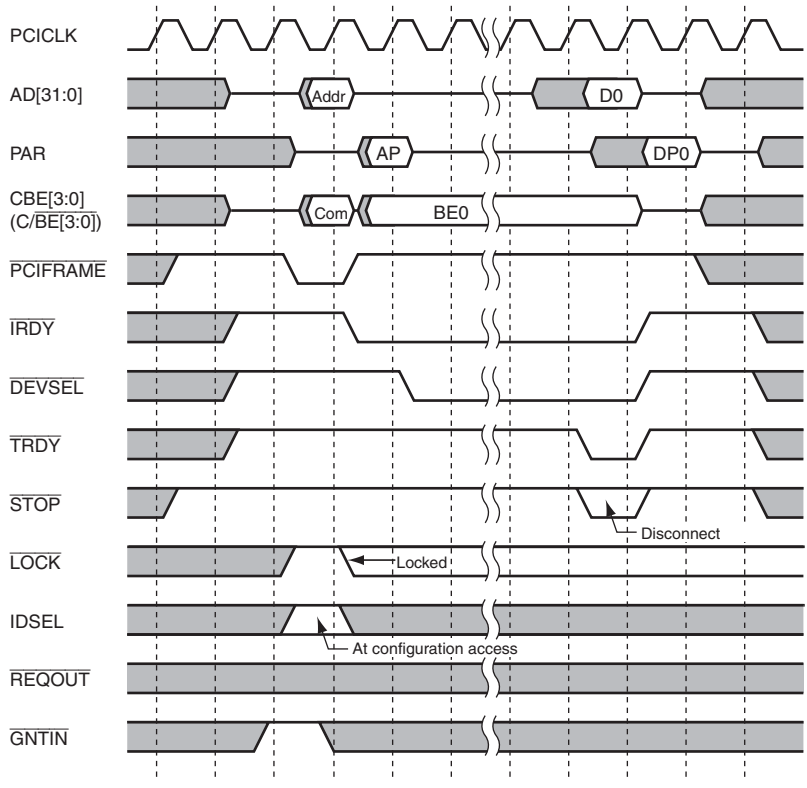
(2) Target Read/Write Cycle Timing

The PCIC responds to target memory burst read accesses from an external master by retries until 8 longword (32-bit) data are prepared in the PCIC's internal FIFO. That is, it always responds to the first target burst read with a retry. For a single read access, the PCIC reaponds as soon as the data is prepared.

Also, when a target memory write access is made, the content of the data is guaranteed until the write data is completely written to the local memory if reading the target write data immediately after write access.

Only single transfers are supported in the case of target accesses of the configuration space and I/O space. If there is a burst access request, the external master is disconnected on completion of the first transfer. Note that the $\overline{\text{DEVSEL}}$ response speed is fixed at 2 clocks (Medium) in the case of target access to the PCIC.

Figure 13.21 shows an example target single read cycle in normal mode. Figure 13.22 shows an example target single write cycle in normal mode. Figure 13.23 is an example of a target burst read cycle in host bus bridge mode. And figure 13.24 is an example of a target burst write cycle in 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)

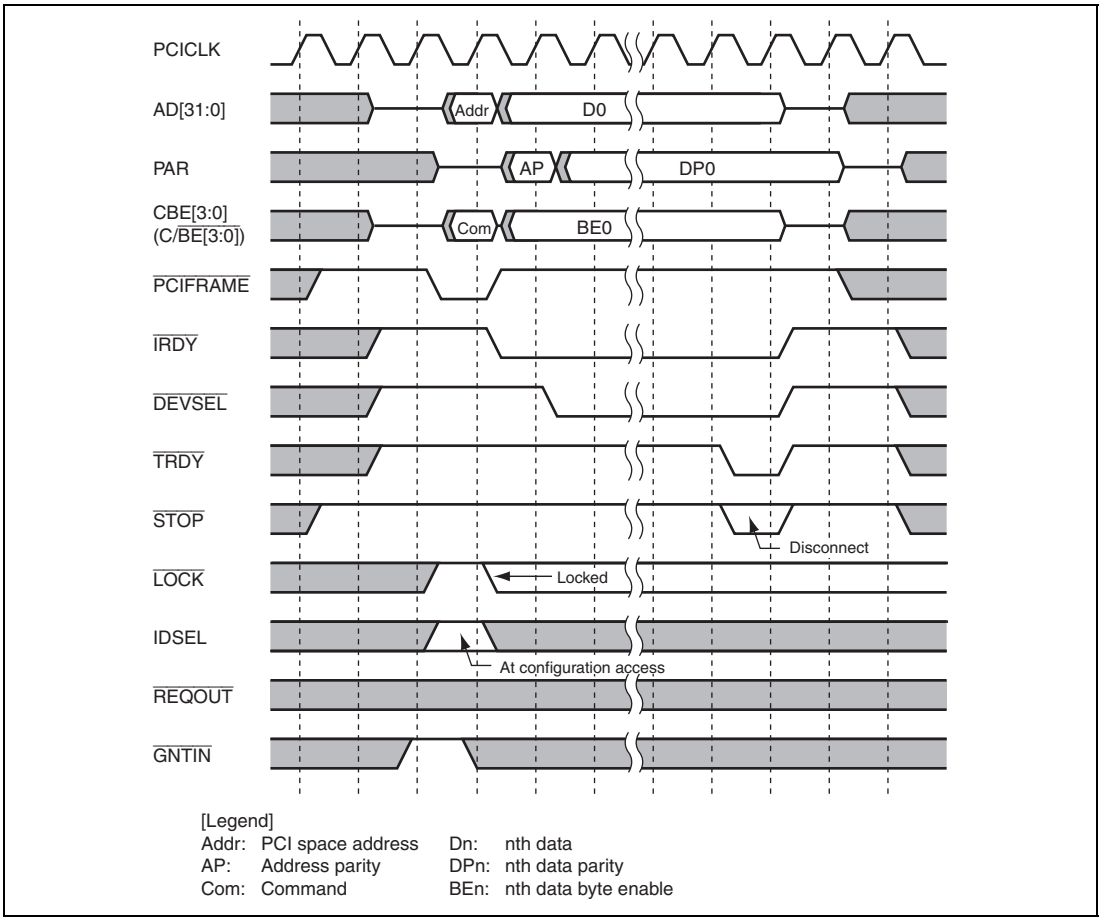


Figure 13.22 Target Write Cycle in Normal Mode (Single)

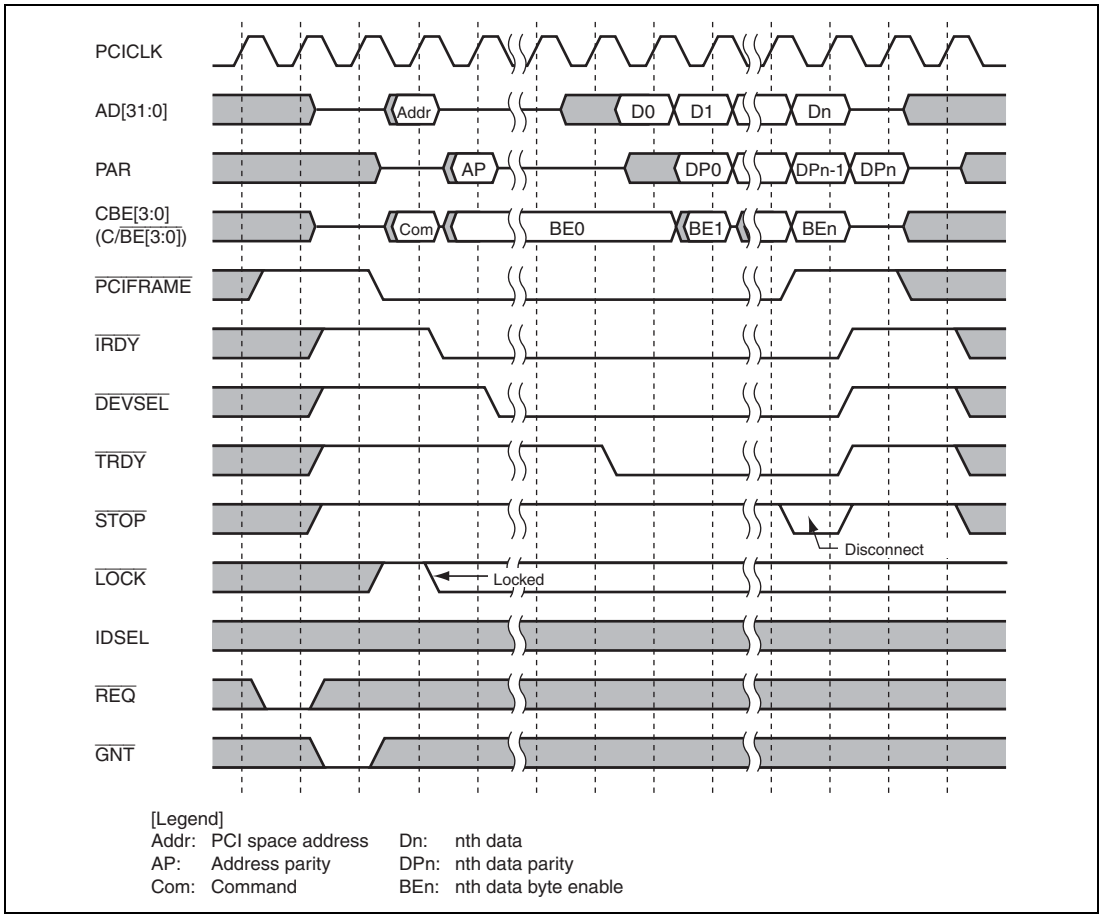


Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burst)

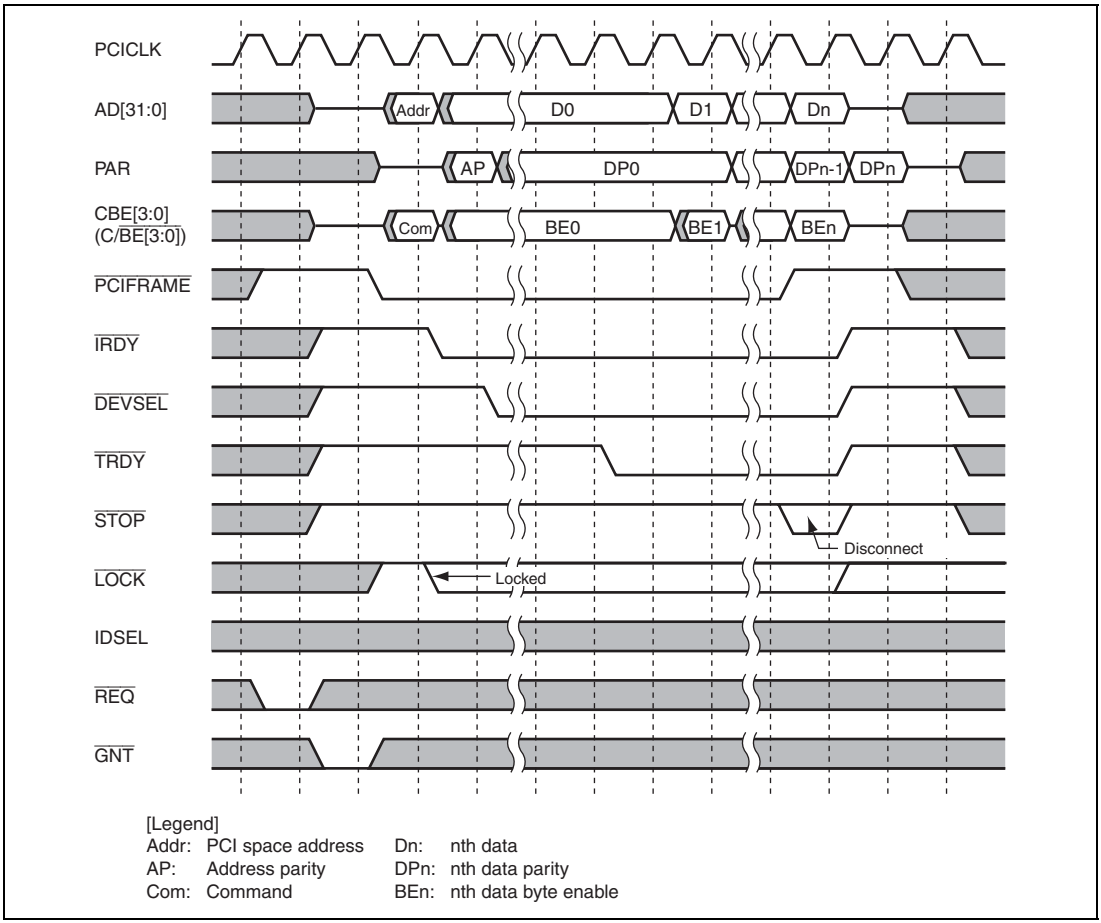


Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst)

(3) Address/Data Stepping Timing

By writing 1 to the SC bit in PCICMD, a wait (stepping) of one clock can be inserted when the PCIC is driving the AD bus. As a result, the PCIC drives the AD bus over 2 clocks. This function can be used when there is a heavy load on the PCI bus and the AD bus does not achieve the stipulated logic level in one clock.

When the PCIC operates as the host bus bridge mode, it is recommended to use this function for the issuance of configuration transfers.

Figure 13.25 is an example of burst memory write cycle with stepping. Figure 13.26 is an example of target burst read cycle with stepping.

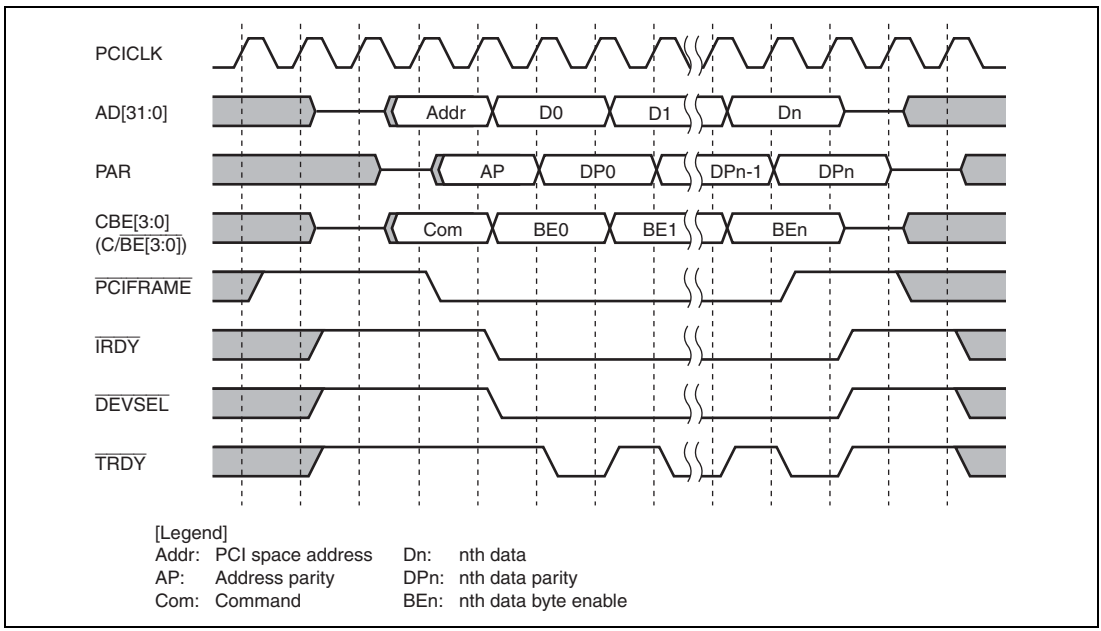


Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with stepping)

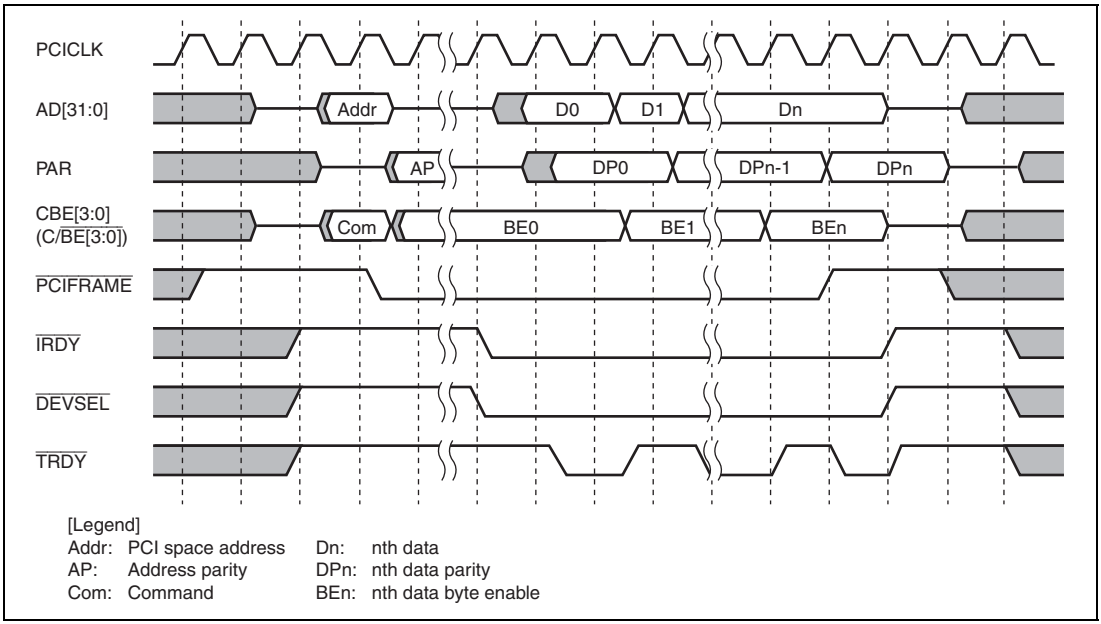


Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with stepping)

Section 14 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (DMA transfer end notification), external memory, on-chip memory, memory-mapped external devices, and peripheral modules.

14.1 Features

- 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, and 32 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 request 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 selected.
- 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 transfers 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)

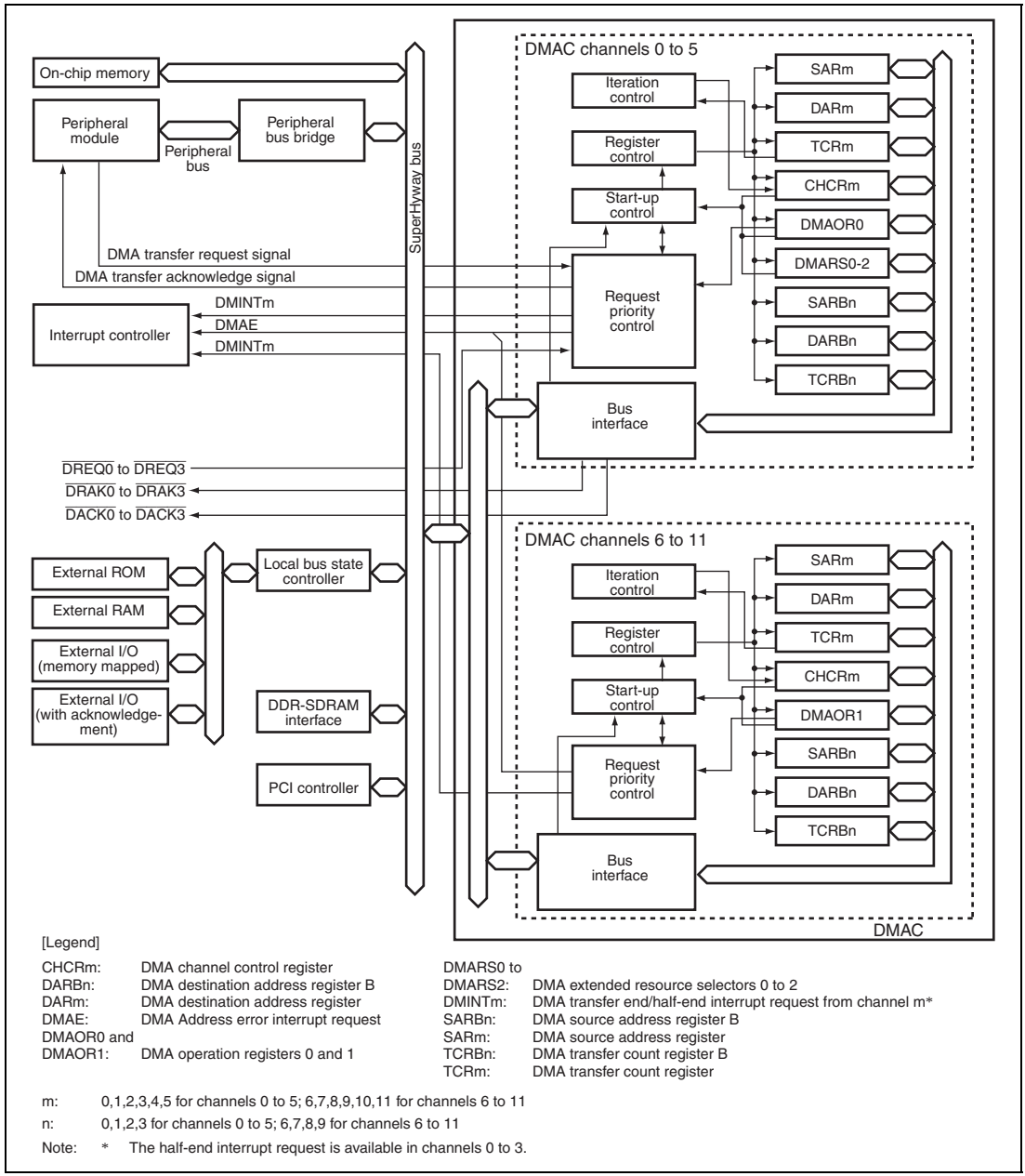
  — High level detection

  — 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)

Figure 14.1 shows the block diagram of the DMAC.



[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 m*
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

14.2 Input/Output Pins

The external pins for the DMAC are described below. Table 14.1 lists the configuration of the pins that are connected to external device. The DMAC has pins for four channels (channel 0 to 3) for external bus use.

Table 14.1 Pin Configuration

Channel	Pin Name	Function	I/O	Description
0	$\overline{\text{DREQ0}}$*1*3	DMA transfer request	Input	DMA transfer request input from external device to channel 0
	$\overline{\text{DRAK0}}$*2*4	$\overline{\text{DREQ0}}$ acceptance confirmation	Output	Notifies acceptance of DMA transfer request and start of execution from channel 0 to external device
	$\overline{\text{DACK0}}$*2*5	DMA transfer end notification	Output	Strobe output from channel 0 to external device which has output, regarding DMA transfer request
1	$\overline{\text{DREQ1}}$*1*6	DMA transfer request	Input	DMA transfer request input from external device to channel 1
	$\overline{\text{DRAK1}}$*2*7	$\overline{\text{DREQ1}}$ acceptance confirmation	Output	Notifies acceptance of DMA transfer request and start of execution from channel 1 to external device
	$\overline{\text{DACK1}}$*2*8	DMA transfer end notification	Output	Strobe output from channel 1 to external device which has output, regarding DMA transfer request
2	$\overline{\text{DREQ2}}$*1*9	DMA transfer request	Input	DMA transfer request input from external device to channel 2
	$\overline{\text{DRAK2}}$*2*10	$\overline{\text{DREQ2}}$ acceptance confirmation	Output	Notifies acceptance of DMA transfer request and start of execution from channel 2 to external device
	$\overline{\text{DACK2}}$*2*11	DMA transfer end notification	Output	Strobe output from channel 2 to external device which has output, regarding DMA transfer request

Channel	Pin Name	Function	I/O	Description
3	$\overline{\text{DREQ3}}$*1*12	DMA transfer request	Input	DMA transfer request input from external device to channel 3
	$\overline{\text{DRAK3}}$*2*13	$\overline{\text{DREQ3}}$ acceptance confirmation	Output	Notifies acceptance of DMA transfer request and start of execution from channel 3 to external device
	$\overline{\text{DACK3}}$*2*14	DMA transfer end notification	Output	Strobe output from channel 3 to external device which has output, regarding DMA transfer request

Notes: 1. Initial value is low level detection.
2. Initial value is low active.
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 pin, and port K5 (GPIO) input/output pin.
10. This pin is multiplexed with $\overline{\text{CE2A}}$ (LBSC) output pin, AUDCK (H-UDI) output pin, and port K1 (GPIO) output pin.
11. This pin is multiplexed with $\overline{\text{MRESETOUT}}$ (RESET) output pin, AUDATA2 (H-UDI) 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 pin, and port K4 (GPIO) input/output pin.
13. This pin is multiplexed with $\overline{\text{CE2B}}$ (LBSC) output pin, AUDSYNC output pin, and port K0 (GPIO) output pin.
14. This pin is multiplexed with $\overline{\text{IRQOUT}}$ (INTC) output pin, AUDATA3 (H-UDI) output pin, and port K2 (GPIO) input/output pin.

14.3 Register Descriptions

Table 14.2 shows the configuration of registers of the DMAC. Table 14.3 shows the register states in each processing mode.

Table 14.2 Register Configuration of DMAC

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*3
0	DMA source address register 0	SAR0	R/W	H'FC80 8020	H'1C80 8020	32
	DMA destination address register 0	DAR0	R/W	H'FC80 8024	H'1C80 8024	32
	DMA transfer count register 0	TCR0	R/W	H'FC80 8028	H'1C80 8028	32
	DMA channel control register 0	CHCR0	R/W*1	H'FC80 802C	H'1C80 802C	32
1	DMA source address register 1	SAR1	R/W	H'FC80 8030	H'1C80 8030	32
	DMA destination address register 1	DAR1	R/W	H'FC80 8034	H'1C80 8034	32
	DMA transfer count register 1	TCR1	R/W	H'FC80 8038	H'1C80 8038	32
	DMA channel control register 1	CHCR1	R/W*1	H'FC80 803C	H'1C80 803C	32
2	DMA source address register 2	SAR2	R/W	H'FC80 8040	H'1C80 8040	32
	DMA destination address register 2	DAR2	R/W	H'FC80 8044	H'1C80 8044	32
	DMA transfer count register 2	TCR2	R/W	H'FC80 8048	H'1C80 8048	32
	DMA channel control register 2	CHCR2	R/W*1	H'FC80 804C	H'1C80 804C	32
3	DMA source address register 3	SAR3	R/W	H'FC80 8050	H'1C80 8050	32
	DMA destination address register 3	DAR3	R/W	H'FC80 8054	H'1C80 8054	32
	DMA transfer count register 3	TCR3	R/W	H'FC80 8058	H'1C80 8058	32
	DMA channel control register 3	CHCR3	R/W*1	H'FC80 805C	H'1C80 805C	32
0 to 5	DMA operation register 0	DMAOR0	R/W*2	H'FC80 8060	H'1C80 8060	16
4	DMA source address register 4	SAR4	R/W	H'FC80 8070	H'1C80 8070	32
	DMA destination address register 4	DAR4	R/W	H'FC80 8074	H'1C80 8074	32
	DMA transfer count register 4	TCR4	R/W	H'FC80 8078	H'1C80 8078	32
	DMA channel control register 4	CHCR4	R/W*1	H'FC80 807C	H'1C80 807C	32
5	DMA source address register 5	SAR5	R/W	H'FC80 8080	H'1C80 8080	32
	DMA destination address register 5	DAR5	R/W	H'FC80 8084	H'1C80 8084	32
	DMA transfer count register 5	TCR5	R/W	H'FC80 8088	H'1C80 8088	32
	DMA channel control register 5	CHCR5	R/W*1	H'FC80 808C	H'1C80 808C	32

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*[3]
0	DMA source address register B0	SARB0	R/W	H'FC80 8120	H'1C80 8120	32
	DMA destination address register B0	DARB0	R/W	H'FC80 8124	H'1C80 8124	32
	DMA transfer count register B0	TCRB0	R/W	H'FC80 8128	H'1C80 8128	32
1	DMA source address register B1	SARB1	R/W	H'FC80 8130	H'1C80 8130	32
	DMA destination address register B1	DARB1	R/W	H'FC80 8134	H'1C80 8134	32
	DMA transfer count register B1	TCRB1	R/W	H'FC80 8138	H'1C80 8138	32
2	DMA source address register B2	SARB2	R/W	H'FC80 8140	H'1C80 8140	32
	DMA destination address register B2	DARB2	R/W	H'FC80 8144	H'1C80 8144	32
	DMA transfer count register B2	TCRB2	R/W	H'FC80 8148	H'1C80 8148	32
3	DMA source address register B3	SARB3	R/W	H'FC80 8150	H'1C80 8150	32
	DMA destination address register B3	DARB3	R/W	H'FC80 8154	H'1C80 8154	32
	DMA transfer count register B3	TCRB3	R/W	H'FC80 8158	H'1C80 8158	32
0, 1	DMA extended resource selector 0	DMARS0	R/W	H'FC80 9000	H'1C80 9000	16
2, 3	DMA extended resource selector 1	DMARS1	R/W	H'FC80 9004	H'1C80 9004	16
4, 5	DMA extended resource selector 2	DMARS2	R/W	H'FC80 9008	H'1C80 9008	16
6	DMA source address register 6	SAR6	R/W	H'FC81 8020	H'1C81 8020	32
	DMA destination address register 6	DAR6	R/W	H'FC81 8024	H'1C81 8024	32
	DMA transfer count register 6	TCR6	R/W	H'FC81 8028	H'1C81 8028	32
	DMA channel control register 6	CHCR6	R/W*[1]	H'FC81 802C	H'1C81 802C	32
7	DMA source address register 7	SAR7	R/W	H'FC81 8030	H'1C81 8030	32
	DMA destination address register 7	DAR7	R/W	H'FC81 8034	H'1C81 8034	32
	DMA transfer count register 7	TCR7	R/W	H'FC81 8038	H'1C81 8038	32
	DMA channel control register 7	CHCR7	R/W*[1]	H'FC81 803C	H'1C81 803C	32
8	DMA source address register 8	SAR8	R/W	H'FC81 8040	H'1C81 8040	32
	DMA destination address register 8	DAR8	R/W	H'FC81 8044	H'1C81 8044	32
	DMA transfer count register 8	TCR8	R/W	H'FC81 8048	H'1C81 8048	32
	DMA channel control register 8	CHCR8	R/W*[1]	H'FC81 804C	H'1C81 804C	32
9	DMA source address register 9	SAR9	R/W	H'FC81 8050	H'1C81 8050	32
	DMA destination address register 9	DAR9	R/W	H'FC81 8054	H'1C81 8054	32
	DMA transfer count register 9	TCR9	R/W	H'FC81 8058	H'1C81 8058	32
	DMA channel control register 9	CHCR9	R/W*[1]	H'FC81 805C	H'1C81 805C	32

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*3
6 to 11	DMA operation register 1	DMAOR1	R/W*2	H'FC81 8060	H'1C81 8060	16
10	DMA source address register 10	SAR10	R/W	H'FC81 8070	H'1C81 8070	32
	DMA destination address register 10	DAR10	R/W	H'FC81 8074	H'1C81 8074	32
	DMA transfer count register 10	TCR10	R/W	H'FC81 8078	H'1C81 8078	32
	DMA channel control register 10	CHCR10	R/W*1	H'FC81 807C	H'1C81 807C	32
11	DMA source address register 11	SAR11	R/W	H'FC81 8080	H'1C81 8080	32
	DMA destination address register 11	DAR11	R/W	H'FC81 8084	H'1C81 8084	32
	DMA transfer count register 11	TCR11	R/W	H'FC81 8088	H'1C81 8088	32
	DMA channel control register 11	CHCR11	R/W*1	H'FC81 808C	H'1C81 808C	32
6	DMA source address register B6	SARB6	R/W	H'FC81 8120	H'1C81 8120	32
	DMA destination address register B6	DARB6	R/W	H'FC81 8124	H'1C81 8124	32
	DMA transfer count register B6	TCRB6	R/W	H'FC81 8128	H'1C81 8128	32
7	DMA source address register B7	SARB7	R/W	H'FC81 8130	H'1C81 8130	32
	DMA destination address register B7	DARB7	R/W	H'FC81 8134	H'1C81 8134	32
	DMA transfer count register B7	TCRB7	R/W	H'FC81 8138	H'1C81 8138	32
8	DMA source address register B8	SARB8	R/W	H'FC81 8140	H'1C81 8140	32
	DMA destination address register B8	DARB8	R/W	H'FC81 8144	H'1C81 8144	32
	DMA transfer count register B8	TCRB8	R/W	H'FC81 8148	H'1C81 8148	32
9	DMA source address register B9	SARB9	R/W	H'FC81 8150	H'1C81 8150	32
	DMA destination address register B9	DARB9	R/W	H'FC81 8154	H'1C81 8154	32
	DMA transfer count register B9	TCRB9	R/W	H'FC81 8158	H'1C81 8158	32

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 flag.
3. Accessing with other access sizes is prohibited.

Table 14.3 Register States in Each Processing Mode

Channel	Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$/WDT/ H-UDI	Manual Reset by WDT/Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby
0	DMA source address register 0	SAR0	Undefined	Undefined	Retained	Retained
	DMA destination address register 0	DAR0	Undefined	Undefined	Retained	Retained
	DMA transfer count register 0	TCR0	Undefined	Undefined	Retained	Retained
	DMA channel control register 0	CHCR0	H'4000 0000	H'4000 0000	Retained	Retained
1	DMA source address register 1	SAR1	Undefined	Undefined	Retained	Retained
	DMA destination address register 1	DAR1	Undefined	Undefined	Retained	Retained
	DMA transfer count register 1	TCR1	Undefined	Undefined	Retained	Retained
	DMA channel control register 1	CHCR1	H'4000 0000	H'4000 0000	Retained	Retained
2	DMA source address register 2	SAR2	Undefined	Undefined	Retained	Retained
	DMA destination address register 2	DAR2	Undefined	Undefined	Retained	Retained
	DMA transfer count register 2	TCR2	Undefined	Undefined	Retained	Retained
	DMA channel control register 2	CHCR2	H'4000 0000	H'4000 0000	Retained	Retained
3	DMA source address register 3	SAR3	Undefined	Undefined	Retained	Retained
	DMA destination address register 3	DAR3	Undefined	Undefined	Retained	Retained
	DMA transfer count register 3	TCR3	Undefined	Undefined	Retained	Retained
	DMA channel control register 3	CHCR3	H'4000 0000	H'4000 0000	Retained	Retained
0 to 5	DMA operation register 0	DMAOR0	Undefined	Undefined	Retained	Retained
4	DMA source address register 4	SAR4	Undefined	Undefined	Retained	Retained
	DMA destination address register 4	DAR4	Undefined	Undefined	Retained	Retained
	DMA transfer count register 4	TCR4	Undefined	Undefined	Retained	Retained
	DMA channel control register 4	CHCR4	H'4000 0000	H'4000 0000	Retained	Retained
5	DMA source address register 5	SAR5	Undefined	Undefined	Retained	Retained
	DMA destination address register 5	DAR5	Undefined	Undefined	Retained	Retained
	DMA transfer count register 5	TCR5	Undefined	Undefined	Retained	Retained
	DMA channel control register 5	CHCR5	H'4000 0000	H'4000 0000	Retained	Retained

Channel	Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$/WDT/ H-UDI	Manual Reset by WDT/Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby
0	DMA source address register B0	SARB0	Undefined	Undefined	Retained	Retained
	DMA destination address register B0	DARB0	Undefined	Undefined	Retained	Retained
	DMA transfer count register B0	TCRB0	Undefined	Undefined	Retained	Retained
1	DMA source address register B1	SARB1	Undefined	Undefined	Retained	Retained
	DMA destination address register B1	DARB1	Undefined	Undefined	Retained	Retained
	DMA transfer count register B1	TCRB1	Undefined	Undefined	Retained	Retained
2	DMA source address register B2	SARB2	Undefined	Undefined	Retained	Retained
	DMA destination address register B2	DARB2	Undefined	Undefined	Retained	Retained
	DMA transfer count register B2	TCRB2	Undefined	Undefined	Retained	Retained
3	DMA source address register B3	SARB3	Undefined	Undefined	Retained	Retained
	DMA destination address register B3	DARB3	Undefined	Undefined	Retained	Retained
	DMA transfer count register B3	TCRB3	Undefined	Undefined	Retained	Retained
0, 1	DMA extended resource selector 0	DMARS0	H'0000 0000	H'0000 0000	Retained	Retained
2, 3	DMA extended resource selector 1	DMARS1	H'0000 0000	H'0000 0000	Retained	Retained
4, 5	DMA extended resource selector 2	DMARS2	H'0000 0000	H'0000 0000	Retained	Retained
6	DMA source address register 6	SAR6	Undefined	Undefined	Retained	Retained
	DMA destination address register 6	DAR6	Undefined	Undefined	Retained	Retained
	DMA transfer count register 6	TCR6	Undefined	Undefined	Retained	Retained
	DMA channel control register 6	CHCR6	H'4000 0000	H'4000 0000	Retained	Retained
7	DMA source address register 7	SAR7	Undefined	Undefined	Retained	Retained
	DMA destination address register 7	DAR7	Undefined	Undefined	Retained	Retained

Channel	Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$/WDT/ H-UDI	Manual Reset by WDT/Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby
7	DMA transfer count register 7	TCR7	Undefined	Undefined	Retained	Retained
	DMA channel control register 7	CHCR7	H'4000 0000	H'4000 0000	Retained	Retained
8	DMA source address register 8	SAR8	Undefined	Undefined	Retained	Retained
	DMA destination address register 8	DAR8	Undefined	Undefined	Retained	Retained
	DMA transfer count register 8	TCR8	Undefined	Undefined	Retained	Retained
	DMA channel control register 8	CHCR8	H'4000 0000	H'4000 0000	Retained	Retained
9	DMA source address register 9	SAR9	Undefined	Undefined	Retained	Retained
	DMA destination address register 9	DAR9	Undefined	Undefined	Retained	Retained
	DMA transfer count register 9	TCR9	Undefined	Undefined	Retained	Retained
	DMA channel control register 9	CHCR9	H'4000 0000	H'4000 0000	Retained	Retained
6 to 11	DMA operation register 1	DMAOR 1	Undefined	Undefined	Retained	Retained
10	DMA source address register 10	SAR10	Undefined	Undefined	Retained	Retained
	DMA destination address register 10	DAR10	Undefined	Undefined	Retained	Retained
	DMA transfer count register 10	TCR10	Undefined	Undefined	Retained	Retained
	DMA channel control register 10	CHCR10	H'4000 0000	H'4000 0000	Retained	Retained
11	DMA source address register 11	SAR11	Undefined	Undefined	Retained	Retained
	DMA destination address register 11	DAR11	Undefined	Undefined	Retained	Retained
	DMA transfer count register 11	TCR11	Undefined	Undefined	Retained	Retained
	DMA channel control register 11	CHCR11	H'4000 0000	H'4000 0000	Retained	Retained
6	DMA source address register B6	SARB6	Undefined	Undefined	Retained	Retained
	DMA destination address register B6	DARB6	Undefined	Undefined	Retained	Retained
	DMA transfer count register B6	TCRB6	Undefined	Undefined	Retained	Retained

Channel	Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$/WDT/ H-UDI	Manual Reset by WDT/Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby
7	DMA source address register B7	SARB7	Undefined	Undefined	Retained	Retained
	DMA destination address register B7	DARB7	Undefined	Undefined	Retained	Retained
	DMA transfer count register B7	TCRB7	Undefined	Undefined	Retained	Retained
8	DMA source address register B8	SARB8	Undefined	Undefined	Retained	Retained
	DMA destination address register B8	DARB8	Undefined	Undefined	Retained	Retained
	DMA transfer count register B8	TCRB8	Undefined	Undefined	Retained	Retained
9	DMA source address register B9	SARB9	Undefined	Undefined	Retained	Retained
	DMA destination address register B9	DARB9	Undefined	Undefined	Retained	Retained
	DMA transfer count register B9	TCRB9	Undefined	Undefined	Retained	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 transfer. 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 longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary 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	17	16
	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	R/W	R/W

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

14.3.2 DMA Source Address Registers B0 to B3, B6 to B9 (SARB0 to SARB3, SARB6 to SARB9)

SARB are 32-bit readable/writable registers that specify the source address of a DMA transfer that is set in SAR again in repeat/reload mode. Data to be written from the CPU to SAR is also written to SARB. To set SARB address that differs from SAR address, write data to SARB after SAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary 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	17	16
	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/W	R/W

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

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 transfer. 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 address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary 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	17	16
	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/W	R/W

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

14.3.4 DMA Destination Address Registers B0 to B3, B6 to B9 (DARB0 to DARB3, DARB6 to DARB9)

DARB are 32-bit readable/writable registers that specify the destination address of a DMA transfer that is set in DAR again in repeat/reload mode. Data to be written from the CPU to DAR is also written to DARB. To set DARB address that differs from DAR address, write data to DARB after DAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary 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	17	16
	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	R/W	R/W

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

14.3.5 DMA Transfer Count Registers 0 to 11 (TCR0 to TCR11)

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

The upper eight bits of TCR (bits 31 to 24) are always read as 0, and the write value should always be 0. The initial value is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	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/W	R/W

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

14.3.6 DMA Transfer Count Registers B0 to B3, B6 to B9 (TCRB0 to TCRB3, TCRB6 to TCRB9)

TCRB are 32-bit readable/writable registers. Data to be written from the CPU to TCR is also written to TCRB. While the half-end* function is used, TCRB are used as the initial value hold registers to detect half-end. Also, TCRB specify the number of DMA transfers which are set in TCR in repeat mode. TCRB specify the number of DMA transfers and are used as transfer count counters in reload mode.

Note: * The "half-end" means the transfer is half finished.

In reload mode, the lower 8 bits (bits 7 to 0) operate as transfer count counters, values of SAR and DAR are updated after the value of the bits 7 to 0 became 0, and then the value of the bits 23 to 16 of TCRB are loaded to the bits 7 to 0. In bits 23 to 16, set the number of transfers which starts reloading. In reload mode, a value from H'FF (255 times) to H'01 (1 time) can be specified to the bits 23 to 16 and 7 to 0 of TCRB, and set the same number in both bits 23 to 16 and bits 7 to 0 and clear to H'00 in bits 15 to 8. Also, set the HIE bit in CHCR to 0 and do not use the half end function.

The upper eight bits of TCRB (bits 31 to 24) are always read as 0, and the write value should 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	17	16
	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	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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

14.3.7 DMA Channel Control Registers 0 to 11 (CHCR0 to CHCR11)

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	—	RPT[2:0]			—	DO	RL	—	TS2	HE	HIE	AM	AL
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R	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	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
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/W

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 should always be 0.
30	LCKN	1	R/W	Bus Lock Signal Disable Specifies whether enable or disable the bus lock signal output when a read instruction for the SuperHyway bus. This bit is effective in cycle steal mode, and should be cleared to 0 in burst mode. To disable the bus lock signal, the bus request from the bus master other than the DMAC could be received, 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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
27 to 25	RPT[2:0]	000	R/W	DMA Setting Renewal Specify These bits are enabled in CHCR0 to CHCR3 and CHCR6 to CHCR9. 000: Normal mode (DMAC operation) 001: Repeat mode SAR/DAR/TCR used as repeat area 010: Repeat mode DAR/TCR used as repeat area 011: Repeat mode 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 should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR0 to CHCR3. 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-high or active-low output. This bit valid only in CHCR0 to 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 should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
20	TS2	0	R/W	DMA Transfer Size Specify With TS1 and TS0 (bits 4 and 3), this bit specifies the DMA transfer size. When the transfer source or transfer destination is a register of an peripheral module that access size is designated, the transfer size for the register should be the same value of its access size. For the transfer source or destination address specified by SAR or DAR, an address boundary should be set according to the transfer data size. TS[2:0] 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

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half End Flag After HIE (bit 18) is set to 1 and the number of transfers become half of TCR (1 bit shift to right) which is set before transfer starts, HE becomes 1. This bit is set to 1 when the TCR value is equal to any of the following: • (TCR set before transfer)/2 (TCR: even number) • (TCR set before transfer - 1)/2 (TCR: odd number) • 8,388,608 (H'0080 0000) (TCR: maximum number H'0000 0000) The HE bit is not set when transfers are ended by an NMI interrupt or address error, or by clearing the DE bit or the DME bit in DMAOR before the number of transfers is decreased to half of the TCR value set preceding the transfer. The HE bit is kept set when the transfer ends by an NMI interrupt or address error, or clearing the DE bit (bit 0) or the DME bit in DMAOR after the HE bit is set to 1. To clear the HE bit, write 0 after reading 1 in the HE bit. This bit is valid only in CHCR0 to CHCR3 and CHCR6 to CHCR9. 0: During the DMA transfer or DMA transfer has been interrupted TCR > (TCR set before transfer)/2 [Clearing condition] Writing 0 after HE = 1 is read. 1: TCR = (TCR set before transfer)/2

Bit	Bit Name	Initial Value	R/W	Descriptions
18	HIE	0	R/W	Half End Interrupt Enable Specifies whether an interrupt request is generated to the CPU when the number of transfers is decreased to half of the TCR value (a read transfer cycle end) set preceding the transfer. When the HIE bit is set to 1 and the HE bit is set, an interrupt request is generated to the CPU. To confirm the half end of the transfer, execute a dummy read of the destination space and issue the SYNCO instruction. Clear this bit to 0 while reload mode is set. This bit is valid in CHCR0 to 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 cycle or in 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 high active 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

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	DM[1:0]	00	R/W	Destination Address Mode 1, 0 Specify whether the DMA destination address is incremented, decremented, or left fixed. 00: Fixed destination address 01: Destination 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: 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 transfer 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 transfer 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11 to 8	RS[3:0]	0000	R/W	Resource Select 3 to 0 Specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that the DMA enable bit (DE) is cleared to 0. 0000: External request, dual address mode 0100: Auto request 1000: Selected by DMA extended resource selector (for peripheral modules) Other than above: Setting prohibited Note: External request specification is valid only in CHCR0 to CHCR3. None of the external request can be selected in CHCR4 to CHCR11. DMA extended resource selector is valid only in 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 input and the detecting level. These bits are valid only in CHCR0 to CHCR3. In channels 0 to 3, also, if the transfer request source is specified as a peripheral module or if an auto-request is 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 data. 0: Cycle steal mode 1: Burst mode Select the cycle steal mode when the peripheral module requests.
4, 3	TS[1:0]	00	R/W	DMA Transfer Size Specify See the description of TS2 (bit 20).

Bit	Bit Name	Initial Value	R/W	Descriptions
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the final DMA transfer. Setting this bit to 1 generates an interrupt request (DMINT) to the CPU when the TE bit is set to 1 and the final DMA transfer of read cycle ended. To confirm the final end of the transfer, execute a dummy read of the destination space and issue the SYNCO instruction. 0: Interrupt request is disabled. 1: Interrupt request is enabled.
1	TE	0	R/(W)*	Transfer End Flag Shows that DMA transfer ends. The TE bit is set to 1 when TCR becomes to 0 (and the DMAC starts executing the final DMA transfer). The TE bit is not set to 1 in either of the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before TCR is cleared to 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMAOR. To clear the TE bit, the TE bit should be written to 0 after reading 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been interrupted [Clearing condition] Writing 0 after TE = 1 read 1: TCR = 0 (During the final DMA transfer or the DMA transfer ends)

Bit	Bit Name	Initial Value	R/W	Descriptions
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this time, all of the bits TE, NMIF, and AE in DMAOR must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0, which is the same as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

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

14.3.8 DMA Operation Register 0, 1 (DMAOR0 and DMAOR1)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register shows the DMA transfer status. DMAOR 0 is for channel 0 to 5, and DMAOR1 is for channel 6 to11.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CMS[1:0]		—	—	PR[1:0]		—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	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/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select 1, 0 Select either normal mode or intermittent mode in cycle steal mode. It is necessary that all channels 0 to 5 (DMAOR0) or 6 to 11 (DMAOR1) bus modes are set to cycle steal mode to make valid intermittent mode. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Issues a bus request after waiting 16 Bck clocks and executes one DMA transfer. 11: Intermittent mode 64 Issues a bus request after waiting 64 Bck clocks and executes one DMA transfer.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
9, 8	PR[1:0]	00	R/W	Priority Mode 1, 0 Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 (DMAOR0) CH6 > CH7 > CH8 > CH9 > CH10 > CH11 (DMAOR1) 01: CH0 > CH2 > CH3 > CH1 > CH4 > CH5 (DMAOR0) CH6 > CH8 > CH9 > CH7 > CH10 > CH11 (DMAOR1) 10: Setting prohibited 11: Round-robin mode When round-robin mode is specified, do not mix the cycle steal mode and the burst mode in channels 0 to 5 or 6 to 11 respectively.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred during DMA transfer. This bit is set under following conditions: • The value set in SAR or DAR does not match to the transfer size boundary. • The transfer source or transfer destination is invalid space. • The transfer source or transfer destination is in module stop mode If this bit is set, the corresponding channels (channels 0 to 5 or 6 to 11) DMA transfer are all disabled even if the DE bit in each CHCR and the DME bit in corresponding DMAOR are set to 1. 0: No DMAC address error [Clearing condition] Writing AE = 0 after AE = 1 read 1: DMAC address error occurs

Bit	Bit Name	Initial Value	R/W	Descriptions
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. When the NMI is input, the DMA transfer in progress can be done in at least one transfer unit. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input. 0: No NMI interrupt [Clearing condition] Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channels 0 to 5 (DMAOR0) or 6 to 11 (DMAOR1). If the DME bit and the DE bit in CHCR are set to 1, transfer is enabled. In this time, all of the bits TE in CHCR, NMIF, and AE in DMAOR must be 0. If this bit is cleared during transfer, transfers in all channels 0 to 5 (DMAOR0) or 6 to 11 (DMAOR1) are terminated. 0: Disables DMA transfers on all channels (Channels 0 to 5 by DMAOR0, 6 to 11 by DMAOR1) 1: Enables DMA transfers on all channels (Channels 0 to 5 by DMAOR0, 6 to 11 by DMAOR1)

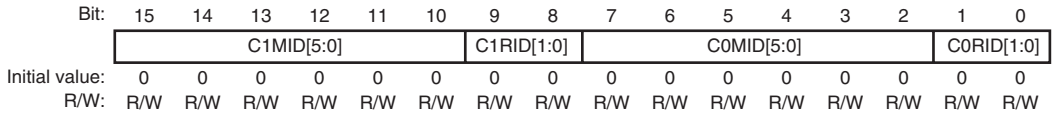
Note: * Writing 0 is possible to clear the flag.

14.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2)

DMARS are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 specifies for channels 0 and 1, DMARS1 specifies for channels 2 and 3, and DMARS2 specifies for channels 4 and 5. This register can set the transfer request of SCIF0, SCIF1, HAC, HSPI, SIOF, SSI, FLCTL, and MMCIF.

When MID/RID other than the values listed in table 14.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits RS[3:0] has been set to B'1000 for CHCR0 to CHCR5 registers. Otherwise, even if DMARS has been set, transfer request source is not accepted.

- DMARS0



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C1MID[5:0]						C1RID[1:0]		C0MID[5:0]						C0RID[1:0]	
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/W

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 channel 1 (MID) See table 14.4.
9, 8	C1RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 1 (RID) See table 14.4.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 0 (MID) See table 14.4
1, 0	C0RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 0 (RID) See table 14.4.

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C3MID[5:0]						C3RID[1:0]		C2MID[5:0]						C2RID[1:0]	
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/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C3MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 3 (MID) See table 14.4.
9, 8	C3RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 3 (RID) See table 14.4.
7 to 2	C2MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 2 (MID) See table 14.4.
1, 0	C2RID[1:0]	00	R/W R/W	Transfer request register ID1 and ID0 for DMA channel 2 (RID) See table 14.4.

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]						C5RID[1:0]		C4MID[5:0]						C4RID[1:0]	
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/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C5MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 5 (MID) See table 14.4.
9, 8	C5RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 5 (RID) See table 14.4.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA channel 4 (MID) See table 14.4.
1, 0	C4RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA channel 4 (RID) See table 14.4.

Table 14.4 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID and RID fields)	MID[5:0]	RID[1:0]	Function
SCIF0	H'21	B'001000	B'01	Transmit
	H'22		B'10	Receive
SCIF1	H'29	B'001010	B'01	Transmit
	H'2A		B'10	Receive
HAC	H'41	B'010000	B'01	Transmit
	H'42		B'10	Receive
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 receive
FLCTL	H'83	B'100000	B'11	Transmit and receive the data part.
	H'87	B'100001	B'11	Transmit and receive the control code part.
MMCIF	H'93	B'100100	B'11	Transmit and receive

14.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

14.4.1 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by external devices or peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto request, external request, and 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 used.

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. Specify B'0100 to the RS [3:0] bits in CHCRn (n = 0 to 11) of the using 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 begins so 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 (DREQ) of an external device. This mode is valid only in channel 0 to 3. Specify B'0000 to the RS [3:0] bits in CHCRn (n = 0 to 3) of the using DMA channel. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a request at the DREQ input.

Choose to detect DREQ by either the edge or level of the signal input with the DL bit and DS bit in CHCRn (n = 0 to 3) as shown in table 14.5. The source of the transfer request does not have to be the data transfer source or destination.

Table 14.5 Selecting External Request Detection with DL, DS Bits

CHCRn (n = 0 to 3)		
DL	**DS**	**Detection of External Request**
0	0	Low level detection (initial value; $\overline{\text{DREQ}}$)
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When DREQ is accepted, the DREQ pin becomes request accept disabled state. After issuing acknowledge signal DACK for the accepted DREQ, the DREQ pin again becomes request accept enabled state.

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

- Overrun 0: Transfer is aborted after the same number of transfer has been performed as requests.
- Overrun 1: Transfer is aborted after transfers have been performed for the number of requests 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 request signal of an peripheral module. This mode is valid only in channel 0 to 5. Specify B'1000 to the RS [3:0] bits in CHCRn (n = 0 to 5) of the using DMA channel. Transfer request signals comprise the transmit data empty transfer request and receive data full transfer request from the SCIF0, SCIF1, HAC, HSPI, SIOF, SSI, and MMCIF set by DMARS0/1/2, and transfer requests from the FLCTL.

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

When a transmit data empty transfer request of the SCIF0 is set as the transfer request, the transfer destination must be the SCIF0's transmit data register. Likewise, when receive data full transfer request of the SCIF0 is set as the transfer request, the transfer source must be the SCIF0's receive data register. These conditions also apply to the SCIF1, HAC, HSPI, SIOF, SSI, and MMCIF.

Table 14.7 Peripheral Module Request Modes

DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
MID	RID					
001000	01	SCI F0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
	10	SCIF0 receiver	RXI (receive FIFO data full interrupt)	SCFRDR0	Any	Cycle steal
001010	01	SCI F1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR1	Cycle steal
	10	SCIF1 receiver	RXI (receive FIFO data full interrupt)	SCFRDR1	Any	Cycle steal
010000	01	HAC transmitter	Transmit data empty request	Any	HACPCML, HACPCMR	Cycle steal
	10	HAC receiver	Receive data is not read	HACPCML, HACPCMR	Any	Cycle steal
010001	01	HSPI transmitter	Transmit data	Any	SPTBR	Cycle steal
	10	HSPI receiver	Receive data	SPRBR	Any	Cycle steal
010100	01	SIOF transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR	Cycle steal
	10	SIOF receiver	RXI (receive FIFO data full interrupt)	SIRDR	Any	Cycle steal
011100	11	SSI transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSITDR	Cycle steal
		SSI receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any	Cycle steal
100000	11	FLCTL data part transmit	Transmit FIFO data empty request	Any	FLDTFIFO	Cycle steal
		FLCTL data part receive	Receive FIFO data full request	FLDTFIFO	Any	Cycle steal
100001	11	FLCTL management code part transmit	Transmit FIFO data empty request	Any	FLECFIFO	Cycle steal
		FLCTL management code part receive	Receive FIFO data full request	FLECFIFO	Any	Cycle steal
100100	11	MMCIF data part transmit	FIFO data write request	Any	DR	Cycle steal
		MMCIF data part receive	FIFO data read request	DR	Any	Cycle steal

14.4.2 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it transfers data according to a predetermined priority. Two modes (fixed mode and round-robin mode) are selected by the bits PR[1:0] in DMAOR0 for channels 0 to 5 and DMAOR1 for channels 6 to 11.

If the DMAC receives simultaneous transfer requests from both any channels 0 to 5 and 6 to 11 respectively, then executes each channels 0 to 5 or 6 to 11 request alternately (initial state: channel 0 to 5 is higher priority).

Fixed Mode: In this mode, the priority levels among the channels remain fixed. There are two 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, word, longword, 16-byte, or 32-byte unit) is transferred on one channel, the priority is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority. The round-robin mode operation is shown in figure 14.2. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5, and CH6 > CH7 > CH8 > CH9 > CH10 > CH11 immediately after reset.

When round-robin mode is specified, do not mix the cycle steal mode and the burst mode in channels 0 to 5 or 6 to 11 respectively.

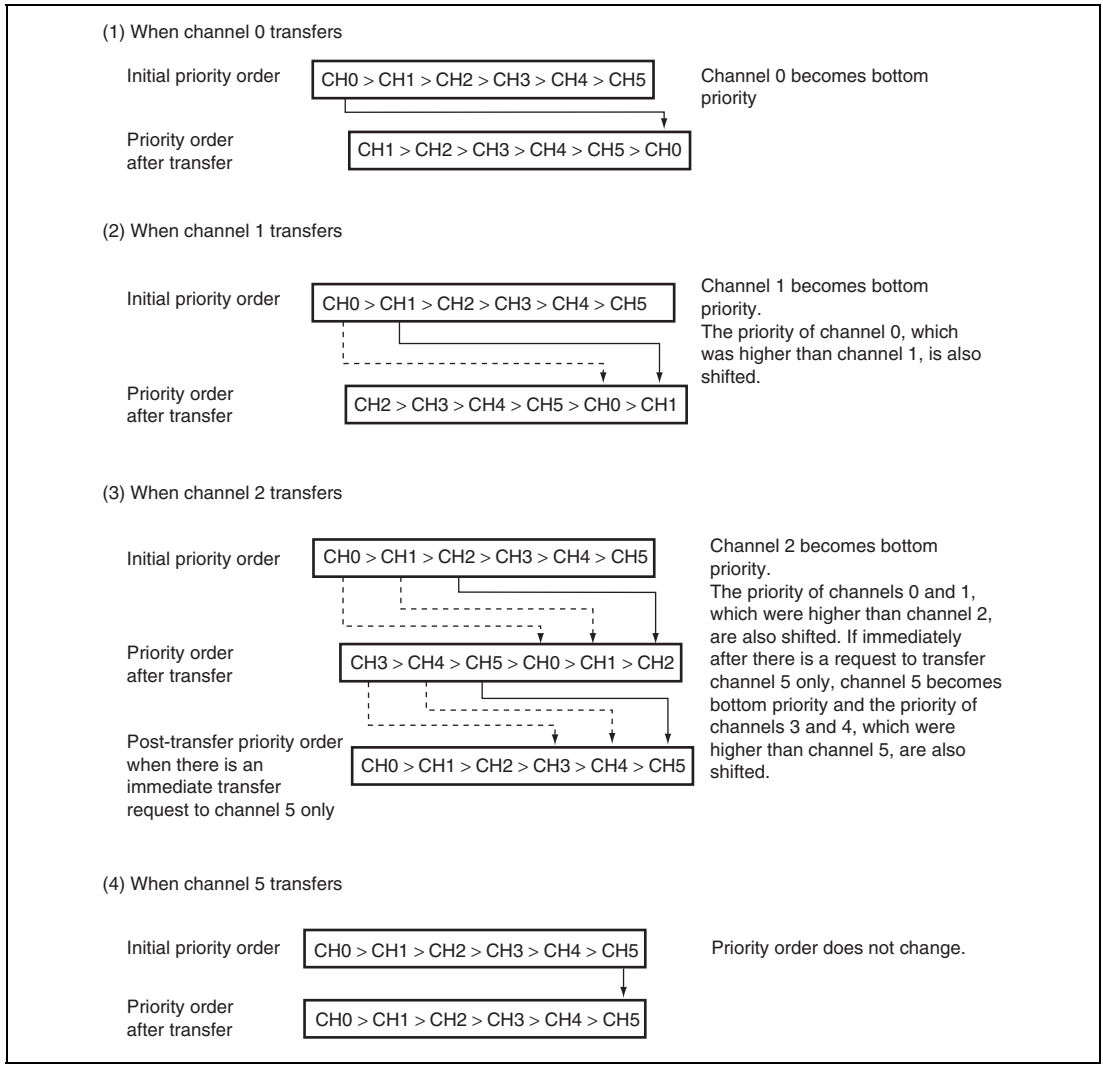


Figure 14.2 Round-Robin Mode (example of channel 0 to 5)

Figure 14.3 shows how the priority changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (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 channel 3 becomes the lowest priority.

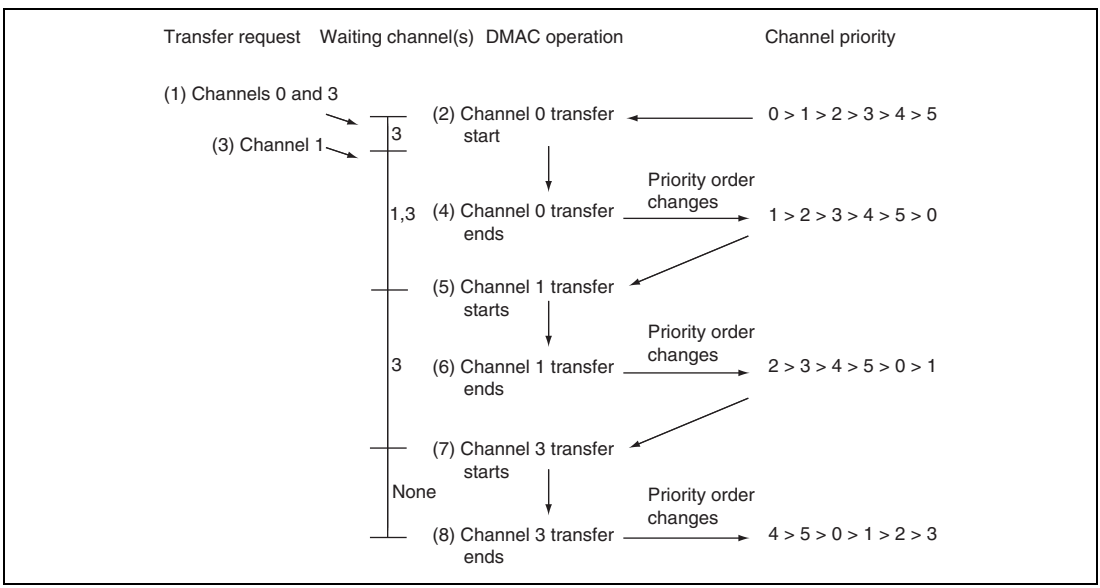


Figure 14.3 Changes in Channel Priority in Round-Robin Mode (example of channel 0 to 5)

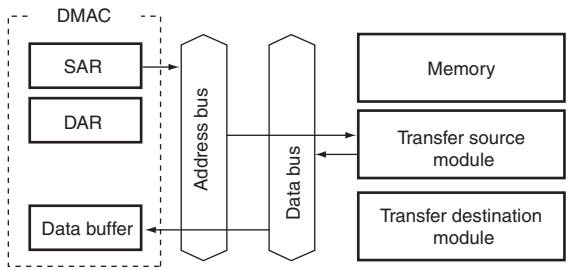
14.4.3 DMA Transfer Types

DMA transfer type is dual address mode transfer. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode.

Dual Address Modes:

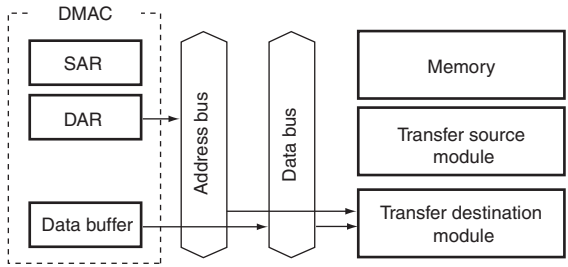
In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 14.4, data is read to the DMAC from one external memory in a data read cycle, and then that data 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



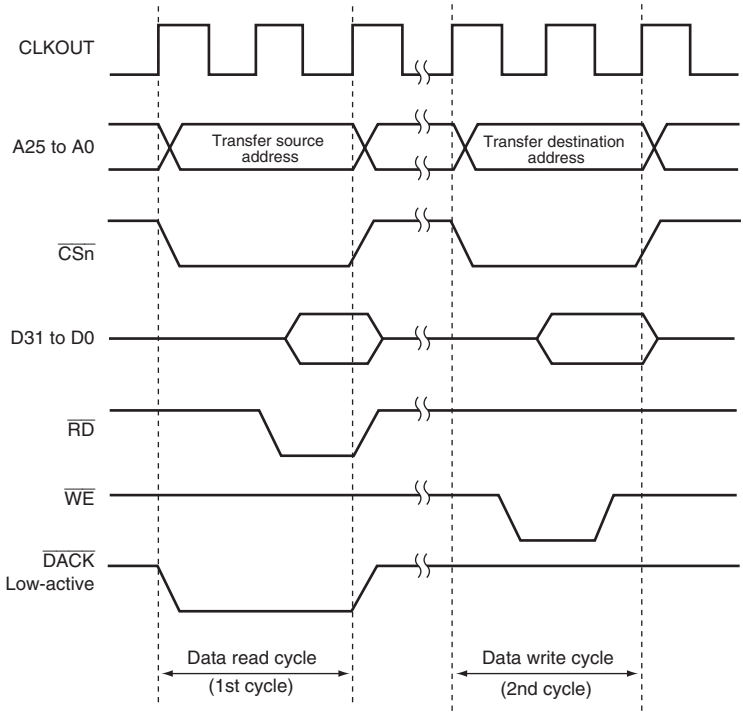
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 14.4 Data Flow of Dual Address Mode

Auto request, external request, and peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. CHCR can specify whether the DACK is output in read cycle or write cycle.

Figure 14.5 shows an example of DMA transfer timing in 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)

Bus Modes: There are two bus modes: cycle steal mode and burst mode. Select the mode in the TB and LCKN bits in CHCR. Moreover, cycle steal mode has normal and intermittent modes that are specified by the CMS bits in DMAOR.

- Cycle-Steal Mode
  - — Normal mode1 (DMAOR.CMS = 00, CHCR.LCKN = 0, CHCR.TB = 0)

    In cycle-steal normal mode, the SuperHyway bus mastership is given to another bus master after a one-transfer unit (byte, word, longword, 16-byte, or 32-byte unit) DMA transfer. When the next transfer request occurs, the DMAC issues the next transfer request, the bus mastership is obtained from the other bus master and a transfer is performed for one-transfer unit. When that transfer ends, the bus mastership is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

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

  Figure 14.6 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

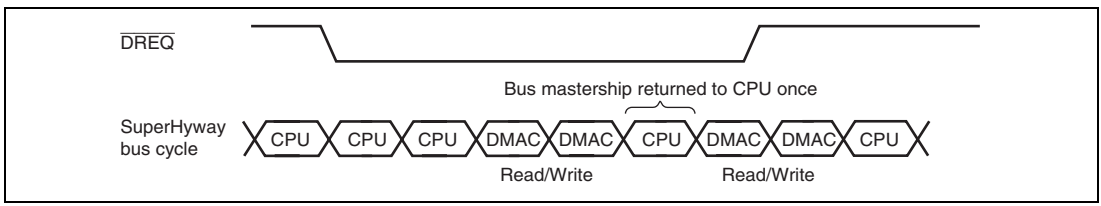


Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection)

  - — 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 mastership, is 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 mode 2.

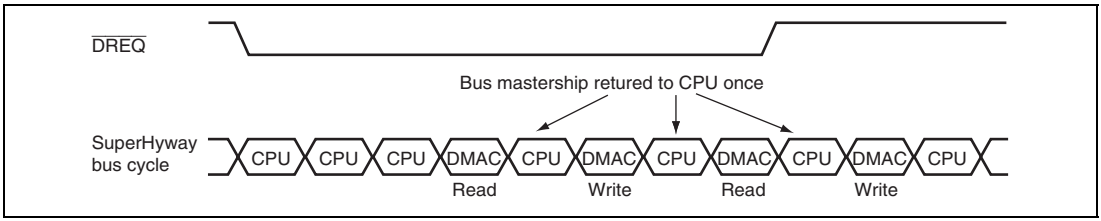


Figure 14.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection)

— Intermittent mode 16 (DMAOR.CMS = 10, CHCR.LCKN = 0 or 1, CHCR.TB = 0), intermittent mode 64 (DMAOR.CMS = 11, CHCR.LCKN = 0 or 1, CHCR.TB = 0)

In intermittent mode of cycle steal, the DMAC returns the SuperHyway bus mastership to other bus master whenever a one-transfer unit (byte, word, longword, or 16-byte or 32-byte unit) is complete. If the next transfer request occurs after that, the DMAC issues the next 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 unit and returns the bus mastership to other bus master. These operations are repeated until the 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 postponed in case of entry updating due to cache miss.

The intermittent modes, however, must be cycle steal mode in all channels 0 to 5 or 6 to11 for the corresponding transfer channel.

Figure 14.8 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:

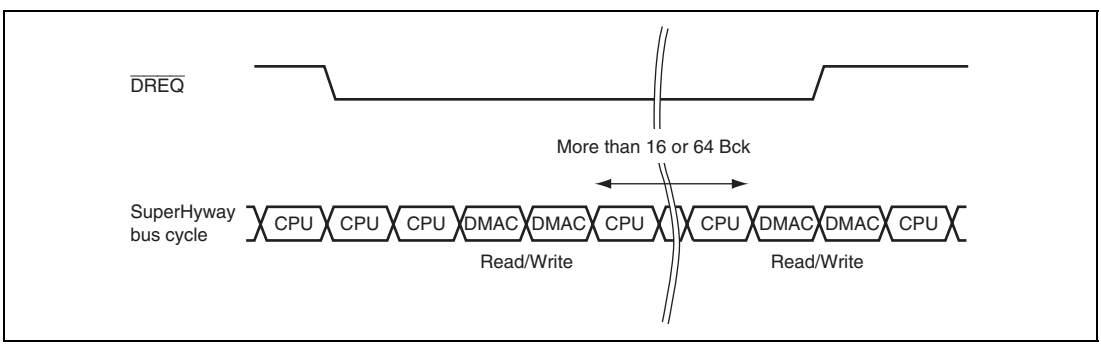


Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection)

- Burst Mode (LCKN = 0, TB = 1)

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

  Burst mode cannot be used when the peripheral module is the transfer request source.

Figure 14.9 shows DMA transfer timing in burst mode.

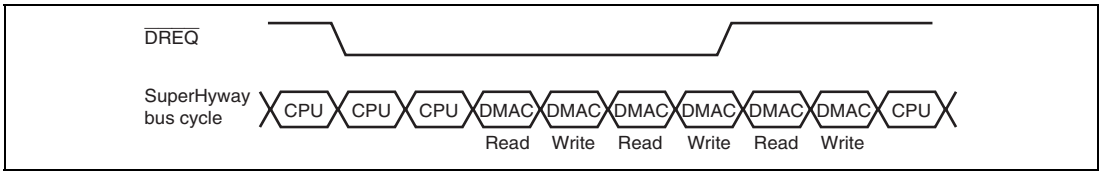


Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection)

DMA Transfer Matrix: Table 14.8 shows the DMA transfer matrix in auto-request mode and table 14.9 shows the DMA transfer matrix in external request mode, and table 14.10 shows the 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 RAM, SuperHyway RAM**
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 transfer size should be the same value of its access size.

Table 14.9 DMA Transfer Matrix in External Request Mode (only channels 0 to 3)

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**Peripheral module*[1]**	**L RAM, SuperHyway RAM**
LBSC space	Yes	Yes*[2]	Yes*[2]	Yes	Yes
DDRIF space	Yes*[3]	No	Yes*[4]	Yes*[3]	Yes*[3]
PCIC space	Yes*[3]	Yes*[5]	Yes*[6]	Yes*[3]	Yes*[3]
Peripheral module*[1]	Yes	Yes*[2]	Yes*[2]	Yes	Yes
L RAM, SuperHyway RAM	Yes	Yes*[2]	Yes*[2]	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 transfer size 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 address of 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 address of 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 destination 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 bit in CHCR is cleared to 0, when the transfer destination address is H'FD00 0000 to H'FDFF FFFF the AM bit in CHCR is set to 1.

Table 14.10 DMA Transfer Matrix in Peripheral module Request Mode

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**Peripheral module***	**L RAM, SuperHyway RAM**
LBSC space	No	No	No	Yes	No
DDRIF space	No	No	No	Yes	No
PCIC space	No	No	No	Yes	No
Peripheral module*	Yes	Yes	Yes	Yes	Yes
L RAM, SuperHyway RAM	No	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 transfer size should be the same value of its register access size.

The transfer source or the transfer destination should be a register of request source in peripheral module request mode. This transfer is available only cycle steal mode, and when the transfer request source is an peripheral module, the transfer is available in channel 0 to 5.

Bus Mode and Channel Priority: When the priority is set in fixed mode (CH0 > CH1) and channel 1 is transferring in burst mode, if there is a transfer request to channel 0 with a higher 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 continue after the channel 0 transfer has completely finished.

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the transfer of one transfer unit and the channel 1 transfer is continuously performed without releasing the bus mastership. The bus mastership will then switch between the two in the order channel 0, channel 1, channel 0, and channel 1.

This example is shown in figure 14.9. When multiple channels are operating in burst modes, the channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership will not be given to the bus master until all competing burst transfers are complete.

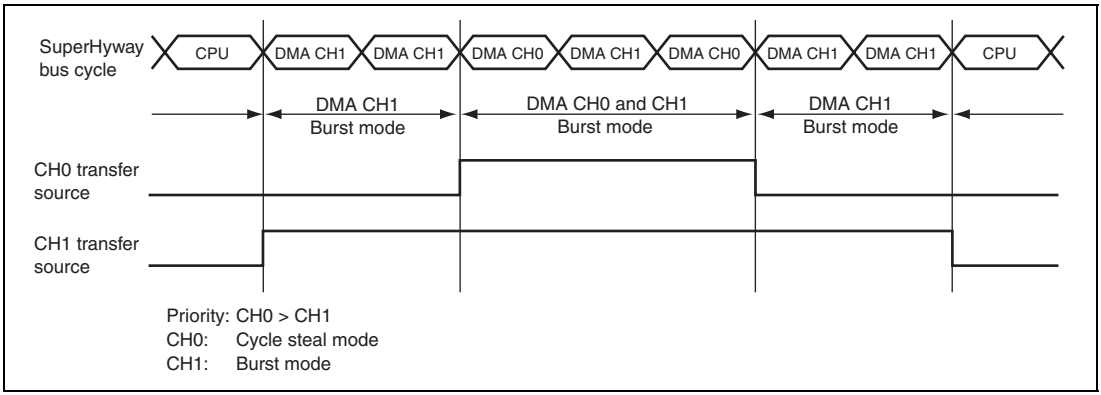


Figure 14.10 Bus State when Multiple Channels are Operating

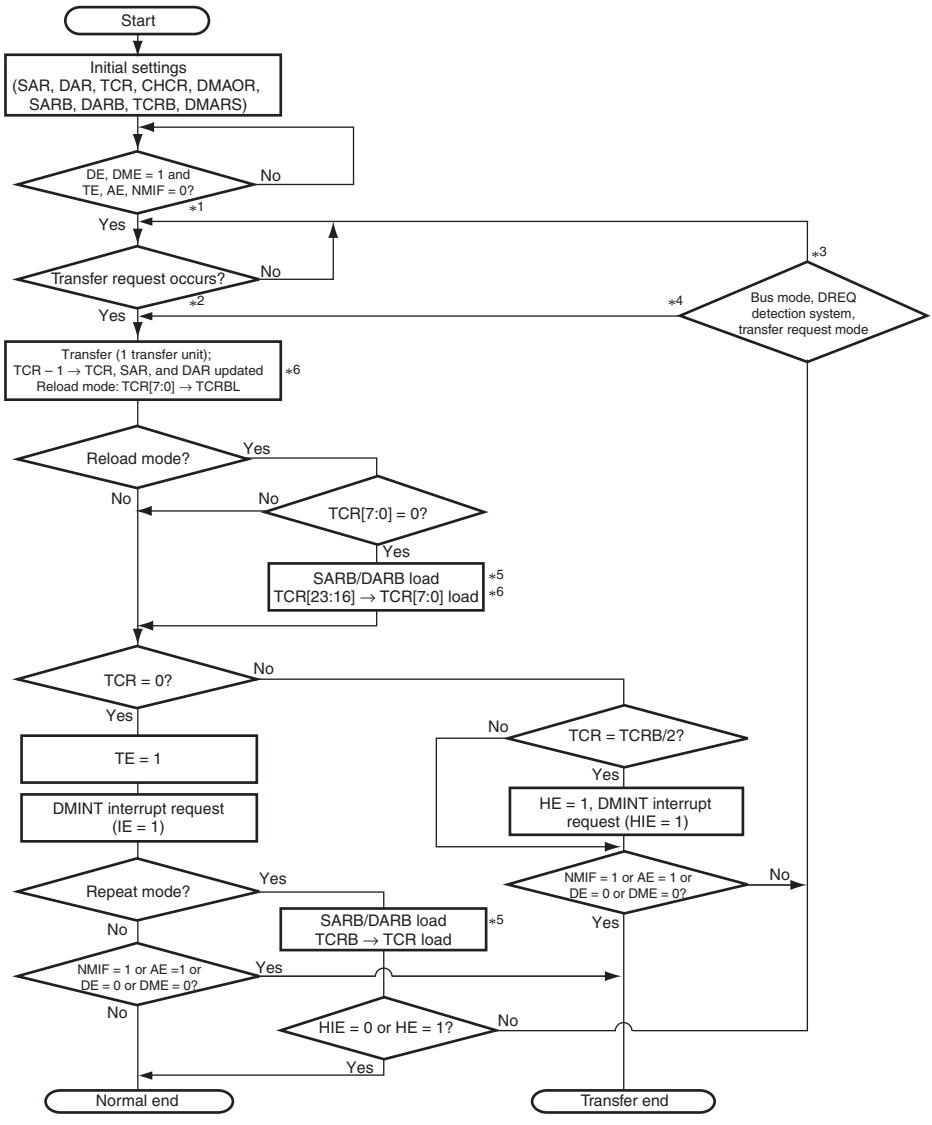
In round-robin mode, the priority changes according to the specification shown in figure 14.2. However, the channel in cycle steal mode cannot be mixed with the channel in burst mode.

14.4.4 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). In auto request mode, the transfer begins 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 bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DMINT interrupt is sent to the CPU.
4. When an address error or an NMI interrupt is generated, the transfer is aborted. Transfers are 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

14.4.5 Repeat Mode Transfer

In a repeat mode transfer, a DMA transfer is repeated without specifying the transfer settings every time before executing a transfer.

Using a repeat mode transfer with the half end function allows a double buffer transfer executed virtually. Following processings can be executed effectively by using a repeat mode transfer. As an example, operation of receiving voice data from the VOICE CODEC and compressing it is explained.

In the following example, processing of compressing 40-word voice data every data reception is 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 DME bit to 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, clear the 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, clear the TE bit to 0 and compress 40-word voice data from the address set in DAR + 40. After this operation, the value of DARB is copied to DAR in DMAC and initialized, and the value of 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 NMI interrupt is generated. Note that if the HE bit is not cleared in the procedure 3 or if the TE bit is

not cleared in the procedure 4, then the transfer is stopped according to the condition of both the HE and the TE bits are set to 1.

As explained above, a repeat mode transfer enables sequential voice compression by changing buffer for storing data received consequentially and a data buffer for processing signals alternately.

14.4.6 Reload Mode Transfer

In a reload mode transfer, according to the settings of bits RPT[2:0] in CHCR, the value set in SARB/DARB is set to SAR/DAR and the value of bits TCRB[23:16] is set in bits TCRB[7:0] at each transfer set in the bits TCRB[7:0], and the transfer is repeated until TCR becomes 0 without specifying the transfer settings again. A reload mode transfer is effective when repeating data transfer with specific area. Figure 14.12 shows the operation of reload mode transfer.

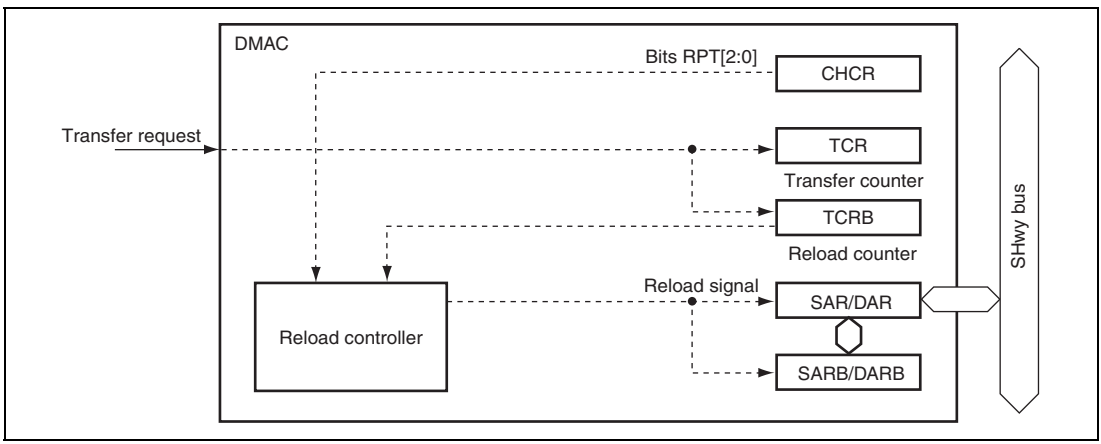


Figure 14.12 Reload Mode Transfer

When a reload mode transfer is executed, TCRB is used as a reload counter. Set TCRB according to section 14.3.6, DMA Transfer Count Registers B0 to B3, B6 to B9 (TCRB0 to TCRB3, TCRB6 to TCRB9).

14.4.7 DREQ Pin Sampling Timing

Figures 14.13 to 14.16 show the sample timing of the DREQ input in each bus mode, respectively.

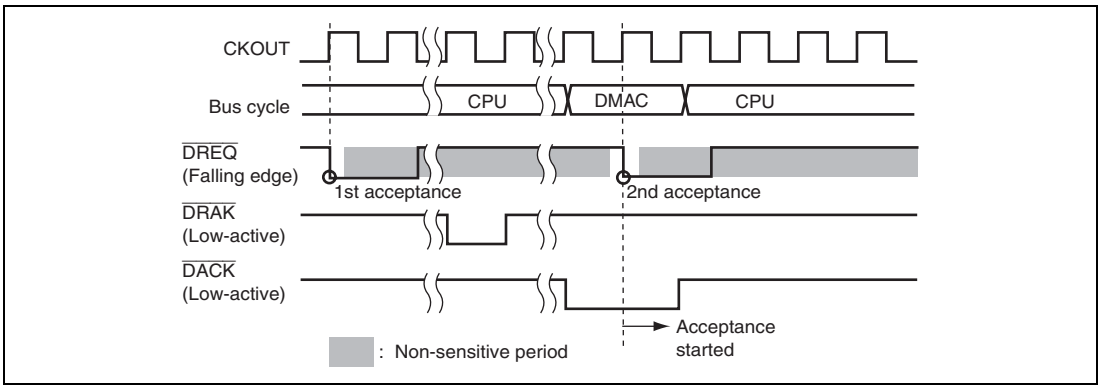


Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

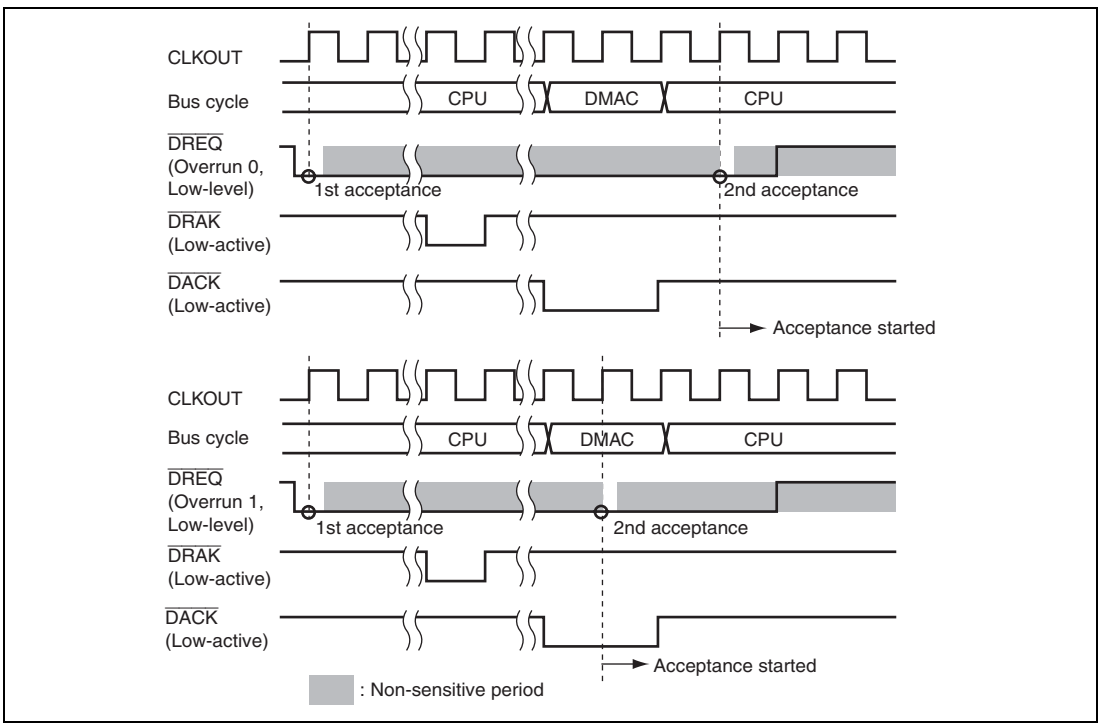


Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

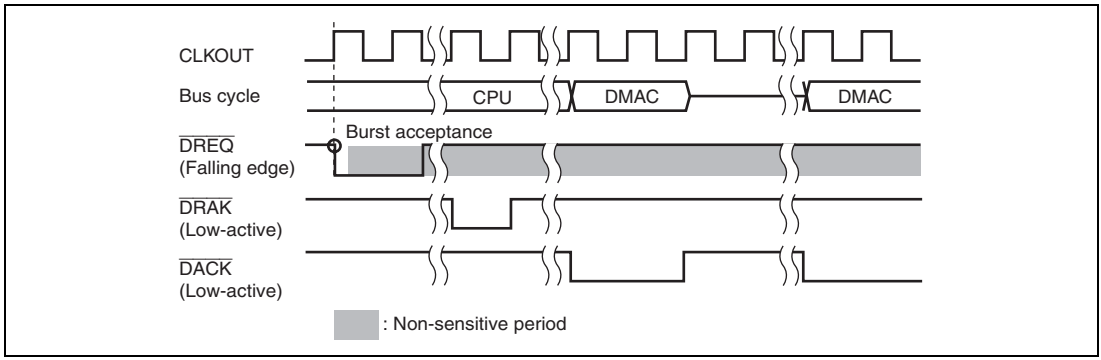


Figure 14.15 Example of DREQ Input Detection in Burst Mode Edge Detection

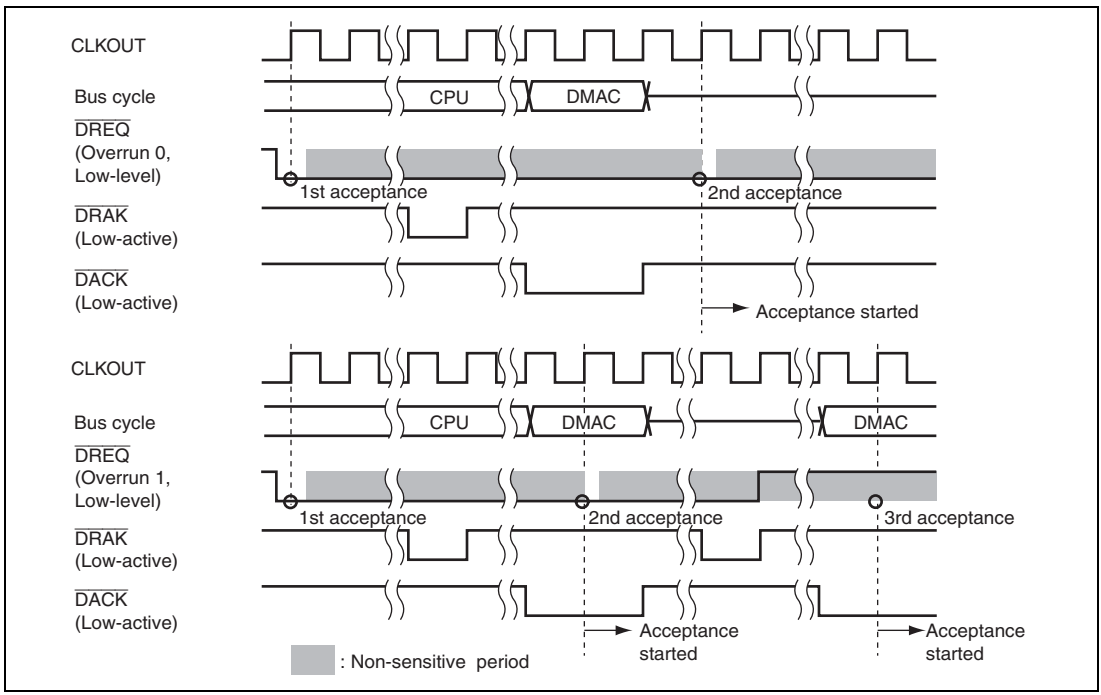


Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detection

14.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

14.5.1 Module Stop

While the DMAC is in operation, modules should not be stopped by setting MSTPCR (transition to the module standby state) .When modules are stopped, transfer contents cannot be guaranteed.

14.5.2 Address Error

When a DMA address error is occurred, after execute the following procedure, and then start a 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 transfer of corresponding channel has completed.

- Frequency should not be changed.
- Transition to sleep mode should not be made.

14.5.4 DACK output division

The DACK output is divided to align the data unit like the $\overline{\text{CSn}}$ output when a DMA transfer unit is divided with multiple bus cycles, for example when an 8-bit or 16-bit external device is accessed in longword units, or when an 8-bit external device is accessed in word units, and the $\overline{\text{CSn}}$ output is negated between these bus cycles.

14.5.5 Clear DMINT Interrupt

To ensure that a DMINT interrupt source that should have been cleared is not inadvertently accepted again, clear the BL bit after confirming the corresponding flag in INT2B3 register becomes 0 or issue the RTE instruction.

14.5.6 $\overline{\text{CS}}$ Output Settings and Transfer Size Larger than External Bus Width

When one DMA transfer is performed by multiple bus cycles*[1], the $\overline{\text{CSn}}$ output should be set not to negate between bus cycles*[2]. For detail of settings, refer to table 11.11 to 11.14. If set the $\overline{\text{CSn}}$ output is negated between bus cycles, the DREQ signal is not sampled correctly and malfunction may occur.

Notes: 1. When a DMA transfer is performed with larger transfer size than the bus width. For example, performing the 16-/32-byte transfer to the 8-/16-/32-bit bus width LBSC space, longword (32-bit) transfer to the 8-/16-bit bus width LBSC space, or word (16-bit) transfer to the 8-bit bus width LBSC space. Note that except for a 32-bit access to the MPX interface. This access generates only one bus cycle (burst).

2. When the $\overline{\text{CSn}}$ output is negated between bus cycles, then the DACK output is also 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 when the DREQ level detection with overrun 1 and the DREQ edge detection. In this case, the DMA transfer is suspended and do not perform correctly, to avoid this insert one or more idle cycle 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

The transfer destination is the LBSC space and the DACK is output during the write cycle:

(1) Set B'001 to B'111 (i.e., other than 000) to the IWW bits in CSnBCR

Note: * The transfer source is the LBSC space and the DACK is output during the read cycle or the transfer destination is the LBSC space and the DACK is output during the write cycle. And then specifies no idle cycle (CSnBCR.IWRRD, IWRRS, IWW are cleared to B'000). Note that the case that both the transfer source and the transfer destination are the LBSC spaces, does not apply this.

Table 14.11 to 14.14 shows the register settings that whether or not the negation of the DACK output with the number of bus cycle generation of the DMA transfer. The DACK is not negated 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 destination is the LBSC space, to avoid the DACK is asserted ceaselessly during between the two or more times DMA transfer, set B'001 to B'111 to the IWRRD, IWRRS or IWW bits in CSnBCR. In this setting, if the 16-byte DMA transfer is performed, multiple bus cycles are generated and the $\overline{\text{CSn}}$ is negated between bus cycles, the DREQ signal is not sampled correctly and malfunction may occur.

Table14.11 Register Settings for SRAM, Burst ROM, Byte Control SRAM Interface

			Register Settings of $\overline{CSn}$ is not negated	
Bus Width [bit]	**DMA Transfer Access Size**	**Bus Cycle Number**	**CSnBCR.IWRRD, IWRRS or IWW**	**CSnWCR.ADS and ADH**
8	Byte	1	Any	Any
	Word	2	Any	B'000
	Longword	4	Any	B'000
	16-Byte	16	B'000	B'000
	32-Byte	32	Any	B'000
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

Table14.12 Register Settings for PCMCIA Interface

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Settings of $\overline{CSn}$ is not negated
			CSnBCR.IWRRD,IWRRS or IWW
8	Byte	1	Any
	Word	2	Any
	Longword	4	Any
	16-Byte	16	B'000
	32-Byte	32	Any
16	Byte	1	Any
	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 $\overline{CSn}$ is not negated
			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 $\overline{CSn}$ is not negated	
			CSnBCR.IWW	CSnWCR.IW[1:0]
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

Section 15 Clock Pulse Generator (CPG)

The CPG generates clocks provided to both the inside and outside of the SH7780, and controls the power-down mode function. The CPG comprises a crystal oscillator circuit, PLLs, and a divider.

15.1 Features

The CPG has the following features.

- Generates SH7780 internal clocks

  SH7780 internal clocks are: the CPU clock (Ick) which is used in the CPU, FPU, cache, and TLB; the SHwy clock (SHck) which is used by the SuperHyway bus; and peripheral clocks (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 the external 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 controlled 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 modules can be stopped.

Figure 15.1 is a block diagram of the CPG.

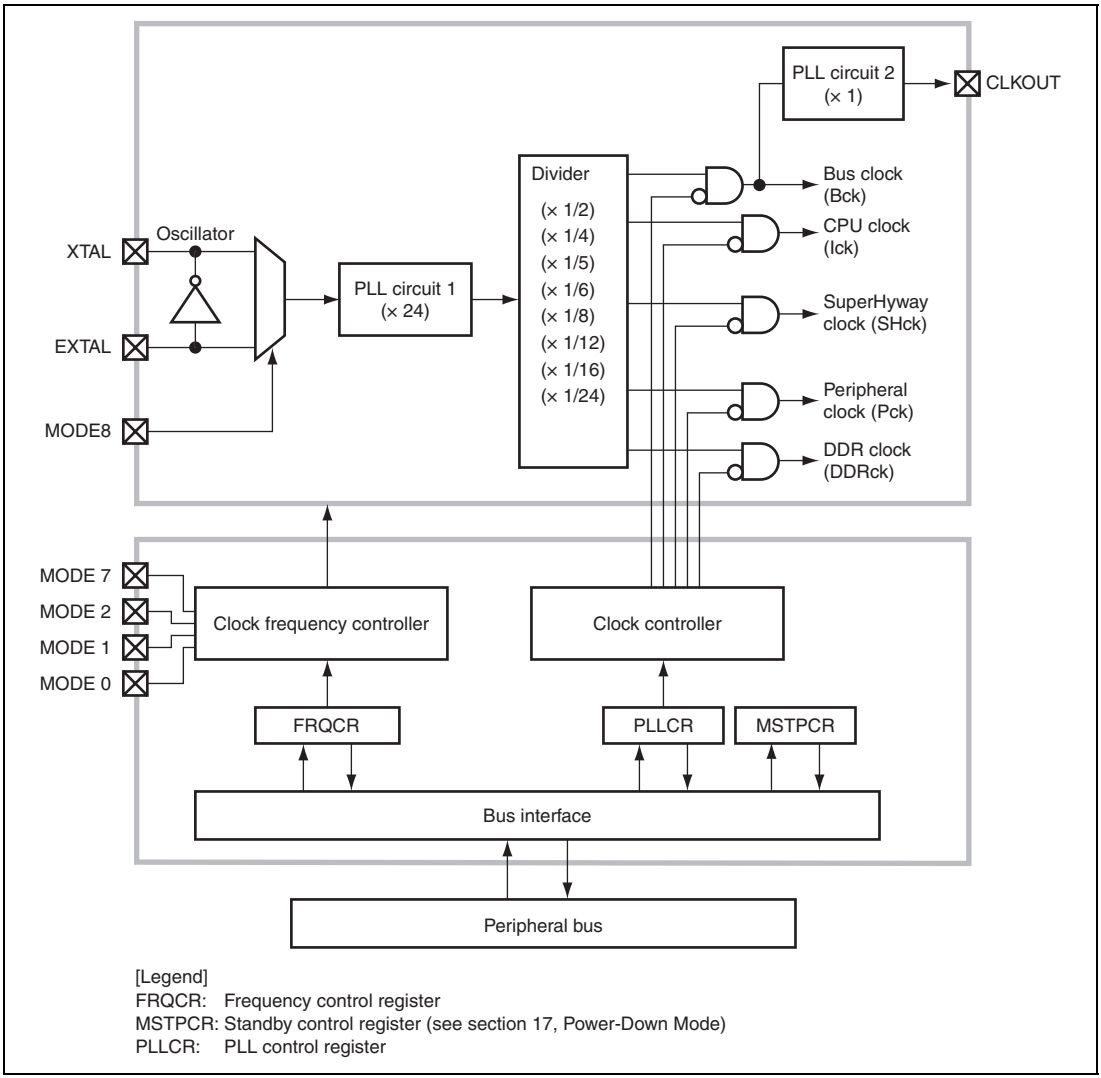


Figure 15.1 Block Diagram of CPG

The function of each block is described below.

- PLL circuit 1

  PLL circuit 1 multiplies the frequency of the external crystal oscillator and the clock input on the EXTAL pin by 24.
- PLL circuit 2

  PLL circuit 2 coordinates the phases of the bus clock (Bck) and the clock signal output from the CLKOUT pin that is used as the external peripheral interface clock.
- Crystal oscillator circuit

  Used when a crystal resonator is connected to the XTAL and EXTAL pins. Use of the crystal oscillator circuit can be selected with the MODE8 pin.
- Divider

  The divider generates the CPU clock (Ick), SuperHyway clock (SHck), on-chip peripheral module clock (Pck), DDR memory clock (DDRck), and external bus clock (Bck). The division ratio is selected by the mode pin MODE0, MODE1, MODE2, MODE7.

15.2 Input/Output Pins

Table 15.1 lists the CPG pin configuration.

Table 15.1 CPG Pin Configuration

Pin Name	Function	I/O	Description
MODE0, MODE1, MODE2, and MODE7*[1]	Mode Control Pins 0,1,2,7	Input	Select the clock operating mode of after power-on reset.
MODE8*[2]	Mode Control Pins 8	Input	Selects use/non-use of crystal resonator. When MODE8 is "low", external clock is input from the EXTAL pin. When MODE8 is "high", crystal resonator connected directly to the EXTAL and XTAL pins.
XTAL	Clock Pins	Output	This pin is connected to a crystal resonator.
EXTAL		Input	This pin is connected to a crystal oscillator to input an external clock or is connected to a crystal resonator.
CLKOUT		Output	This pin is used to output the external bus clock.

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.

15.3 Clock Operating Modes

The correspondence between settings of the mode control pins (MODE7 and MODE2 to MODE0) and clock operating modes after power-on reset is shown in table 15.2.

Table 15.2 Clock Operating Modes

Clock Operating Mode	Mode Control Pin Setting*				PLL1, PLL2	Frequency Multiplication Ratio (to Input Clock)					FRQCR Initial Value
	MODE7	MODE2	MODE1	MODE0		Ick	SHck	Pck	DDRck	Bck	
0	Low	Low	Low	Low	On	× 12	× 6	× 3/2	× 24/5	× 3	H'1023 3335
1	Low	Low	Low	High	On	× 12	× 6	× 1	× 24/5	× 2	H'1024 4336
2	Low	Low	High	Low	On	× 12	× 6	× 3/2	× 24/5	× 3/2	H'1025 5335
3	Low	Low	High	High	On	× 12	× 6	× 1	× 24/5	× 1	H'1026 6336
12	High	High	Low	Low	On	× 12	× 4	× 1	× 4	× 2	H'1044 4346

Note: Other than above: setting prohibited.

15.4 Register Descriptions

Table 15.3 shows the CPG register configuration. Table 15.4 shows the register states in each processing mode.

Table 15.3 Register configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync clock
Frequency control register	FRQCR	R/W	H'FFC8 0000	H'1FC8 0000	32	Pck
PLL control register	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32	Pck
Standby control register	MSTPCR	R/W	H'FFC8 0030	H'1FC8 0030	32	Pck

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 $\overline{\text{PRESET}}$ Pin	Power-on Reset by WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep by Sleep Instruction
Frequency control register	FRQCR	H'1xxx x3xx*[2]	H'1xxx x3xx*[2]	Retained	Retained
PLL control register	PLLCR	H'0000 E001	Retained	Retained	Retained
Standby control register*[1]	MSTPCR	H'0000 0000	Retained	Retained	Retained

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 setting (See table 15.2).

15.4.1 Frequency Control Register (FRQCR)

FRQCR is a 32-bit readable/writable register that selects the frequency division ratio of the SuperHyway clock (SHck), the peripheral clock (Pck), the DDR clock (DDRck) and the bus clock (Bck). Refer to the clock operating mode table about the frequency multiplication ratio. FRQCR can only be accessed in longwords. FRQCR is initialized by a power-on reset via the $\overline{\text{PRESET}}$ pin and WDT over-flow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	IFC0*	CFC[3:0]*				BFC[3:0]*			
Initial value:	0	0	0	1	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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	P1FC[3:0]*			
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	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 write value should always be 0001.
27 to 25	—	000	R	Reserved These bits are always read as 0. The write value should always be 0. Writing to other than 000, the operation of this LSI is not guaranteed.
24	IFC0*	Undefined	R/W	CPU Clock (Ick) and SuperHyway Clock (SHck) Frequency Division Ratio Setting 00010: ×12 (Ick), ×6 (SHck) Clock operating mode 0, 1, 2 or 3 (after power-on reset) 00100: ×12 (Ick), ×4 (SHck) Clock operating mode 12 (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 reset depends on the mode pins setting (see table 15.2).
23	CFC3*	0	R/W	
22	CFC2*	Undefined		
21	CFC1*	Undefined		
20	CFC0*	0		

Bit	Bit Name	Initial Value	R/W	Description
19	BFC3	0	R	Bus Clock (Bck) Frequency Division Ratio Setting
18	BFC2	Undefined	R	0011: ×3
17	BFC1	Undefined	R	0100: ×2
16	BFC0	Undefined	R	0101: ×3/2
				0110: ×1
				Other than above: Setting prohibited
				The initial value of this field after power-on reset depends on the mode pin setting (see table 15.2). Writing is ignored.
15 to 12	—	0	R	Reserved
		Undefined		The initial value of this field after power-on reset depends on the mode pins setting (see table 15.2). Writing is ignored.
		Undefined		
		Undefined		
11 to 8	—	0011	R	Reserved
				These bits are always read as 0011. The write value should always be 0011.
7 to 4	—	0	R	Reserved
		Undefined		The initial value of this field after power-on reset depends on the mode pins setting (see table 15.2). Writing is ignored.
		Undefined		
		Undefined		
3	P1FC3	0	R	Indicates the division ratio of the external bus clock frequency.
2	P1FC2	1	R	
1	P1FC1	Undefined	R	0101: ×3/2
0	P1FC0	Undefined	R	0110: ×1
				The initial value of this field after power-on reset depends on the mode pin setting (see table 15.2). Writing is ignored.

Note: * Bits IFC and CFC in FRQCR should be modified together.

15.4.2 PLL Control Register (PLLCR)

PLLCR is a 32-bit readable/writable register that controls the clock output on the CLKOUT pin. This register can only be accessed in longwords.

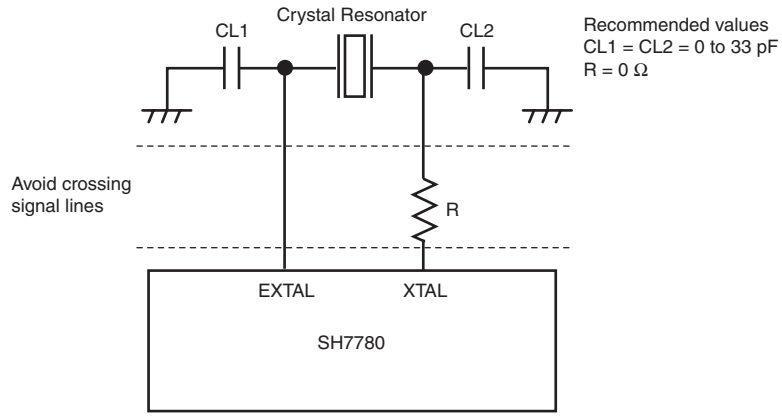
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKOFF	—
Initial value:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	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 value should always be 0.
15 to 13	—	All 1	R	Reserved This bit is always read as 1. The write value should always be 1. Writing 0 to any of these bits, the operation of this LSI is not guaranteed.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value 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 should always be 1.

15.5 Notes on Board Design

When Using Crystal Resonator: Place the crystal resonator and capacitors close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, ensure that no other signal lines cross the signal lines for these pins.



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 pin.

When Using PLL and DLL circuit: Separate each VDD-PLL, VDD-DLL, VSS-PLL, and VSS-DLL from the other VDD and VSS lines at the board power supply source, and insert resistors (RCB and RD) and bypass capacitors (CPB and CD) close to the PLL and DLL pins as noise filters.

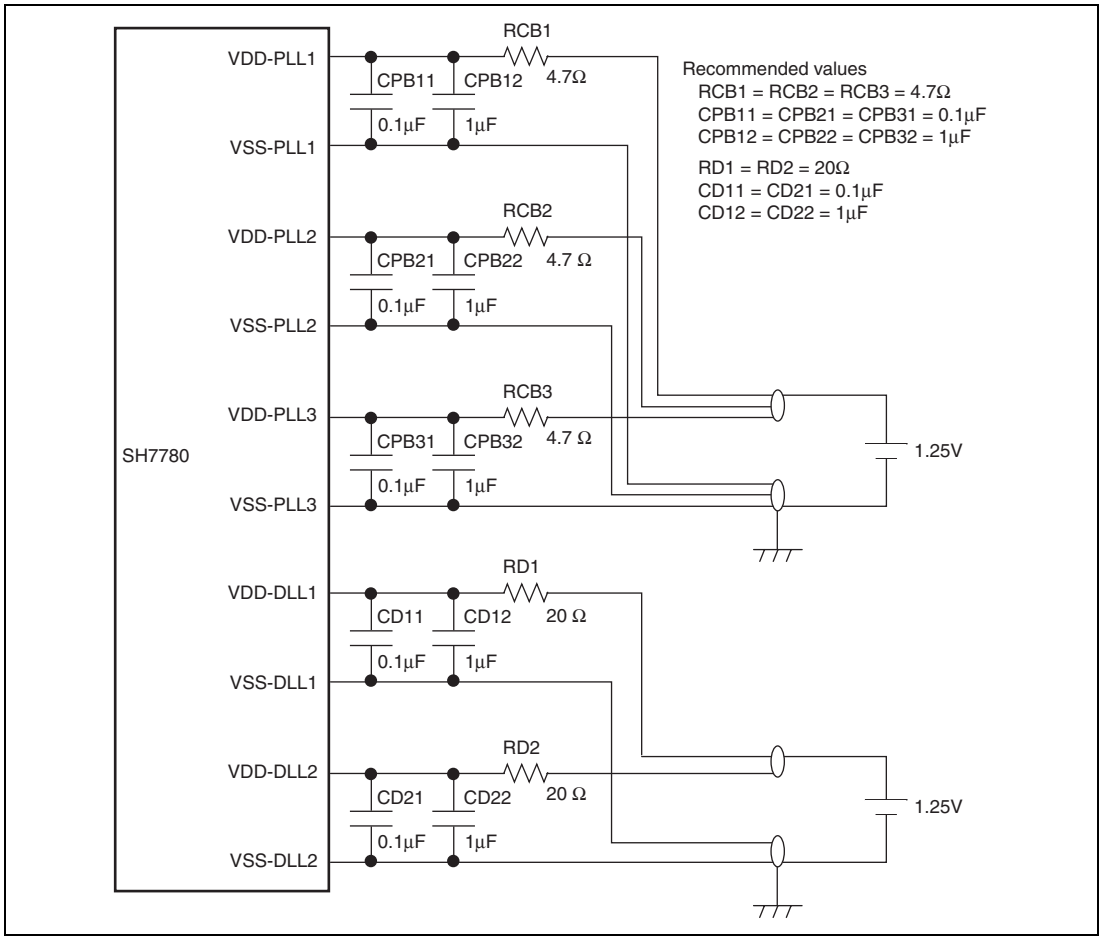


Figure 15.3 Points for Attention when Using PLL and DLL Circuit

Section 16 Watchdog Timer and Reset

The reset and watchdog timer (WDT) control circuit comprises the reset control unit and WDT control unit which control the power-on reset sequence and a reset for on-chip peripheral modules and external devices.

The WDT is a one-channel timer which can be used as the watchdog timer or interval timer.

16.1 Features

- The watchdog timer unit monitors a system crash using a timer counting at specified intervals.
- The watchdog timer unit generates a reset for on-chip peripheral modules when a WDT overflow occurs.
- A power-on reset or a manual reset can be selectable, when a manual reset is selected, the $\overline{\text{MRESETOUT}}$ pin is asserted.
- Generates the interval timer interrupt when counter overflow occurs in interval timer mode.
- The maximum time until the watchdog timer overflows is approximately 21 seconds (when the peripheral clock Pck is 50 MHz).
- Writing to WDT-related registers is not normally allowed. A specified code in the upper bits of write data enables writing to the registers.
  WTCNT and WTCSR differ from other registers in being more difficult to write to. The procedure for writing to these registers is given below.

Figure 16.1 shows a block diagram of the WDT.

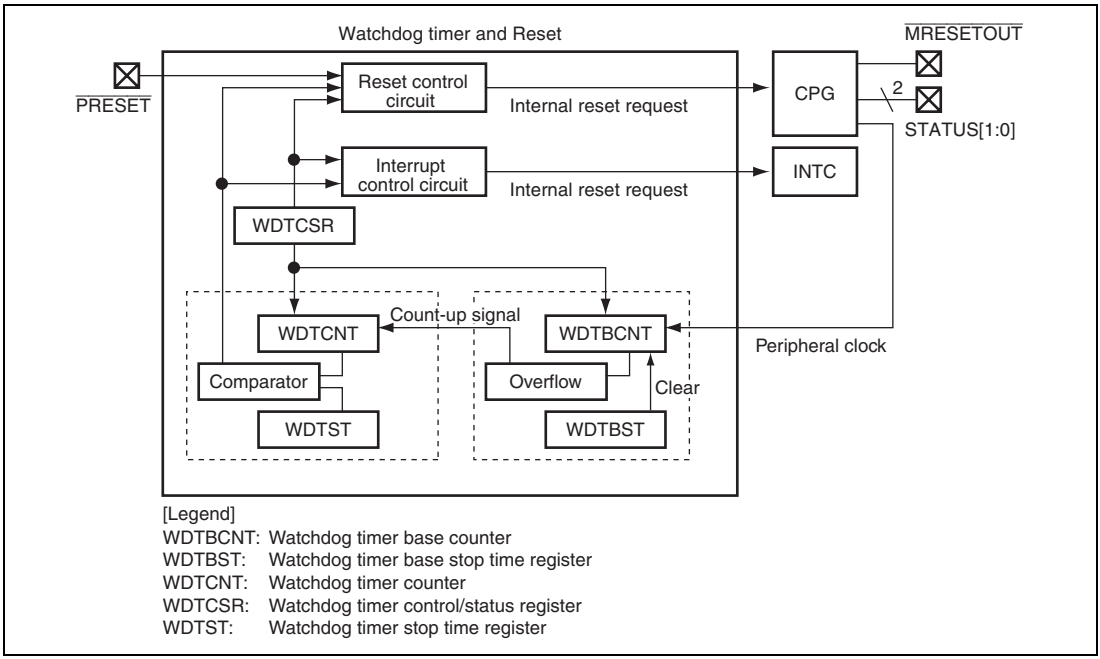


Figure 16.1 Block Diagram of WDT

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the reset control unit.

Table 16.1 Pin Configuration

Pin name	Function	I/O	Description		
$\overline{\text{PRESET}}$	Reset	Input	Power-on reset		
$\overline{\text{MRESETOUT}}$*1	Manual reset output	Output	Low level output during manual reset execution		
STATUS1*2	Processing state 1	Output	Indicate the processor's operating status		
STATUS0*3	Processing state 0		STATUS1	STATUS0	Operating Status
			High	High	Reset
			High	Low	Sleep mode
			Low	Low	Normal operation

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.

16.3 Register Descriptions

Table 16.2 shows the registers of the reset and watchdog timer. Table 16.3 shows the register states in each processing mode.

Table 16.2 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
Watchdog timer stop time register	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32	Pck
Watchdog timer control/status register	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32	Pck
Watchdog timer base stop time register	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32	Pck
Watchdog timer counter	WDTCNT	R	H'FFCC 0010	H'1FCC 0010	32	Pck
Watchdog timer base counter	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32	Pck

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	Sleep by SLEEP Instruction
Watchdog timer stop time register	WDTST	H'0000 0000	Retained	Retained	Retained
Watchdog timer control/status register	WDTCSR	H'0000 0000	Retained	Retained	Retained
Watchdog timer base stop time register	WDTBST	H'0000 0000	Retained	Retained	Retained
Watchdog timer counter	WDTCNT	H'0000 0000	Retained	Retained	Retained
Watchdog timer base counter	WDTBCNT	H'0000 0000	Retained	Retained	Retained

16.3.1 Watchdog Timer Stop Time Register (WDTST)

WDTST is a readable/writable 32-bit register that specifies the time until a watchdog timer overflows. The time until WDTCNT overflows becomes the minimum value when set H'001 to the bits 11 to 0, and the maximum value when set H'000 to the bits 11 to 0. Use a longword access to write to the WDTST, with H'5A in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTST											
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/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	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value must be H'5A.
23 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTST	All 0	R/W	Counter value

16.3.2 Watchdog Timer Control/Status Register (WDTCSR)

WDTCSR is a readable/writable 32-bit register that comprises the timer mode-selecting bit and overflow flags. Use a longword access to write to the WDTCSR, with H'A5 in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TME	WT/IT	RSTS	WOVF	IOVF	—	—	—
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/W	R/W	R/W	R/W	R/W	R	R	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 to this register, the write value must be H'A5.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TME	0	R/W	Timer Enable Specifies starting and stopping of timer operation. 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 watchdog timer or interval timer. Up counting may not be performed correctly if this bit is modified while the WDT is running. 0: Interval timer mode 1: Watchdog timer mode

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	0	R/W	Reset Select Specifies the kind of reset to be performed when WDTCNT overflows in watchdog timer mode. This setting is ignored in interval timer mode. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Flag Indicates that WDTCNT has overflowed in watchdog timer mode. This flag is not set in interval timer mode. 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 interval timer mode. This flag is not set in watchdog timer 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 value 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 write access to clear the WDTBCNT, with H'55 in the bits 31 to 24. The reading value of this register is always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	(Given code)								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

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

16.3.4 Watchdog Timer Counter (WDTCNT)

WDTCNT is a 32-bit read-only register that comprises 12-bit watchdog timer counter and counts up on the WDTBCNT overflow signal. When WDTCNT overflows, a reset is generated in watchdog timer mode, or an interrupt is generated in interval timer mode. Writing to WDTCNT is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTCNT											
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	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 the peripheral clock (Pck). When WDTBCNT overflows, WDTCNT is counted up and WDTBCNT is cleared to 0. Writing to WDTBCNT is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTBCNT	
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBCNT															
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	R	R	R

16.4 Operation

16.4.1 Reset request

Power-on reset and manual reset are available. These sources are follows.

(1) Power-on reset

- Input low level via $\overline{\text{PRESET}}$ pin.
- The WDTCNT overflows when the WT/IT bit in the WTCSR is 1, and the RSTS bit is 0.
- The H-UDI reset occurs (for details, see section 30, User Debugging Interface (H-UDI)).

```
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 in SR
- When the WDTCNT overflows while the WT/IT bit and the RSTS bit are set to 1 in WTCSR.

```
Manual_reset()
{
    EXPEVT = H'0000 0020;
    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(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 periodically so that WDTCNT does not overflow. See section 16.4.5, Clearing WDT Counter for WDT 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 WDTCNT 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 generated each 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 sends an interval timer interrupt (ITI) request to INTC. The counter continues counting.

Figure 16.2 shows a WDT counting up operation.

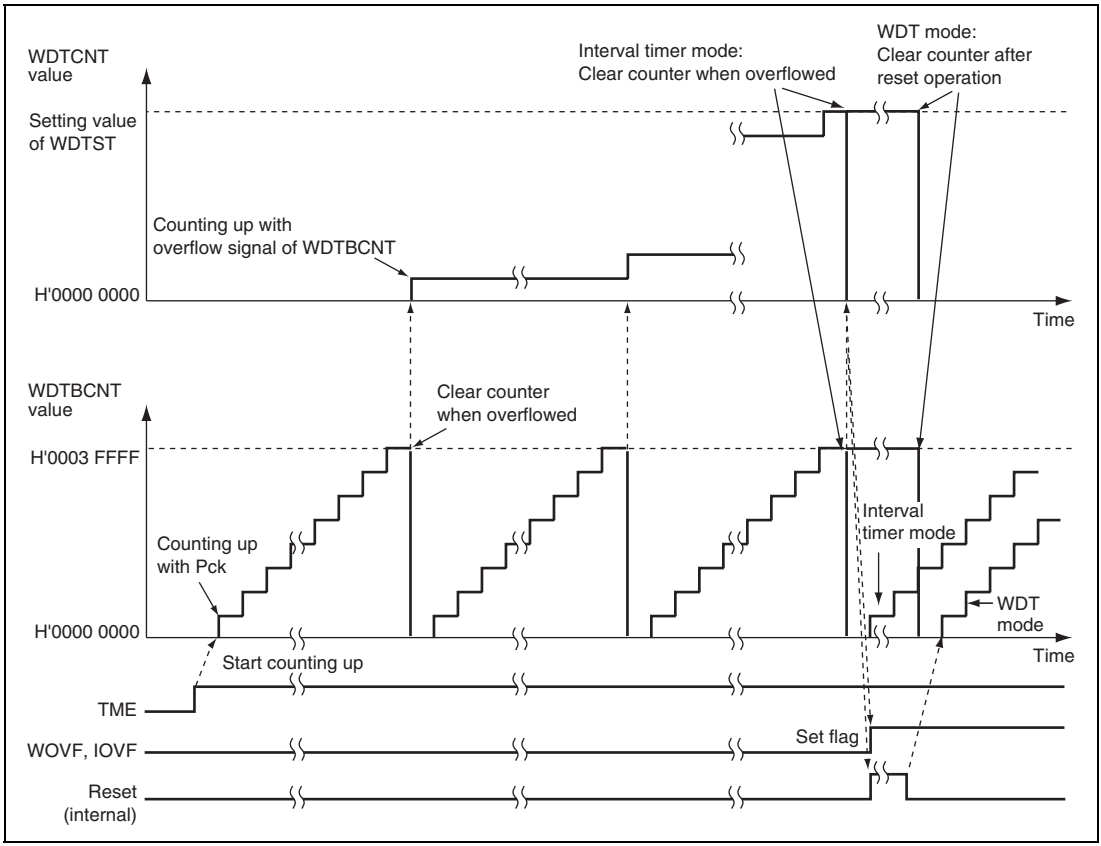


Figure 16.2 WDT Counting Up Operation

16.4.4 Time for WDT Overflow

WDTBCNT is a 18-bit up-counter operated on the peripheral clock (Pck). WDTBCNT is cleared 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 approximately 5.243 ms (= 2^18 [bit] × 1/50 [MHz]).

WDTCNT is a 12-bit counter, starts count up operation when overflow occurs in WDTBCNT. The time until WDTCNT overflows becomes the maximum value when H'000 are set to WDTST.

Where the peripheral clock frequency is 50 MHz, the maximum overflow time is approximately 21.475 s (= 2^12 [bit] × 5.243 [ms]).

And the time until WDTCNT overflows becomes the minimum value when H'001 is set to WDTST. The minimum overflow time is approximately 5.243 ms (= 2^1 [bit] × 5.243 [ms]).

16.4.5 Clearing WDT Counter

Writing H'55 to WDTBST with longword access clears WDTBCNT and writing the overflow setting value to WDTST clears WDTCNT.

16.5 Status Pin Change Timing during Reset

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-on reset state by the $\overline{\text{PERSET}}$ pin low level input and then it is necessary to ensure the synchronization settling time of the PLL circuit. Therefore, do not input high level to the $\overline{\text{PRESET}}$ pin during the synchronization settling time of the PLL. The PLL synchronization settling time is the total value 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 is continued during the reset holding time in the LSI. The reset holding time is 20 clock cycles of the XTAL clock and thereafter equal to or more than 45 clock cycles of the peripheral clock (Pck).

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and that indicates a normal operation is synchronous with the peripheral clock (Pck) and asynchronous 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 the $\overline{\text{TRST}}$ pin input level should be low level to initialize the H-UDI.

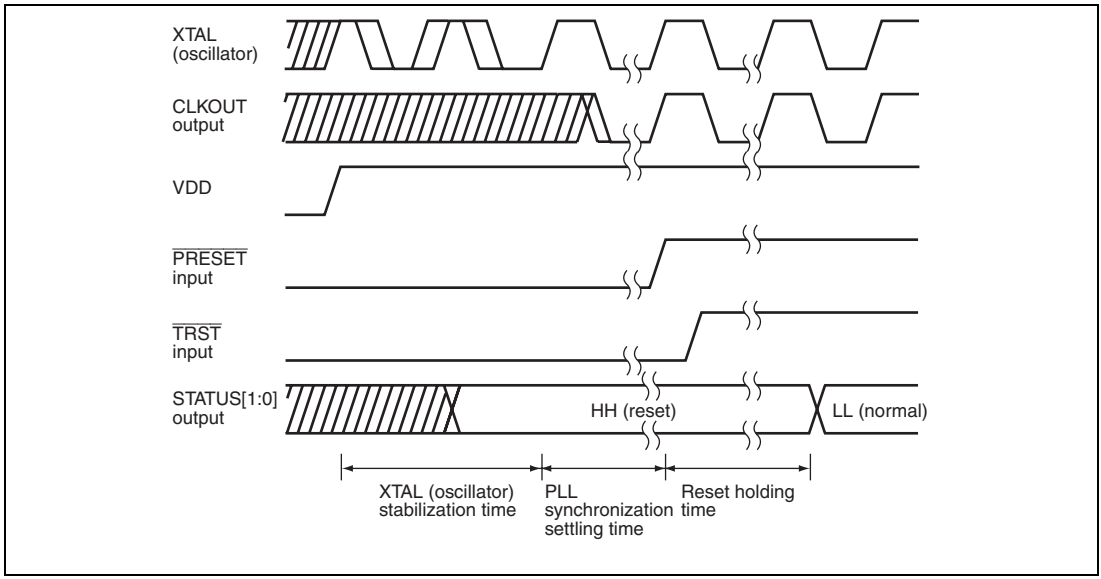


Figure 16.3 STATUS Output during Power-on

PRESET input during normal operation

It is necessary to ensure the PLL synchronization settling time when the PRESET input during normal operation.

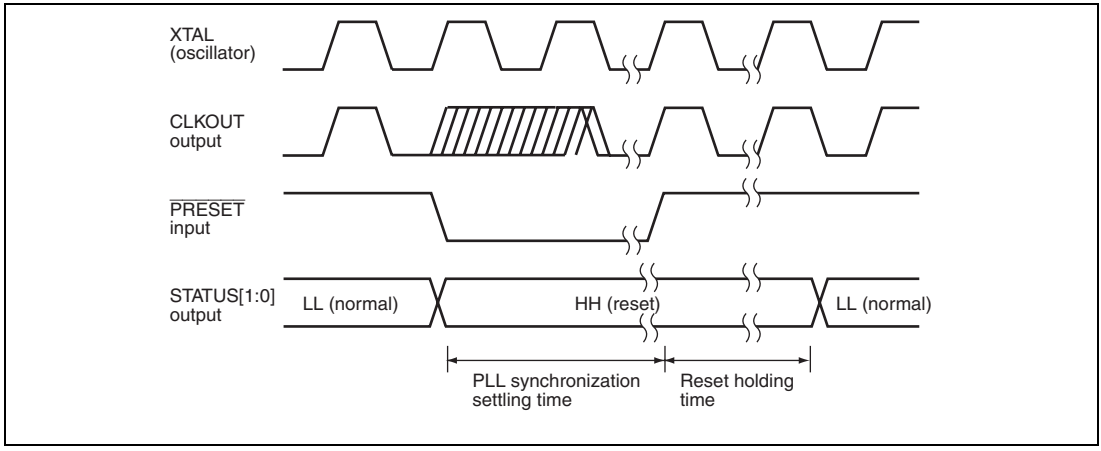


Figure 16.4 STATUS Output by Reset input during Normal Operation

$\overline{\text{PRESET}}$ input during Sleep Mode

It is necessary to ensure the PLL oscillation time when power-on reset generates by the $\overline{\text{PRESET}}$ pin low revel input during sleep mode.

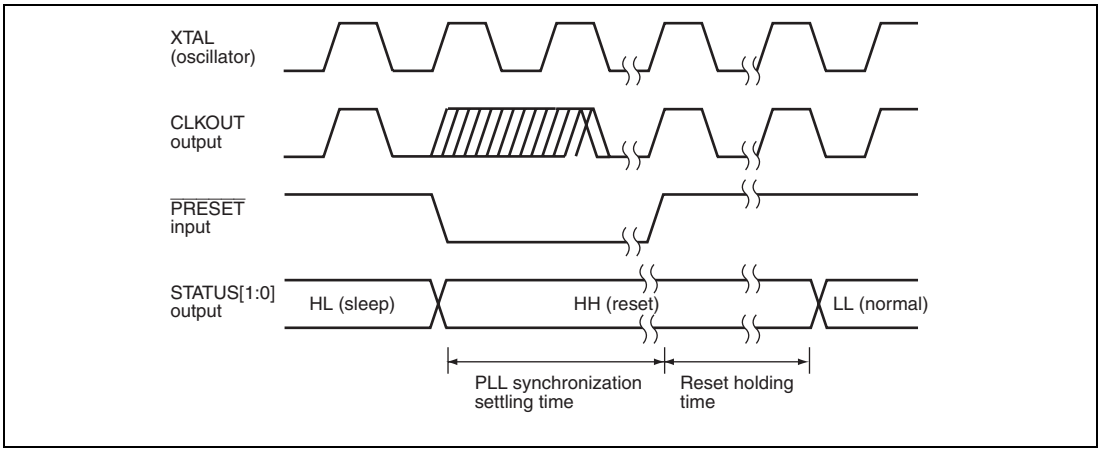


Figure 16.5 STATUS Output by Reset input during Sleep Mode

16.5.2 Power-On Reset by Watchdog Timer Overflow

The transition time from the watchdog timer overflowed to the power-on reset state (watchdog timer reset setup time) is 1 clock cycle of the XTAL clock and thereafter equal to or more than 5 clock cycles of the peripheral clock (Pck).

The power-on reset time (watchdog timer reset holding time) by the watchdog timer overflowed is 3774 clock cycles of the XTAL clock and thereafter equal to or more than 45 clock cycles of the peripheral clock (Pck).

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the XTAL clock and the CLKOUT pin output clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck).

Power-On Reset by Watchdog timer Overflowed in Normal Operation

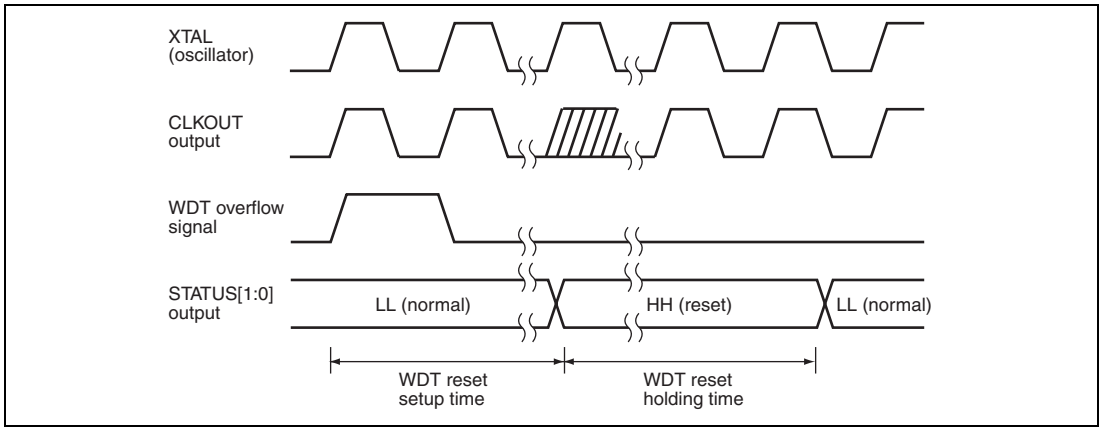


Figure 16.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation

Power-On Reset by Watchdog timer Overflowed in Sleep Mode

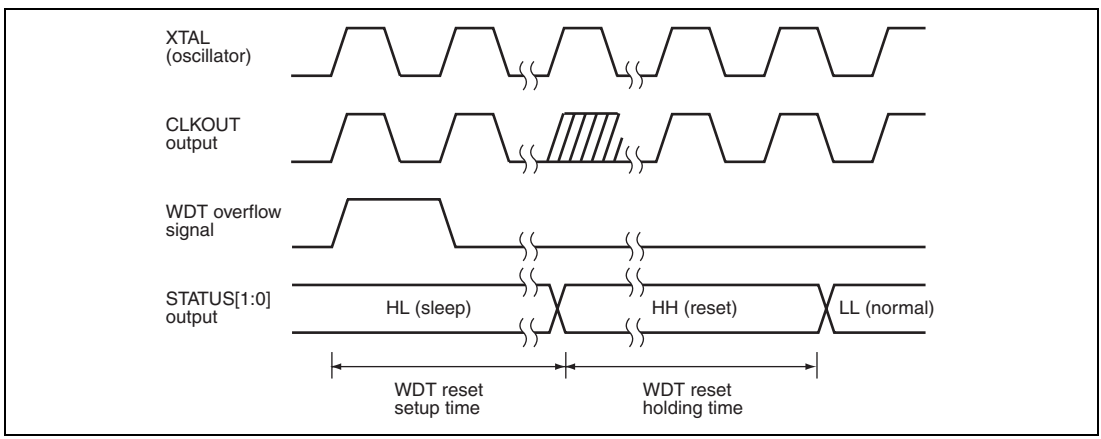


Figure 16.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode

16.5.3 Manual Reset by Watchdog Timer Overflow

The transition time from watchdog timer overflowed to manual reset state (watchdog timer reset setup time) is 1 clock cycle of the XTAL clock and thereafter equal to or more than 5 clock cycles of the peripheral clock (Pck).

The manual reset time (watchdog timer manual reset holding time) by the watchdog timer overflowed is equal to or more than 3774 clock cycles of the XTAL clock.

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the XTAL clock and the CLKOUT pin output clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck).

Manual Reset by Watchdog timer Overflowed in Normal Operation

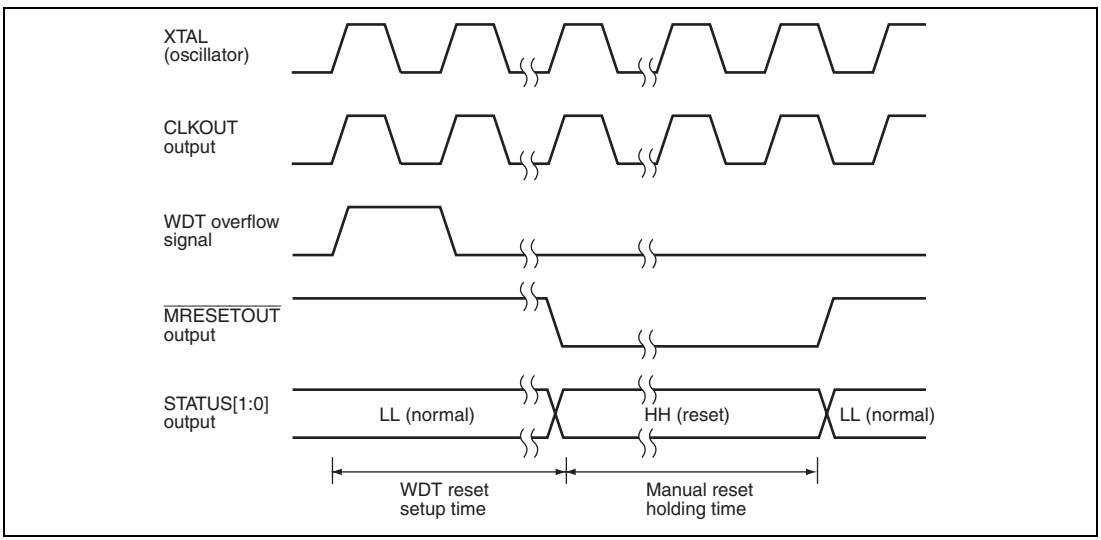


Figure 16.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation

Manual Reset by Watchdog timer Overflowed in Sleep Mode

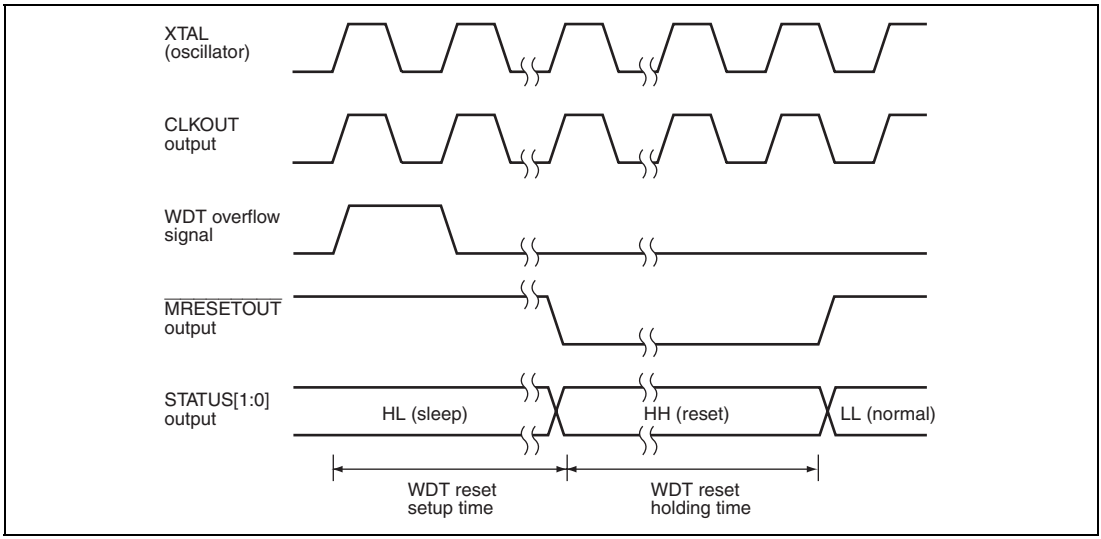


Figure 16.9 STATUS Output by Watchdog timer overflow Manual Reset during Sleep Mode

Section 17 Power-Down Mode

In power-down modes, some of the on-chip peripheral modules and the CPU functions are halted, enabling power consumption to be reduced.

17.1 Features

The SH7780 power-down mode has the following features.

- Supports sleep mode and module standby mode
- Supports RTC power supply backup mode where the power supply for only the RTC is held and other power supplies are turned off
- Supports DDR-SDRAM power supply backup mode where the power supply for only the 2.5-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 state to a power-down mode, states of the CPU and on-chip peripheral modules in each mode, and methods to leave each power-down mode.

Table 17.1 Power-Down Modes

		State								
					On-Chip Module					
Power-Down Mode	Transition Condition	CPG	CPU	On-Chip Memory	DMAC	RTC	Others	Pin	DDR-SDRAM	Cancellation
Sleep	SLEEP instruction executed	Operated	Halted (register contents retained)	Retained	Operated	Operated	Operated	Retained	Auto-refresh or self-refresh*3	- Interrupt - Power-on reset - manual reset
Module standby	Corresponding bit in MSTPCR set to 1 (see section 17.3.1)	Operated	Operated	Retained	Selected modules halted	Operated	Selected modules halted	Retained	Auto-refresh or self-refresh	Clear corresponding bit in MSTPCR to 0 (see section 17.3.1)
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	Power-on reset
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)	Power-on reset

Notes: 1. Because power supplies (1.25 V and 3.3 V) other than the 2.5V power supply are 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 supply are stopped in this mode, modules other than the RTC stop operating and do not hold register information.

3. Satisfy both transition conditions when both backups by the RTC and DDR-SDRAM power supplies are necessary.

4. Do not input signals to I/O pins while the I/O power supply (VDDQ) is stopped.

17.2 Input/Output Pins

Table 17.2 shows the pin configuration of the Power-Down Modes.

Table 17.2 Pin Configuration

Pin name	Function	I/O	Description		
STATUS1	Processing state 1	Output	Indicate the processor's operating status		
STATUS0	Processing state 2		STATUS1	STATUS0	Operating Status
			High	High	Reset
			High	Low	Sleep mode
			Low	Low	Normal operation

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 the register states in each processing mode.

Table 17.3 Register configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync clock
Standby control register	MSTPCR	R/W	H'FFC8 0030	H'1FC8 0030	32	Pck

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	Sleep by SLEEP Instruction
Standby control register	MSTPCR	H'0000 0000	Retained	Retained	Retained

17.3.1 Standby Control Register (MSTPCR)

MSTPCR is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	MSTP21	—	—	—	—	—
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/W	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	—	—	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value:	0	0	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/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	MSTP21	0	R/W	Module Stop Bit 21 To make a transition to the DMAC module standby mode, confirm that the DME bits in DMA operation registers (DMAOR0 and DMAOR1) are cleared to 0 or all TE bits in DMA channel control registers (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 module
20 to 14	—	All 0	R	Reserved These bits are is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13 to 8	MSTP[13:8]	All 0	R/W	Module Stop Bit [13:8] 0: Supplies the clock to the corresponding module 1: Stops the clock supply to the corresponding module [13]: MMCIF, [12]: FLCTL, [11]: RTC, [10]: TMU channels 0 to 2, [9]: TMU channels 3 to 5, [8]: CMT
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value 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 module 1: Stops the clock supply to the corresponding module [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 below (1) 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 finished.

(2) Perform two dummy read operations for MSTPR before the sleep instruction is issued or the operating frequency is changed.

17.4 Sleep Mode

17.4.1 Transition to Sleep mode

A transition to the sleep mode is made by executing the SLEEP instruction in the program execution state. Although the CPU stops operating after execution of the SLEEP instruction, the contents of the CPU registers are held.

On-chip peripheral modules other than the CPU continue to operate and the clock continues to be output on the CLKOUT pin.

In sleep mode, a high-level signal is output at the STATUS1 pin, and a low-level signal at the 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 a reset.

Since an interrupt is accepted in sleep mode even if the BL bit in SR is set to 1, save the contents 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], or on-chip module interrupt, and the interrupt exception handling then starts. A corresponding code to the interrupt is stored in INTVENT.

Cancellation by Reset: The sleep mode is canceled with a power-on reset by the $\overline{\text{PRESET}}$ pin, a power-on reset by a watchdog timer overflow, or a manual reset.

17.5 Module Standby State

This LSI supports the module standby state, where the clock supplied to on-chip modules is stopped.

17.5.1 Transition to Module Standby Mode

Setting a corresponding bit in the standby control register (MSTPCR) to 1 will stop the clock supply.

Modules in module standby state keep the state immediately before the transition to the module standby state. The registers keep the contents before halted, and the external pins keep the functions before halted. At waking up from the module standby state, operation is restarted from 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 operation and are in an idle state, with no interrupt sources from the external pins or other modules.

17.5.2 Cancellation of Module Standby Mode and Resume

The module standby mode can be canceled by clearing a corresponding bit in the standby control register (MSTPCR) to 0.

17.6 DDR-SDRAM Power Supply Backup

17.6.1 Self-Refresh and Initialization

To preserve the contents of the DDR-SDRAM with battery backup, make sure that the DDR-SDRAM is in the self-refresh mode before turning off the system power supply. When the system power supply is turned on, whether initialization of the DDR-SDRAM and cancellation of the self-refresh mode is needed will depend on whether the DDR-SDRAM has been in self-refresh mode or has not been initialized. Both a transition to and a cancellation of the self-refresh mode are done for the DDR-SDRAM by a command.

RMODE Bit: Bit 33 in MIM. The initial value is 0. Setting this bit to 1 after setting the DRE bit in MIM to 1 causes the DDRIF to start the sequence for a transition to the self-refresh mode. For 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 (high) 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 the LSI power supply, the BKPRST signal must be driven to low in synchronization with turning the LSI power supply on or off. The $\overline{\text{BKPRST}}$ signal must be kept low while the system power supply is turned off.

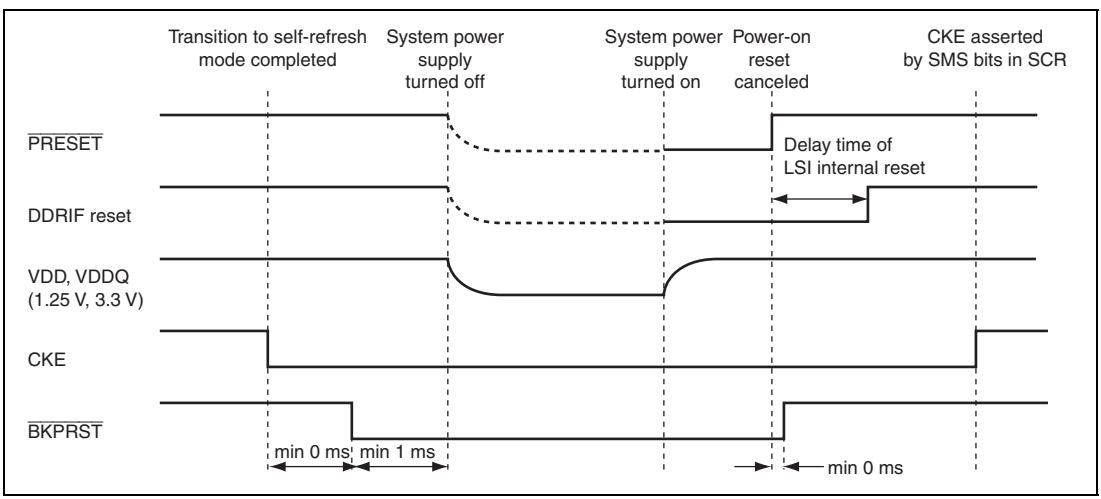


Figure 17.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off

17.6.2 DDR-SDRAM Backup Sequence when Turning Off System Power Supply

The sequence when the system power supply is turned off is shown below.

Figure 17.1 shows the sequence of a transition to the self-refresh mode to turn off the system power supply.

(A) Confirm that all transactions of the DDRIF by on-chip peripheral modules are completed.

(B) Issue the all bank precharge command (PREALL) with bits SMS2 to SMS0 in SCR by software. Activated banks will be closed. After that, issue the auto-refresh command (REFA) 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 the self-refresh mode. At this time, keep the DCE bit set to 1. The self-refresh command will be automatically issued and the CKE signal will be driven to low by the DDRIF. After that, the 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 supply is turned off, the CKE output may be unstable. Before turning off the system power supply, use the external $\overline{\text{BKPRST}}$ signal to keep the CKE signal input of the DDR-SDRAM low until 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 auto-refresh 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 until the clock is supplied after the LSI power supply has become stable. Use the external $\overline{\text{BKPRST}}$ signal to keep the CKE signal input of the DDR-SDRAM low until canceling the power-on reset as shown in figure 17.1.

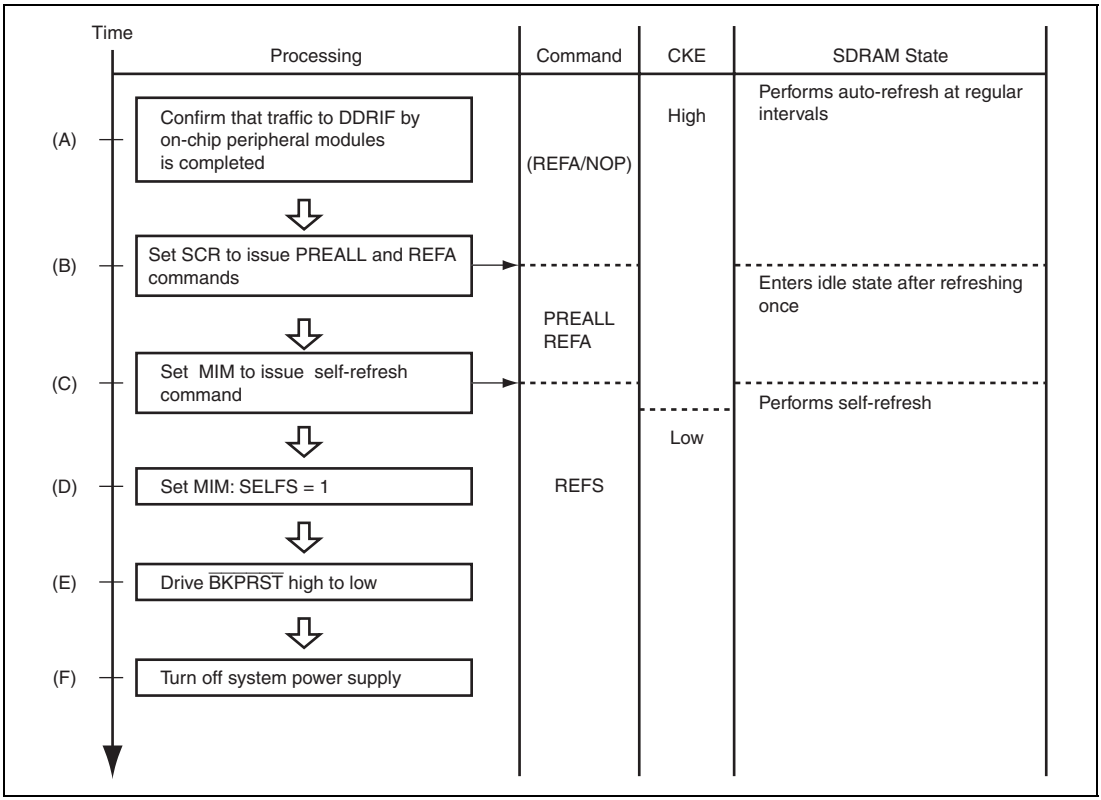


Figure 17.2 Sequence for Turning Off System Power Supply in Self-Refresh Mode

17.7 RTC Power Supply Backup

17.7.1 Transition to RTC Power Supply Backup

To turn on the RTC battery backup with the system power supply turned off, assert the $\overline{\text{XRTCSTBI}}$ signal before the voltage of the VDD (1.25V) power supply starts to drop. This function can be used to reduce the VDD current. When the RTC clock is supplied to the RTC crystal oscillator, each counter of the RTC is still being counted up while the VDD power supply is not being supplied.

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 interrupt occurs during the RTC power supply backup mode, it is invalid because of the power-on reset. The 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-on oscillation settling time to prevent the LSI from being damaged by the transient current caused 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 $\overline{\text{PRESET}}$ signal.

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 goes to the RTC power supply backup mode.

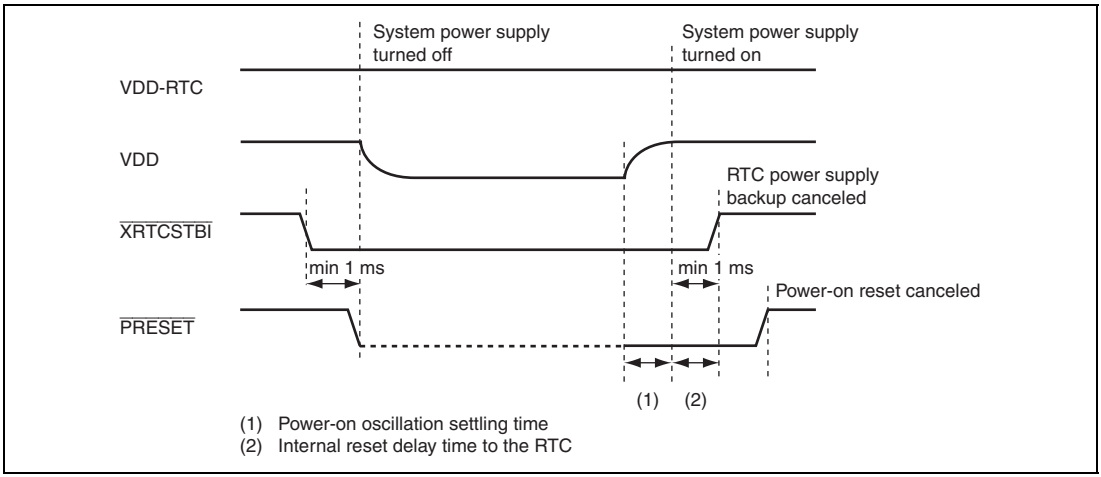


Figure 17.3 Sequence for Turning System Power Supply On/Off

17.8 Mode Transitions

Figure 17.4 shows the mode transitions.

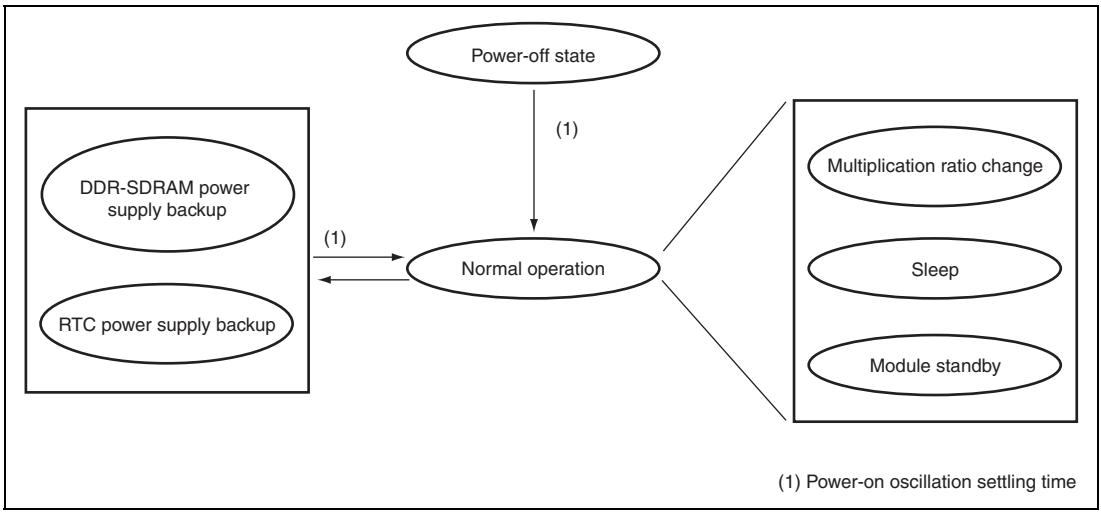


Figure 17.4 Mode Transition Diagram

17.9 STATUS Pin Change Timing

17.9.1 In Reset

Refer to section 16.5, Status Pin Change Timing during Reset.

17.9.2 In Sleep

Figure 17.5 shows the state of output pins in sleep mode.

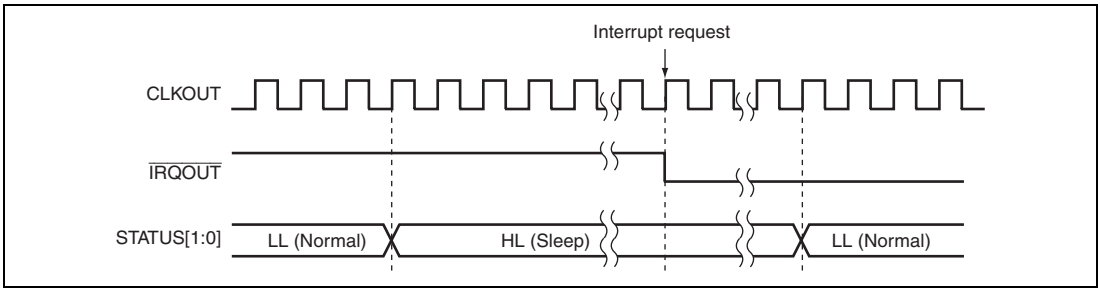


Figure 17.5 Status Pins Output from Sleep to Interrupt

Section 18 Timer Unit (TMU)

This LSI includes an on-chip 32-bit timer unit (TMU), which has six channels (channels 0 to 5).

18.1 Features

The TMU has the following features.

- Auto-reload type 32-bit down-counter provided for each channel
- Input capture function provided in channel 2
- Selection of rising edge or falling edge as external clock input edge when external clock is 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 32-bit 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, and 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, and Pck/1024) (Pck is the peripheral clock).
- Two interrupt sources

  One underflow source (each channel) and one input capture source (channel 2).

Figure 18.1 shows a block diagram of the TMU.

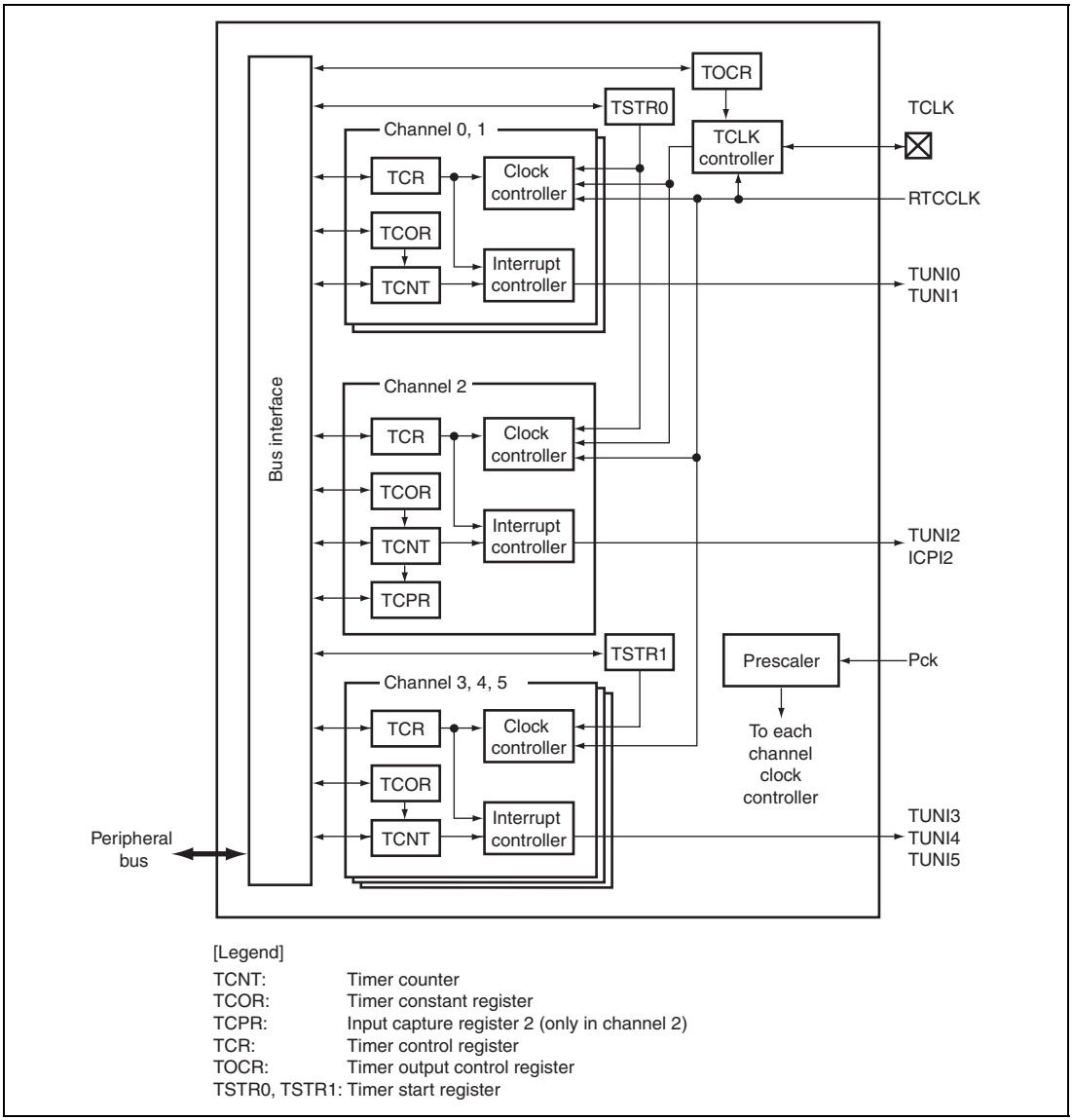


Figure 18.1 Block Diagram of TMU

18.2 Input/Output Pins

Table 18.1 shows the TMU pin configuration.

Table 18.1 Pin Configuration

Pin Name	Function	I/O	Description
TCLK*	Clock input/output	I/O	Channel 0, 1 and 2 external clock input pin/channel 2 input capture control input pin/RTC output pin (shared with RTC)

Note: This pin is multiplexed with the LBSC and GPIO pins.

18.3 Register Descriptions

Table 18.2 shows the TMU register configuration. Table 18.3 shows the register states in each processing mode.

Table 18.2 Register Configuration

Channel	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0, 1, 2 Common	Timer output control register	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8	Pck
	Timer start register 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8	Pck
0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32	Pck
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32	Pck
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16	Pck
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32	Pck
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32	Pck
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16	Pck
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32	Pck
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32	Pck
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16	Pck
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32	Pck
3, 4, 5 Common	Timer start register 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8	Pck
3	Timer constant register 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32	Pck
	Timer counter 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32	Pck
	Timer control register 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16	Pck
4	Timer constant register 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32	Pck
	Timer counter 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32	Pck
	Timer control register 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16	Pck
5	Timer constant register 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32	Pck
	Timer counter 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32	Pck
	Timer control register 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16	Pck

Table 18.3 Register States in Each Processing Mode

Channel	Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep by SLEEP Instruction	Module Standby
0, 1, 2 Common	Timer output control register	TOCR	H'00	H'00	Retained	Retained
	Timer start register 0	TSTR0	H'00	H'00	Retained	Retained
0	Timer constant register 0	TCOR0	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 0	TCNT0	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 0	TCR0	H'0000	H'0000	Retained	Retained
1	Timer constant register 1	TCOR1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 1	TCNT1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 1	TCR1	H'0000	H'0000	Retained	Retained
2	Timer constant register 2	TCOR2	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 2	TCNT2	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 2	TCR2	H'0000	H'0000	Retained	Retained
	Input capture register 2	TCPR2	Retained	Retained	Retained	Retained
3, 4, 5 Common	Timer start register 1	TSTR1	H'00	H'00	Retained	Retained
3	Timer constant register3	TCOR3	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 3	TCNT3	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 3	TCR3	H'0000	H'0000	Retained	Retained
4	Timer constant register 4	TCOR4	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 4	TCNT4	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 4	TCR4	H'0000	H'0000	Retained	Retained
5	Timer constant register 5	TCOR5	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 5	TCNT5	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 5	TCR5	H'0000	H'0000	Retained	Retained

18.3.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit readable/writable register that specifies whether external pin TCLK is used as the external clock or input capture control input pin, or as the on-chip RTC output clock output pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCOE
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 value should always be 0.
0	TCOE	0	R/W	Timer Clock Pin Control Specifies whether timer clock pin TCLK is used as the external clock or input capture control input pin, or as the on-chip RTC output clock output pin. 0: Timer clock pin (TCLK) is used as external clock input or input capture control input pin 1: Timer clock pin (TCLK) is used as on-chip RTC output clock output pin

18.3.2 Timer Start Register (TSTR0, TSTR1)

TSTR is an 8-bit readable/writable register that specifies whether TCNT in each channel is operated or stopped.

- TSTR0

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Specifies whether TCNT2 is operated or stopped. 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 stopped. 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 stopped. 0: TCNT0 count operation is stopped 1: TCNT0 performs count operation

- TSTR1

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR5	STR4	STR3
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR5	0	R/W	Counter Start 5 Specifies whether TCNT5 is operated or stopped. 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 stopped. 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 stopped. 0: TCNT3 count operation is stopped 1: TCNT3 performs count operation

18.3.3 Timer Constant Register (TCORn) (n = 0 to 5)

The TCOR registers are 32-bit readable/writable registers. When a TCNT counter underflows while counting down, the TCOR value is set in that TCNT, which continues counting down from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	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/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 the input 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 the corresponding channel. At the same time, the TCOR value is set in TCNT, and the count-down operation continues from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	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/W

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

18.3.5 Timer Control Registers (TCRn) (n = 0 to 5)

The TCR registers are 16-bit readable/writable registers. Each TCR selects the count clock, specifies the edge when an external clock is selected, and controls interrupt generation when the flag indicating TCNT underflow is set to 1. TCR2 is also used for input capture control and control of interrupt generation in the event of input capture.

- TCR0, TCR1, TCR3, TCR4 and TCR5

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UNF	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- TCR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ICPF	UNF	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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/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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ICPF*[1]	0	R/W	Input Capture Interrupt Flag Status flag, provided in channel 2 only, which indicates 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 TCNT 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	Bit Name	Initial Value	R/W	Description
7	ICPE1*[1]	0	R/W	Input Capture Control
6	ICPE0*[1]	0	R/W	These bits, provided in channel 2 only, specify whether the input capture function is used, and control enabling or disabling of interrupt generation when the function is used. The CKEG bits specify whether the rising edge or falling edge of the TCLK pin is used to set the TCNT2 value in TCPR2. The TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. When the ICPF bit is 1, TCPR2 is not set in the event of input capture. 00: Input capture function is not used. 01: Setting prohibited 10: Input capture function is used, but interrupt due to input capture (TICPI2) is not enabled. Data transfer request is sent to the DMAC in the event of input capture. 11: Input capture function is used, and interrupt due to input capture (TICPI2) is enabled.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling or disabling of interrupt generation when the UNF status flag is set to 1, indicating TCNT underflow. 0: Interrupt due to underflow (TUNI) is disabled 1: Interrupt due to underflow (TUNI) is enabled
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock input edge when an external clock is selected or the input capture function is used. 00: Count/input capture register set on rising edge 01: Count/input capture register set on falling edge 1X: Count/input capture register set on both rising and falling edges

Bit	Bit Name	Initial Value	R/W	Description
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT count clock.
0	TPSC0	0	R/W	000: Counts on Pck/4
				001: Counts on Pck/16
				010: Counts on Pck/64
				011: Counts on Pck/256
				100: Counts on Pck/1024
				101: Setting prohibited
				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 only in channel 2. The input capture function is controlled by means of the ICPE and CKEG bits in TCR2. When input capture occurs, the TCNT2 value is copied into TCPR2. The value is set in TCPR2 only when the ICPF bit in TCR2 is 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	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	1	0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

18.4 Operation

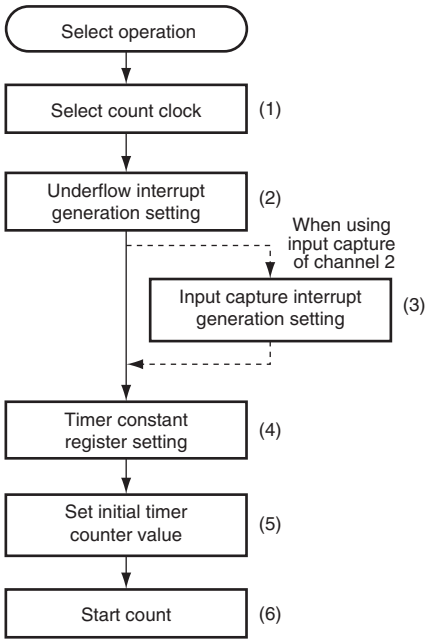
Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). Each TCNT performs count-down operation. The channels have an auto-reload function that allows cyclic count operations, and can also perform external event counting. Channel 2 also has an input capture function.

18.4.1 Counter Operation

When one of bits STR0 to STR2 in TSTR is set to 1, the TCNT for the corresponding channel starts counting. When TCNT underflows, the UNF flag in TCR is set. If the UNIE bit in TCR is set to 1 at this time, an interrupt request is sent to the CPU. At the same time, the value is copied 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.



(1) Select the count clock with the TPSC2 to TPSC0 bits in TCR. When the external clock (TCLK) is selected, specify the external clock edge with the CKEG1 and CKEG0 bits in TCR.

(2) Specify whether an interrupt is to be generated on TCNT underflow with the UNIE bit in TCR.

(3) When the input capture function is used, set the ICPE bits in TCR, including specification of whether the interrupt function is to be used.

(4) Set a value in TCOR.

(5) Set the initial value inTCNT.

(6) Set the STR bit to 1 in TSTR to start the count.

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

(2) Auto-Reload Count Operation

Figure 18.3 shows the TCNT auto-reload operation.

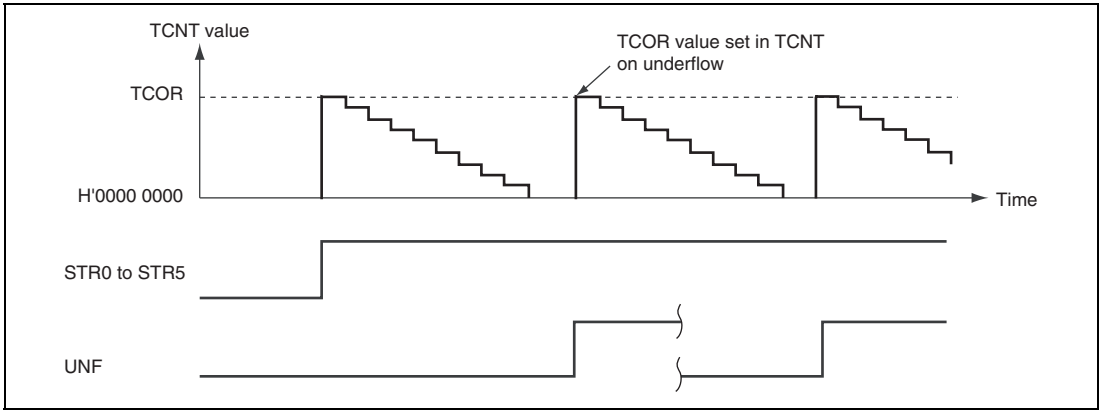


Figure 18.3 TCNT Auto-Reload Operation

(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 the peripheral clock can be selected as the count clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 18.4 shows the timing in this case.

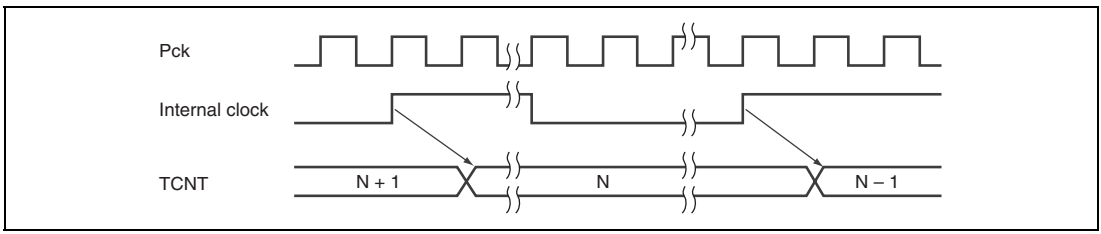


Figure 18.4 Count Timing when Operating on Internal Clock

- Operating on external clock

  In channels 0, 1, and 2, the external clock pin (TCLK) input can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR. The detected edge (rising, falling, or both edges) can be selected with the CKEG1 and CKEG0 bits in TCR.

  Figure 18.5 shows the timing for both-edge detection.

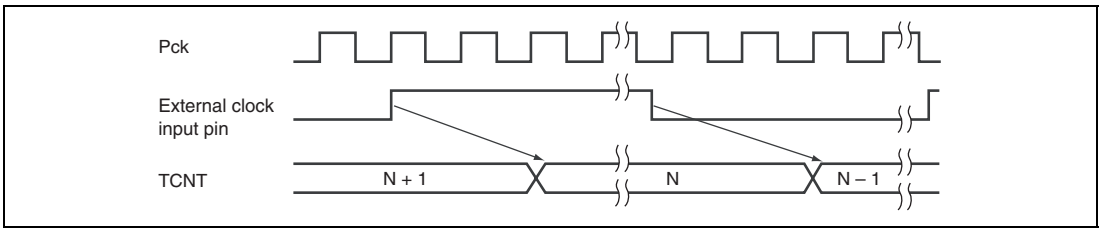


Figure 18.5 Count Timing when Operating on External Clock

- Operating on on-chip RTC output clock

  The on-chip RTC output clock can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 18.6 shows the timing for both-edge detection.

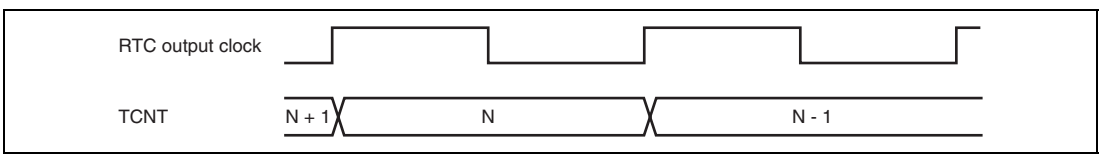


Figure 18.6 Count Timing when Operating on on-chip RTC output Clock

18.4.2 Input Capture Function

Channel 2 has an input capture function.

The procedure for using the input capture function is as follows:

1. Use bits TPSC2 to TPSC0 in TCR to set an internal clock as the timer operating clock.
2. Use bits IPCE1 and IPCE0 in TCR to specify use of the input capture function, and whether interrupts are to be generated when this function is used.
3. Use bits CKEG1 and CKEG0 in TCR to specify whether the rising or falling edge of the TCLK pin is to be used to set the TCNT value in TCPR2.

When input capture occurs, the TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. A new DMAC transfer request is not generated until processing of the previous request is finished.

Figure 18.7 shows the operation timing when the input capture function is used (with TCLK rising edge detection).

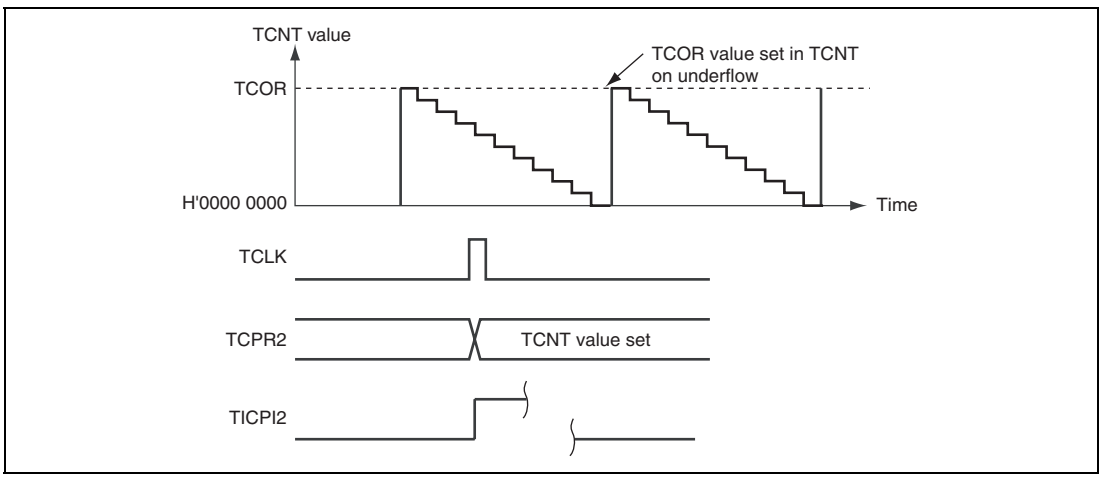


Figure 18.7 Operation Timing when Using Input Capture Function

18.5 Interrupts

There are seven TMU interrupt sources: underflow interrupts and the input capture interrupt when the input capture function is used. Underflow interrupts are generated on each of the channels, and input capture interrupts on channel 2 only.

An underflow interrupt request is generated (for each channel) when both the UNF bit and the interrupt enable bit (UNIE) for that channel are set to 1.

When the input capture function is used and an input capture request is generated, an interrupt is requested if the ICPF bit in TCR2 is 1 and the input capture control bits (ICPE1 and ICPE0) in 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 Usage Notes

18.6.1 Register Writes

When writing to a TMU register, timer count operation must be stopped by clearing the start bit (STR5 to STR0) for the relevant channel in TSTR.

Note that TSTR can be written to, and the UNF and ICPF bits in TCR can be cleared while the count is in progress. When the flags (UNF and ICPF) are cleared while the count is in progress, make sure not to change the values of bits other than those being cleared.

18.6.2 Reading from TCNT

Reading from TCNT is performed synchronously with the timer count operation. Note that when 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 read as the 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 frequency divider circuit.

18.6.4 External Clock Frequency

Ensure that the external clock (TCLK) input frequency for channels 0, 1 and 2 does not exceed Pck/4.

Section 19 Compare Match Timer (CMT)

This LSI includes the 32-bit compare match timer, which has four channels (channel 0 to 3).

There are two mode of operation: one is four channels 32-bit free running timer mode that has common 32-bit free running time base and the other is four channels 16-bit timer/counter mode that operates as four channels timer or counter individually.

19.1 Features

- 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 shows a block diagram of the CMT.

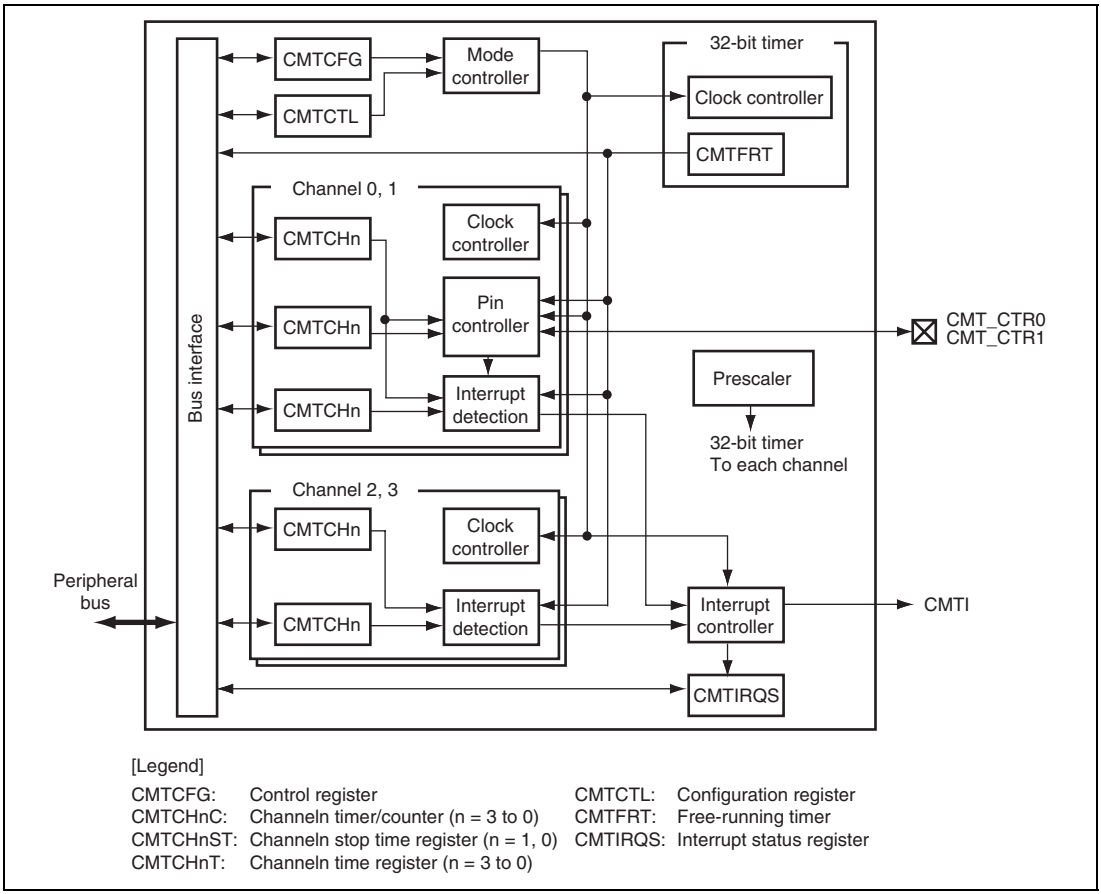


Figure 19.1 Block Diagram of CMT

19.2 Input/Output Pins

Table 19.1 shows the CMT pin configuration.

Table 19.1 Pin Configuration

Pin Name	Function	I/O	Description
CMT_CTR0*	Channel 0 timer/counter input/output	I/O	32-bit free-running timer or 16-bit timer/counter input capture input, output compare output or external trigger input.
CMT_CTR1*	Channel 1 timer/counter input/output	I/O	

Note: These pins are multiplexed with the STATUS0 and STATUS1 pins.

19.3 Register Descriptions

Table 19.2 shows the CMT register configuration. Table 19.3 shows the register states in each processing mode.

Table 19.2 Register Configuration

Ch.	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
Common	Configuration register	CMTCFG	R/W	H'FFE3 0000	H'1FE3 0000	32	Pck
	Free-running timer	CMTFRT	R	H'FFE3 0004	H'1FE3 0004	32	Pck
	Control register	CMTCTL	R/W	H'FFE3 0008	H'1FE3 0008	32	Pck
	Interrupt status register	CMTIRQS	R/W	H'FFE3 000C	H'1FE3 000C	32	Pck
0	Channel 0 time register	CMTCH0T	R/W	H'FFE3 0010	H'1FE3 0010	32	Pck
	Channel 0 stop time register	CMTCH0ST	R/W	H'FFE3 0020	H'1FE3 0020	32	Pck
	Channel 0 timer/counter	CMTCH0C	R/W	H'FFE3 0030	H'1FE3 0030	32	Pck
1	Channel 1 time register	CMTCH1T	R/W	H'FFE3 0014	H'1FE3 0014	32	Pck
	Channel 1 stop time register	CMTCH1ST	R/W	H'FFE3 0024	H'1FE3 0024	32	Pck
	Channel 1 timer/counter	CMTCH1C	R/W	H'FFE3 0034	H'1FE3 0034	32	Pck
2	Channel 2 time register	CMTCH2T	R/W	H'FFE3 0018	H'1FE3 0018	32	Pck
	Channel 2 timer/counter	CMTCH2C	R/W	H'FFE3 0038	H'1FE3 0038	32	Pck
3	Channel 3 time register	CMTCH3T	R/W	H'FFE3 001C	H'1FE3 001C	32	Pck
	Channel 3 timer/counter	CMTCH3C	R/W	H'FFE3 003C	H'1FE3 003C	32	Pck

Table 19.3 Register States of CMT in Each Processing Mode

Ch.	Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep by SLEEP Instruction	Module Standby
Common	Configuration register	CMTCFG	H'0000 0000	H'0000 0000	Retained	Retained
	Free-running timer	CMTFRT	H'0000 0000	H'0000 0000	Retained	Retained
	Control register	CMTCTL	H'0000 0000	H'0000 0000	Retained	Retained
	Interrupt status register	CMTIRQS	H'0000 0000	H'0000 0000	Retained	Retained
0	Channel 0 time register	CMTCH0T	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 0 stop time register	CMTCH0ST	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 0 timer/counter	CMTCH0C	H'0000 0000	H'0000 0000	Retained	Retained
1	Channel 1 time register	CMTCH1T	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 1 stop time register	CMTCH1ST	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 1 timer/counter	CMTCH1C	H'0000 0000	H'0000 0000	Retained	Retained
2	Channel 2 time register	CMTCH2T	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 2 timer/counter	CMTCH2C	H'0000 0000	H'0000 0000	Retained	Retained
3	Channel 3 time register	CMTCH3T	H'0000 0000	H'0000 0000	Retained	Retained
	Channel 3 timer/counter	CMTCH3C	H'0000 0000	H'0000 0000	Retained	Retained

19.3.1 Configuration Register (CMTCFG)

CMTCFG is a 32-bit readable/writable register. The possible operations for a pin are timer compare, timer input capture, up or down count, and capture input, where one pin is used for capture while the second is used to enable the count.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROT0
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	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ED1		ED0		—	—	FRTM	—	—	—	T01	
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/W	R/W	R/W	R/W	R	R	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	ROT0	0	R/W	Channel 0,1 Rotation Enable [Updown-counter mode (T01 = 11)] 0: Counting up by CMT_CTR0 signal, counting down by CMT_CTR1 signal 1: Rotary mode operation by CMT_CTR[1:0] signal (Then the settings of ED0 and ED1 are invalid) [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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11, 10	ED1	All 0	R/W	Channel 1 Pin Active Control [Input capture mode] 00: Setting prohibited* 01: Edge detection on rising edge of CMT_CTR1 pin input 10: Edge detection on falling edge of CMT_CTR1 pin input 11: Edge detection on either edge of CMT_CTR1 pin input [Output compare mode] 00: Setting prohibited* 01: High level is output from CMT_CTR1 pin during active period 10: Low level is output from CMT_CTR1 pin during 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_CTR0 pin input 10: Edge detection on falling edge of CMT_CTR0 pin input 11: Edge detection on either edge of CMT_CTR0 pin input [Output compare mode] 00: Setting prohibited * 01: High level is output from CMT_CTR0 pin during active period 10: Low level is output from CMT_CTR0 pin during active period 11: Setting prohibited*

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	FRTM	0	R/W	Free-Running Timer Mode Determines whether the timer works as a common 32-bit free-running timer or four independent 16-bit timers/counters. 0: 16-bit timer/counter mode (channel 0, 1) 16-bit timer mode (channel 2, 3) 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 value should always be 0.
1, 0	T01	All 0	R/W	Timer 0,1 Configuration Specifies the channel 0 and channel 1 operation mode. Clear to 00 in 32-bit free-running timer mode (FRTM = 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 prohibited.

19.3.2 Free-Running Timer (CMTFRT)

CMTFRT is a 32-bit read only register that is the common time base of the capture/compare register (channel 0, 1) and compare register (channel 2, 3) in 32-bit free-running timer (FRT) mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRT															
Initial value:	0	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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRT															
Initial value:	0	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	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-running timer (FRT).

19.3.3 Control Register (CMTCTL)

CMTCTL is a 32-bit readable/writable register that controls interrupts, makes settings for the clocks, and selects the operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TE3	TE2	TE1	TE0	IOE3	IOE2	IOE1	IOE0	ICE3	ICE2	ICE1	ICE0	—	—	IEE1	IEE0
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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CC3		CC2		CC1		CC0		—	—	SL1	SL0	OP3	OP2	OP1	OP0
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	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TE3	0	R/W	Channel 3 to 0 timer enable
30	TE2	0	R/W	Enables the counting of each of the 16-bit counters. If these bits are inactive when operating in timer mode or in counter mode, the counters are reset to 0. In updown-counter mode, channel 1 needs to be disabled (TE1 = 0). 0: Counting disabled; counter will be reset to H'0000 1: Counter is incremented n = 3 to 0
29	TE1	0	R/W	
28	TE0	0	R/W	
27	IOE3	0	R/W	Channel 3 to 0 Interrupt Overflow Enable
26	IOE2	0	R/W	These bits enable an interrupt to be generated when the relevant IOn bit is set in CMTIRQS register. 0: Interrupt generation disabled 1: Interrupt generation enabled n = 3 to 0
25	IOE1	0	R/W	
24	IOE0	0	R/W	
23	ICE3	0	R/W	Channel 3 to 0 Interrupt Compare Enable
22	ICE2	0	R/W	These bits enable an interrupt to be generated when the relevant ICn bit is set in the CMTIRQS register. 0: Interrupt generation disabled 1: Interrupt generation enabled n = 3 to 0
21	ICE1	0	R/W	
20	ICE0	0	R/W	
19	—	All 0	R	Reserved
18				These bits are always read as 0. The write value should 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 generated when the relevant IEn bit is set in CMTIRQS register. 0: Interrupt generation disabled 1: Interrupt generation enabled When a channel is in output compare mode, the corresponding IEEn has to be set to 0. n = 1, 0

Bit	Bit Name	Initial Value	R/W	Description
15, 14	CC3	All 0	R/W	Timer Clock Control Channel 3 These bits specify the clock input for the 16-bit timer in channel 3.* 00: Clock for timer 3 is 1/32 of peripheral clock (Pck) 01: Clock for timer 3 is 1/128 of peripheral clock (Pck) 10: Clock for timer 3 is 1/512 of peripheral clock (Pck) 11: Clock for timer 3 is 1/1024 of peripheral clock (Pck) The clock which divided from the peripheral clock (Pck) 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 timer in channel 2.* 00: Clock for timer 2 is 1/32 of peripheral clock (Pck) 01: Clock for timer 2 is 1/128 of peripheral clock (Pck) 10: Clock for timer 2 is 1/512 of peripheral clock (Pck) 11: Clock for timer 2 is 1/1024 of peripheral clock (Pck) The clock which divided from the peripheral clock (Pck) 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 (Pck) 01: Clock for timer 1 is 1/128 of peripheral clock (Pck) 10: Clock for timer 1 is 1/512 of peripheral clock (Pck) 11: Clock for timer 1 is 1/1024 of peripheral clock (Pck) Set the same value as the CC0 bit when using 16-bit input capture mode. Set the same value as the CC0 bit when using 16-bit input capture mode. The clock which divided from the peripheral clock (Pck) is the timer/counter resolution.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	CC0	All 0	R/W	Free-Running Timer Clock Control This clock is used for the 32-bit free-running timer (FRT) and also for the 16-bit timer/counter in channel 0.* 00: Clock for FRT and timer 0 is 1/32 of peripheral clock (Pck) 01: Clock for FRT and timer 0 is 1/128 of peripheral clock (Pck) 10: Clock for FRT and timer 0 is 1/512 of peripheral clock (Pck) 11: Clock for FRT and timer 0 is 1/1024 of peripheral clock (Pck) The clock which divided from the peripheral clock (Pck) is the timer/counter resolution.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 whether in output compare mode with 32-bit free-running timer mode, the output remains active for half the maximum time or until the stop value is reached. 0: Output remains active until the channel n stop time value is reached 1: Output remains active for half the total time of the FRT 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 determine whether the timer is used in output compare or input capture mode. Set 1 to the corresponding bit when using channel 2 or 3 as the timer. 0: Input capture mode (can be set in channel 0, 1) 1: Output compare mode When a channel is in output compare mode, the corresponding IEEn bits has to be set to 0. n = 3 to 0
1	OP1	0	R/W	
0	OP0	0	R/W	

Note: * The source clock is the peripheral clock (Pck). The clock which divided from the source clock is the timer/counter resolution of the channel.

19.3.4 Interrupt Status Register (CMTIRQS)

CMTIRQS, once set, can only be cleared by a write. Writing 0 to these bits clears the interrupt status bits. These conditions only create an interrupt if the relevant interrupt enable bit is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	IO3	IO2	IO1	IO0	IC3	IC2	IC1	IC0	—	—	IE1	IE0
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 value should 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-counters or updown-counters have wrapped i.e. overflowed from H'FFFF to H'0000 or underflowed from H'0000 to H'FFFF. 0: The counter is not overflowed or underflowed 1: The counter is overflowed or underflowed
9	IO1	0	R/W	
8	IO0	0	R/W	
7	IC3	0	R/W	Channel 3 to 0 Interrupt Compare
6	IC2	0	R/W	A bit for each channel indicates whether in timer mode, the free-running timer has become equal to the channel times. 0: Timer has not become equal to the channel time value 1: Timer has become equal to the channel time value
5	IC1	0	R/W	
4	IC0	0	R/W	
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 edge that will cause an action (active edge) has been defected. 0: Channel 1 to 0 has not received an active edge 1: Channel 1 to 0 has received an active edge

19.3.5 Channels 0 to 3 Time Registers (CMTCH0T to CMTCH3T)

In output compare mode, these registers specify the value to be compared with the free-running timer. In input capture mode, this register stores the free-running timer values or the 16-bit timer values on the active edge of the input. Every time an edge is detected, these registers are updated and the new captured value will be saved.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n time															
Initial value:	0	0	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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n time															
Initial value:	0	0	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/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-running timer. When clearing the STCn bit in CMTCTL, the output state that specifies by CMTCFG is 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	17	16
	Channel n stop time															
Initial value:	0	0	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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n stop time															
Initial value:	0	0	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/W	R/W

Note: n = 1 to 0

19.3.7 Channels 0 to 3 Counters (CMTCH0C to CMTCH3C)

Each channel register indicates the current value of the timer/counter. Writing to this register, it can be set the timer counter. When reading this register, the timer/counter value is not affected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n counter															
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/W

19.4 Operation

The CMT has two operation modes: one is four channels free-running timer that operates with the common time base of 32-bit free-running timer operating between approximately 1.5MHz (Pck/32 selected at Pck = 50MHz) to 30kHz (Pck/1024 selected at Pck = 33MHz). The other is 16-bit timer/counter that operating as two channels 16-bit timer/counter and two channels 16-bit timer. When operating as the timer, it can be selectable input capture or output compare. They differ from the free-running timer mode that they are initialized to H'0000 when capture input or compare match occurs on that channel.

19.4.1 Edge Detection

The timers and counters are based on edge detection on the input pins. An active edge can be selectable by setting CMTCFG to be a rising edge, falling edge, or both edges. In addition, the edge detection logic can operate in rotary switch operation where the combination of two inputs indicates whether the switch has been turned right or left and the updown counter is incremented or decremented. The edge detection input can either work independently for the timers or the up-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 least two cycles of the clock divided from the peripheral clock (Pck) for that channel, as shown in figure 19.2.

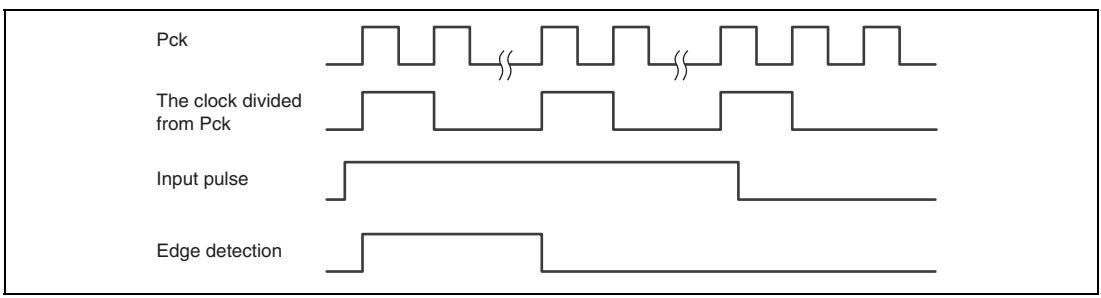


Figure 19.2 Edge Detection (example of rising edge)

19.4.2 32-Bit Timer: Input Capture

When rising edge or falling edge is detected while the timer CMTFRT is operating, the value of CMTFRT is captured in the corresponding CMTCHnT (n = 1, 0). Then the IEn flag in CMTIRQS is set to 1 and the interrupt is generated when the IEEn bit in CMTCTL is set to 1.

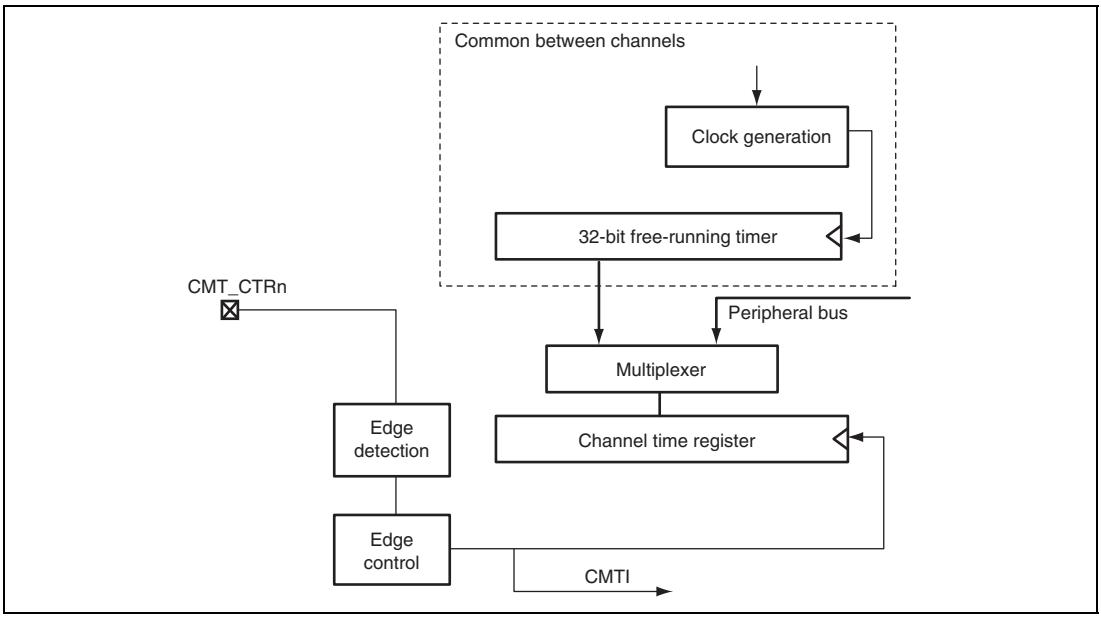


Figure 19.3 32-Bit Timer Mode: Input Capture (channel 1 and channel 0)

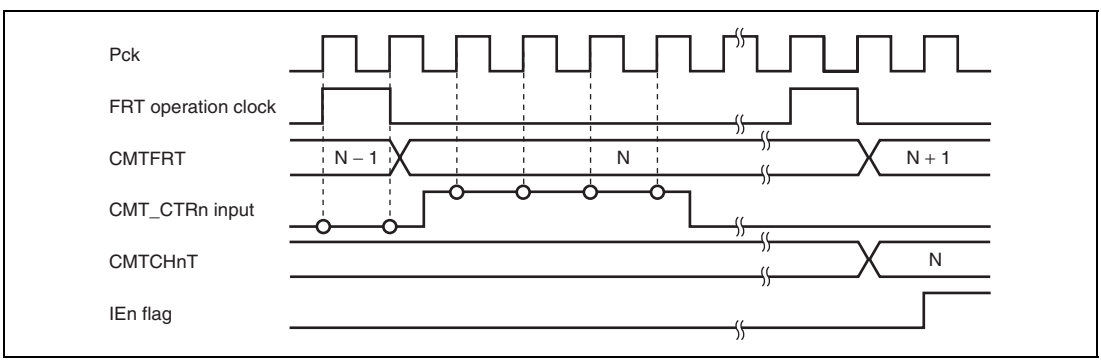


Figure 19.4 32-bit Timer mode: Input Capture Operation Timing

Table 19.4 32-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	1 (32-bit free-running timer)
	4 to 0	All 0
CMTCTL	31 to 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each channel)
	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 channels)

19.4.3 32-Bit Timer: Output Compare

When the value of the CMTFRT matches value of CMTCHnT plus 1 while the timer CMTFRT is operating, the CMT_CTR output state becomes equal to the setting of the ED1 and ED0 bits in CMTCFG. Then the ICn flag in CMTIRQS is set to 1 and the interrupt is generated when the ICEn bit in CMTCTLis set to 1.

The CMT_CTR output is being asserted until the value of CMTFRT matches CMTCHnT plus 1 or 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 generate an interrupt with regular period.

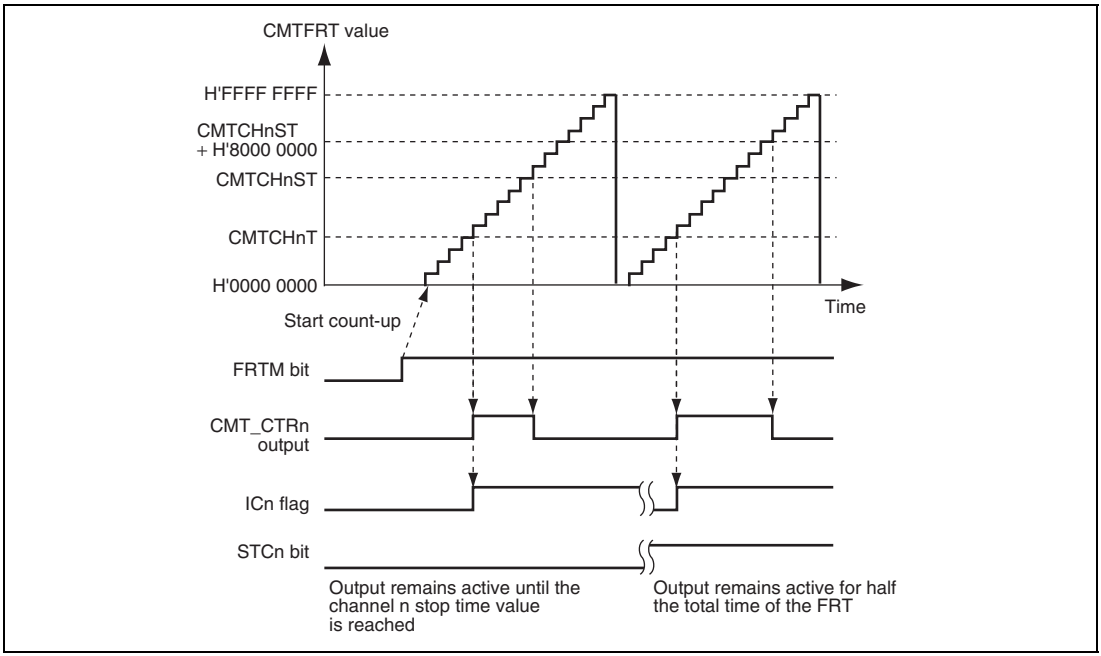


Figure 19.5 CMT_CTRn Assert Timing (channel 0 and 1)

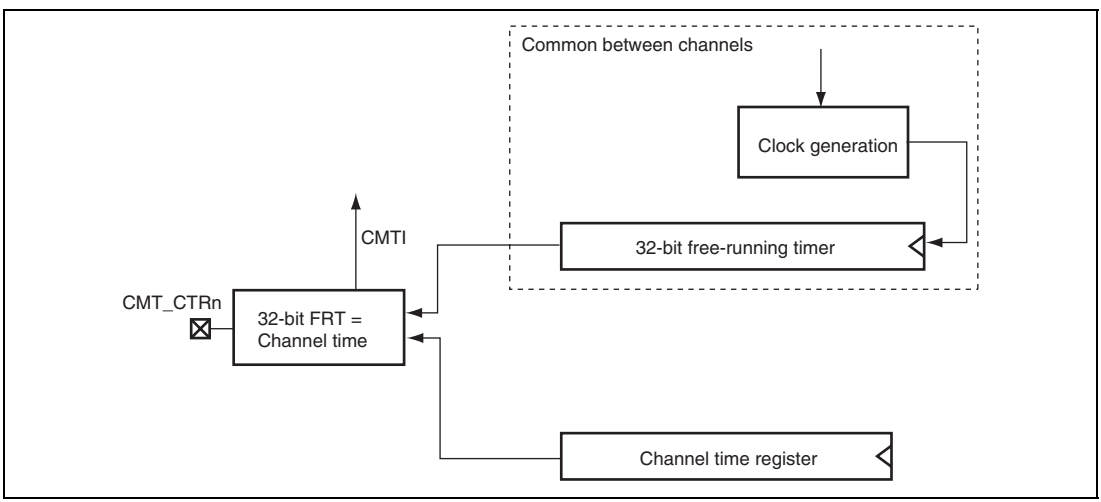


Figure 19.6 32-Bit Timer Mode: Output Compare (channel 1 and channel 0)

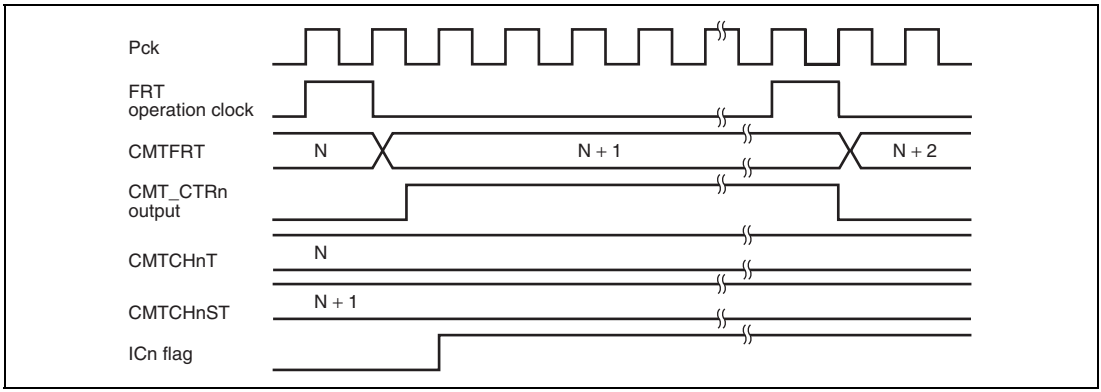


Figure 19.7 32-bit Timer Mode: Output Compare Operation Timing (Example of High output in Active and Not Active by CMTCHnST)

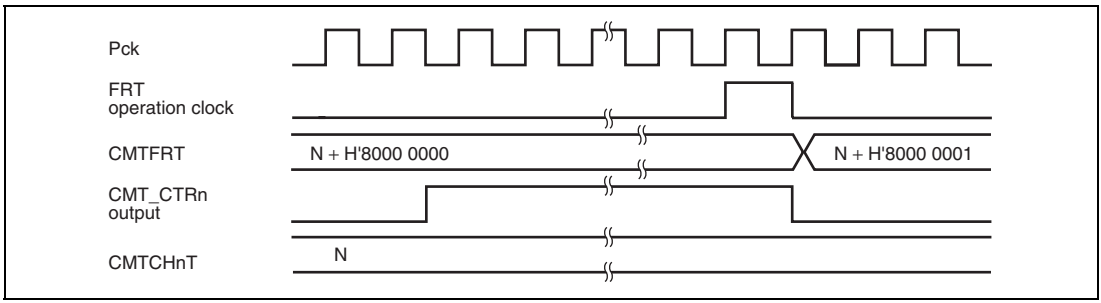


Figure 19.8 32-bit Timer Mode: Output Compare Operation Timing (Example of High output in Active and Not Active by CMTFRT)

Table 19.5 32-bit Timer Mode: Example of Output Compare 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	1 (32-bit free-running timer)
	4 to 0	All 0
CMTCTL	31 to 24	All 0
	23 to 20	Arbitrary value (compare interrupt setting of each channel)
	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)

19.4.4 16-Bit Timer: Input Capture

When rising edge or falling edge is detected while the timer CMTCH0C is operating, the value of CMTCHnC (n = 0, 1) is captured in the corresponding CMTCHnT (n = 1, 0). Then the IEn flag in CMTIRQS is set to 1 and the interrupt is generated when the IEEn bit in CMTCTL is set to 1.

Each channel timer overflowed after the timer is counting up at H'FFFF that is the value of CMTCHnC (n = 1, 0). Then the IOn flag in CMTIRQS is set to 1 and the interrupt is generated when the IOEn bit in CMTCTL is set to 1.

The 16-bit timer CMTCHnC (n = 1, 0) is initialized to H'0000 when the TEn (n = 1, 0) bit in CMTCL is cleared to 0 or an input capture occurs.

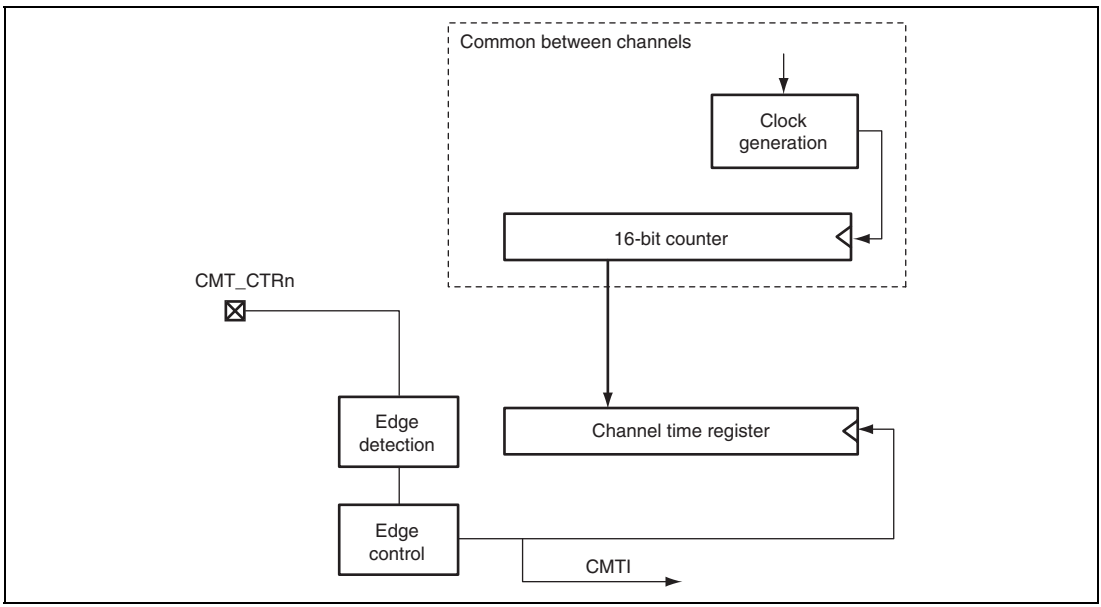


Figure 19.9 16-Bit Timer Mode: Input Capture (channel 1 and channel 0)

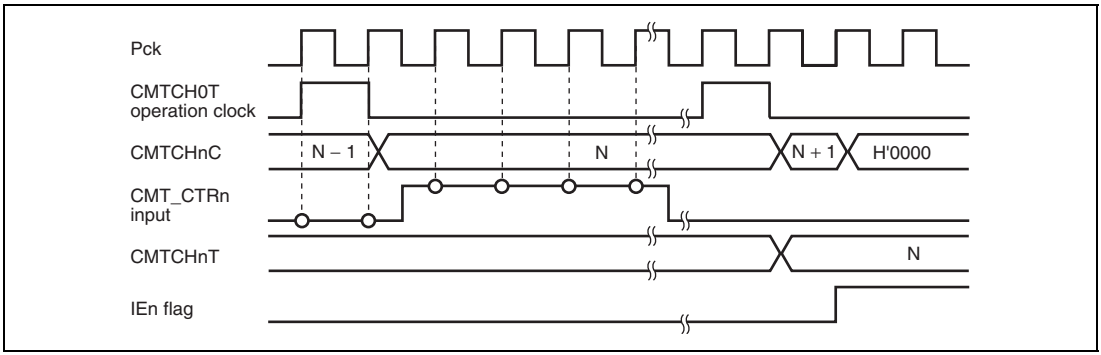


Figure19.10 16-Bit Timer Mode: Input Capture Operation Timing

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 channel)
	23 to 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each channel)
	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)

19.4.5 16-Bit Timer: Output Compare

When the value of CMTCHnC (n = 1, 0) matches the lower 16-bit of CMTCHnT while the timer CMTCH0C is operating, the output is inverted. Then the ICn flag in CMTIRQS is set to 1 and the interrupt is generated when the ICEn bit in CMTCTL is set to 1. However, when the lower 16-bit of time register CMTCHnT is H'0000, the compare match does not occur.

Each channel timer overflowed after the timer is counting up at H'FFFF that is the value of CMTCHnC (n = 3 to 0). Then the IOn flag in CMTIRQS is set to 1 and the interrupt is generated when the IOEn bit in CMTCTL is set to 1.

The 16-bit timer CMTCHnC (n = 3 to 0) is initialized to H'0000 when the TEn (n = 1, 0) bit in 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 generate an interrupt with regular period.

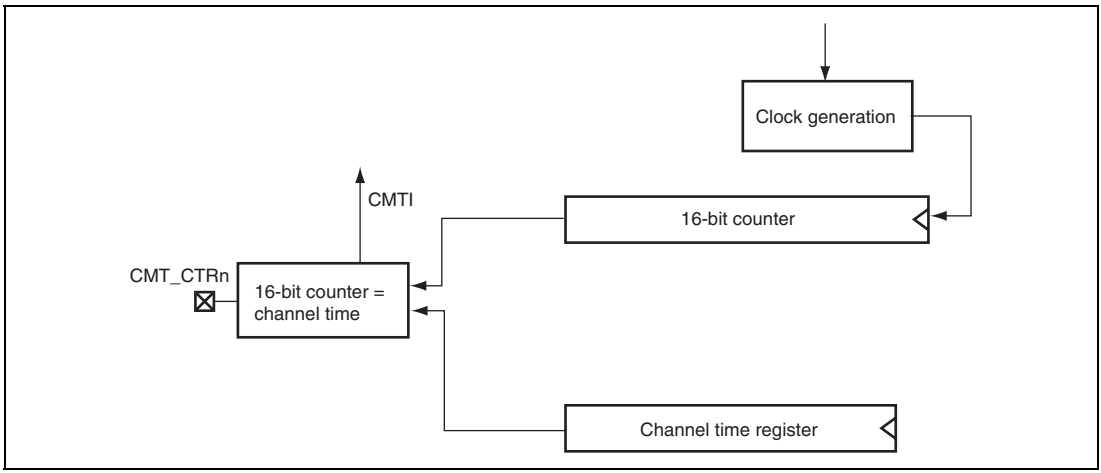


Figure 19.11 16-Bit Timer Mode: Output Compare
(CMT_CTR pins are available for channel 1 and channel 0)

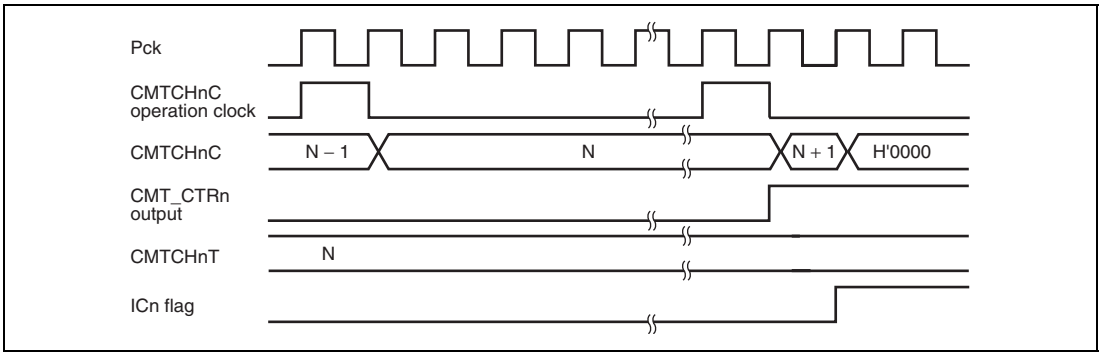


Figure19.12 16-Bit Timer Mode: Output Compare Operation Timing

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 channel)
	23 to 20	Arbitrary value (compare interrupt setting of each channel)
	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)

19.4.6 Counter: Up-counter

When rising edge or falling edge of the CMT_CTR input signal is detected at the rising edge of each channel operation clock, the channel counter CMTCHnC value is captured in the corresponding CMTCHnT (n = 1, 0) and that counter is counted up. Then the IEn flag in CMTIRQS is set to 1 and the interrupt is generated when the IEEn bit in CMTCTL is set to 1.

And each channel counter overflowed, the IOn flag in CMTIRQS is set to 1 and the interrupt is generated when the IOEn bit in CMTCTL is set to 1.

The counter CMTCHnC (n = 1, 0) is initialized to H'0000 when the TEn (n = 1, 0) bit in CMTCL is cleared to 0 or an input capture occurs.

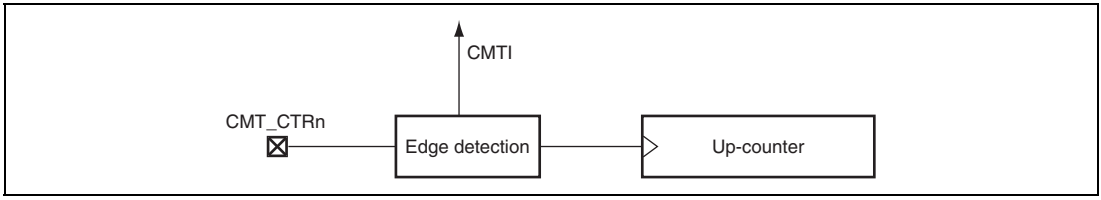


Figure 19.13 Up-Counter Mode (channel 1 and channel 0)

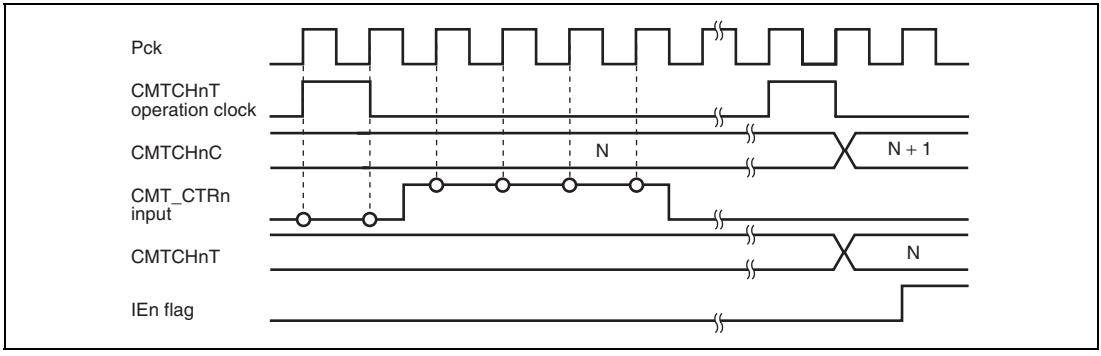


Figure 19.14 Up-counter Mode Operation Timing

Table 19.8 Setting Example of Up-counter Mode

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 2	All 0
	1, 0	10 (Up-counter 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 channel)
	23 t o 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each channel)
	15 to 12	All 0
	11 to 8	Arbitrary value (clock setting of each channel)
	7 to 0	All 0

19.4.7 Counter: Updown-counter

Channel 0 can be used as an updown-counter. However, the CMT_CRT1 pin is to connect to the channel 0, the channel 1 timer/counter needs to be disabled.

When rising edge or falling edge of the CMT_CTR input signal is detected at the rising edge of the channel 0 operation clock, the channel counter CMTCH0C is counted up or down. Then the IEn flag in CMTIRQS is set to 1 and the interrupt is generated when the IEEn bit in CMTCTL is set to 1. If both the count up pin (CMT_CTR0) and count down pin (CMT_CTR1) detect rising edge or falling edge, the counter value is not updated but the IEn bit in CMTIRQS is set to 1.

And the counter CMTCH0C overflowed or underflowed, the IO0 flag in CMTIRQS is set to 1 and 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 to 0.

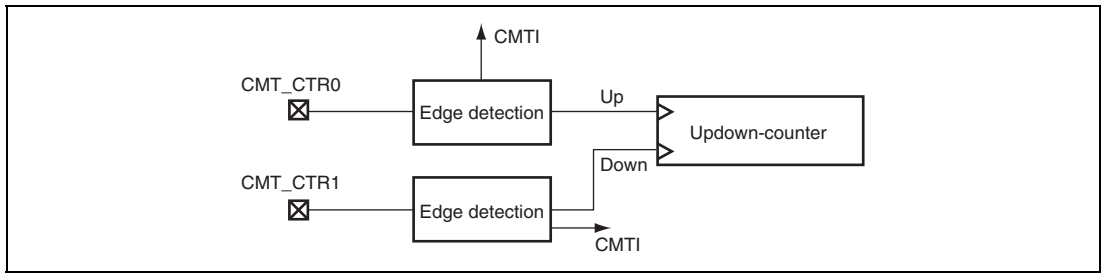


Figure 19.15 Updown-Counter Mode (only channel 0)

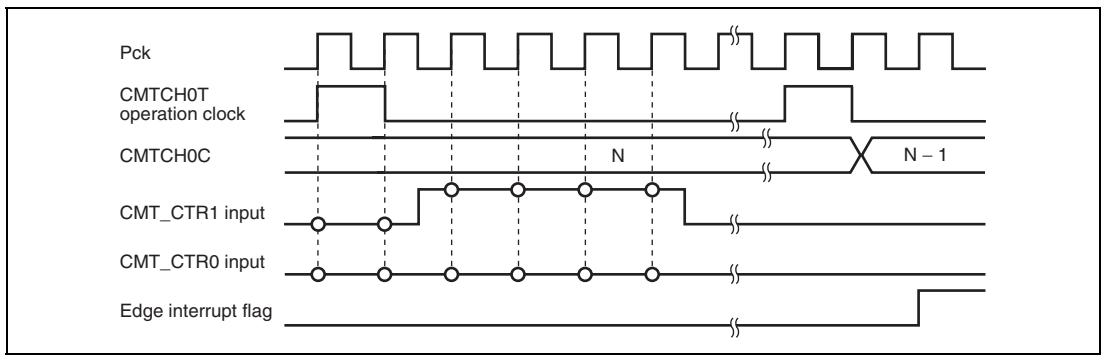


Figure 19.16 Updown-Counter Mode: Countdown Operation Timing (only channel 0)

Table 19.9 Setting Example of Updown-counter Mode

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 2	All 0
	1, 0	11 (Updown-counter mode setting of channel 0)
CMTCTL	31 to 29	All 0
	28	1 (counter enable of channel 0)
	27 to 25	All 0
	24	Arbitrary value (overflow interrupt setting of channel 0)
	23 t o 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each channel)
	15 to 10	All 0
	9, 8	Arbitrary value (clock setting of channel 0)
	7 to 0	All 0

19.4.8 Counter: Rotary Switch Operation of Updown-counter

The Updown-counter can operate as a rotary switch. When the falling edge of the control pin is detected and then the data pin input level is low, the counter is counted up, or the data pin input level is high, the counter is counted down. Then the timer is counted up, the IE0 flag in CMTIRQS is set to 1, or the timer is counted down, the IE1 flag in CMTIRQS is set to 1 and the interrupt is generated when the IEEn bit in CMTCTL is set to 1. If both the count up and count down are detected, both the IE0 and the IE1 flags are set to 1 and the counter CMTCH0C is updated by the most recently detected edge.

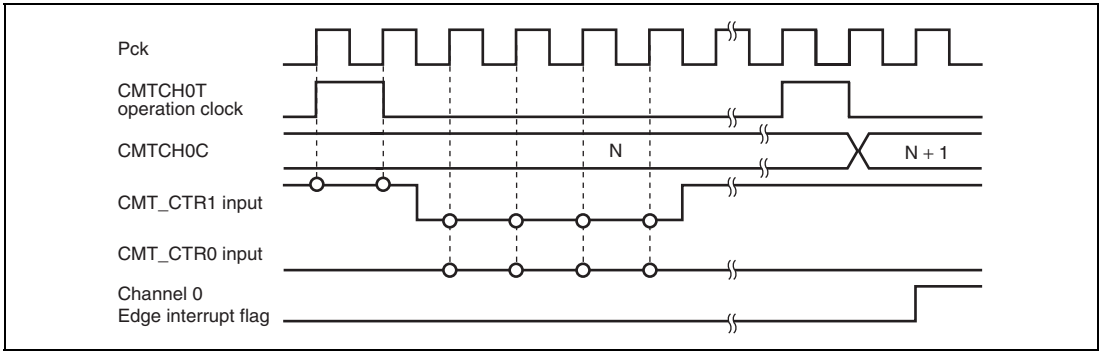


Figure 19.17 Rotary Switch Operation Count-Up Timing

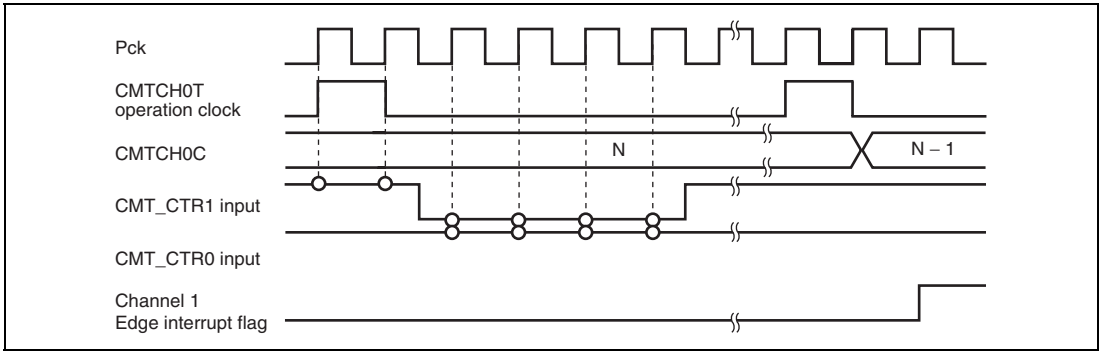


Figure 19.18 Rotary Switch Operation Count-Down Timing

Table 19.10 Setting Example of Updown-counter Mode

Register	Bit	Settings
CMTCFG	31 t o 17	All 0
	16	All 1 (rotary switch operation setting of channel 0)
	15 to 6	All 0
	5	0 (16-bit timer/counter)
	4 to 2	All 0
	1, 0	11 (Updown-counter mode setting of channel 0)
CMTCTL	31 to 29	All 0
	28	1 (counter enable of channel 0)
	27 to 25	All 0
	24	Arbitrary value (overflow interrupt setting of channel 0)
	23 t o 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each channel)
	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 one 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 available
16-bit timer	Input capture	Available	Not Available	Available
	Output compare	Available	Available	Not available
Counter	Up-counter	Available	Not Available	Available
	Updown-counter	Available	Not Available	Available

Section 20 Realtime Clock (RTC)

The SH7780 includes an on-chip realtime clock (RTC) and a 32.768 kHz crystal oscillator for use by the RTC.

20.1 Features

The RTC has the following features.

- Clock and calendar functions (BCD display)
  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 frequency divider
- Start/stop function
- 30-second adjustment function
- Alarm interrupts
  Comparison with second, minute, hour, day-of-week, day, month, or year can be selected as the alarm interrupt condition
- Periodic interrupts
  An interrupt period of 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds can be selected
- Carry interrupt
  Carry interrupt function indicating a second counter carry, or a 64 Hz counter carry when the 64 Hz counter is read
- Automatic leap year adjustment

20.1.1 Block Diagram

Figure 20.1 shows a block diagram of the RTC.

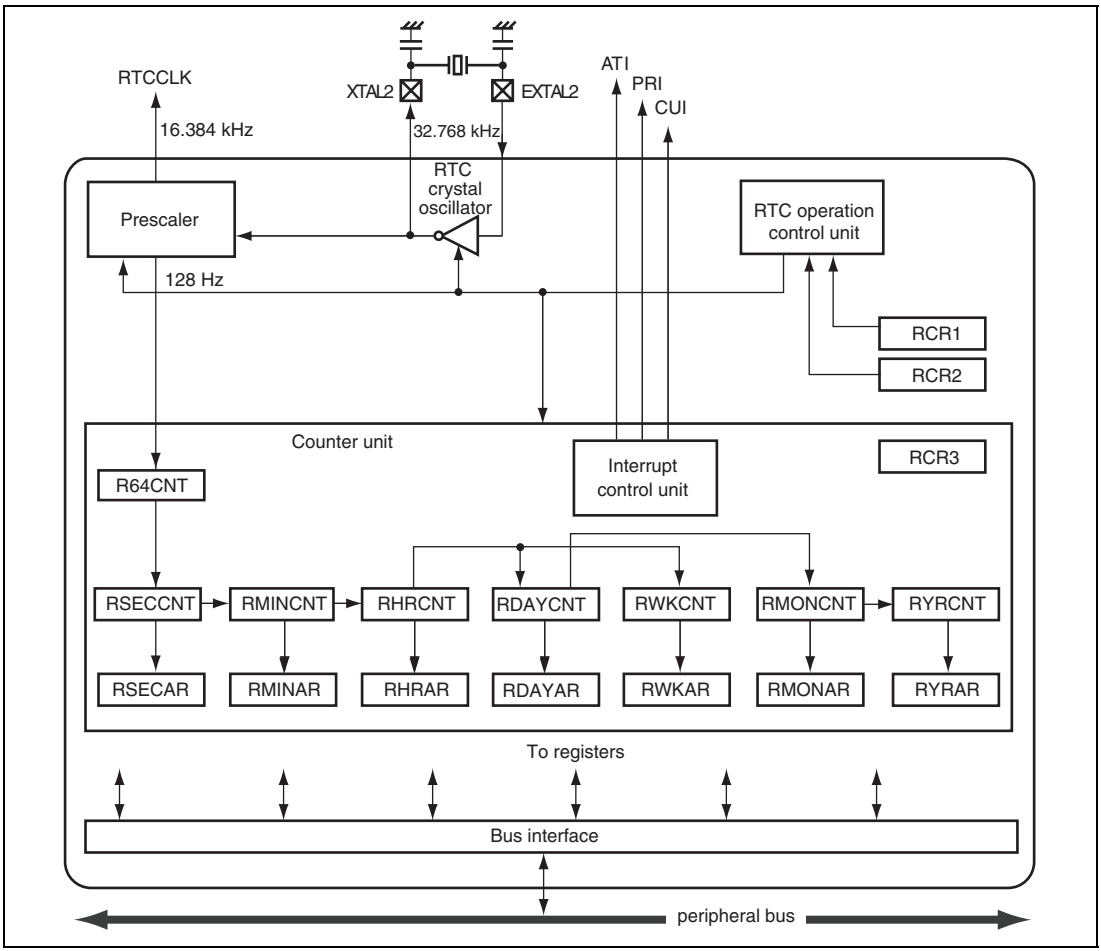


Figure 20.1 Block Diagram of RTC

20.2 Input/Output Pins

Table 20.1 shows the RTC pins.

Table 20.1 RTC Pins

Pin Name	Function	I/O	Description
EXTAL2	RTC oscillator crystal pin	Input	Connects crystal to RTC oscillator
XTAL2	RTC oscillator crystal pin	Output	Connects crystal to RTC oscillator
TCLK*[1]	TMU clock input/RTC clock output	I/O	TMU external clock input pin/input capture control input pin/RTC output pin (shared with TMU)
VDD-RTC	Dedicated RTC power supply	—	RTC oscillator power supply pin*[2]
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 not used.

20.3 Register Descriptions

Table 20.2 shows the RTC registers. Table 20.3 shows the register states in each processing mode.

Table 20.2 RTC Registers

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size	Sync Clock
64 Hz counter	R64CNT	R	H'FFE80000	H'1FE80000	8	Pck
Second counter	RSECCNT	R/W	H'FFE80004	H'1FE80004	8	Pck
Minute counter	RMINCNT	R/W	H'FFE80008	H'1FE80008	8	Pck
Hour counter	RHRCNT	R/W	H'FFE8000C	H'1FE8000C	8	Pck
Day-of-week counter	RWKCNT	R/W	H'FFE80010	H'1FE80010	8	Pck
Day counter	RDAYCNT	R/W	H'FFE80014	H'1FE80014	8	Pck
Month counter	RMONCNT	R/W	H'FFE80018	H'1FE80018	8	Pck
Year counter	RYRCNT	R/W	H'FFE8001C	H'1FE8001C	16	Pck
Second alarm register	RSECAR	R/W	H'FFE80020	H'1FE80020	8	Pck
Minute alarm register	RMINAR	R/W	H'FFE80024	H'1FE80024	8	Pck
Hour alarm register	RHRAR	R/W	H'FFE80028	H'1FE80028	8	Pck
Day-of-week alarm register	RWKAR	R/W	H'FFE8002C	H'1FE8002C	8	Pck
Day alarm register	RDAYAR	R/W	H'FFE80030	H'1FE80030	8	Pck
Month alarm register	RMONAR	R/W	H'FFE80034	H'1FE80034	8	Pck
RTC control register 1	RCR1	R/W	H'FFE80038	H'1FE80038	8	Pck
RTC control register 2	RCR2	R/W	H'FFE8003C	H'1FE8003C	8	Pck
RTC control register 3	RCR3	R/W	H'FFE80050	H'1FE80050	8	Pck
Year alarm register	RYRAR	R/W	H'FFE80054	H'1FE80054	16	Pck

Table 20.3 Register States of RTC in Each Processing Mode

Register Name	Abbreviation	Initial Value	Power-on Reset	Manual Reset	Sleep
64 Hz counter	R64CNT	Undefined	Counts	Counts	Counts
Second counter	RSECCNT	Undefined	Counts	Counts	Counts
Minute counter	RMINCNT	Undefined	Counts	Counts	Counts
Hour counter	RHRCNT	Undefined	Counts	Counts	Counts
Day-of-week counter	RWKCNT	Undefined	Counts	Counts	Counts
Day counter	RDAYCNT	Undefined	Counts	Counts	Counts
Month counter	RMONCNT	Undefined	Counts	Counts	Counts
Year counter	RYRCNT	Undefined	Counts	Counts	Counts
Second alarm register	RSECAR	Undefined*[1]	Initialized*[1]	Retained	Retained
Minute alarm register	RMINAR	Undefined*[1]	Initialized*[1]	Retained	Retained
Hour alarm register	RHRAR	Undefined*[1]	Initialized*[1]	Retained	Retained
Day-of-week alarm register	RWKAR	Undefined*[1]	Initialized*[1]	Retained	Retained
Day alarm register	RDAYAR	Undefined*[1]	Initialized*[1]	Retained	Retained
Month alarm register	RMONAR	Undefined*[1]	Initialized*[1]	Retained	Retained
RTC control register 1	RCR1	H'00*[3]	Initialized	Initialized	Retained
RTC control register 2	RCR2	H'09*[4]	Initialized	Initialized*[2]	Retained
RTC control register 3	RCR3	H'00	Initialized	Retained	Retained
Year alarm register	RYRAR	Undefined	Retained	Retained	Retained

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.

20.3.1 64 Hz Counter (R64CNT)

R64CNT is an 8-bit read-only register that indicates a state of 64 Hz to 1 Hz within the RTC frequency divider.

If this register is read when a carry is generated from the 128 Hz frequency division stage, bit 7 (CF) in RTC control register 1 (RCR1) is set to 1, indicating the simultaneous occurrence of the carry and the 64 Hz counter read. In this case, the read value is not valid, and so R64CNT must be read again after first writing 0 to the CF bit in RCR1 to clear it.

When the RESET bit or ADJ bit in RTC control register 2 (RCR2) is set to 1, the RTC frequency 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 counting the BCD-coded second value in the RTC. It counts on the carry (transition of the R64CNT.1Hz bit 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 value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or 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 be 0.

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

20.3.3 Minute Counter (RMINCNT)

RMINCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded minute value in the RTC. It counts on the carry generated once per minute by the second counter.

The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RMINCNT 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 be 0.

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 counting the BCD-coded hour value in the RTC. It counts on the carry generated once per hour by the minute counter.

The setting range is decimal 00 to 23. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or 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 should always 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

20.3.5 Day-of-Week Counter (RWKCNT)

RWKCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded day-of-week value in the RTC. It counts on the carry generated once per day by the hour counter.

The setting range is decimal 0 to 6. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RWKCNT is not initialized by a power-on or manual reset.

Bits 7 to 3 are always read as 0. A write to these bits is invalid, but the write value should always 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	6
Day of week	Sun	Mon	Tue	Wed	Thu	Fri	Sat

20.3.6 Day Counter (RDAYCNT)

RDAYCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded day value in the RTC. It counts on the carry generated once per day by the hour counter.

The setting range is decimal 01 to 31. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RDAYCNT is not initialized by a power-on or manual reset.

The setting range for RDAYCNT depends on the month and whether the year is a leap year, so care is required when making the setting. Taking the year counter (RYRCNT) value as the year, leap year calculation is performed according to whether or not the value is divisible by 400, 100, and 4.

Bits 7 and 6 are always read as 0. A write to these bits is invalid, but the write value should always 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

20.3.7 Month Counter (RMONCNT)

RMONCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded month value in the RTC. It counts on the carry generated once per month by the day counter.

The setting range is decimal 01 to 12. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RMONCNT is not initialized by a power-on or manual reset.

Bits 7 to 5 are always read as 0. A write to these bits is invalid, but the write value should always 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 the BCD-coded year value in the RTC. It counts on the carry generated once per year by the month counter.

The setting range is decimal 0000 to 9999. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in 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	1	0
	1000-year units				100-year units				10-year units				1-year units			
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

20.3.9 Second Alarm Register (RSECAR)

RSECAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded second value counter, RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 59 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The other fields in RSECAR are 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 BCD-coded minute value counter, RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 59 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The other fields in RMINAR are not 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

20.3.11 Hour Alarm Register (RHRAR)

RHRAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded hour value counter, RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 23 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The other fields in RHRAR are not 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 be 0.

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 BCD-coded day-of-week value counter, RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 0 to 6 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The other fields in RWKAR are not initialized by a power-on or manual reset.

Bits 6 to 3 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Day-of-week code	0	1	2	3	4	5	6
Day of week	Sun	Mon	Tue	Wed	Thu	Fri	Sat

20.3.13 Day Alarm Register (RDAYAR)

RDAYAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded day value counter, RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 01 to 31 + ENB bit. The RTC will not operate normally if any other value is set. The setting range for RDAYAR depends on the month and whether the year is a leap 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 RDAYAR are not 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 be 0.

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

20.3.14 Month Alarm Register (RMONAR)

RMONAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded month value counter, RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 01 to 12 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The other fields in RMONAR are 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 should always 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. When the 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 registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR and RMONAR) in which the ENB and YENB bits are set to 1. The alarm flag of RCR1 is only set to 1 when the respective values all match.

The setting range of RYRAR is decimal 0000 to 9999, and normal operation is not obtained if a value beyond this range is set here.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1000 years				100 years				10 years				1 year			
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

20.3.16 RTC Control Register 1 (RCR1)

RCR1 is an 8-bit readable/writable register containing a carry flag and alarm flag, plus flags to enable or disable interrupts for these flags.

The CIE and AIE bits are initialized to 0 by a power-on or manual reset; the value of bits other than CIE and AIE is undefined.

Bit:	7	6	5	4	3	2	1	0
	CF	—	—	CIE	AIE	—	CRF	AF
Initial value:	—	—	—	0	0	—	—	—
R/W:	R/W	R	R	R/W	R/W	R	R	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 counter carry, or a 64 Hz counter carry when the 64 Hz counter is read. The count register value read at this time is not guaranteed, and so the count register must be read again. 0: No second counter carry, or 64 Hz counter carry when 64 Hz counter is read [Clearing condition] When 0 is written to CF 1: Second counter carry, or 64 Hz counter carry when 64 Hz counter is read [Setting conditions] Generation of a second counter carry, or a 64 Hz 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 write to these bits is invalid, but the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	CIE	0	R/W	Carry Interrupt Enable Flag Enables or disables interrupt generation when the carry flag (CF) is set to 1. 0: Carry interrupt is not generated when CF flag is set to 1 1: Carry interrupt is generated when CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag Enables or disables interrupt generation when the alarm flag (AF) is set to 1. 0: Alarm interrupt is not generated when AF flag is set to 1 1: Alarm interrupt is generated when AF flag is set to 1
2	—	Undefined	R	Reserved The initial value of these bits is undefined. A write to these bits is invalid, but the write value should always be 0.
1	CRF	Undefined	R	Carry Ready Flag Indicates whether or not RSECCNT (second counter) is in the state of the carry ready period. This flag is set to 1 when the second counter value is to be incremented after the 1 Hz bit in R64CNT (64 Hz counter) has changed from 1 to 0. However, writing to this bit is invalid, the write value 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

Bit	Bit Name	Initial Value	R/W	Description
0	AF	Undefined	R/W	Alarm Flag Set to 1 when the alarm time set in those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1 matches the respective counter values 0: Alarm registers and counter values do not match (Initial value) [Clearing condition] When 0 is written to AF 1: Alarm registers and counter values match* [Setting condition] When alarm registers in which the ENB bit is set to 1 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-second 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 PEF bit is undefined. In a manual reset, bits other than RTCEN and START are initialized, while the value of the PEF bit is undefined. In standby mode RCR2 is not initialized, and retains its current value.

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

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	Undefined	R/W	Periodic Interrupt Flag Indicates interrupt generation at the interval specified by bits PES2–PES0. When this flag is set to 1, a periodic interrupt is generated. 0: Interrupt is not generated at interval specified by bits PES2–PES0 [Clearing condition] When 0 is written to PEF 1: Interrupt is generated at interval specified by bits PES2–PES0 [Setting conditions] Generation of interrupt at interval specified by bits 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 interrupts. 000: No periodic interrupt generation 001: Periodic interrupt generated at 1/256-second intervals 010: Periodic interrupt generated at 1/64-second intervals 011: Periodic interrupt generated at 1/16-second 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 intervals 111: Periodic interrupt generated at 2-second intervals
3	RTCEN	1	R/W	Oscillator Enable Controls the operation of the RTC's crystal oscillator. 0: RTC crystal oscillator is halted 1: RTC crystal oscillator is operated

Bit	Bit Name	Initial Value	R/W	Description
2	ADJ	0	R/W	30-Second Adjustment Used for 30-second adjustment. When 1 is written to this bit, a value up to 29 seconds is rounded down to 00 seconds, and a value of 30 seconds or more is rounded up to 1 minute. The frequency divider circuits (RTC prescaler and R64CNT) are also reset at this time. This bit always returns 0 if read. 0: Normal clock operation 1: 30-second adjustment performed
1	RESET	0	R/W	Reset The frequency divider circuits are initialized by writing 1 to this bit. When 1 is written to the RESET bit, the frequency divider circuits (RTC prescaler and R64CNT) are reset and the RESET bit is automatically cleared to 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, month, and year counters are stopped* 1: Second, minute, hour, day, day-of-week, month, and year counters operate normally* Note: * The 64 Hz counter continues to operate unless stopped by means of the RTCEN bit.

20.3.18 RTC Control Register (RCR3)

RCR3 is readable/writable register that specifies enable or disable the alarm function of RYRCNT that is the RTC's BCD-coded year-value counter. When the YENB bit of RCR3 is set to 1, the RYRCNT value is compared with the RYRAR value.

RCR3 is initialized by a power-on reset.

Bits 6 to 0 of RCR3 are always read as 0. A write to these bits is invalid. If a value is written to these bits, it should always be 0.

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

20.4 Operation

Examples of the use of the RTC are shown below.

20.4.1 Time Setting Procedures

Figure 20.2 shows examples of the time setting procedures.

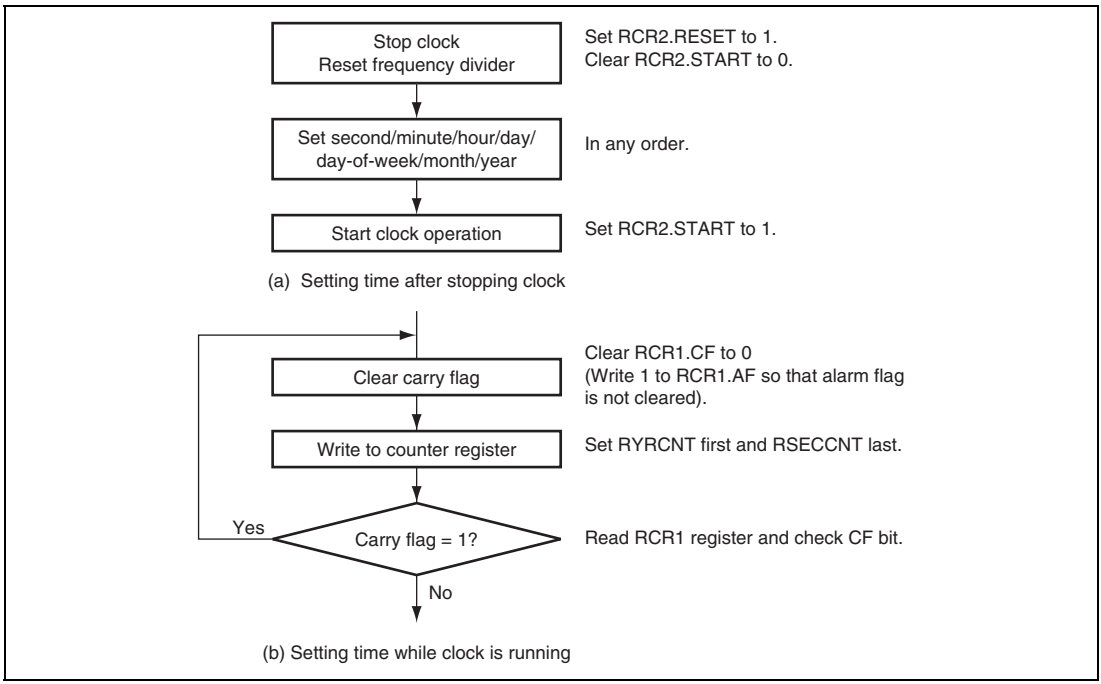


Figure 20.2 Examples of Time Setting Procedures

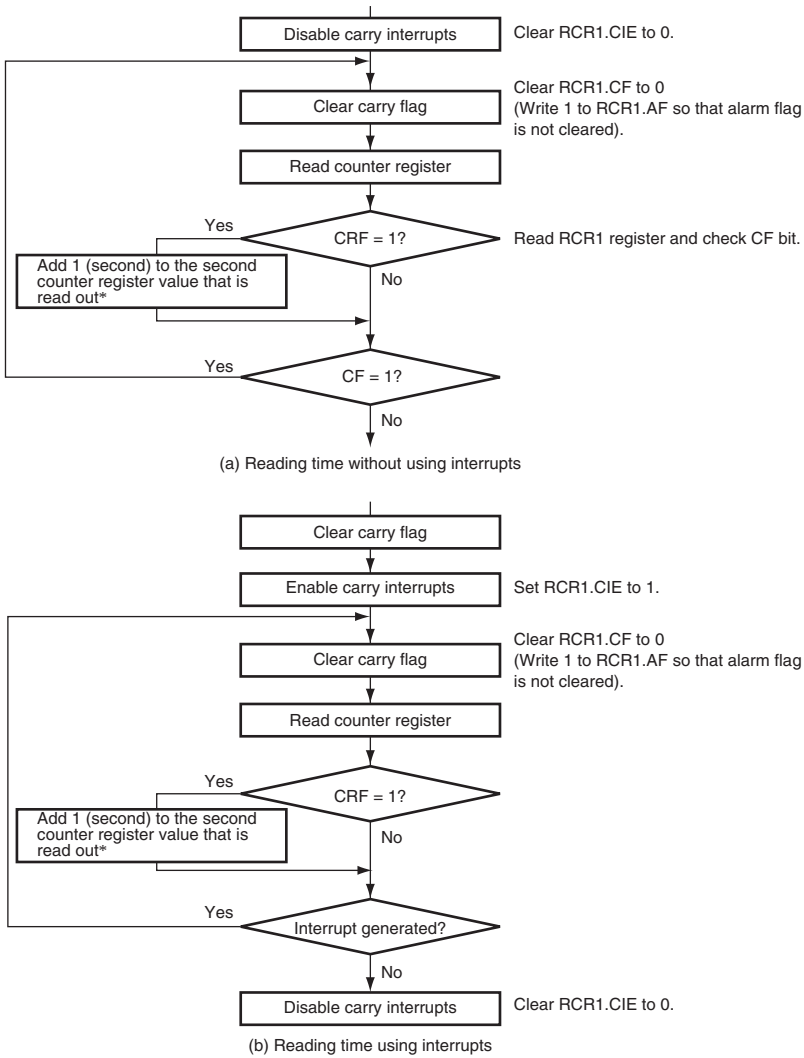
The procedure for setting the time after stopping the clock is shown in figure 20.2 (a). The programming for this method is simple, and it is useful for setting all the counters, from second to year.

The procedure for setting the time while the clock is running is shown in figure 20.2 (b). This method is useful for modifying only certain counter values (for example, only the second data or hour data). If a carry occurs during the write operation, the write data is automatically updated and there will be an error in the set data. The carry flag should therefore be used to check the write 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.

20.4.2 Time Reading Procedures

Figure 20.3 shows examples of the time reading procedures.



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

If a carry occurs or being the carry ready period (RCR1.CRF = 1) while the time is being read, the correct time will not be obtained and the read must be repeated. The procedure for reading the time without using interrupts is shown in figure 20.3 (a), and the procedure using carry interrupts in figure 20.3 (b). The method without using interrupts is normally used to keep the program simple.

20.4.3 Alarm Function

The use of the alarm function is illustrated in figure 20.4.

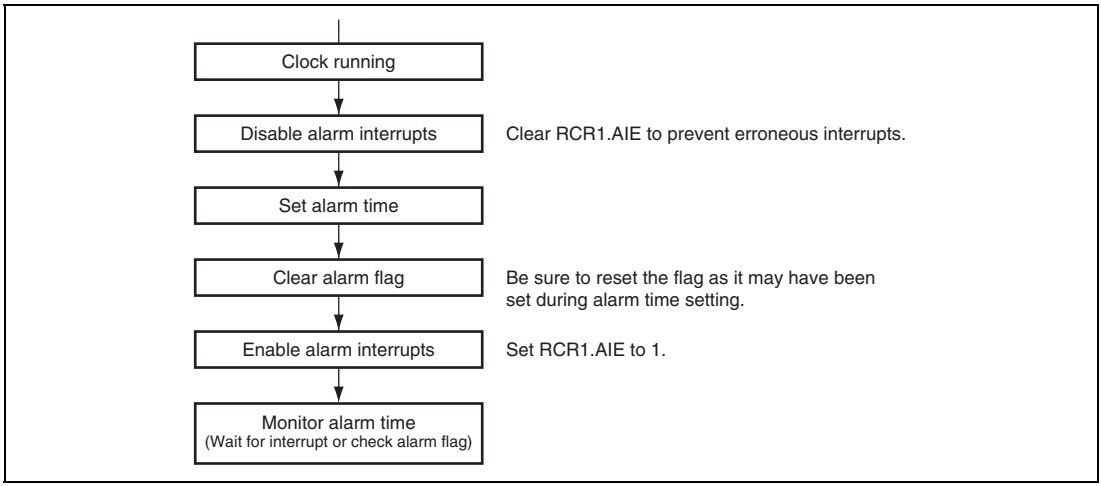


Figure 20.4 Example of Use of Alarm Function

An alarm can be generated by the second, minute, hour, day-of-week, day, month, or year value, or a combination of these. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

When the counter and the alarm time match, RCR1.AF is set to 1. Alarm detection can be 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 be detected.

20.5 Interrupts

There are three kinds of RTC interrupt: alarm interrupts, periodic interrupts, and carry interrupts.

An alarm interrupt request (ATI) is generated when the alarm flag (AF) in RCR1 is set to 1 while the alarm interrupt enable bit (AIE) is also set to 1.

A periodic interrupt request (PRI) is generated when the periodic interrupt enable bits (PES2–PES0) in RCR2 are set to a value other than 000 and the periodic interrupt flag (PEF) is set to 1.

A carry interrupt request (CUI) is generated when the carry flag (CF) in RCR1 is set to 1 while the carry interrupt enable bit (CIE) is also set to 1.

20.6 Usage Notes

20.6.1 Register Initialization

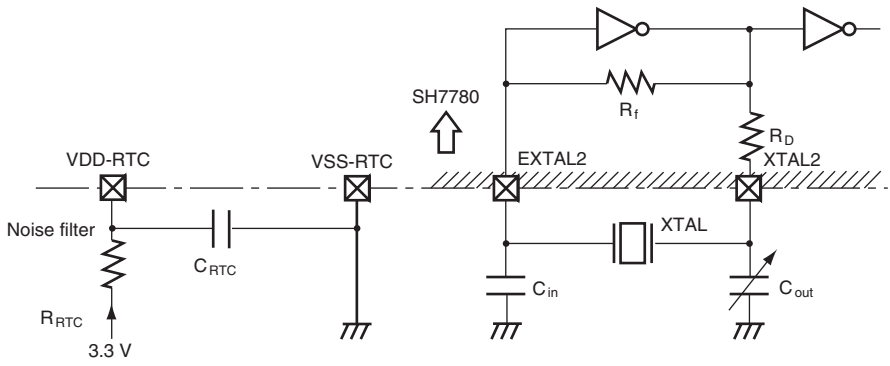
After powering on and making the RCR1 register settings, reset the frequency divider (by setting RCR2.RESET to 1) and make initial settings for all the other registers.

20.6.2 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 20.4, and the RTC 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 requirements 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 board.
4. The crystal oscillation stabilization time depends on the mounted circuit constants, floating capacitance, etc., 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 XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2 and XTAL2) wiring is routed as far away as possible 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

20.6.3 Interrupt source and request generating order

If it occurs complex interrupt source of alarm interrupts (ATI), periodic interrupts (PFI), and carry interrupts (CUI) at the same time, the RTC generates interrupt request as shown in figure 20.6.

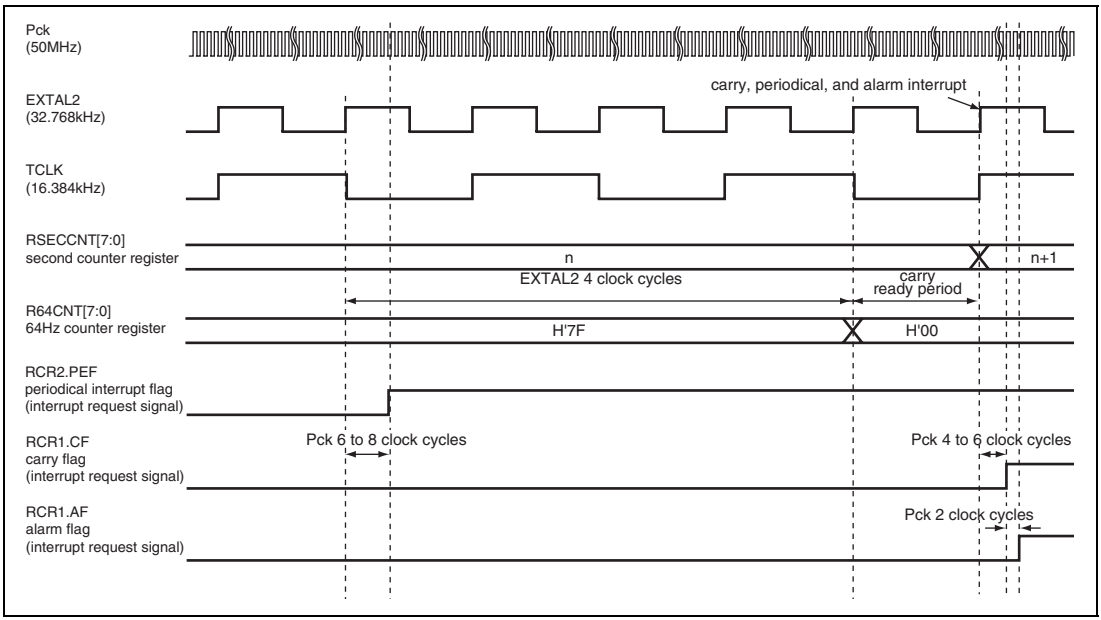


Figure 20.6 Interrupt Request Signal Generation Timing of Complex Sources

Section 21 Serial Communication Interface with FIFO (SCIF)

This LSI is equipped with a 2-channel serial communication interface with built-in FIFO buffers (Serial Communication Interface with FIFO: SCIF). The SCIF can perform both asynchronous and clocked synchronous serial communications.

64-stage FIFO buffers are provided for both transmission and reception, enabling fast, efficient, and continuous communication.

Channels 0 has modem control functions ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$).

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 be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).

  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 frame length at Space 0 (low level). When a framing error occurs, a break can also be detected by 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 can be 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

- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and reception to be performed simultaneously.

  The transmitter and receiver both have a 64-stage FIFO buffer structure, enabling continuous serial data transmission and reception.
- LSB first for data transmission/reception.
- On-chip baud rate generator allows any bit rate to be selected.
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCIF0_SCK or SCIF1_SCK pin.
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error—that can issue requests independently.
- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- 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 in channel 0)
- The amount of data in the transmit/receive FIFO registers, and the number of receive errors in 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 diagrams of the I/O ports in SCIF. There are two channels in this LSI (channel n = 0, 1).

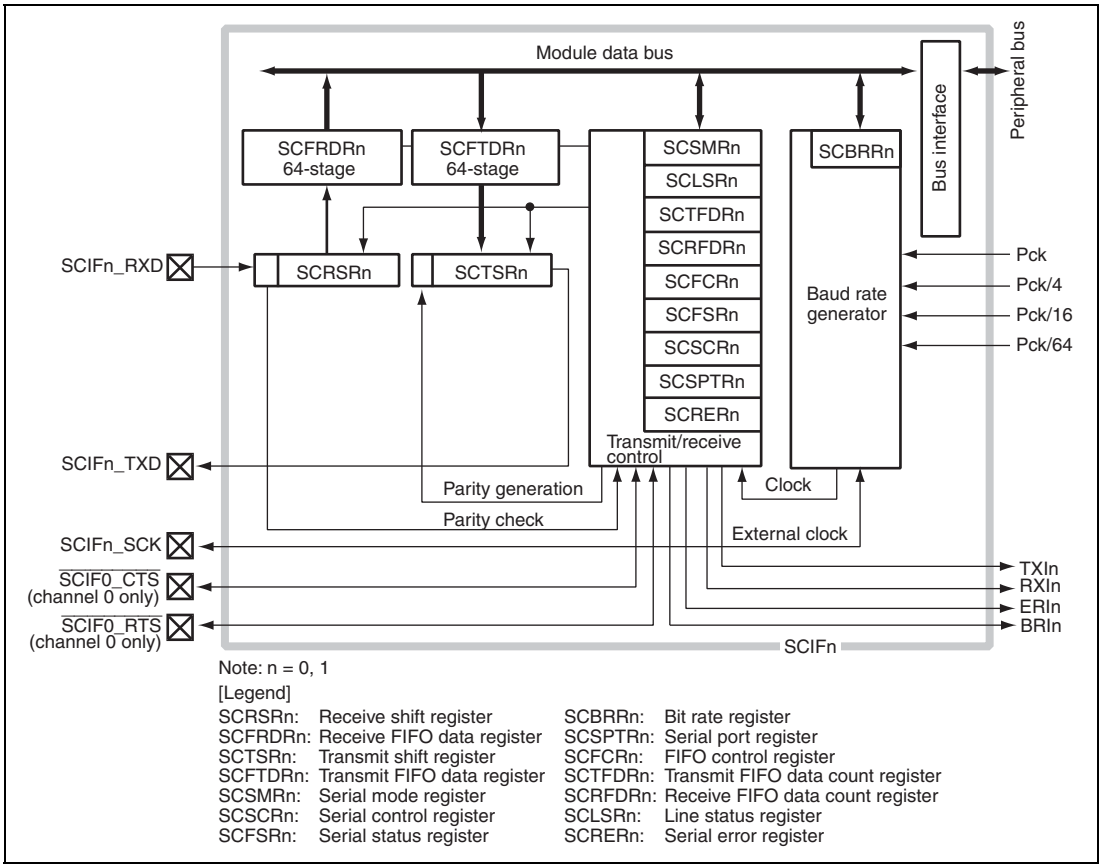
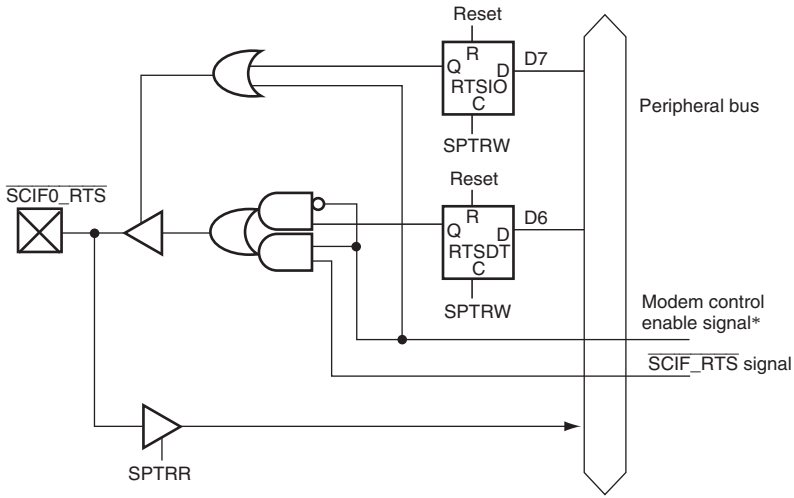


Figure 21.1 Block Diagram of SCIF

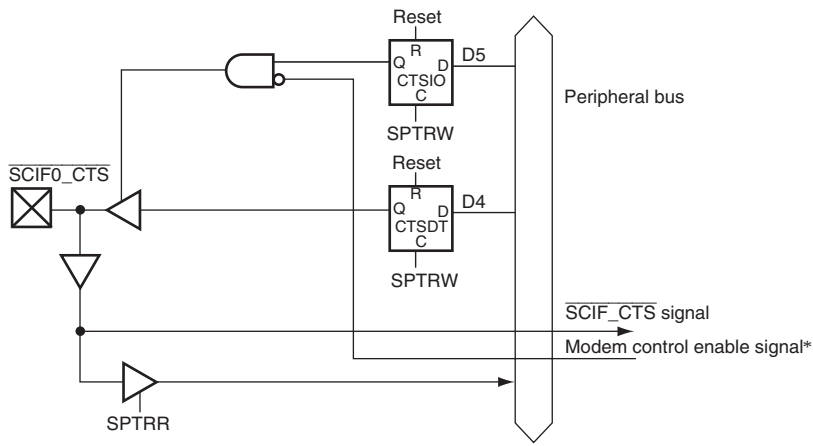
Figures 21.2 to 21.6 show block diagrams of the I/O ports in SCIF.



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The $\overline{\text{SCIF0\_RTS}}$ pin function is designated as modem control by the MCE bit in SCFCR.

Figure 21.2 $\overline{\text{SCIF0\_RTS}}$ Pin (Only in Channel 0)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The $\overline{\text{SCIF0\_CTS}}$ pin function is designated as modem control by the MCE bit in SCFCR.

Figure 21.3 $\overline{\text{SCIF0\_CTS}}$ Pin (Only in Channel 0)

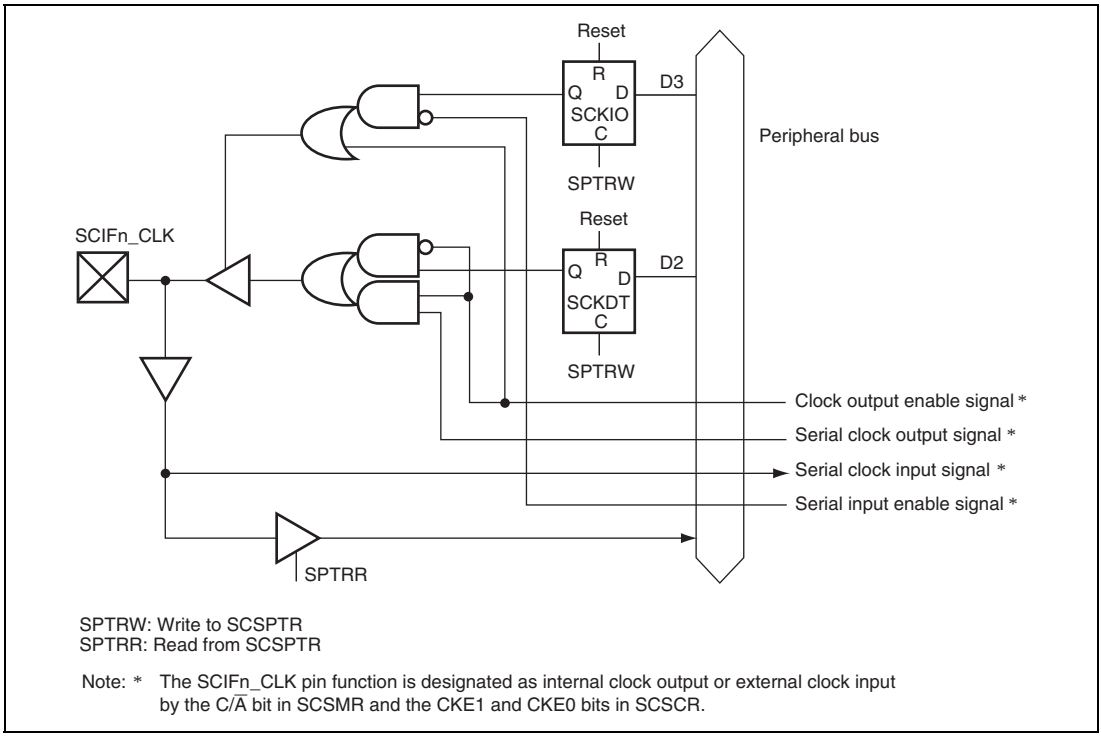


Figure 21.4 SCIFn_SCK Pin (n = 0, 1)

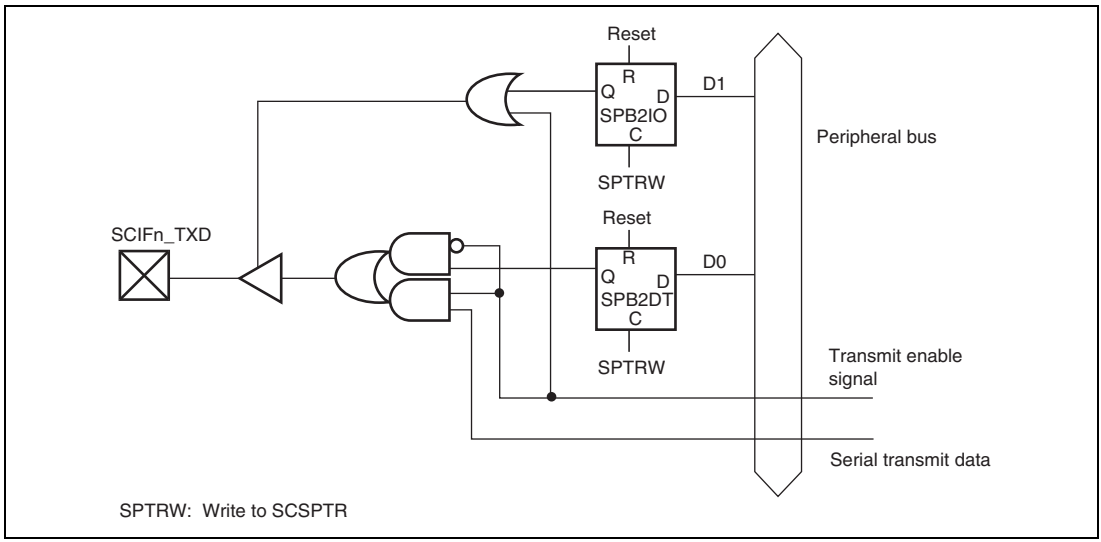


Figure 21.5 SCIFn_TXD Pin (n = 0, 1)

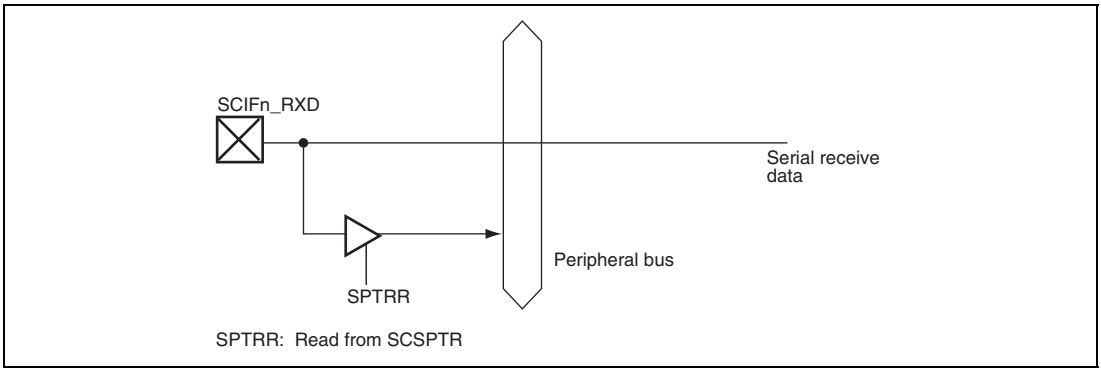


Figure 21.6 SCIFn_RXD Pin (n = 0, 1)

21.2 Input/Output Pins

Table 21.1 shows the SCIF pin configuration.

Table 21.1 Pin Configuration

Pin Name	Function	I/O	Description
SCIF0_SCK	Channel 0 serial clock pin	I/O	Clock input/output
SCIF0_RXD	Channel 0 receive data pin	Input	Receive data input
SCIF0_TXD	Channel 0 transmit data pin	Output	Transmit data output
$\overline{\text{SCIF0\_CTS}}$	Channel 0 modem control pin	I/O	Transmission enabled
$\overline{\text{SCIF0\_RTS}}$	Channel 0 modem control pin	I/O	Transmission request
SCIF1_SCK	Channel 1 serial clock pin	I/O	Clock input/output
SCIF1_RXD	Channel 1 receive data pin	Input	Receive data input
SCIF1_TXD	Channel 1 transmit data pin	Output	Transmit data output

Notes: These pins are made to function as serial pins by performing SCIF operation settings with the C/$\overline{\text{A}}$ bit in SCSMR, the TE, RE, CKE1, and CKE0 bits in SCSCR, and the MCE bit in SCFCR. Break state transmission and detection can be set in SCSPTR of the SCIF.

Channel 0 pins are multiplexed with the PCIC, HSPI, FLCTL, GPIO and mode control pins, and channel 1 pins are multiplexed with the MMCIF, GPIO and mode control pins.

21.3 Register Descriptions

Table 21.2 shows the register configuration. Table 21.3 shows the register states in each processing mode.

Table 21.2 Register Configuration

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Serial mode register 0	SCSMR0	R/W	H'FFE0 0000	H'1FE0 0000	16	Pck
	Bit rate register 0	SCBRR0	R/W	H'FFE0 0004	H'1FE0 0004	8	Pck
	Serial control register 0	SCSCR0	R/W	H'FFE0 0008	H'1FE0 0008	16	Pck
	Transmit FIFO data register 0	SCFTDR0	W	H'FFE0 000C	H'1FE0 000C	8	Pck
	Serial status register 0	SCFSR0	R/W*1	H'FFE0 0010	H'1FE0 0010	16	Pck
	Receive FIFO data register 0	SCFRDR0	R	H'FFE0 0014	H'1FE0 0014	8	Pck
	FIFO control register 0	SCFCR0	R/W	H'FFE0 0018	H'1FE0 0018	16	Pck
	Transmit FIFO data count register 0	SCTFDR0	R	H'FFE0 001C	H'1FE0 001C	16	Pck
	Receive FIFO data count register 0	SCRFDR0	R	H'FFE0 0020	H'1FE0 0020	16	Pck
	Serial port register 0	SCSPTR0	R/W	H'FFE0 0024	H'1FE0 0024	16	Pck
	Line status register 0	SCLSR0	R/W*2	H'FFE0 0028	H'1FE0 0028	16	Pck
	Serial error register 0	SCRER0	R	H'FFE0 002C	H'1FE0 002C	16	Pck
1	Serial mode register 1	SCSMR1	R/W	H'FFE1 0000	H'1FE1 0000	16	Pck
	Bit rate register 1	SCBRR1	R/W	H'FFE1 0004	H'1FE1 0004	8	Pck
	Serial control register 1	SCSCR1	R/W	H'FFE1 0008	H'1FE1 0008	16	Pck
	Transmit FIFO data register 1	SCFTDR1	W	H'FFE1 000C	H'1FE1 000C	8	Pck
	Serial status register 1	SCFSR1	R/W*1	H'FFE1 0010	H'1FE1 0010	16	Pck
	Receive FIFO data register 1	SCFRDR1	R	H'FFE1 0014	H'1FE1 0014	8	Pck
	FIFO control register 1	SCFCR1	R/W	H'FFE1 0018	H'1FE1 0018	16	Pck
	Transmit FIFO data count register 1	SCTFDR1	R	H'FFE1 001C	H'1FE1 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	R	H'FFE1 0020	H'1FE1 0020	16	Pck
	Serial port register 1	SCSPTR1	R/W	H'FFE1 0024	H'1FE1 0024	16	Pck
	Line status register 1	SCLSR1	R/W*2	H'FFE1 0028	H'1FE1 0028	16	Pck
	Serial error register 1	SCRER1	R	H'FFE1 002C	H'1FE1 002C	16	Pck

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.

Table 21.3 Register States of SCIF in Each Processing Mode

Ch.	Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/ WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep by SLEEP Instruction	Module Standby
0	Serial mode register 0	SCSMR0	H'0000	H'0000	Retained	Retained
	Bit rate register 0	SCBRR0	H'FF	H'FF	Retained	Retained
	Serial control register 0	SCSCR0	H'0000	H'0000	Retained	Retained
	Transmit FIFO data register 0	SCFTDR0	Undefined	Undefined	Retained	Retained
	Serial status register 0	SCFSR0	H'0060	H'0060	Retained	Retained
	Receive FIFO data register 0	SCFRDR0	Undefined	Undefined	Retained	Retained
	FIFO control register 0	SCFCR0	H'0000	H'0000	Retained	Retained
	Transmit FIFO data count register 0	SCTFDR0	H'0000	H'0000	Retained	Retained
	Receive FIFO data count register 0	SCRFDR0	H'0000	H'0000	Retained	Retained
	Serial port register 0	SCSPTR0	H'0000*[1]	H'0000*[1]	Retained	Retained
	Line status register 0	SCLSR0	H'0000	H'0000	Retained	Retained
	Serial error register 0	SCRER0	H'0000	H'0000	Retained	Retained
1	Serial mode register 1	SCSMR1	H'0000	H'0000	Retained	Retained
	Bit rate register 1	SCBRR1	H'FF	H'FF	Retained	Retained
	Serial control register 1	SCSCR1	H'0000	H'0000	Retained	Retained
	Transmit FIFO data register 1	SCFTDR1	Undefined	Undefined	Retained	Retained
	Serial status register 1	SCFSR1	H'0060	H'0060	Retained	Retained
	Receive FIFO data register 1	SCFRDR1	Undefined	Undefined	Retained	Retained
	FIFO control register 1	SCFCR1	H'0000	H'0000	Retained	Retained
	Transmit FIFO data count register 1	SCTFDR1	H'0000	H'0000	Retained	Retained
	Receive FIFO data count register 1	SCRFDR1	H'0000	H'0000	Retained	Retained
	Serial port register 1	SCSPTR1	H'0000*[2]	H'0000*[2]	Retained	Retained
	Line status register 1	SCLSR1	H'0000	H'0000	Retained	Retained
	Serial error register 1	SCRER1	H'0000	H'0000	Retained	Retained

Notes: 1. Bits 2 and 0 are undefined.
2. Bits 6, 4, 2, and 0 are undefined.

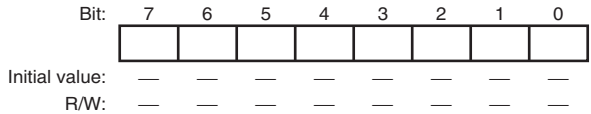
Since the register functions, pin functions, and interrupt requests are the same in each channel except for the modem control, the channel number n (n = 0, 1) is omitted in the description below.

21.3.1 Receive Shift Register (SCRSR)

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the SCIF_RXD pin in SCRSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to SCFRDR, automatically.

SCRSR cannot be directly read from and written to by the CPU.



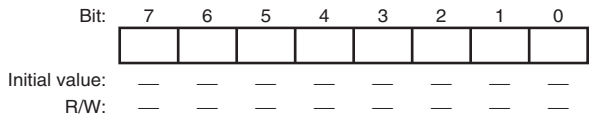
21.3.2 Receive FIFO Data Register (SCFRDR)

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 SCRSR to SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled for reception, and consecutive receive operations can be performed until SCFRDR is full (64 data 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 be returned. When SCFRDR is full of receive data, subsequent serial data is lost.



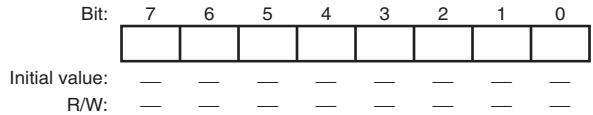
21.3.3 Transmit Shift Register (SCTSR)

SCTSR is the register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTDR to SCTSR, then sends the data to the SCIF_TXD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from SCFTDR to SCTSR, and transmission started, automatically.

SCTSR cannot be directly read from and written to by the CPU.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

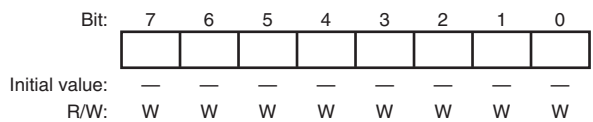
21.3.4 Transmit FIFO Data Register (SCFTDR)

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 the 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. Data written in this case is ignored.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

21.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial transfer format and select the baud rate generator clock source.

SCSMR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
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/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or clocked synchronous 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 data length. In clocked synchronous mode, the data length is fixed at 8 bits regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of SCFTDR is not transmitted. 0: 8-bit data 1: 7-bit data

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking is performed in reception. In clocked synchronous mode, parity bit addition and checking is disabled regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled* Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. In asynchronous mode, the O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking. In clocked synchronous mode or when parity addition and checking is disabled in asynchronous mode, the O/$\overline{E}$ bit setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is even. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is odd.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length In asynchronous mode, selects 1 or 2 bits as the stop bit length. The stop bit setting is valid only in asynchronous mode. Since the stop bit is not added in clocked synchronous mode, the STOP bit setting is invalid. 0: 1 stop bit*[1] 1: 2 stop bits*[2] In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character. Note: 1. In transmission, a single 1-bit (stop bit) is added to the end of a transmit character before it is sent. 2. In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. The clock source can be selected from Pck, Pck/4, Pck/16, and Pck/64, according to the setting of bits CKS1 and CKS0. For details of the relationship between clock sources, 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

21.3.6 Serial Control Register (SCSCR)

SCSCR is a register used to enable/disable transmission/reception by SCIF, serial clock output, interrupt requests, and to select transmission/reception clock source for the SCIF.

SCSCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
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/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when serial transmit data is transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR falls to or below the transmit trigger set number, and the TDFE flag in SCFSR is set to 1. TXI interrupt requests can be cleared using the following methods: Either by reading 1 from the TDFE flag in SCFSR, writing transmit data exceeding the transmit trigger set number to SCFTDR and then clearing the TDFE flag in SCFSR to 0, or by clearing the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag or DR flag in SCFSR is set to 1, a receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, and a break interrupt (BRI) request when the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1. 0: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request disabled 1: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request enabled Note: An RXI interrupt request can be cleared by reading 1 from the RDF or DR flag in SCFSR, then clearing the flag to 0, or by clearing the RIE bit to 0. ERI and BRI interrupt requests can be cleared by reading 1 from the ER, BRK, or ORER flag in SCFSR, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmission by the SCIF. Serial transmission is started when transmit data is written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled* Note: SCSMR and SCFCR settings must be made, the transmission format decided, and the transmit FIFO reset, before the TE bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the start of serial reception by the SCIF. Serial reception is started when a start bit is detected in this state in asynchronous mode or a synchronization clock is input while the RE bit is set to 1. It should be noted that clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER flags in SCFSR, and ORER flag in SCLSR, which retain their states. Serial reception begins once the start bit is detected in these states. 0: Reception disabled 1: Reception enabled* Note: * SCSMR and SCFCR settings must be made, the reception format decided, and the receive FIFO reset, before the RE bit is set to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-error interrupt (ERI) and break interrupt (BRI) requests. The REIE bit setting is valid only when the RIE bit is 0. Receive-error interrupt (ERI) and break interrupt (BRI) requests can be cleared by reading 1 from the ER, BRK in SCFSR, or ORER flag in SCLSR, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0. When REIE is set to 1, ERI and BRI interrupt requests will be generated even if RIE is cleared to 0. In DMA transfer, this setting is made if the interrupt controller is to be notified of ERI and BRI interrupt requests. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	These bits select the SCIF clock source and whether to enable or disable the clock output from the SCIF_SCK pin. The CKE1 and CKE0 bits are used together to specify whether the SCIF_SCK pin functions as a serial clock output pin or a serial clock input pin. Note however that the CKE0 bit setting is valid only when an internal clock is selected as the SCIF clock source (CKE1 = 0). When an external clock is selected (CKE1 = 1), the CKE0 bit setting is invalid. The CKE1 and CKE0 bits must be set before determining the SCIF's operating mode with SCSMR. • Asynchronous mode 00: Internal clock/SCIF_SCK pin functions as port by setting SCSPTR register 01: Internal clock/SCIF_SCK pin functions as clock output*[1] 1x: External clock/SCIF_SCK pin functions as clock 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 times the bit rate. 2. Inputs a clock with a frequency 16 times the bit rate.

21.3.7 Serial Status Register n (SCFSR)

SCFSR is a 16-bit register that consists of status flags that indicate the operating status of the SCIF.

SCFSR can be read from or written to by the CPU at all times. However, 1 cannot be written to flags ER, TEND, TDFE, BRK, RDF, and DR. Also note that in order to clear these flags they must be read as 1 beforehand. The FER flag and PER flag are read-only flags and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	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	R/W*	R/W*

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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/W*	Receive Error Indicates that a framing error or parity error occurred during reception. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a receive error occurs, the receive data is still transferred to SCFRDR, and reception continues. The FER and PER bits in SCFSR can be used to determine whether there is a receive error in the readout data from SCFRDR. 0: No framing error or parity error occurred during reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during reception [Setting conditions] • When the SCIF checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0* • When, in reception, the number of 1-bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SCSMR Note: In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the second stop bit is not checked.
6	TEND	1	R/W*	Transmit End Indicates that transmission has been ended without valid data in SCFTDR after transmission of the last bit of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, and 0 is written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DMAC 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 after transmission of the last bit of a 1-byte serial transmit character

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/W*	Transmit FIFO Data Empty Indicates that data has been transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR has fallen to or below the transmit trigger data number set by bits TTRG1 and TTRG0 in SCFCR, and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the transmit trigger set number have been written to SCFTDR [Clearing conditions] • When transmit data exceeding the transmit trigger set number is written to SCFTDR after reading TDFE = 1, and 0 is written to TDFE • When transmit data exceeding the transmit trigger set number is written to SCFTDR by the DMAC 1: The number of transmit data bytes in SCFTDR does not exceed the transmit trigger set number (Initial value) [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger set number as the result of a transmit operation* Note: As SCFTDR is a 64-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 64 - (transmit trigger set number). Data written in excess of this will be ignored. SCTFDR indicates the number of data bytes transmitted to SCFTDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/W*	Break Detect Indicates that a receive data break signal has been detected. 0: A break signal has not been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after reading BRK = 1 1: A break signal has been received* [Setting condition] • When data with a framing error is received, followed by the space "0" level (low level) for at least one frame length Note: When a break is detected, the receive data (H'00) following detection is not transferred to SCFRDR. When the break ends and the receive signal returns to mark "1", receive data transfer is resumed.
3	FER	0	R	Framing Error In asynchronous mode, indicates whether or not a framing error has been found in the data that is to be read next from SCFRDR. 0: There is no framing error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data that is to be read next from SCFRDR 1: There is a framing error that is to be read from SCFRDR [Setting condition] • When there is a framing error in the data that is to be read next from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error In asynchronous mode, indicates whether or not a parity error has been found in the data that is to be read next from SCFRDR. 0: There is no parity error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in the data that is to be read next from SCFRDR 1: There is a parity error in the receive data that is to be read from SCFRDR [Setting condition] • When there is a parity error in the data that is to be read next from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/W*	Receive FIFO Data Full Indicates that the received data has been transferred from SCRSR to SCFRDR, and the number of receive data bytes in SCFRDR is equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. 0: The number of receive data bytes in SCFRDR is less than the receive trigger set number [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger set number after reading RDF = 1, and 0 is written to RDF • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR falls below the receive trigger set number 1: The number of receive data bytes in SCFRDR is equal to or greater than the receive trigger set number [Setting condition] • When SCFRDR contains at least the receive trigger set number of receive data bytes* Note: SCFRDR is a 64-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If all the data in SCFRDR is read and another read is performed, the data value will be undefined. The number of receive data bytes in SCFRDR is indicated by SCRFDR.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/W*	Receive Data Ready In asynchronous mode, indicates that there are fewer than the receive trigger set number of data bytes in SCFRDR, and no further data has arrived for at least 15 etu after the stop bit of the last data received. This is not set when using clocked synchronous mode. 0: Reception is in progress or has ended normally and there is no receive data left in SCFRDR [Clearing conditions] • Power-on reset or manual reset • When all the receive data in SCFRDR has been read after reading DR = 1, and 0 is written to DR • When all the receive data in SCFRDR has been read by the DMAC 1: No further receive data has arrived [Setting condition] • When SCFRDR contains fewer than the receive trigger set number of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received* [Legend] etu: Elementary time unit (time for transfer of 1 bit) Note: Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.

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

21.3.8 Bit Rate Register n (SCBRR)

SCBRR is an 8-bit register that set the serial transmission/reception bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

SCBRR can always be read from and written to by the CPU.

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{Pck}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

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:

$$\text{Error (\%)} = \left\{ \frac{Pck \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

21.3.9 FIFO Control Register n (SCFCR)

SCFCR performs data count resetting and trigger data number setting for transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RST RG2*	RST RG1*	RST RG0*	RTRG1	RTRG0	TTRG1	TTRG0	MCE*	TFCL	RFCL	LOOP
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 value 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 the number of receive data stored in SCFRDR exceeds the trigger number shown below.
8	RSTRG0*	0	R/W	
				000:63
				001:1
				010:8
				011:16
				100:32
				101:48
				110:54
				111:60

Bit	Bit Name	Initial Value	R/W	Description
7	RTRG1	0	R/W	Receive FIFO Data Number Trigger
6	RTRG0	0	R/W	These bits are used to set the number of receive data bytes that sets the RDF flag in SCFSR. The RDF flag is set when the number of receive data bytes in SCFRDR is equal to or greater than the trigger set number shown below. 00:1 01:16 10:32 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 remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when the number of transmit data bytes in SCFTDR is equal to or less than the trigger set number shown below. 00: 32 (32) 01:16 (48) 10: 2 (62) 11: 0 (64) Note: Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.
3	MCE*	0	R/W	Modem Control Enable Enables the $\overline{\text{SCIF0\_CTS}}$ and $\overline{\text{SCIF0\_RTS}}$ modem control signals. Always set the MCE bit to 0 in clocked synchronous mode. 0: Modem signals disabled 1: Modem signals enabled Note: When the MCE bit is 0, $\overline{\text{SCIF0\_CTS}}$ is fixed at active-0 regardless of the input value, and $\overline{\text{SCIF0\_RTS}}$ output is also fixed at 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TFCL	0	R/W	Transmit FIFO Data Count Register Clear Clears the transmit FIFO data count register to 0. 0: Clear operation disabled 1: Clear operation enabled Note: A reset operation is performed in the event of a power-on reset or manual reset.
1	RFCL	0	R/W	Receive FIFO Data Count Register Clear Clears the transmit FIFO data count register to 0. 0: Clear operation disabled 1: Clear operation enabled Note: A reset operation is performed in the event of a power-on reset or manual reset.
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SCIF_TXD) and receive input pin (SCIF_RXD), and the $\overline{\text{SCIF0\_RTS}}$ pin and $\overline{\text{SCIF0\_CTS}}$ pin (for channel 0), enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Note: * Only channel 0. Reserved bit in channel 1.

21.3.10 Transmit FIFO Data Count Register n (SCTFDR)

SCTFDR is a 16-bit register that indicates the number of transmit data bytes stored in SCFTDR.

SCTFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	T6	T5	T4	T3	T2	T1	T0
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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	T6 to T0	All 0	R	These bits show the number of untransmitted data bytes in SCFTDR. A value of H'00 indicates that there is no transmit data, and a value of H'40 indicates that SCFTDR is full of transmit data.

21.3.11 Receive FIFO Data Count Register n (SCRFDR)

SCRFDR is a 16-bit register that indicates the number of receive data bytes stored in SCFRDR.

SCRFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	R6	R5	R4	R3	R2	R1	R0
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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	R6 to R0	All 0	R	These bits show the number of receive data bytes in SCFRDR. A value of H'00 indicates that there is no receive data, and a value of H'40 indicates that SCFRDR is full of receive data.

21.3.12 Serial Port Register n (SCSPTR)

SCSPTR is a 16-bit readable/writable register that controls input/output and data for the port pins multiplexed with the serial communication interface (SCIF) pins at all times. Input data can be read from the SCIF_RXD pin, output data written to the SCIF_TXD pin, and breaks in serial transmission/reception controlled, by means of bits 1 and 0.

All SCSPTR bits except bits 6, 4, 2, and 0 are initialized to 0 by a power-on reset or manual reset; the value of bits 6, 4, 2, and 0 is undefined. SCSPTR is not initialized in the module standby state.

Note that when reading data via a serial port pin in the SCIF, the peripheral clock value from 2 cycles before is read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RTS IO*	RTS DT*	CTS IO*	CTS DT*	SCK IO	SCK DT	SPB2 IO	SPB2 DT
Initial value:	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/W	R/W

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 value 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/output condition. When actually setting the $\overline{\text{SCIF0\_RTS}}$ pin as a port output pin to output the value set by the RTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value is not output to $\overline{\text{SCIF0\_RTS}}$ pin 1: RTSDT bit value is output to $\overline{\text{SCIF0\_RTS}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
6	RTSDT*	—	R/W	Serial Port $\overline{\text{SCIF0\_RTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF0\_RTS}}$ pin input/output data. Input or output is specified by the RTSIO bit. In output mode, the RTSDT bit value is output to the $\overline{\text{SCIF0\_RTS}}$ pin. The $\overline{\text{SCIF0\_RTS}}$ pin value is read from the RTSDT bit regardless of the value of the RTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
5	CTSIO*	0	R/W	Serial Port $\overline{\text{SCIF0\_CTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF0\_CTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF0\_CTS}}$ pin as a port output pin to output the value set by the CTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value is not output to $\overline{\text{SCIF0\_CTS}}$ pin 1: CTSDT bit value is output to $\overline{\text{SCIF0\_CTS}}$ pin
4	CTSDT*	—	R/W	Serial Port $\overline{\text{SCIF0\_CTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF0\_CTS}}$ pin input/output data. Input or output is specified by the CTSIO bit. In output mode, the CTSDT bit value is output to the $\overline{\text{SCIF0\_CTS}}$ pin. The $\overline{\text{SCIF0\_CTS}}$ pin value is read from the CTSDT bit regardless of the value of the CTSIO bit. The initial value of this bit after a power-on 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/output condition. When actually setting the SCIF_SCK pin as a port output pin to output the value set by the SCKDT bit, the CKE1 and CKE0 bits in SCSCR should be cleared to 0. 0: SCKDT bit value is not output to SCIF_SCK pin 1: SCKDT bit value is output to SCIF_SCK pin

Bit	Bit Name	Initial Value	R/W	Description
2	SCKDT	—	R/W	Serial Port Clock Port Data Specifies the serial port SCIF_SCK pin input/output data. Input or output is specified by the SCKIO bit. In output mode, the SCKDT bit value is output to the SCIF_SCK pin. The SCIF_SCK pin value is read from the SCKDT bit regardless of the value of the SCKIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 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 condition. When actually setting the SCIF_TXD pin as a port output pin to output the value set by the SPB2DT bit, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_TXD pin 1: SPB2DT bit value is output to the SCIF_TXD pin
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input data and SCIF_TXD pin output data. The SCIF_TXD pin output condition is specified by the SPB2IO bit. When the SCIF_TXD pin is designated as an output, the value of the SPB2DT bit is output to the SCIF_TXD pin. The SCIF_RXD pin value is read from the SPB2DT bit regardless of the value of the SPB2IO bit. The initial value of this bit after a power-on reset or manual reset 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.

21.3.13 Line Status Register n (SCLSR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ORER
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	R	R	R/W*

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/W*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Reception in progress, or reception has ended normally [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. 1: An overrun error occurred during reception [Setting condition] • When the next serial reception is completed while SCFRDR receives 64-byte data (SCFRDR is full) The receive data prior to the overrun error is retained in SCFRDR, and the data received subsequently is lost. Serial reception cannot be continued while the ORER flag is set to 1.

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

21.3.14 Serial Error Register n (SCRER)

SCRER is a 16-bit register that indicates the number of receive errors in the data in SCFRDR. SCRER can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PER5	PER4	PER3	PER2	PER1	PER0	—	—	FER5	FER4	FER3	FER2	FER1	FER0
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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value 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 which a parity error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits PER5 to PER0 is the number of data bytes in which a parity error occurred. If all 64 bytes of receive data in SCFRDR have parity errors, the value indicated by bits PER5 to PER0 will be 0.
11	PER3	0	R	
10	PER2	0	R	
9	PER1	0	R	
8	PER0	0	R	
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value 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 which a framing error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits FER5 to FER0 is the number of data bytes in which a framing error occurred. If all 64 bytes of receive data in SCFRDR have framing errors, the value indicated by bits FER5 to FER0 will be 0.
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

21.4 Operation

21.4.1 Overview

The SCIF can carry out serial communication in asynchronous mode, in which synchronization is achieved character by character and in synchronous mode, in which synchronization is achieved with clock pulses. For details on asynchronous mode, see section 21.4.2, Operation in Asynchronous Mode.

64-stage FIFO buffers are provided for both transmission and reception, reducing the CPU 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 (channel 0 only).

The serial transfer format is selected using SCSMR, as shown in table 21.4. The SCIF clock source is determined by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as shown in table 21.5.

Note: Since the operations are the same in each channel except for the modem control, the 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 parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun errors, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Choice of internal (peripheral clock: Pck) or external clock (SCIF_SCK input clock) as SCIF clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock and can 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 must be input (the on-chip baud rate generator is not used).

Clocked Synchronous Mode:

- Data length: Fixed at 8 bits
- LSB first for data transmission/reception
- Detection of overrun errors during reception
- Choice of internal or external clock input from SCIF_SCK pin as SCIF clock source

  When internal clock (peripheral clock: Pck) is selected:

  The SCIF operates on the baud rate generator clock and a serial clock is output to external devices.

  When external clock (SCIF_SCK input clock) is selected:

  The on-chip baud rate generator is not used and the SCIF operates on the input serial clock.

Table 21.5 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					SCIF Transfer Format		
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous mode	8-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
	1	0	0		7-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous mode	8-bit data	No	No

Note: x: Don't care

Table 21.6 SCSMR and SCSCR Settings for SCIF Clock Source Selection

SCSMR	SCSCR Settings				
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCK Pin Function**
0	0	0	Asynchronous mode	Internal	SCIF does not use SCIF_SCK pin
		1			Outputs clock with frequency of 16 times the bit rate
	1	0		External	Inputs clock with frequency of 16
		1			times the bit rate
1	0	0	Clocked synchronous mode	Internal	Outputs synchronization clock
		1			
	1	0		External	Inputs synchronization clock
		1			

21.4.2 Operation in Asynchronous Mode

In asynchronous mode, a character that consists of data with a start bit indicating the start of communication and a stop bit indicating the end of communication is transmitted or received. In this mode, serial communication is performed with synchronization achieved character by character.

Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

Figure 21.7 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCIF monitors the transmission line, and when it goes to the space state (low level), 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 finally stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. Receive data can be latched at the middle of each bit because the SCIF samples data at the eighth clock which has a frequency of 16 times the bit rate.

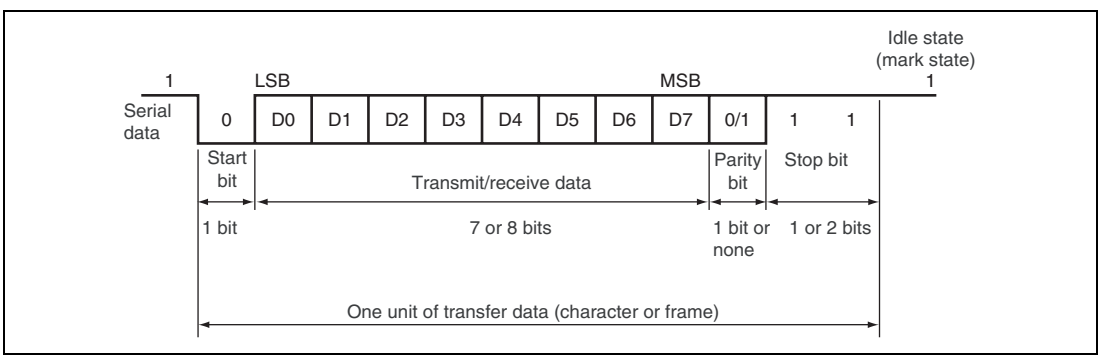


Figure 21.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

(1) Data Transfer Format

Table 21.7 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 21.7 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
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	STOP	

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

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 21.5.

When an external clock is input at the SCIF_SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCIF is operated on an internal clock, a clock whose frequency is 16 times the bit rate is 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 SCSCR to 0, then initialize the SCIF as described below.

When the operating mode or transfer format, etc., is changed, the TE and RE bits must be cleared 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 RE bits to 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 flag in SCFSR has been set. TEND can also be cleared to 0 during transmission, but the data being transmitted will go to the mark state after the clearance. Before setting TE again to start 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, including initialization, since operation will be unreliable in this case.

Figure 21.8 shows a sample SCIF initialization flowchart.

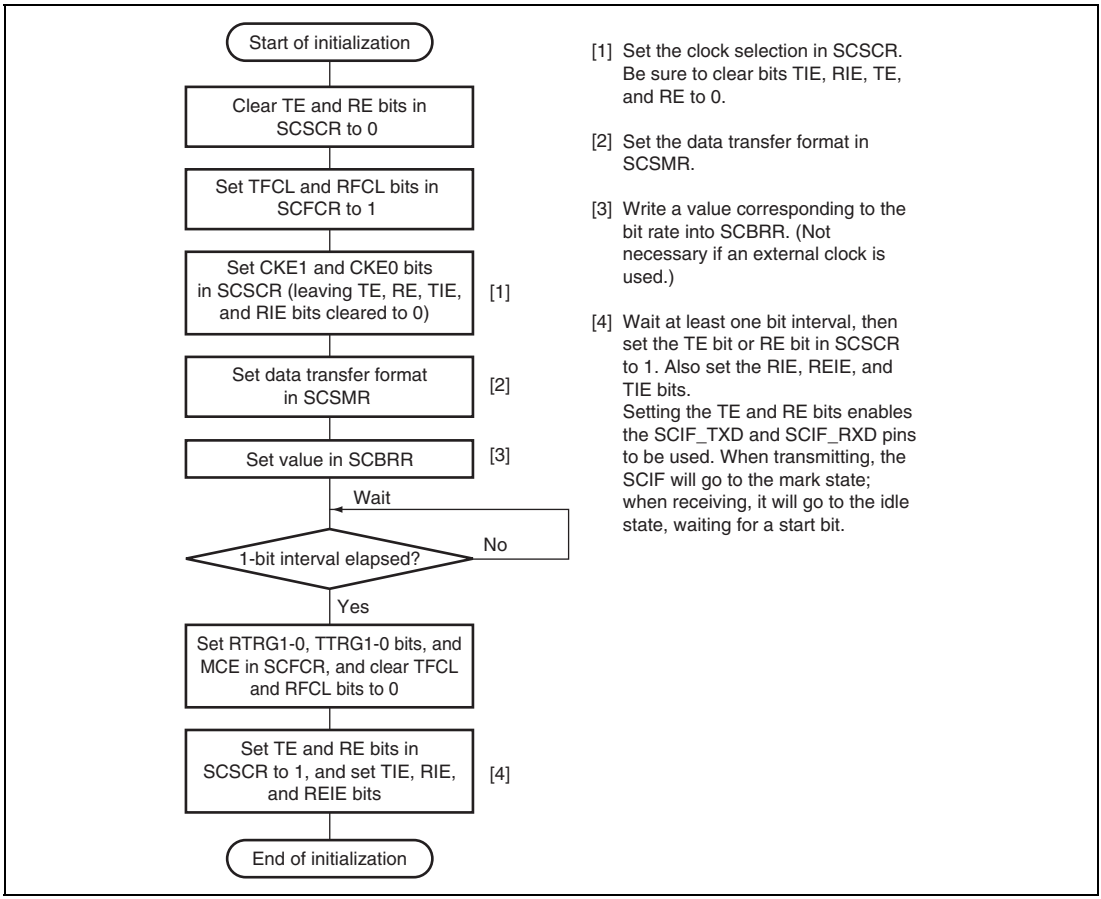


Figure 21.8 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Asynchronous Mode):

Figure 21.9 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

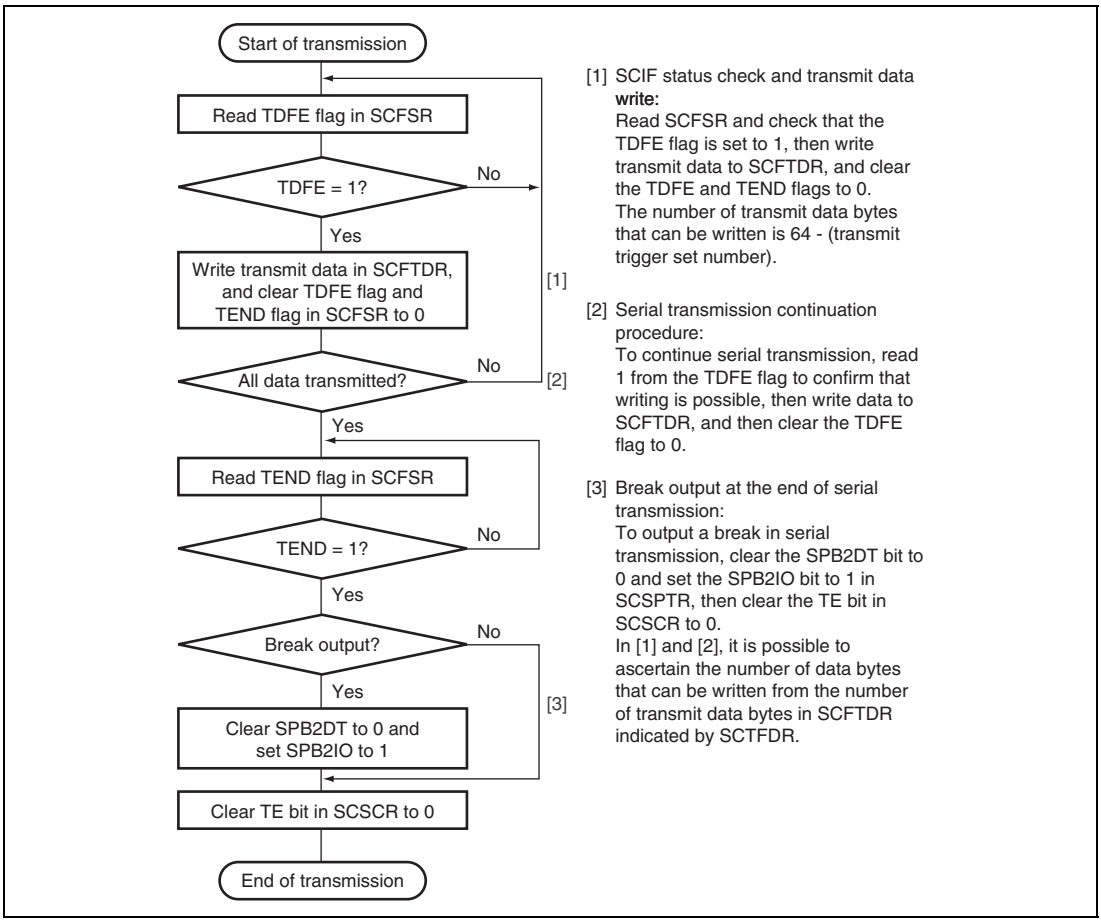


Figure 21.9 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 − (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the 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 parity bit is not output can also be selected.

   (d) Stop bit(s): One or two 1-bits (stop bits) are output.

   (e) Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output from the SCIF_TXD pin.

Figure 21.10 shows an example of the operation for transmission in asynchronous mode.

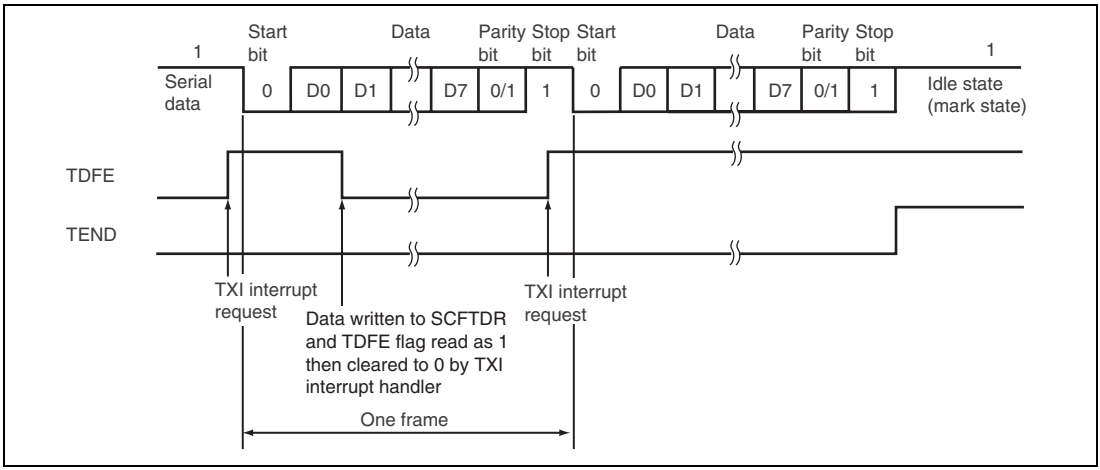


Figure 21.10 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{SCIF0\_CTS}}$ input value. When $\overline{\text{SCIF0\_CTS}}$ is set to 1 during transmission , the line goes to the mark state after transmission of one frame. When $\overline{\text{SCIF0\_CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 21.11 shows an example of the operation when modem control is used.

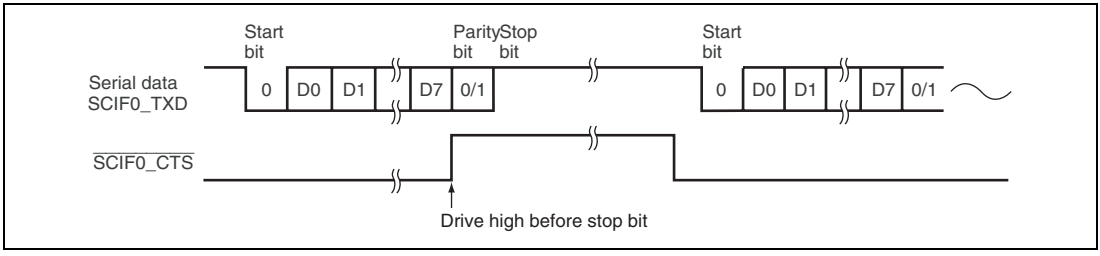


Figure 21.11 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_CTS}}$) (Only in Channel 0)

(5) Serial Data Reception (Asynchronous Mode)

Figure 21.12 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

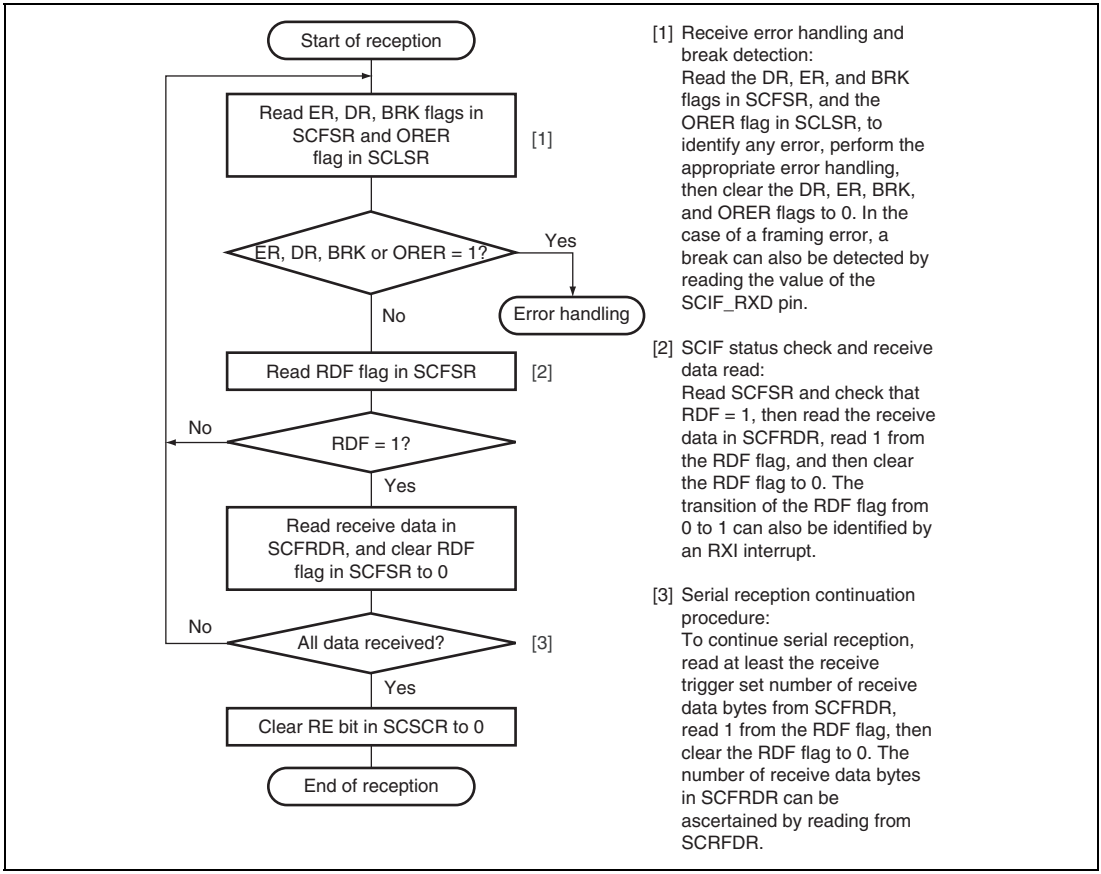
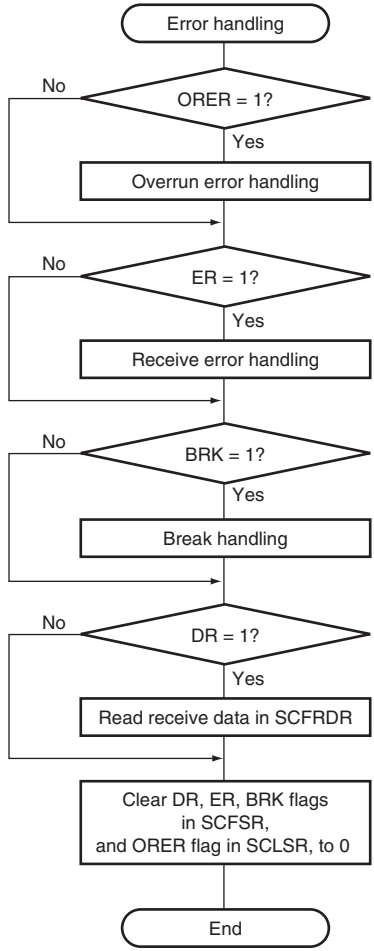


Figure 21.12 Sample Serial Reception Flowchart (1)



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 21.12 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0-start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCIF carries out the following checks.

   (a) Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   (b) The SCIF checks whether receive data can be transferred from SCRSR to SCFRDR.*

   (c) Overrun error check: The SCIF checks that the ORER flag is 0, indicating that no overrun error has occurred.*

   (d) Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not 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-FIFO-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 receive-error 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 to 1, a break reception interrupt (BRI) request is generated.

Figure 21.13 shows an example of the operation for reception in asynchronous mode.

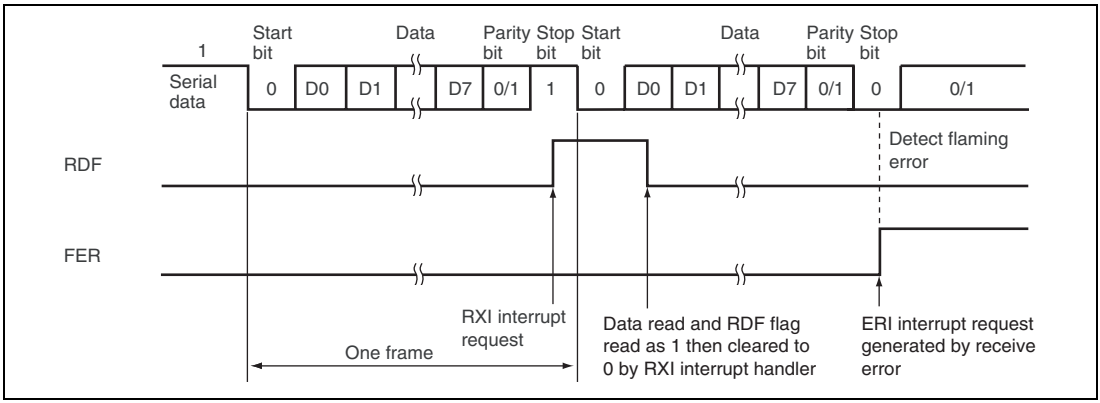


Figure 21.13 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{SCIF0\_RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{SCIF0\_RTS}}$ is 0, reception is possible. When $\overline{\text{SCIF0\_RTS}}$ is 1, this indicates that SCFRDR contains bytes of data equal to or more than the $\overline{\text{SCIF0\_RTS}}$ output active trigger number. The $\overline{\text{SCIF0\_RTS}}$ output active trigger value is specified by bits 10 to 8 in the FIFO control register (SCFCR). For details, see section 21.3.9, FIFO Control Register n (SCFCR). In 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.

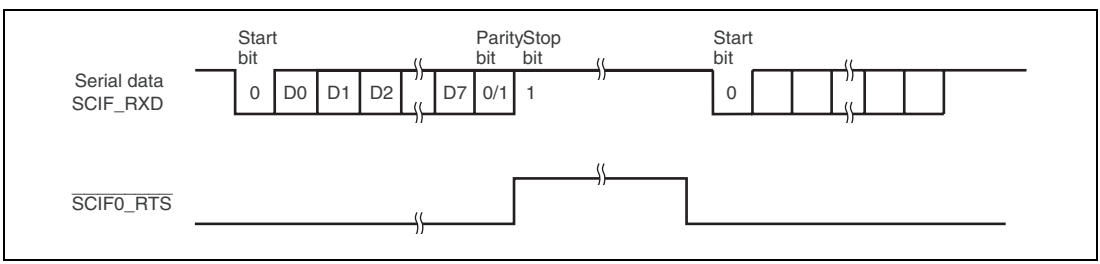


Figure 21.14 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_RTS}}$) (Only in Channel 0)

21.4.3 Operation in Clocked Synchronous Mode

Clocked synchronous mode, in which data is transmitted or received in synchronization with clock pulses, is suitable for fast serial communication.

Since the transmitter and receiver are independent units in the SCIF, full-duplex communication can be achieved by sharing the clock. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

Figure 21.15 shows the general format for clocked synchronous communication.

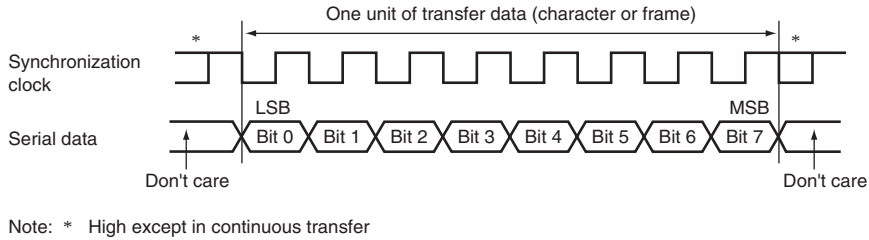


Figure 21.15 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, data on the communication line is output from one fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the start of the synchronization clock.

In serial communication, each character is output starting with the LSB and ending with the MSB. 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 of the synchronization clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity bit can be added.

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 17.5.

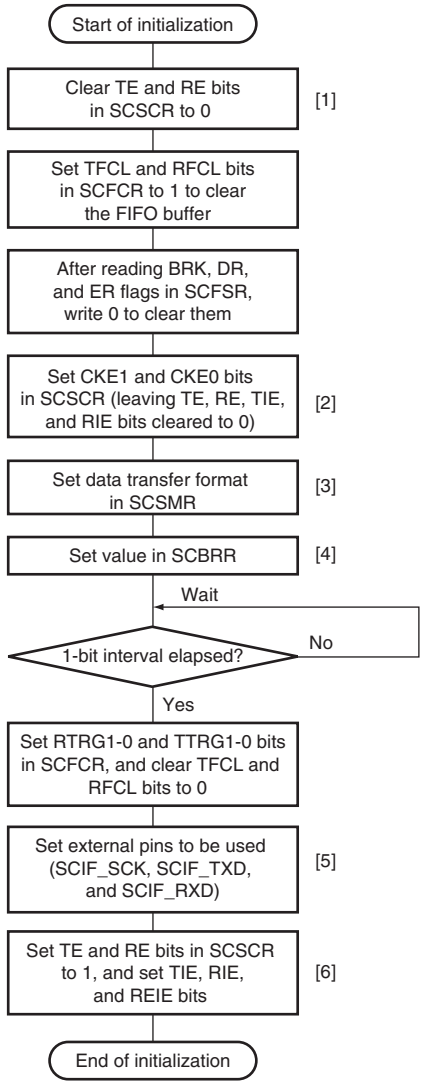
When the SCIF is operated on an internal clock, the synchronization clock is output from the SCIF_SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When an internal clock is selected in a receive operation only, as long as the RE bit in SCSCR is set to 1, clock pulses are output until the number of receive data bytes in the receive FIFO data register reaches the receive trigger number.

(3) SCIF Initialization (Clocked Synchronous Mode):

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When changing the operating mode or transfer format, etc., the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, FER, or ORER flag state or change the contents of SCFRDR.

Figure 21.16 shows a sample SCIF initialization flowchart.



[1] Leave the TE and RE bits cleared to 0 until the initialization almost ends. Be sure to clear the TIE, RIE, TE, and RE bits to 0.

[2] Set the CKE1 and CKE0 bits.

[3] Set the data transfer format in SCSMR.

[4] Write a value corresponding to the bit rate into SCBRR. This is not necessary if an external clock is used. Wait at least one bit interval after this write before moving to the next step.

[5] Set the 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

(4) Serial Data Transmission (Clocked Synchronous Mode)

Figure 21.17 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

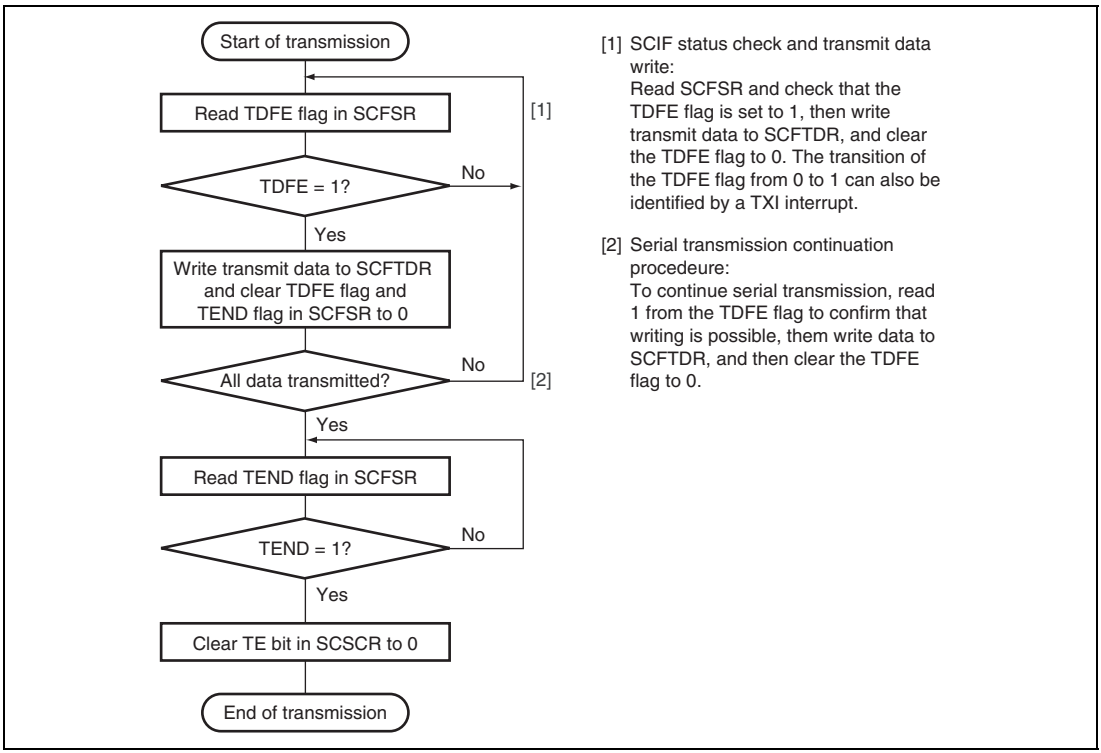


Figure 21.17 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 (transmit trigger setting).

2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronization clock pulses for each data.

   When the external clock is selected, data is output in synchronization with the input clock.

   The serial transmit data is sent from the SCIF_TXD pin in the LSB-first order.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the last bit. If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of the last bit.
4. After serial transmission ends, the SCIF_SCK pin is fixed high when the CKE1 bit in SCSCR is 0.

Figure 21.18 shows an example of the operation for transmission in clocked synchronous mode.

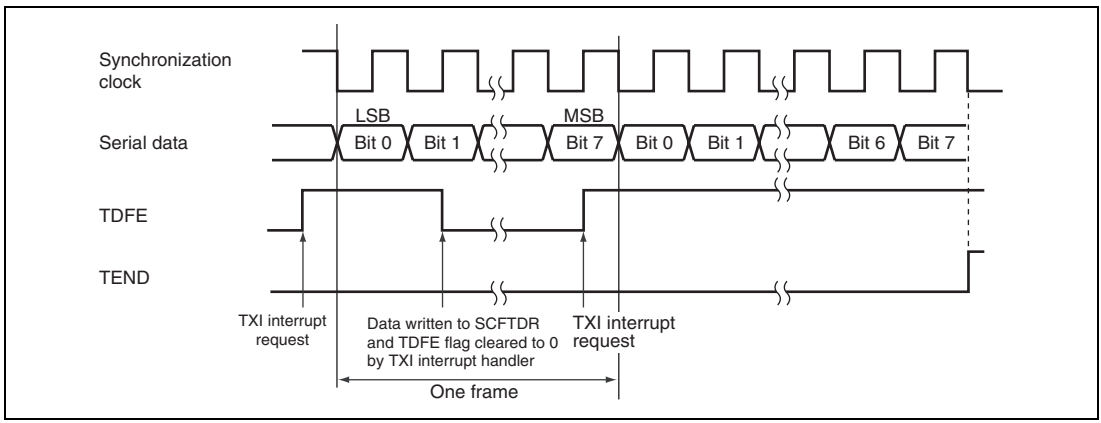


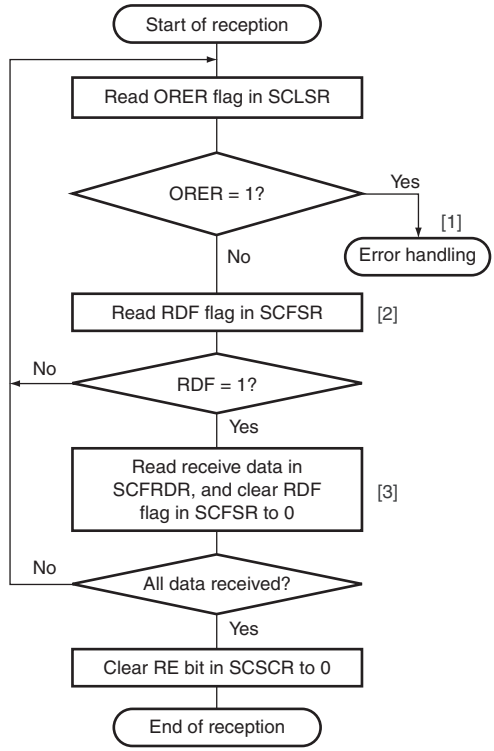
Figure 21.18 Sample SCIF Transmission Operation in Clocked Synchronous Mode

(5) Serial Data Reception (Clocked Synchronous Mode)

Figure 21.19 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching the operating mode from asynchronous mode to clocked synchronous mode without initializing the SCIF, make sure that the ORER, PER7 to PER0, and FER7 to FER0 flags are cleared to 0.



[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)

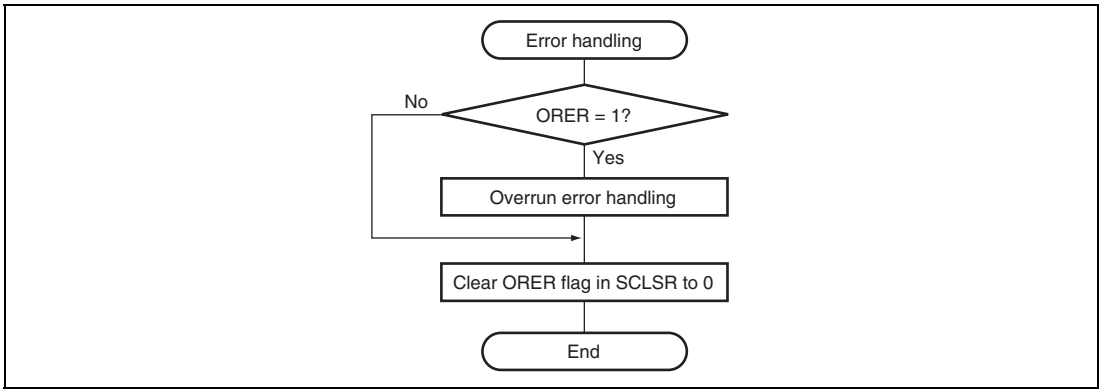


Figure 21.19 Sample Serial Reception Flowchart (2)

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 from SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. If an 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-data-full interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interrupt (BRI) request is generated.

Figure 21.20 shows an example of the operation for reception in clocked synchronous mode.

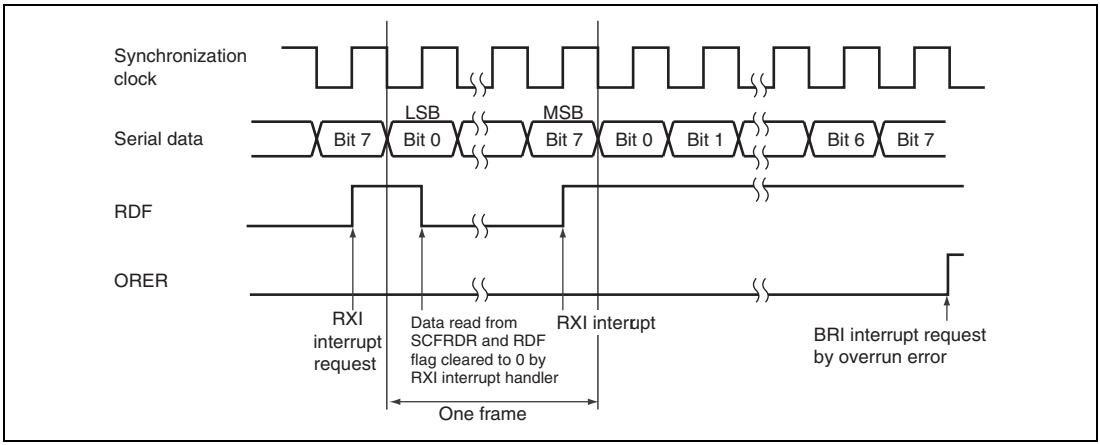
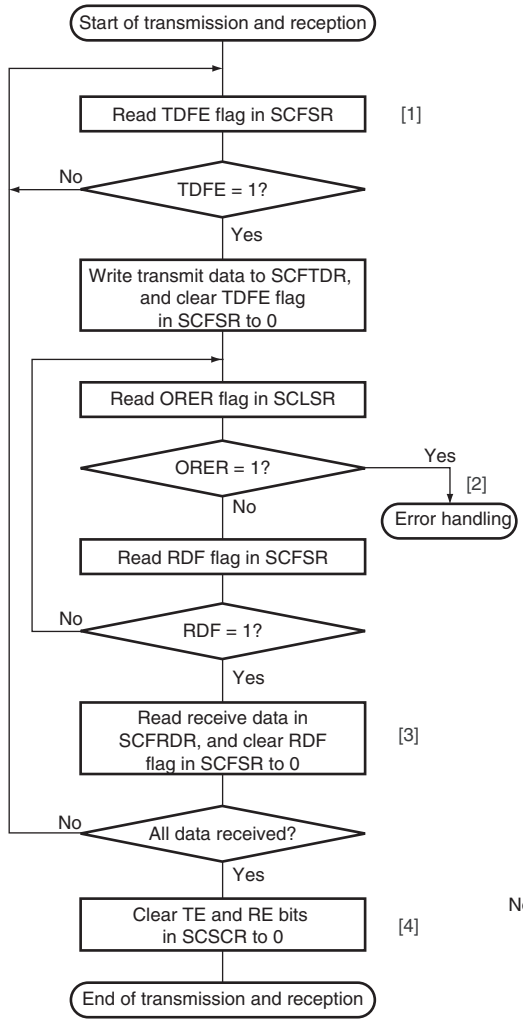


Figure 21.20 Sample SCIF Reception Operation in Clocked Synchronous Mode

(6) Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 21.21 shows a sample flowchart for simultaneous serial data transmission and reception.

Use the following procedure for simultaneous serial transmission and reception after enabling the SCIF for both transmission and reception.



[1] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:
Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0.
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 operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 21.21 Sample Simultaneous Serial Transmission and Reception Flowchart

21.5 SCIF Interrupt Sources and the DMAC

The SCIF has four interrupt sources in each channel: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and break interrupt (BRI) request.

Table 21.7 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

If the TDFE flag in SCFSR is set to 1 when a TXI interrupt is enabled by the TIE bit, a TXI interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated. If the TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIFO-data-empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can activate 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 bit, an RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generated. If the RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receive-FIFO-data-full request for DMA transfer is generated. A receive-FIFO-data-full request can activate the DMAC to perform data transfer. Note that generation of an RXI interrupt request or a receive-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 request is generated. If transmission/reception is carried out using the DMAC, set and enable the DMAC before making the SCIF settings. Also make settings to inhibit output of RXI and TXI interrupt requests to the interrupt controller. If output of interrupt requests is enabled, these interrupt requests to the interrupt controller can be cleared by the DMAC regardless of the interrupt handler.

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to output ERI interrupt requests, but not RXI interrupt requests.

Table 21.8 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready flag (DR)*	Possible	
BRI	Interrupt initiated by break flag (BRK) or overrun error flag (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous mode.

21.6 Usage Notes

Note the following when using the SCIF.

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in SCFCR. After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again, even after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from 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 become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive data is read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFRDR to 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 SCIF_RXD pin value directly. In the break state the input from the SCIF_RXD pin consists of all 0s, so the 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 receive operation continues.

(4) Sending a Break Signal

The input/output condition and level of the SCIF_TXD pin are determined by bits SPB2IO and SPB2DT in SCSPTR. This feature can be used to send a break signal.

After the serial transmitter is initialized and until the TE bit is set to 1 (enabling transmission), the SCIF_TXD pin function is not selected and the value of the SPB2DT bit substitutes for the mark state. The SPB2IO and SPB2DT bits should therefore be set to 1 (designating output and high level) in the beginning.

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), and then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized, regardless of the current transmission state, and 0 is output from the 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 the bit rate.

In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

The timing is shown in figure 21.22.

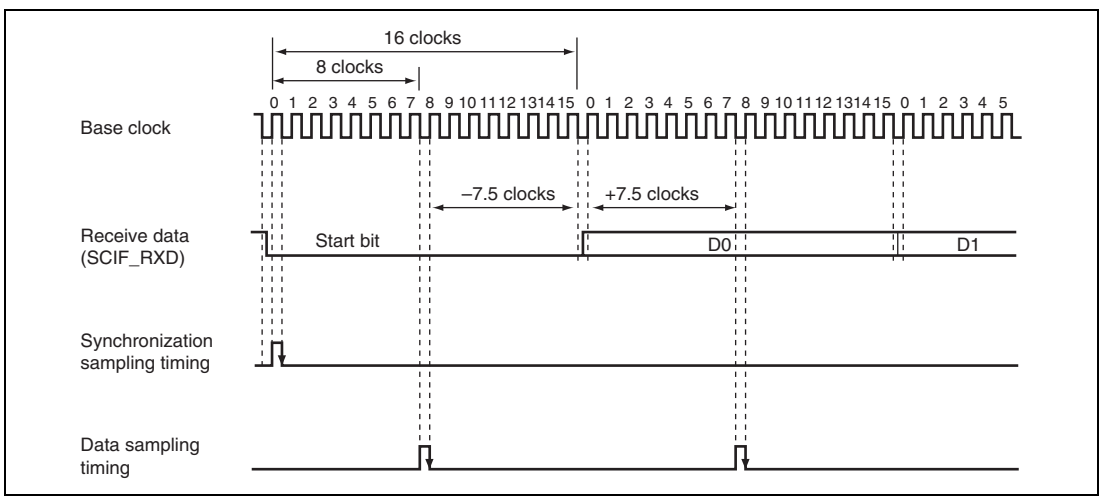


Figure 21.22 Receive Data Sampling Timing in Asynchronous Mode

Thus, the reception margin in asynchronous mode is given by formula (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

M: Receive margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

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 formula (2).

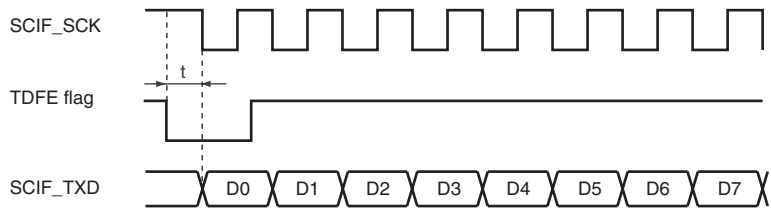
When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \text{..... (2)}$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20% to 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 the DMAC, an external clock should be input after at least five peripheral clock (Pck) cycles. A malfunction may occur when the transfer clock is input within four cycles after updating SCFTDR (see figure 21.23).



Note: When the SCIF is operated on an external clock, set t to ensure 5 Pck clock cycles or more.

Figure 21.23 Example of Synchronization Clock Transfer by DMAC

Section 22 Serial I/O with FIFO (SIOF)

This LSI includes a clock-synchronized serial I/O module with FIFO (SIOF).

22.1 Features

- Serial transfer
  16-stage 32-bit FIFOs (transmission and reception are independent of each other)
  Supports 8-bit data/16-bit data/16-bit stereo audio input/output
  MSB first for data transmission
  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 shows a block diagram of the SIOF.

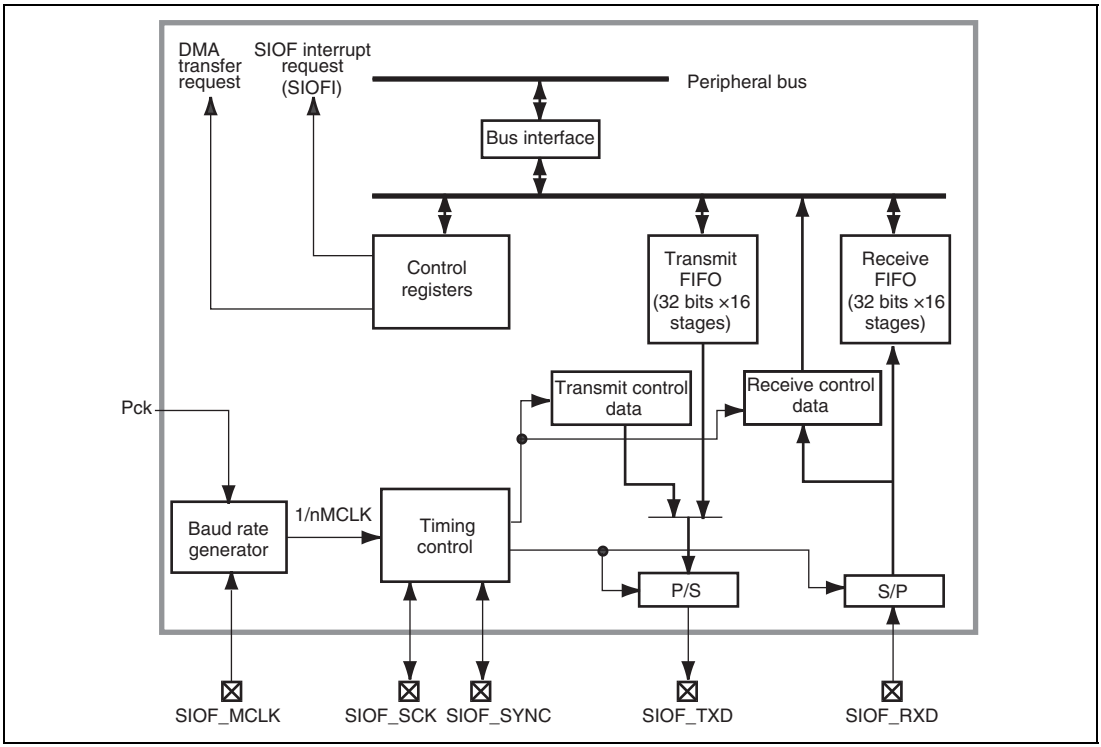


Figure 22.1 Block Diagram of SIOF

22.2 Input/Output Pins

The pin configuration in this module is shown in table 22.1.

Table 22.1 Pin Configuration

Pin Name	Function	I/O	Description
SIOF_MCLK	Master clock	Input	Master clock input pin
SIOF_SCK	Serial clock	I/O	Serial clock pin (common to transmission/reception)
SIOF_SYNC	Frame synchronous signal	I/O	Frame synchronous signal (common to transmission/reception)
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.

22.3 Register Descriptions

Table 22.2 shows the SIOF register configuration. Table 22.3 shows the register states in each processing mode.

Table 22.2 Register Configuration of SIOF

Name	Abbreviation	R/W	P4 Address	Area7 Address	Access Size	Sync Clock
Mode register	SIMDR	R/W	H'FFE2 0000	H'1FE2 0000	16	Pck
Clock select register	SISCR	R/W	H'FFE2 0002	H'1FE2 0002	16	Pck
Transmit data assign register	SITDAR	R/W	H'FFE2 0004	H'1FE2 0004	16	Pck
Receive data assign register	SIRDAR	R/W	H'FFE2 0006	H'1FE2 0006	16	Pck
Control data assign register	SICDAR	R/W	H'FFE2 0008	H'1FE2 0008	16	Pck
Control register	SICTR	R/W	H'FFE2 000C	H'1FE2 000C	16	Pck
FIFO control register	SIFCTR	R/W	H'FFE2 0010	H'1FE2 0010	16	Pck
Status register	SISTR	R/W	H'FFE2 0014	H'1FE2 0014	16	Pck
Interrupt enable register	SIIER	R/W	H'FFE2 0016	H'1FE2 0016	16	Pck
Transmit data register	SITDR	W	H'FFE2 0020	H'1FE2 0020	32	Pck
Receive data register	SIRDR	R	H'FFE2 0024	H'1FE2 0024	32	Pck
Transmit control data register	SITCR	R/W	H'FFE2 0028	H'1FE2 0028	32	Pck
Receive control data register	SIRCR	R/W	H'FFE2 002C	H'1FE2 002C	32	Pck

Table 22.3 Register States of SIOF in Each Processing Mode

Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/Multiple Exceptions	Sleep by SLEEP Instruction	Module Standby
Mode register	SIMDR	H'8000	H'8000	Retained	Retained
Clock select register	SISCR	H'C000	H'C000	Retained	Retained
Transmit data assign register	SITDAR	H'0000	H'0000	Retained	Retained
Receive data assign register	SIRDAR	H'0000	H'0000	Retained	Retained
Control data assign register	SICDAR	H'0000	H'0000	Retained	Retained
Control register	SICTR	H'0000	H'0000	Retained	Retained
FIFO control register	SIFCTR	H'1000	H'1000	Retained	Retained
Status register	SISTR	H'0000	H'0000	Retained	Retained
Interrupt enable register	SIIER	H'0000	H'0000	Retained	Retained
Transmit data register	SITDR	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
Receive data register	SIRDR	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
Transmit control data register	SITCR	H'0000 0000	H'0000 0000	Retained	Retained
Receive control data register	SIRCR	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained

[Legend] x: Undefined

22.3.1 Mode Register (SIMDR)

SIMDR is a 16-bit readable/writable register that sets the SIOF operating mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRMD[1:0]		SYN CAT	REDG	FL[3:0]				TXDIZ	RCIM	SYN CAC	SYN CDL	—	—	—	—
Initial value:	1	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
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 signal to be 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 falling edge of SIOF_SCK 1: The SIOF_RXD signal is sampled at the rising edge of SIOF_SCK Note: The timing to transmit the SIOF_TXD signal is at the opposite edge of the timing that samples the SIOF_RXD. This bit is valid only in master mode.
11 to 8	FL[3:0]	0000	R/W	Frame Length 3 to 0 Specify the flame length and transfer data format. For details, refer to table 22.7.

Bit	Bit Name	Initial Value	R/W	Description
7	TXDIZ	0	R/W	SIOF_TXD Pin Output when Transmission is Invalid* 0: High output (1 output) when invalid 1: High-impedance state when invalid Note: Invalid means when disabled, and when a slot that is not assigned as transmit data or control data is being transmitted.
6	RCIM	0	R/W	Receive Control Data Interrupt Mode 0: Sets the RCRDY bit in SISTR when the contents of SIRCR change. 1: Sets the RCRDY bit in SISTR each time when the SIRCR receives the control data.
5	SYNCAC	0	R/W	SIOF_SYNC Pin Polarity Valid when the SIOF_SYNC signal is output as 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 as synchronous pulse. Only one-bit delay is valid for 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 value should always be 0.

Table 22.4 shows the operation in each transfer mode.

Table 22.4 Operation in Each Transfer Mode

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

Note: * The control data method is valid only when the FL bits are specified as B'1xxx. (x: don't care)

22.3.2 Clock Select Register (SISCR)

SISCR is a 16-bit readable/writable register that sets the serial clock generation conditions for the master clock. SISCR can be specified when the bits TRMD[1:0] in SIMDR are specified as B'10 or B'11.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSSEL	MSIMM	—	BRPS[4:0]					—	—	—	—	—	BRDV[2:0]		
Initial value:	1	1	0	0	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/W	R/W

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 the baud rate generator (prescaler). 0: Uses the input clock signal of the SIOF_MCLK pin as the master clock 1: Uses peripheral clock (Pck) as the master clock
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate generator as the serial clock 1: Uses the master clock itself as the serial clock
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 8	BRPS[4:0]	00000	R/W	Prescalar Setting Set the master clock division ratio according to the count value of the prescalar of the baud rate generator. The range of settings is from 00000 (× 1/1) to 11111 (× 1/32).
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 stage of 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 BRPS[4:0] are set to 00001. The frequency division ratio of the baud rate generator is finally determined by the value of BRPS by BRDV (maximum 1/1024).

22.3.3 Control Register (SICTR)

SICTR is a 16-bit readable/writable register that sets the SIOF operating state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKE	FSE	—	—	—	—	TXE	RXE	—	—	—	—	—	—	TXRST	RXRST
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	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
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 baud rate generator and initiates the operation. At the same time, the SIOF outputs the clock generated by the baud rate 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 frame counter and initiates the operation.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	TXE	0	R/W	Transmit Enable 0: Disables data transmission from the SIOF_TXD pin 1: Enables data transmission from the SIOF_TXD pin • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOF_SYNC signal). • When the 1 setting for this bit becomes valid, the SIOF issues a transmit transfer request according to the setting of the TFWM bit in SIFCTR. When transmit data is stored in the transmit FIFO, transmission of data from the SIOF_TXD pin begins. 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 of the next frame (at the rising edge of the SIOF_SYNC signal). • When the 1 setting for this bit becomes valid, the SIOF begins the reception of data from the SIOF_RXD pin. When receive data is stored in the receive FIFO, the SIOF issues a reception transfer request according to the setting of the RFWM bit in SIFCTR. This bit is initialized upon receive reset.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	TXRST	0	R/W	Transmit Reset 0: Does not reset transmit operation 1: Resets transmit operation • This bit setting becomes valid immediately. For details of transmit reset, refer to table 22.13. • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. For details of receive reset, refer to table 22.13. • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.

22.3.4 Transmit Data Register (SITDR)

SITDR is a 32-bit write-only register that specifies the SIOF operating status.

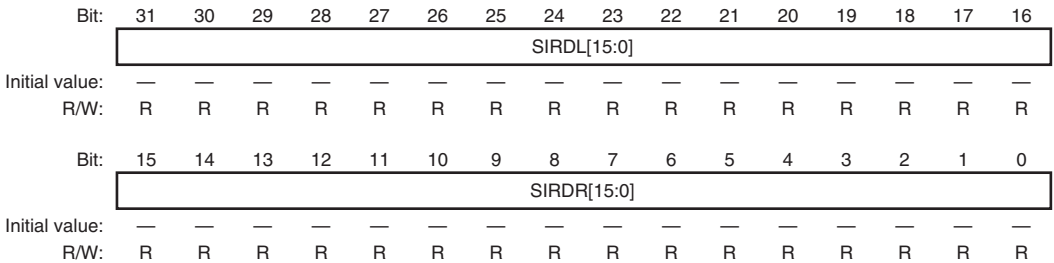
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITDL[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SITDR[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	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 pin as left-channel data. The position of the left-channel data in the transmit frame is specified by the TDLA bit in SITDAR. • These bits are valid only when the TDLE bit in 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 pin as right-channel data. The position of the right-channel data in the transmit frame is specified by the TDRA bit in SITDAR. • These bits are valid only when the TDRE bit and TLREP bit in SITDAR are set to 1 and cleared to 0, respectively.

22.3.5 Receive Data Register (SIRDR)

SIRDR is a 32-bit read-only register that reads receive data of the SIOF. SIRDR stores data in the receive FIFO.



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 left-channel data. The position of the left-channel data in the receive frame is specified by the RDLA bit in SIRDAR. • These bits are valid only when the RDLE bit in 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 right-channel data. The position of the right-channel data in the receive frame is specified by the RDRA bit in SIRDAR. • These bits are valid only when the RDRE bit in SIRDAR is set to 1.

22.3.6 Transmit Control Data Register (SITCR)

SITCR is a 32-bit readable/writable register that specifies transmit control data of the SIOF. SITCR can be specified only when the FL bits in SIMDR are specified as B'1xxx (x: don't care).

SITCR is initialized by the conditions specified in table 22.3, Register State of SIOF in Each Processing Mode, or by a transmit reset caused by the TXRST bit in SICTR.

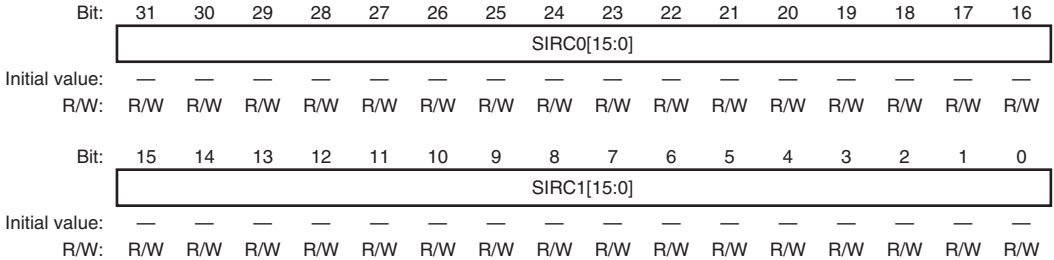
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITC0[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SITC1[15:0]															
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

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 pin as control channel 0 transmit data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in 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 pin as control channel 1 transmit data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

22.3.7 Receive Control Data Register (SIRCR)

SIRCR is a 32-bit readable/writable register that stores receive control data of the SIOF. SIRCR can be specified only when the FL bits in SIMDR are specified as B'1xxx (x: don't care).



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 control channel 0 receive data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in 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 control channel 1 receive data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

22.3.8 Status Register (SISTR)

SISTR is a 16-bit readable/writable register that shows the SIOF state. Each bit in this register becomes an SIOF interrupt source when the corresponding bit in SIIER is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared to 0, SITCR is over-written and the previous contents of SITCR are not output from the SIOF_TXD pin. • This bit is valid when the TXE bit in SITCR is set to 1. • This bit indicates a state of the SIOF. If SITCR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an 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 is 1. • This bit indicates a state; if SITDR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
12	TDREQ	0	R	Transmit Data Transfer Request 0: Indicates that the size of empty space in the transmit FIFO does not exceed the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. When using transmit data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if the size of empty space in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	RCRDY	0	R	Receive Control Data Ready 0: Indicates that the SIRCR stores no valid data. 1: Indicates that the SIRCR stores valid data. • If SIRCR is written when this bit is set to 1, SIRCR is modified by the latest data. • This bit is valid when the RXE bit in SICTR is set to 1. • This bit indicates a state of the SIOF. If SIRCR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
9	RFFUL	0	R	Receive FIFO Full 0: Receive FIFO not full 1: Receive FIFO full • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if SIRDR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid space in the receive FIFO does not exceed the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid data space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. When using receive data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if the size of valid data space in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SAERR	0	R/W	Slot Assign Error 0: Indicates that no slot assign error occurs 1: Indicates that a slot assign error occurs A slot assign error occurs when the specifications in SITDAR, SIRDAR, and SICDAR overlap. If a slot assign error occurs, the SIOF does not transmit data to the SIOF_TXD pin and does not receive data from the SIOF_RXD pin. Note that the SIOF does not clear the TXE bit or RXE bit in SICTR at a slot assign error. • This bit is valid when the TXE bit or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs A frame synchronization error occurs when the next frame synchronization timing appears before the previous data or control data transfers have been completed. If a frame synchronization error occurs, the SIOF performs transmission or reception for slots that can be transferred. • This bit is valid when the TXE or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
3	TFOVF	0	R/W	Transmit FIFO Overflow 0: No transmit FIFO overflow 1: Transmit FIFO overflow A transmit FIFO overflow means that there has been an attempt to write to SITDR when the transmit FIFO is full. When a transmit FIFO overflow occurs, the SIOF indicates overflow, and writing is invalid. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit FIFO underflow occurs, the SIOF repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
1	RFUDF	0	R/W	Receive FIFO Underflow 0: No receive FIFO underflow 1: Receive FIFO underflow A receive FIFO underflow means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive FIFO underflow occurs, the value of data read from SIRDR is not guaranteed. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF indicates overflow, and receive data is lost. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

22.3.9 Interrupt Enable Register (SIIER)

SIIER is a 16-bit readable/writable register that enables the issue of SIOF interrupts. When each interrupt enable bit in this register is set to 1 and the corresponding bit in SISTR is set to 1, the SIOF issues an interrupt.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TD MAE	TCR DYE	TFE MPE	TDR EQE	RD MAE	RC RDYE	RF FULE	RD REQE	—	—	SA ERRE	FS ERRE	TF OVFE	TF UDFE	RF UDFE	RF OVFE
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/W

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

Bit	Bit Name	Initial Value	R/W	Description
10	RCRDYE	0	R/W	Receive Control Data Ready Enable 0: Disables interrupts due to receive control data ready 1: Enables interrupts due to receive control data ready
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive Data Transfer Request Enable 0: Disables interrupts due to receive data transfer requests 1: Enables interrupts due to receive data transfer requests
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

22.3.10 FIFO Control Register (SIFCTR)

SIFCTR is a 16-bit readable/writable register that indicates the area available for the transmit/receive FIFO transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TFWM[2:0]			TFUA[4:0]					RFWM[2:0]			RFUA[4:0]				
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TFWM[2:0]	000	R/W	Transmit FIFO Watermark 000: Issue a transfer request when 16 stages of the transmit FIFO are empty. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or more stages of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages of the transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages of the transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages of transmit FIFO are empty. • A transfer request to the transmit FIFO is issued by the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 stages of 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 transferred by the CPU or DMAC as 00000 (full) to 10000 (empty).

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	RFWM[2:0]	000	R/W	Receive FIFO Watermark 000: Issue a transfer request when 1 stage or more of the receive FIFO are valid. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 4 or more stages of the receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of the receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of the receive FIFO are valid. 111: Issue a transfer request when 16 stages of the receive FIFO are valid. • A transfer request to the receive FIFO is issued by the RDREQE bit in SISTR. • The receive FIFO is always used as 16 stages of 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 transferred by the CPU or DMAC as 00000 (empty) to 10000 (full).

22.3.11 Transmit Data Assign Register (SITDAR)

SITDAR is a 16-bit readable/writable register that specifies the position of the transmit data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLE	—	—	—	TDLA[3:0]				TDRE	TLREP	—	—	TDRA[3:0]			
Initial value:	0	0	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/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 transmit frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Transmit data for the left channel is specified in the 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 in SITDR as right-channel data 1: Repeatedly transmits data specified in the SITDL bit in SITDR as right-channel data • This bit setting is valid when the TDRE bit is set to 1. • When this bit is set to 1, the SITDR settings are ignored.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	TDRA[3:0]	0000	R/W	Transmit Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a transmit frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Transmit data for the right channel is specified in the 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 data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLE	—	—	—	RDLA[3:0]				RDRE	—	—	—	RDRA[3:0]			
Initial value:	0	0	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	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 value should 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 receive frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Receive data for the left channel is stored in the SIRDL bit in SIRDR.

Bit	Bit Name	Initial Value	R/W	Description
7	RDRE	0	R/W	Receive Right-Channel Data Enable 0: Disables right-channel data reception 1: Enables right-channel data reception
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	RDRA[3:0]	0000	R/W	Receive Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a receive frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Receive data for the right channel is stored in the 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 data in a frame. SICDAR can be specified only when the FL bits in SIMDR are specified as B'1xxx (x: don't care).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD0E	—	—	—	CD0A[3:0]				CD1E	—	—	—	CD1A[3:0]			
Initial value:	0	0	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	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 control channel 0 data 1: Enables transmission and reception of control channel 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	CD0A[3:0]	0000	R/W	Control Channel 0 Data Assigns 3 to 0 Specify the position of control channel 0 data in a receive or transmit frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Transmit data for the control channel 0 data is specified in the SITD0 bit in SITCR. • Receive data for the control channel 0 data is stored in the SIRD0 bit in SIRCR.
7	CD1E	0	R/W	Control Channel 1 Data Enable 0: Disables transmission and reception of control channel 1 data 1: Enables transmission and reception of control channel 1 data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 in a receive or transmit frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Transmit data for the control channel 1 data is specified in the SITD1 bit in SITCR. • Receive data for the control channel 1 data is stored in the SIRD1 bit in SIRCR.

22.4 Operation

22.4.1 Serial Clocks

Master/Slave Modes: The following modes are available as the SIOF clock mode.

- Slave mode: SIOF_SCK, SIOF_SYNC input
- Master mode: SIOF_SCK, SIOF_SYNC output

Baud Rate Generator: In SIOF master mode, the baud rate generator (BRG) is used to generate 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 (division ratio: 1/1), the MSIMM bit in SISCR should be set to 1.

Figure 22.2 shows connections for supply of the serial clock.

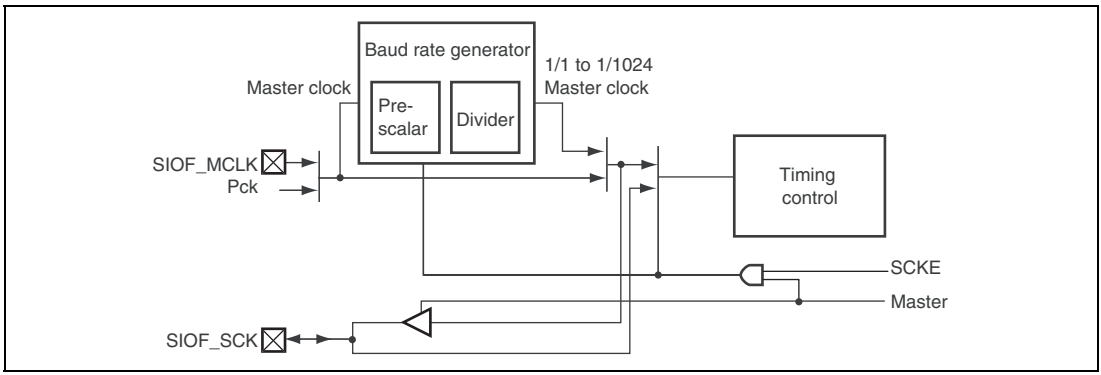


Figure 22.2 Serial Clock Supply

Table 22.5 shows an example of serial clock frequency.

Table 22.5 SIOF Serial Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.2896 MHz	12.288 MHz

22.4.2 Serial Timing

SIOF_SYNC**:** The SIOF_SYNC is a frame synchronous signal. Depending on the transfer mode, it has the following two functions.

- Synchronous pulse: 1-bit-width pulse indicating the start of the frame
- L/R: 1/2-frame-width pulse indicating the left-channel stereo data (L) in high level and the right-channel stereo data (R) in low level

Figure 22.3 shows the SIOF_SYNC synchronization timing.

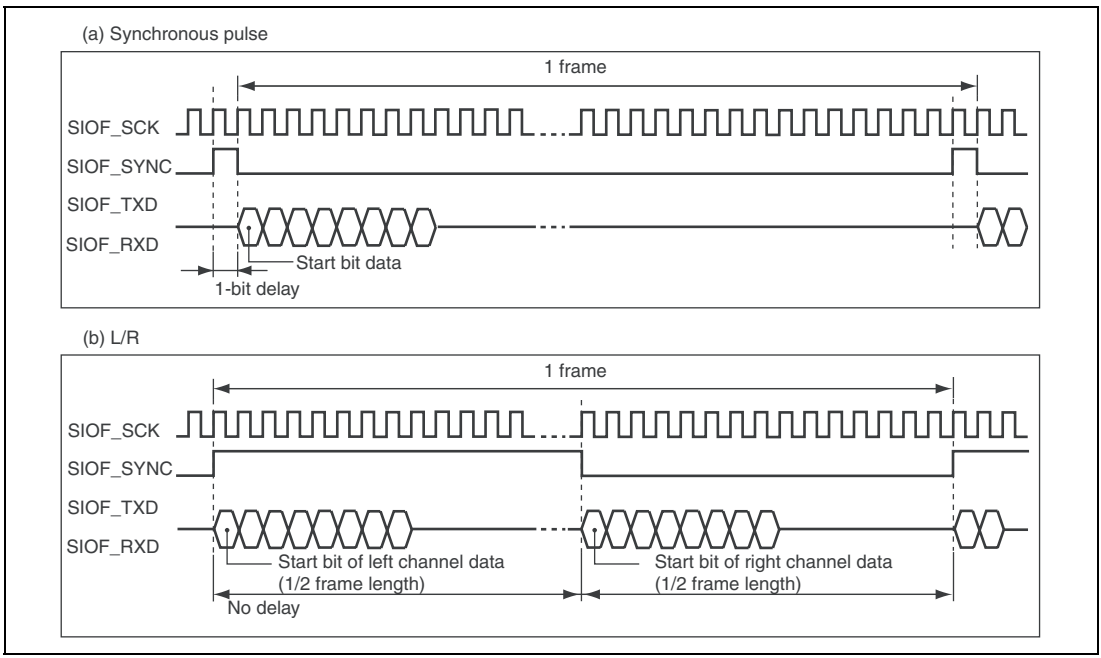


Figure 22.3 Serial Data Synchronization Timing

Transmit/Receive Timing: The SIOF_TXD transmit timing and SIOF_RXD receive timing relative to the SIOF_SCK can be set as the sampling timing in the following two ways. The transmit/receive timing is set using the REDG bit in SIMDR.

- Falling-edge sampling
- Rising-edge sampling

Figure 22.4 shows the transmit/receive timing.

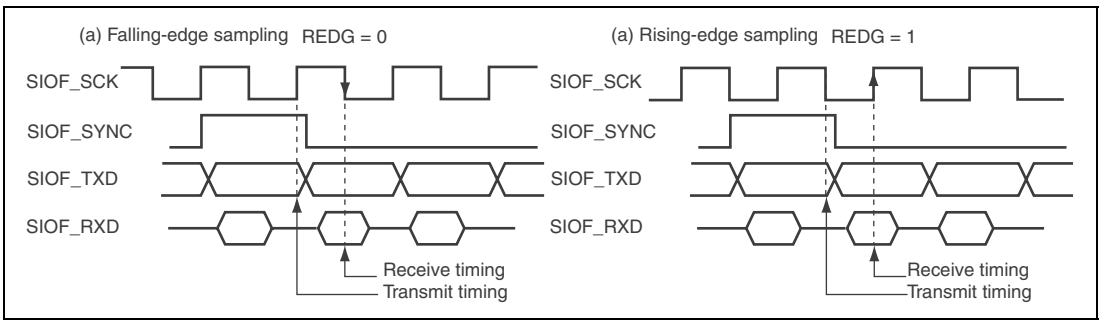


Figure 22.4 SIOF Transmit/Receive Timing

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 22.6. The 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 Data*
00	Slave mode 1	Synchronous pulse	SYNCDL bit	Slot position
01	Slave mode 2	Synchronous pulse		Secondary FS
10	Master mode 1	Synchronous pulse		Slot position
11	Master mode 2	L/R	No	Not supported

Note: * The control data method is valid only when the FL bits are specified as B'1xxx (x: don't care).

Frame Length: The length of the frame to be transferred by the SIOF is specified by the bits FL[3:0] in SIMDR. Table 22.7 shows the relationship between the bits FL[3:0] settings and frame length.

Table 22.7 Frame Length

FL[3:0]	Slot Length	Number of Bits in a Frame	Transfer Data
00xx	8	8	8-bit monaural data
0100	8	16	8-bit monaural data
0101	8	32	8-bit monaural data
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 control data in a frame (common to transmission and reception) by slot numbers. The slot number of each data is 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 the same slot number both in transmission and reception.

22.4.4 Register Allocation of Transfer Data

Transmit/Receive Data: Writing and reading of transmit/receive data is performed for the following registers.

- Transmit data writing: SITDR (32-bit access)
- Receive data reading: SIRDR (32-bit access)

Figure 22.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

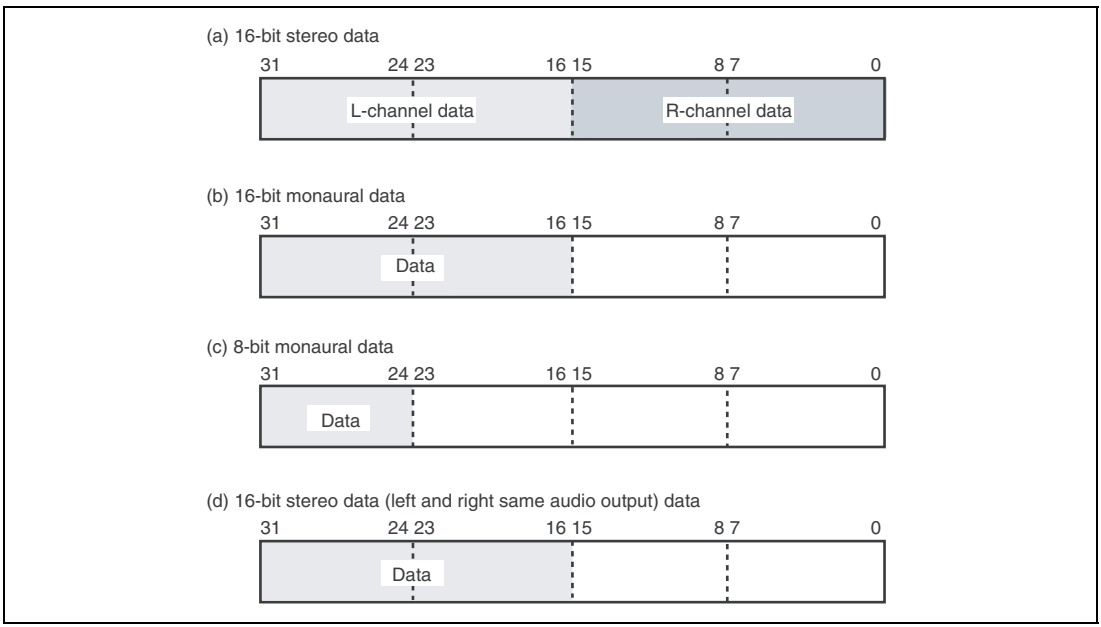


Figure 22.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Therefore, access must be made in byte units for 8-bit data, and in word units for 16-bit data. Data in unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in SITDAR. Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SIRDAR. To achieve left and right same audio output while stereo is specified for transmit data, specify the TLREP bit in SITDAR. Table 22.8 and table 22.9 show the audio mode specification for transmit data and that for receive data, respectively.

Table 22.8 Audio Mode Specification for Transmit Data

	Bit		
Mode	**TDLE**	**TDRE**	**TLREP**
Monaural	1	0	x
Stereo	1	1	0
Left and right same audio output	1	1	1

Note: x: Don't care

Table 22.9 Audio Mode Specification for Receive Data

	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.

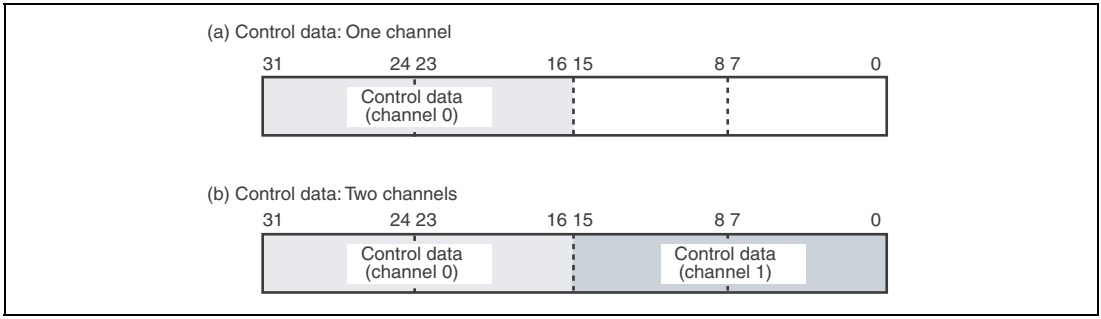


Figure 22.6 Control Data Bit Alignment

The number of channels in control data is specified by the CD0E and CD1E bits in SICDAR. Table 22.10 shows the relationship between the number of channels in control data and bit settings.

Table 22.10 Setting Number of Channels in Control Data

	Bit	
Number of Channels	**CD0E**	**CD1E**
1	1	0
2	1	1

Note: To use only one channel in control data, use channel 0.

22.4.5 Control Data Interface

Control data performs control command output to the CODEC and status input from the CODEC. 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 for all frames transmitted or received by the SIOF by specifying the slot position of control data. This method can be used in both SIOF master and slave modes. Figure 22.7 shows an example of the control data interface timing by slot position control.

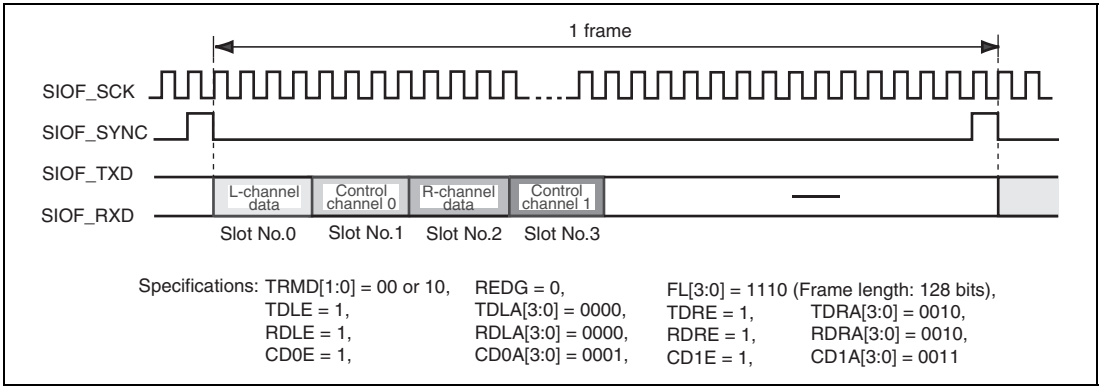


Figure 22.7 Control Data Interface (Slot Position)

Control by Secondary FS (Slave Mode 2): The CODEC normally outputs the SIOF_SYNC signal as synchronization pulse (FS). In this method, the CODEC outputs the secondary FS specific to the control data transfer after 1/2 frame time has been passed (not the normal FS output timing) to transmit or receive control data. This method is valid for SIOF slave mode. The following summarizes the control data interface procedure by the secondary FS.

- Transmit normal transmit data of LSB = 0 (the SIOF forcibly clears 0).
- To execute control data transmission, send transmit data of LSB = 1 (the SIOF forcibly set to 1 by writing SITCR).
- The CODEC outputs the secondary FS.
- The SIOF transmits or receives (stores in SIRCR) control data (data specified by SITCR) synchronously with the secondary FS.

Figure 22.8 shows an example of the control data interface timing by the secondary FS.

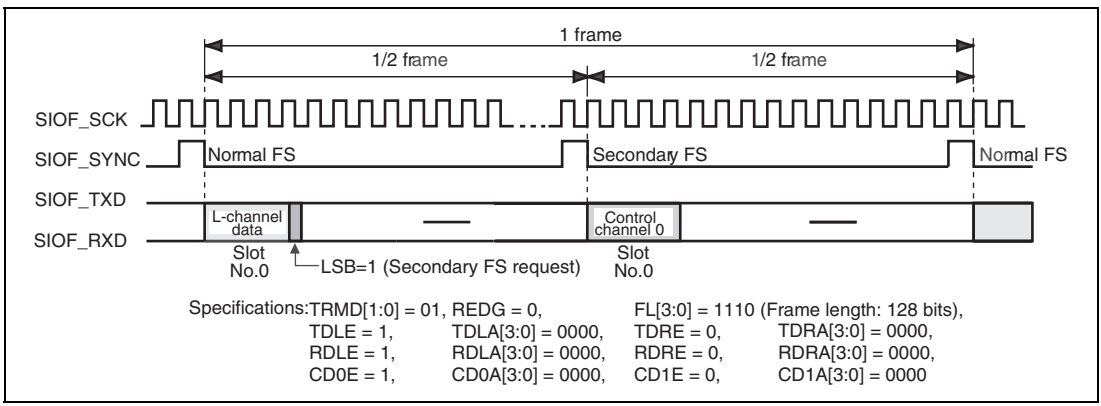


Figure 22.8 Control Data Interface (Secondary FS)

22.4.6 FIFO

Overview: The transmit and receive FIFOs of the SIOF have the following features.

- 16-stage 32-bit FIFOs for transmission and reception
- The FIFO pointer can be updated in one read or write cycle regardless of access size of the CPU and DMAC. (One-stage 32-bit FIFO access cannot be divided into multiple accesses.)

Transfer Request: The transfer request of the FIFO can be issued to the CPU or DMAC as the following interrupt sources.

- 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 request conditions for the FIFO transmit and receive are specified by the bits TFWM[2:0] and the bits RFWM[2:0] in SIFCTR, respectively. Table 22.11 and table 22.12 summarize the conditions specified by SIFCTR.

Table 22.11 Conditions to Issue Transmit Request

TFWM[2:0]	Number of Requested Stages	Transmit Request	Used Areas
000	1	Empty area is 16 stages	Smallest
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	Largest

Table 22.12 Conditions to Issue Receive Request

RFWM[2:0]	Number of Requested Stages	Receive Request	Used Areas
000	1	Valid data is 1 stage or more	Smallest
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	Largest

The number of stages of the FIFO is always sixteen even if the data area or empty area exceeds the FIFO size (the number of FIFOs). Accordingly, an overflow error or underflow error occurs if data area or empty area exceeds sixteen FIFO stages. The FIFO transmit or receive request is canceled when the above condition is not satisfied even if the FIFO is not empty or full.

Number of FIFOs: The number of FIFO stages used in transmission and reception is indicated by the following register.

- Transmit FIFO: The number of empty FIFO stages is indicated by the bits TFUA[4:0] in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the bits RFUA[4:0] in SIFCTR.

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

22.4.7 Transmit and Receive Procedures

Transmission in Master Mode: Figure 22.9 shows an example of settings and operation for master mode transmission.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SICDAR, SITCR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit data, slot position for control data, control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	
3	Start SIOF_SCK output		Output serial clock
4	Set the FSE and TXE bits in SICTR to 1	Set the start for frame synchronous signal output and enable transmission	Output frame synchronous signal and issue transmit transfer request*
5	TDREQ = 1? No / Yes		
6	Set SITDR	Set transmit data	
7	Transmit SITDR from SIOF_TXD synchronously with SIOF_SYNC		Transmit
8	Transfer ended? No / Yes Clear the TXE bit in SICTR to 0 End	Set to disable transmission	End transmission

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

Reception in Master Mode: Figure 22.10 shows an example of settings and operation for master mode reception.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	
3	Start SIOFSCK output		Output serial clock
4	Set the FSE and RXE bits in SICTR to 1	Set the start for frame synchronous signal output and enable reception	Output frame synchronous signal
5	Store SIOFRXD receive data in SIRDR synchronously with SIOF_SYNC		Issue receive transfer request according to the receive FIFO threshold value
6	RDREQ = 1? No / Yes		Reception
7	Read SIRDR	Read receive data	
8	Transfer ended? No / Yes Clear the RXE bit in SICTR to 0 End	Set to disable reception	End reception

Figure 22.10 Example of Receive Operation in Master Mode

Transmission in Slave Mode: Figure 22.11 shows an example of settings and operation for slave mode transmission.

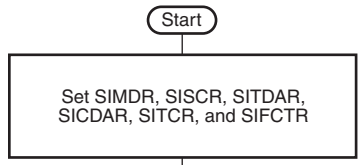
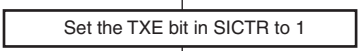
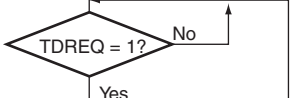
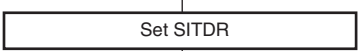
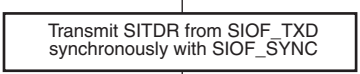
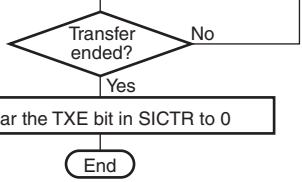
No.	Flow Chart	SIOF Settings	SIOF Operation
1		Set operating mode, serial clock, slot positions for transmit data, slot position for control data, control data, and FIFO request threshold value	
2		Set to enable transmission	Issue transmit transfer request to enable transmission when frame synchronous signal is input
3			
4		Set transmit data	
5			Transmit
6		Set to disable transmission	End transmission

Figure 22.11 Example of Transmit Operation in Slave Mode

Reception in Slave Mode: Figure 22.12 shows an example of settings and operation for slave mode reception.

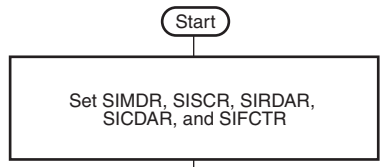
No.	Flow Chart	SIOF Settings	SIOF Operation
1		Set operating mode, serial clock, slot positions for receive data, slot position for control data, and FIFO request threshold value	
2	Set the RXE bit in SICTR to 1	Set to enable reception	Enable reception when the frame synchronous signal is input
3	Store SIOFRXD receive data in SIRDR synchronously with SIOF_SYNC		Issue receive transfer request according to the receive FIFO threshold value
4	RDREQ = 1? No Yes		Reception
5	Read SIRDR	Read receive data	
6	Transfer ended? No Yes Clear the RXE bit in SICTR to 0 End	Set to disable reception	End reception

Figure 22.12 Example of Receive Operation in Slave Mode

Transmit/Receive Reset: The SIOF can separately reset the transmit and receive units by setting the following bits to 1.

- Transmit reset: TXRST bit in SICTR
- Receive reset: RXRST bit in SICTR

Table 22.13 shows the details of initialization upon transmit or receive reset.

Table 22.13 Transmit and Receive Reset

Type	Objects Initialized
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

22.4.8 Interrupts

The SIOF has one type of interrupt.

Interrupt Sources: Interrupts can be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 22.14 lists the SIOF interrupt sources.

Table 22.14 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register is ready to be written.
6		RCRDY	Receive control data ready	The receive control data register stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in both serial data and control data.

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

Regarding Interrupt Source: The transmit sources and receive sources are signals indicating the SIOF state; after being set, if the state changes, they are automatically cleared by the SIOF.

When the DMA transfer is used, a DMA transfer request of the FIFO is disabled for one cycle at the end of that DMA transfer.

Processing when Errors Occur: On occurrence of each of the errors indicated as a status in SISTR, the SIOF performs the following operations.

- Transmit FIFO underflow (TFUDF)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)
  The contents of the transmit FIFO are protected, and the write operation causing the overflow 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 to serial data.
  — If the same slot is assigned to two control data items, data cannot be transferred correctly.

22.4.9 Transmit and Receive Timing

Examples of the SIOF serial transmission and reception are shown in figure 22.13 to figure 22.19.

8-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, an frame length = 8 bits

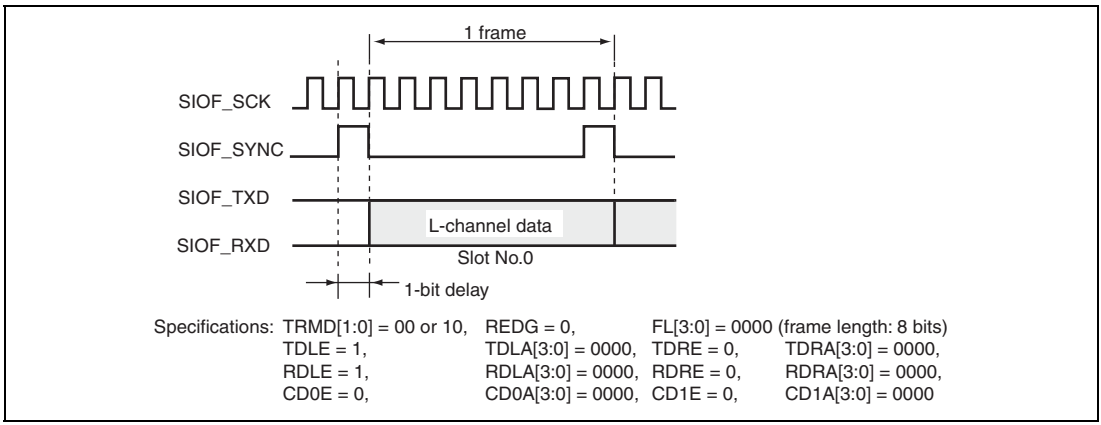


Figure 22.13 Transmit and Receive Timing (8-Bit Monaural Data (1))

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

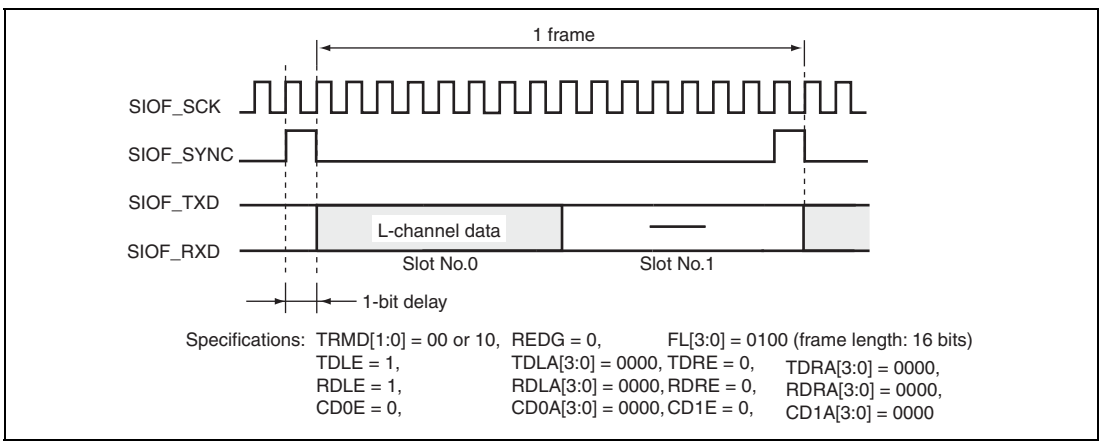


Figure 22.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

16-bit Monaural Data: Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 64 bits

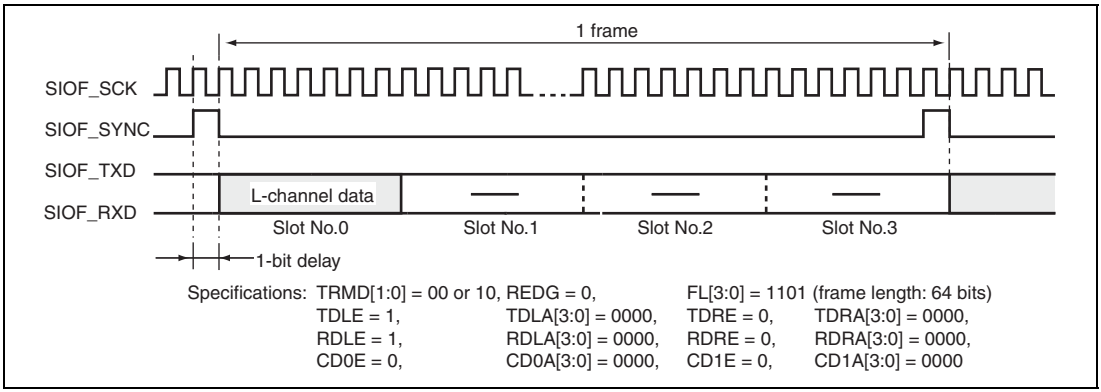


Figure 22.15 Transmit and Receive Timing (16-Bit Monaural Data)

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

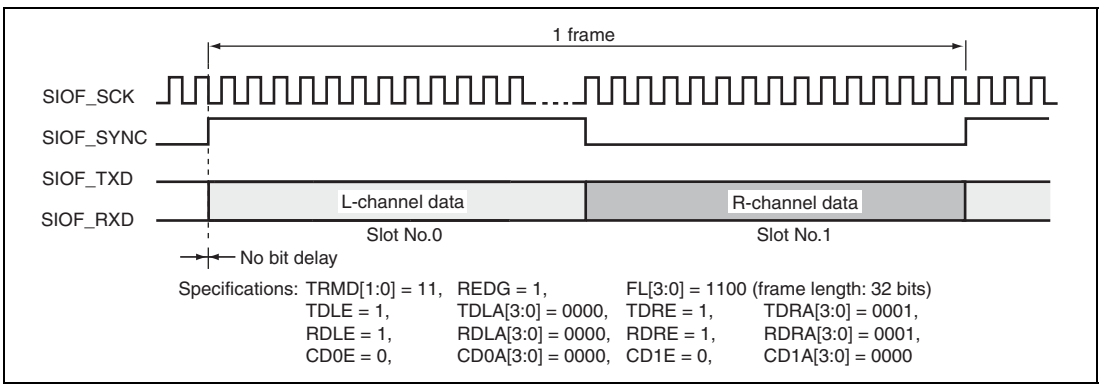


Figure 22.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

16-bit Stereo Data (2): L/R method, rising edge sampling, slot No.0 used for left-channel transmit data, slot No.1 used for left-channel receive data, slot No.2 used for right-channel transmit data, slot No.3 used for right-channel receive data, and frame length = 64 bits

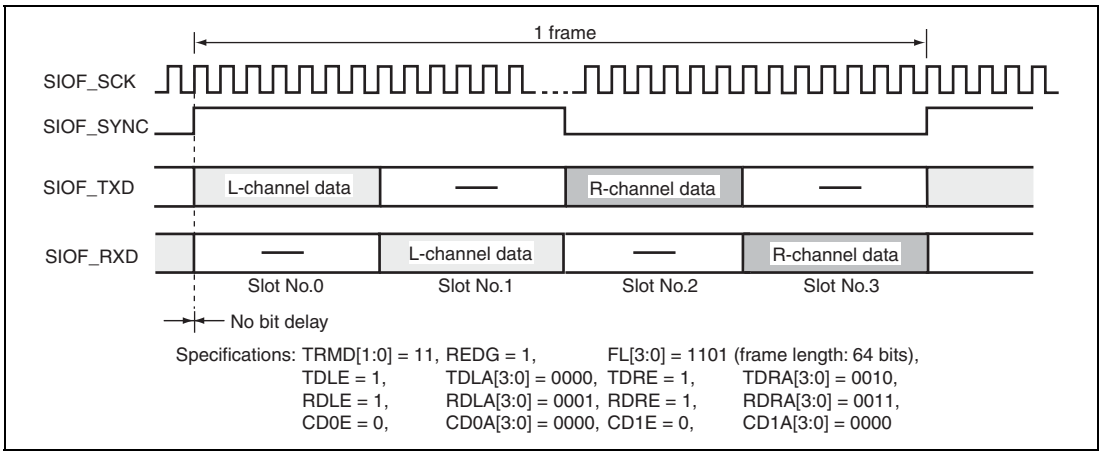


Figure 22.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

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

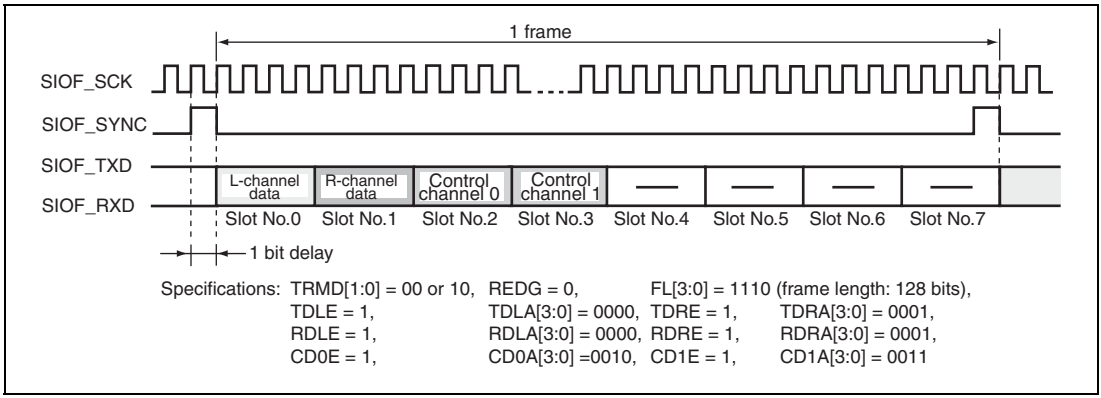


Figure 22.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

16-bit Stereo Data (4): Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.2 used for right-channel data, slot No.1 used for control data for channel 0 , slot No.3 used for control data for channel 1, and frame length = 128 bits

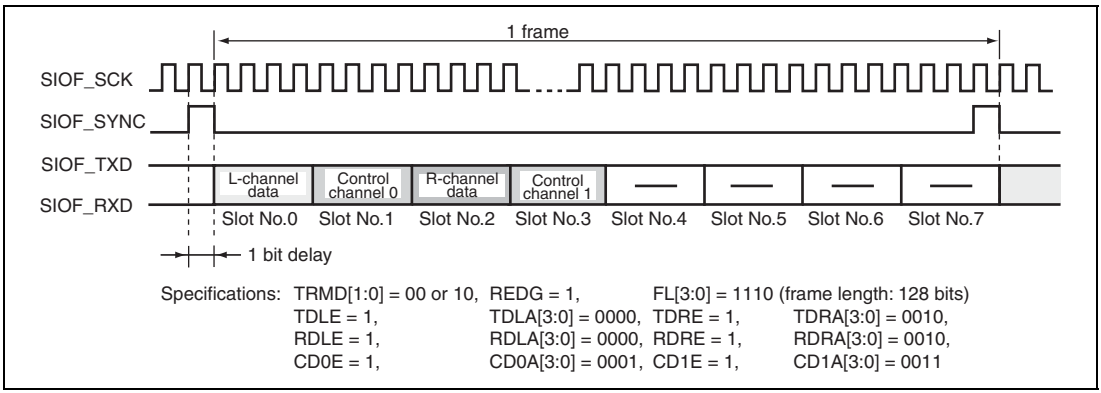


Figure 22.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

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

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

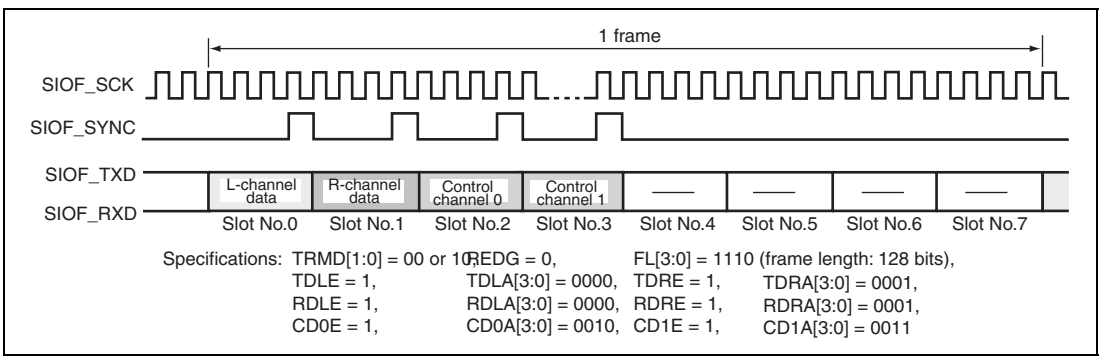


Figure 22.20 Transmit and Receive Timing (16-Bit Stereo Data)

Section 23 Serial Protocol Interface (HSPI)

This LSI incorporates one channel of the Serial Protocol Interface (HSPI).

23.1 Features

The HSPI has the following features.

- Operating mode: Master mode or Slave mode.
- The transmit and receive sections within the module are double buffered to allow duplex communication.
- A flexible peripheral clock division strategy allows a wide range of bit rates to be supported.
- The programmable clock control logic allows setting for two different transmit protocols and 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 configured as a master mode either automatically as part of the data transfer process, or under the manual control of the host processor.

Figure 23.1 is a block diagram of the HSPI.

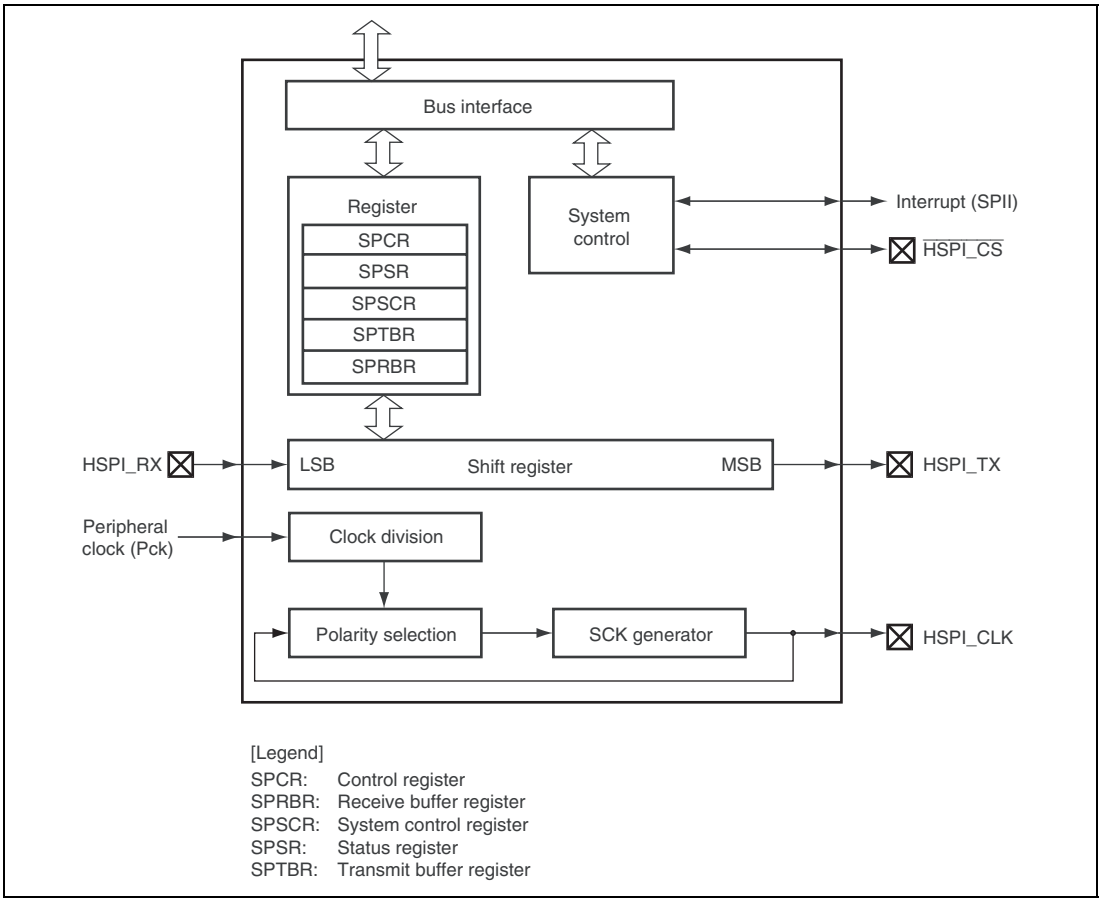


Figure 23.1 Block Diagram of HSPI

23.2 Input/Output Pins

The input/output pins of the HSPI is shown in table 23.1.

Table 23.1 Pin Configuration

Pin Name	Function	I/O	Description
HSPI_CLK	Serial bit clock pin	Input/Output	Clock input/output
HSPI_TX	Transmit data pin	Output	Transmit data output
HSPI_RX	Receive data pin	Input	Receive data input
$\overline{\text{HSPI\_CS}}$	Chip select pin	Input/Output	Chip select

Note: These pins are multiplexed with the SCIF channel 0, FLCTL, GPIO and mode control pins.

23.3 Register Descriptions

Table 23.2 shows the HSPI register configuration. Table 23.3 shows the register states in each processing mode.

Table 23.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Control register	SPCR	R/W	H'FFE5 0000	H'1FE5 0000	32	Pck
Status register	SPSR	R/W*	H'FFE5 0004	H'1FE5 0004	32	Pck
System control register	SPSCR	R/W	H'FFE5 0008	H'1FE5 0008	32	Pck
Transmit buffer register	SPTBR	R/W	H'FFE5 000C	H'1FE5 000C	32	Pck
Receive buffer register	SPRBR	R	H'FFE5 0010	H'1FE5 0010	32	Pck

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	Module Standby
Control register	SPCR	H'0000 0000	H'0000 0000	Retained	Retained
Status register	SPSR	H'0000 0020	H'xxxx xx20	Retained	Retained
System control register	SPSCR	H'0000 0040	H'0000 0040	Retained	Retained
Transmit buffer register	SPTBR	H'0000 0000	H'0000 0000	Retained	Retained
Receive buffer register	SPRBR	H'0000 0000	H'0000 0000	Retained	Retained

23.3.1 Control Register (SPCR)

SPCR is a 32-bit readable/writable register that controls the transfer data of shift timing and specifies the clock polarity and frequency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FBS	CLKP	IDIV	CLKC4	CLKC3	CLKC2	CLKC1	CLKC0
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/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 8	—	All 0	R	Reserved Although the initial value is 0, these bits will be read as an undefined value. The write value should always be 0.
7	FBS	0	R/W	First Bit Start Controls the timing relationship between each bit of transferred data and the serial bit clock. 0: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the first edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled on the first edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. 1: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the second edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled on the second edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low.
6	CLKP	0	R/W	Serial Clock Polarity 0: HSPI_CLK signal is not inverted and so is low when inactive. 1: HSPI_CLK signal is inverted and so is high when inactive.

Bit	Bit Name	Initial Value	R/W	Description
5	IDIV	0	R/W	Initial Clock Division Ratio 0: The peripheral clock is divided by a factor of 4 initially to create an intermediate frequency, which is further divided to create the serial bit clock when master mode. 1: The peripheral clock is divided by a factor of 32 initially to create an intermediate frequency, which is further divided to create the serial bit clock when master mode.
4 to 0	CLKC4 to CLKC0	All 0	R/W	Clock Division Count These bits determine the number of intermediate frequency cycles long both the high and low periods of the serial bit clock. 00000: 1 intermediate frequency cycle. Serial bit clock frequency = Intermediate frequency / 2. 00001: 2 Intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 4. 00010: 3 intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 6. : : 11111: 32 intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 64.

The serial bit clock frequency can be computed using the following formula:

$$\text{Serial bit clock frequency} = \frac{\text{Peripheral clock frequency}}{\text{Initial division ratio} \times ((\text{Clock division count} + 1) \times 2)}$$

When the HSPI is configured as a slave, the IDIV and CLKC bits are ignored and the HSPI synchronizes to the externally supplied serial bit clock. The maximum value of the external serial 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 undergo the HSPI software reset.

23.3.2 Status Register (SPSR)

SPSR is a 32-bit readable/writable register. The status flag in SPSR can confirm whether the HSPI correctly operates or not. If the ROIE bit in SPSCR is set to 1, an interrupt request is generated due to the occurrence of the receive buffer overrun error or the warning of the receive buffer overrun error. When the TFIE bit in SPSCR is set to 1, an interrupt request is generated by the transmit complete status flag. If the appropriate enable bit in SPSCR is set to 1, an interrupt request is generated due to the receive FIFO halfway, receive FIFO full, transmit FIFO empty, or transmit FIFO halfway flag. If the RNIE bit in SPSCR is set to 1, an interrupt request is generated when the receive FIFO is not empty.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	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	1	0
	—	—	—	—	—	TXFU	TXHA	TXEM	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	TXFN	TXFL
Initial value:	—	—	—	—	—	0	0	1	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	Undefined	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
10	TXFU	0	R	Transmit FIFO Full Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO is full of bytes for transmission and cannot accept any more. It is cleared to 0 when data is transmitted from the transmit FIFO.
9	TXHA	0	R	Transmit FIFO Halfway Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO reaches the halfway point, that is, it has 4 bytes of data and 4 spaces for more data. It is cleared to 0 when more data is written to the transmit FIFO. It remains set to 1 until cleared to 0 even if the subsequent FIFO level becomes under the halfway point (4 bytes). If TXHA = 1 and THIE = 1 then the interrupt is generated.

Bit	Bit Name	Initial Value	R/W	Description
8	TXEM	1	R	Transmit FIFO Empty Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO is empty of data to transmit. It is cleared to 0 when more data is written to the transmit FIFO. If TXEM = 1 and TEIE = 1 then the interrupt is generated.
7	RXFU	0	R	Receive FIFO Full Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the receive FIFO is full of received bytes and cannot accept any more. It is cleared to 0 when data is read out of the receive FIFO. 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 FIFO mode. The flag is set to 1 when the receive FIFO reaches the halfway point, that is, it has 4 bytes of data and 4 spaces for more data. This flag is cleared to 0 when the receive data is read from receive FIFO and the FIFO level becomes under 4 bytes (halfway point). It remain set to 1 until cleared to 0 regardless of the 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 FIFO mode. The flag is set to 1 when the receive FIFO is empty of received data. It is cleared to 0 when more data is received into to the receive FIFO. If RXEM = 0 and RNIE = 1 then the interrupt is generated.
4	RXOO	0	R/W*	Receive Buffer Overrun Occurred Flag This status flag is set to 1 when new data has been received but the previous received data has not been read from SPRBR. The previously received data will not be overwritten by the newly received data. The RXOO flag remain set to 1 until writing a 0 to its bit position. If RXOO = 1 and ROIE = 1 then the interrupt is generated.

Bit	Bit Name	Initial Value	R/W	Description
3	RXOW	0	R/W*	Receive Buffer Overrun Warning Flag This status flag is set to 1 when a new serial data transfer starts and the previous received data has not been read from SPRBR. The RXOW remain set to 1 until writing a 0 to its bit position. If RXOW= 1 and ROIE = 1 then the interrupt is generated.
2	RXFL	0	R	Receive Buffer Full Status Flag This status flag indicates that new data is available in the SPRBR and has not yet been read. It is set to 1 at the completion of a serial bus transfer at the point the shift register contents are loaded into the SPRBR. This bit is cleared to 0 by reading SPRBR. If RXFL = 1 and RXDE = 1 then the DMA transfer request enabled.
1	TXFN	0	R	Transmit Complete Status Flag This status flag indicates that the last transmission has completed. It is set to 1 when SPTBR is able to write more data from the peripheral bus. This bit is cleared to 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 transmitted data. It is set to 1 when SPTBR is written with data from the peripheral bus. This bit is cleared to 0 when SPTBR is able to accept more data from the peripheral bus. If TXFL = 0 (i.e. the SPTBR is empty) and TXDE = 1 then the DMA transfer request enabled.

Note: * These bits are readable/writable bits. When writing 0, these bits are initialized, while writing 1 is ignored.

23.3.3 System Control Register (SPSCR)

SPSCR is a 32-bit readable/writable register that enables or disables interrupts or FIFO mode, selects either LSB first or MSB first in transmitting/receiving date, and master or slave mode.

If any of the FFEN, LMSB, CSA or MASL bit values are changed, then the module will undergo the HSPI software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TEIE	THIE	RNIE	RHIE	RFIE	FFEN	LMSB	CSV	CSA	TFIE	ROIE	RXDE	TXDE	MASL
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	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	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 read as an undefined value. The write value should always be 0.
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

Bit	Bit Name	Initial Value	R/W	Description
8	FFEN	0	R/W	FIFO Mode Enable Enables or disables the FIFO mode. When FIFO mode is enabled two 8-entry deep FIFOs are made available, one for transmit data and one for receive data. These FIFOs are read and written via SPTBR and SPRBR. When FIFO mode is disabled the SPTBR and SPRBR are used directly so new data must be written to SPTBR and read from SPRBR for each and every transfer. FIFO mode must be disabled if DMA requests are also going to be used to service SPTBR and SPRBR. 0: FIFO mode disabled 1: FIFO mode enabled
7	LMSB	0	R/W	LSB/MSB First Control 0: Data is transmitted and received most significant bit (MSB) first. 1: Data is transmitted and received least significant bit (LSB) first.
6	CSV	1	R/W	Chip Select Value Controls the value output from the chip select when the HSPI is a master and the chip select generation has 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 generated during data transfer. 1: Chip select output is manually controlled, with its 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 Enable 0: Receive overrun occurred / warning interrupt disabled 1: Receive overrun occurred / warning interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RXDE	0	R/W	Receive DMA Enable 0: Receive DMA transfer request disabled 1: Receive DMA transfer request enabled
1	TXDE	0	R/W	Transmit DMA Enable 0: Transmit DMA transfer request disabled 1: Transmit DMA transfer request enabled
0	MASL	0	R/W	Master/Slave Select Bit 0: HSPI module configured as a slave 1: HSPI module configured as a master

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	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TD							
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/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 8	—	All 0	R	Reserved Although the initial value is 0, these bits will be read as an undefined value. The write value should always be 0.
7 to 0	TD	All 0	R/W	Transmit Data Data written to this register is transferred to the shift register for transmission. When reading these bits, always read as data stored in the transmit buffer.

23.3.5 Receive Buffer Register (SPRBR)

SPRBR is a 32-bit read-only register that stores the number of received data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RD							
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	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 read as an undefined value. The write value should always be 0.
7 to 0	RD	All 0	R	Receive Data Data from the shift register, which is stored as each byte, is received, if the previously received data has been read.

23.4 Operation

23.4.1 Operation Overview without DMA (FIFO Mode Disabled)

Figure 23.2 shows the flow of a transmit/receive operation procedure.

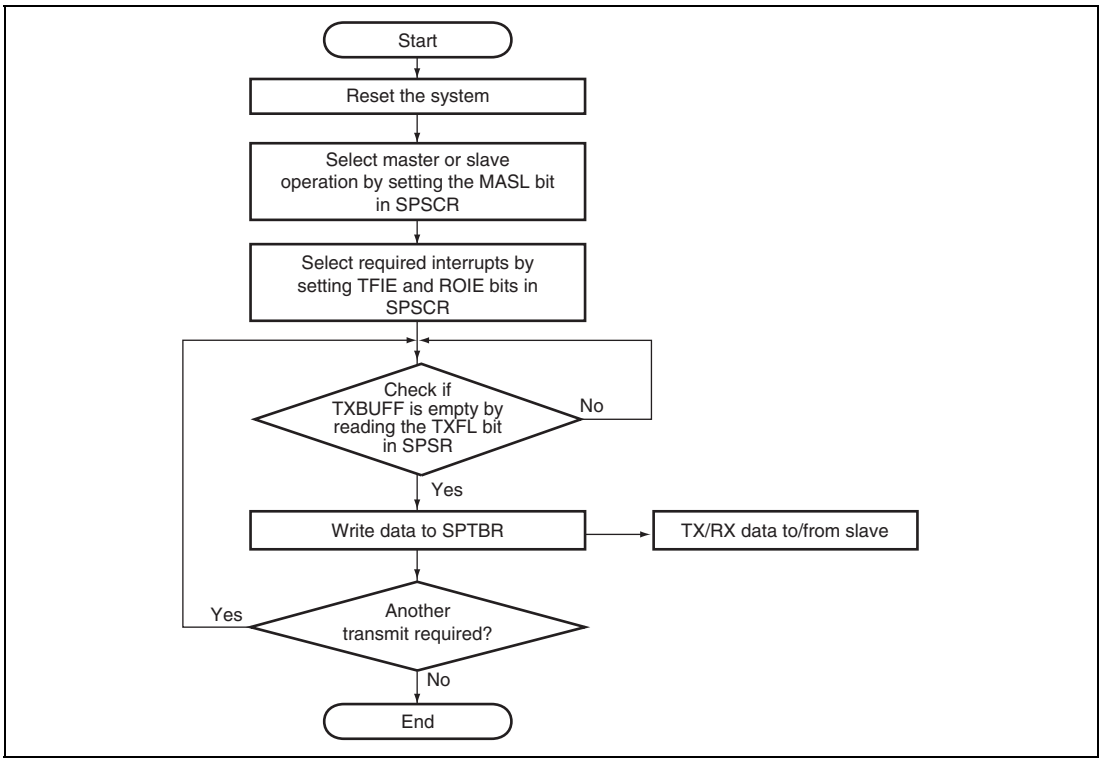


Figure 23.2 Operational Flowchart

Depending on the settings of SPCR, the master transmits data to the slave on either the falling or rising edge of HSPI_CLK and samples data from the slave on the opposite edge. The data transfer between the master and slave completes when the transmit complete status flag (TXFN) in SPSR is set to 1. This flag should be used to identify when an HSPI transfer event (byte transmitted and byte received) has occurred, even in the case where the HSPI module is being used to receive data only (null data being transmitted). By default data is transmitted MSB first, but LSB first is also possible depending on how the LMSB bit in SPSCR is set.

During the transmit function the slave responds by sending data to the master synchronized with the HSPI_CLK from the master transmitted. Data from the slave is sampled and transferred to the shift register in the module and on completion of the transmit function, is transferred to SPRBR.

The $\overline{\text{HSPI\_CS}}$ pin is used to select the HSPI module when the HSPI is configured as a slave, and prepare it to receive data from an external master. When the FBS bit in SPCR is 0, the $\overline{\text{HSPI\_CS}}$ pin must be driven high between successive bytes. When the FBS = 1, the $\overline{\text{HSPI\_CS}}$ pin can stay low for several byte transmissions. In this case, if the system is configured such that the FBS is always 1, then the $\overline{\text{HSPI\_CS}}$ line can be fixed at ground (if the HSPI will only be used as a slave).

23.4.2 Operation Overview with DMA

The operation of the HSPI when DMA is used to perform transmit and receive data transfers is simpler than when DMA is not used. The HSPI must be configured as in the case for transfers without DMA. FIFO mode must be disabled. The DMA controller (DMAC) should then be configured to transfer the required amount of data. DMA requests can then be enabled in the HSPI module and the transfers will then take place without further processor intervention. When the DMAC indicates that all transfers have ended then the DMA request signals in the HSPI module should be disabled to remove any remaining DMA requests. This is necessary as the HSPI module 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 without DMA mode, FIFO mode has been provided. When FIFO mode is enabled, up to 8 bytes can be written in advance for transmission and up to 8 bytes can be received before the receive FIFO needs to be read. To transfer the specified amount of data between the HSPI module and an external device, follow the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity etc.) and enable FIFO mode.
2. Write bytes into the transmit FIFO via SPTBR. If more than 8 bytes are to be transmitted then 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 to the 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 transmit FIFO halfway interrupt and read the contents of the receive FIFO until it is empty. Enable the receive FIFO not empty interrupt to keep track of when the final bytes of the transfer are received.
5. Respond to the receive FIFO not empty interrupt until all the expected data has been received.

6. Disable the module until it is required again.

In some applications, an undefined amount of data will received from an external HSPI device. If this is the case, follow the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity etc.) and enable FIFO mode.
2. Fill the transmit FIFO with the data to transmit. Enable the receive FIFO not empty interrupt.
3. Respond to the receive FIFO not empty interrupt and read data from the receive FIFO until it is empty. Write more data to the transmit FIFO if required.
4. Disable the module when the transfer is to stop.

23.4.4 Timing Diagrams

The following diagrams explain the timing relationship of all shift and sample processes in the HSPI. Figure 23.3 shows the conditions when FBS = 0, while figure 23.4 shows the conditions when FBS = 1. It can be seen that if CLKP in SPCR is 0 then transmit data is shifted on the falling edge of HSPI_CLK and receive data is sampled on the rising edge. The opposite is true when CLKP = 1.

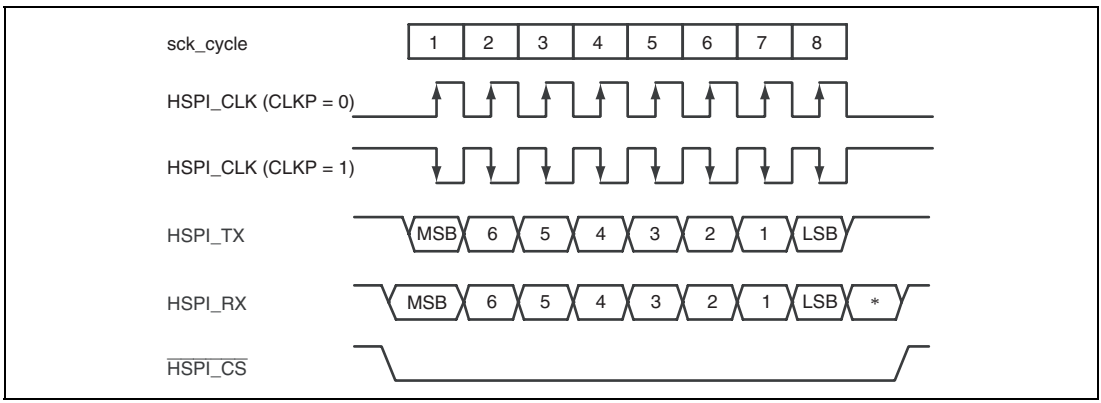


Figure 23.3 Timing Conditions when FBS = 0

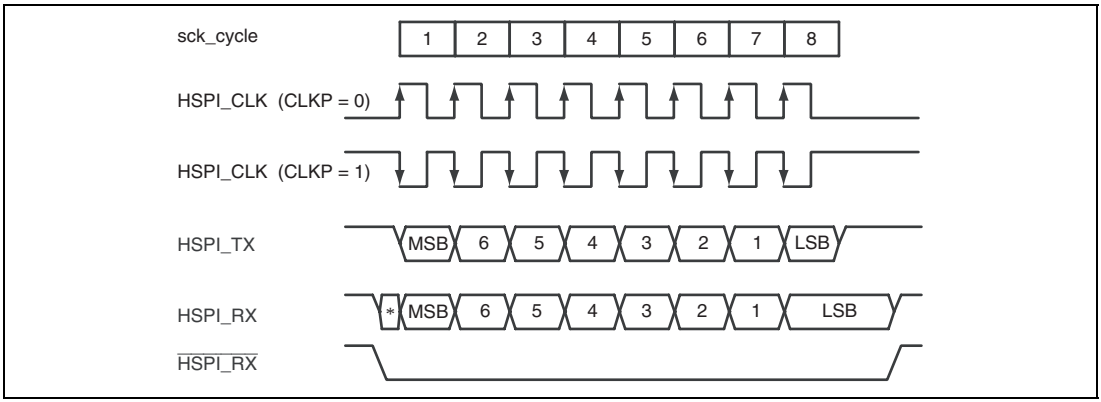


Figure 23.4 Timing Conditions when FBS = 1

23.4.5 HSPI Software Reset

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the HSPI software reset is generated. The receive and transmit FIFO pointers can be initialized by the HSPI software reset. The data transmission after the HSPI software reset should protect transmitting and receiving protocol of HSPI, and please perform it from the first. A guarantee of operation is not offered other than it.

While the master device is not transferring data and the HSPI is in slave mode, the HSPI software reset should be generated before asserting the $\overline{\text{HSPI\_CS}}$. This prevent the HSPI from receiving an 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 FBS bit in SPCR allows selection between two different transfer formats. The MSB or LSB is valid on the falling edge of $\overline{\text{HSPI\_CS}}$. The CLKP bit in SPCR allows for control of the polarity select block 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 synchronized with HSPI_CLK. The transmit byte from the master is replaced with the receive byte from the slave in eight HSPI_CLK cycles. Both the transmit and receive functions are double buffered to allow for continuous reads and writes. When FIFO mode is enabled eight entry FIFOs are available for both transmit and receive data.

Section 24 Multimedia Card Interface (MMCIF)

This LSI supports a multimedia card interface (MMCIF). The MMC mode interface can be utilized. The MMCIF is a clock-synchronous serial interface that transmits/receives data that is distinguished in terms of command and response. A number of commands/responses are predefined in the multimedia card. As the MMCIF specifies a command code and command type/response type upon the issuance of a command, commands extended by the secure multimedia card (Secure-MMC) and additional commands can be supported in the future within the range of combinations of currently defined command types/response types.

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), transfer error (ERR), and FIFO ready (FRDY)
- Interface via the MCCLK output (transfer clock output) pin, the MCCMD input/output (command output/response input) pin, and the MCDAT input/output (data input/output) pin

Figure 24.1 shows a block diagram of the MMCIF.

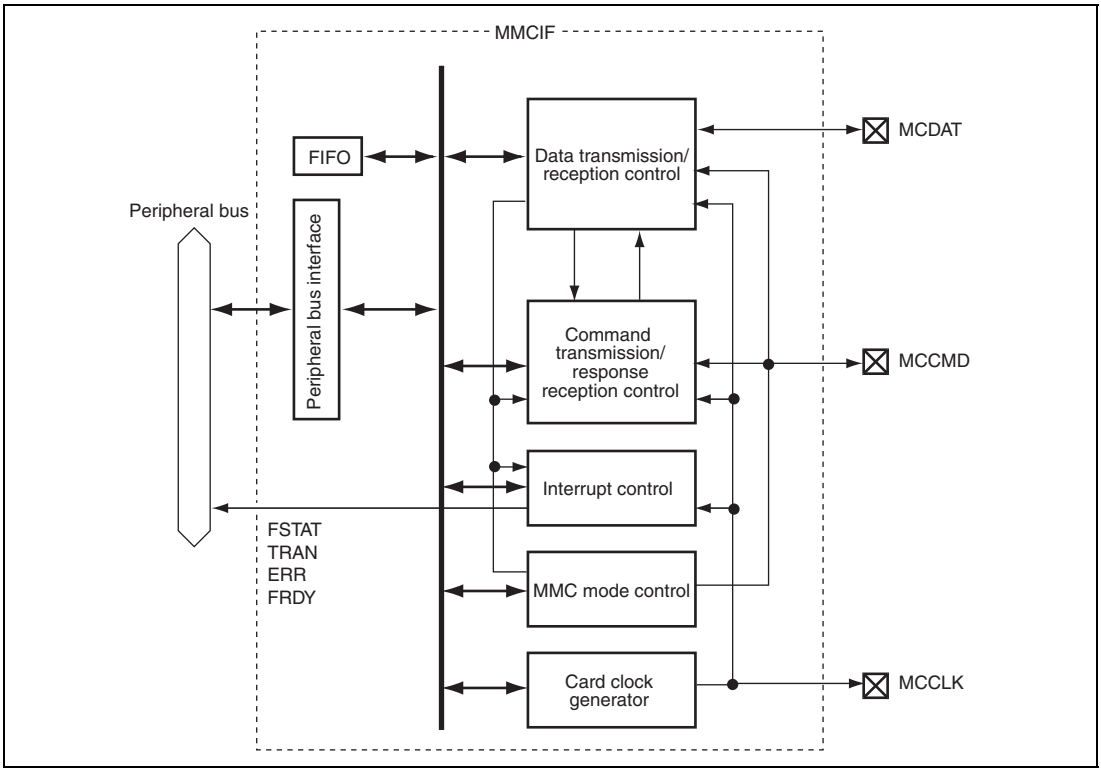


Figure 24.1 Block Diagram of MMCIF

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-drain and CMOS modes, use ports of this LSI.
These pins are multiplexed with the SCIF channel 1, GPIO, and mode control pins.

24.3 Register Descriptions

Table 24.2 shows the MMCIF register configuration. Table 24.3 shows the register states in each processing mode.

Table 24.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Command register 0	CMDR0	R/W	H'FFE6 0000	H'1FE6 0000	8	Pck
Command register 1	CMDR1	R/W	H'FFE6 0001	H'1FE6 0001	8	Pck
Command register 2	CMDR2	R/W	H'FFE6 0002	H'1FE6 0002	8	Pck
Command register 3	CMDR3	R/W	H'FFE6 0003	H'1FE6 0003	8	Pck
Command register 4	CMDR4	R/W	H'FFE6 0004	H'1FE6 0004	8	Pck
Command register 5	CMDR5	R	H'FFE6 0005	H'1FE6 0005	8	Pck
Command start register	CMDSTRT	R/W	H'FFE6 0006	H'1FE6 0006	8	Pck
Operation control register	OPCR	R/W	H'FFE6 000A	H'1FE6 000A	8	Pck
Card status register	CSTR	R	H'FFE6 000B	H'1FE6 000B	8	Pck
Interrupt control register 0	INTCR0	R/W	H'FFE6 000C	H'1FE6 000C	8	Pck
Interrupt control register 1	INTCR1	R/W	H'FFE6 000D	H'1FE6 000D	8	Pck
Interrupt status register 0	INTSTR0	R/W	H'FFE6 000E	H'1FE6 000E	8	Pck
Interrupt status register 1	INTSTR1	R/W	H'FFE6 000F	H'1FE6 000F	8	Pck
Transfer clock control register	CLKON	R/W	H'FFE6 0010	H'1FE6 0010	8	Pck
Command timeout control register	CTOCR	R/W	H'FFE6 0011	H'1FE6 0011	8	Pck
Transfer byte number count register	TBCR	R/W	H'FFE6 0014	H'1FE6 0014	8	Pck
Mode register	MODER	R/W	H'FFE6 0016	H'1FE6 0016	8	Pck
Command type register	CMDTYR	R/W	H'FFE6 0018	H'1FE6 0018	8	Pck
Response type register	RSPTYR	R/W	H'FFE6 0019	H'1FE6 0019	8	Pck
Transfer block number counter	TBNCR	R/W	H'FFE6 001A	H'1FE6 001A	16	Pck
Response register 0	RSPR0	R/W	H'FFE6 0020	H'1FE6 0020	8	Pck
Response register 1	RSPR1	R/W	H'FFE6 0021	H'1FE6 0021	8	Pck
Response register 2	RSPR2	R/W	H'FFE6 0022	H'1FE6 0022	8	Pck
Response register 3	RSPR3	R/W	H'FFE6 0023	H'1FE6 0023	8	Pck
Response register 4	RSPR4	R/W	H'FFE6 0024	H'1FE6 0024	8	Pck

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Response register 5	RSPR5	R/W	H'FFE6 0025	H'1FE6 0025	8	Pck
Response register 6	RSPR6	R/W	H'FFE6 0026	H'1FE6 0026	8	Pck
Response register 7	RSPR7	R/W	H'FFE6 0027	H'1FE6 0027	8	Pck
Response register 8	RSPR8	R/W	H'FFE6 0028	H'1FE6 0028	8	Pck
Response register 9	RSPR9	R/W	H'FFE6 0029	H'1FE6 0029	8	Pck
Response register 10	RSPR10	R/W	H'FFE6 002A	H'1FE6 002A	8	Pck
Response register 11	RSPR11	R/W	H'FFE6 002B	H'1FE6 002B	8	Pck
Response register 12	RSPR12	R/W	H'FFE6 002C	H'1FE6 002C	8	Pck
Response register 13	RSPR13	R/W	H'FFE6 002D	H'1FE6 002D	8	Pck
Response register 14	RSPR14	R/W	H'FFE6 002E	H'1FE6 002E	8	Pck
Response register 15	RSPR15	R/W	H'FFE6 002F	H'1FE6 002F	8	Pck
Response register 16	RSPR16	R/W	H'FFE6 0030	H'1FE6 0030	8	Pck
CRC status register	RSPRD	R/W	H'FFE6 0031	H'1FE6 0031	8	Pck
Data timeout register	DTOUTR	R/W	H'FFE6 0032	H'1FE6 0032	16	Pck
Data register	DR	R/W	H'FFE6 0040	H'1FE6 0040	16	Pck
FIFO pointer clear register	FIFOCLR	W	H'FFE6 0042	H'1FE6 0042	8	Pck
DMA control register	DMACR	R/W	H'FFE6 0044	H'1FE6 0044	8	Pck
Interrupt control register 2	INTCR2	R/W	H'FFE6 0046	H'1FE6 0046	8	Pck
Interrupt status register 2	INTSTR2	R/W	H'FFE6 0048	H'1FE6 0048	8	Pck

Table 24.3 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	Module Standby
Command register 0	CMDR0	H'00	H'00	Retained	Retained
Command register 1	CMDR1	H'00	H'00	Retained	Retained
Command register 2	CMDR2	H'00	H'00	Retained	Retained
Command register 3	CMDR3	H'00	H'00	Retained	Retained
Command register 4	CMDR4	H'00	H'00	Retained	Retained
Command register 5	CMDR5	H'00	H'00	Retained	Retained
Command start register	CMDSTRT	H'00	H'00	Retained	Retained
Operation control register	OPCR	H'00	H'00	Retained	Retained
Card status register	CSTR	H'0x	H'0x	Retained	Retained
Interrupt control register 0	INTCR0	H'00	H'00	Retained	Retained
Interrupt control register 1	INTCR1	H'00	H'00	Retained	Retained
Interrupt status register 0	INTSTR0	H'00	H'00	Retained	Retained
Interrupt status register 1	INTSTR1	H'00	H'00	Retained	Retained
Transfer clock control register	CLKON	H'00	H'00	Retained	Retained
Command timeout control register	CTOCR	H'01	H'01	Retained	Retained
Transfer byte number count register	TBCR	H'00	H'00	Retained	Retained
Mode register	MODER	H'00	H'00	Retained	Retained
Command type register	CMDTYR	H'00	H'00	Retained	Retained
Response type register	RSPTYR	H'00	H'00	Retained	Retained
Transfer block number counter	TBNCR	H'0000	H'0000	Retained	Retained
Response register 0	RSPR0	H'00	H'00	Retained	Retained
Response register 1	RSPR1	H'00	H'00	Retained	Retained
Response register 2	RSPR2	H'00	H'00	Retained	Retained
Response register 3	RSPR3	H'00	H'00	Retained	Retained
Response register 4	RSPR4	H'00	H'00	Retained	Retained
Response register 5	RSPR5	H'00	H'00	Retained	Retained

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exception	Sleep by SLEEP Instruction	Module Standby
Response register 6	RSPR6	H'00	H'00	Retained	Retained
Response register 7	RSPR7	H'00	H'00	Retained	Retained
Response register 8	RSPR8	H'00	H'00	Retained	Retained
Response register 9	RSPR9	H'00	H'00	Retained	Retained
Response register 10	RSPR10	H'00	H'00	Retained	Retained
Response register 11	RSPR11	H'00	H'00	Retained	Retained
Response register 12	RSPR12	H'00	H'00	Retained	Retained
Response register 13	RSPR13	H'00	H'00	Retained	Retained
Response register 14	RSPR14	H'00	H'00	Retained	Retained
Response register 15	RSPR15	H'00	H'00	Retained	Retained
Response register 16	RSPR16	H'00	H'00	Retained	Retained
CRC status register	RSPRD	H'00	H'00	Retained	Retained
Data timeout register	DTOUTR	H'FFFF	H'FFFF	Retained	Retained
Data register	DR	H'xxxx	H'xxxx	Retained	Retained
FIFO pointer clear register	FIFOCLR	H'00	H'00	Retained	Retained
DMA control register	DMACR	H'00	H'00	Retained	Retained
Interrupt control register 2	INTCR2	H'00	H'00	Retained	Retained
Interrupt status register 2	INTSTR2	H'0x	H'0x	Retained	Retained

[Legend] x: Undefined

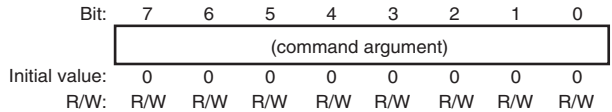
24.3.1 Command Registers 0 to 5 (CMDR0 to CMDR5)

The CMDR registers are six 8-bit registers. A command is written to CMDR as shown in table 24.4, and the command is transmitted when the START bit in CMDSTRT is set to 1. Each command is transmitted in order form the MSB (bit 7) in CMDR0 to the LSB (bit 0) in CMDR5.

Table 24.4 CMDR Configuration

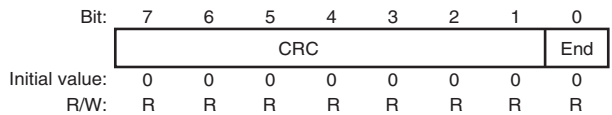
Register	Contents	Operation
CMDR0 to CMDR4	Command argument	Write command arguments.
CMDR5	CRC and End bit	Setting of CRC is unnecessary (automatic calculation). End bit is fixed to 1 and its setting is unnecessary (automatic setting). The read value is 0.

- 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 value should always be 0.
0	End	0	R	This bit is always read as 0. The write value should always be 0.

24.3.2 Command Start Register (CMDSTRT)

CMDSTRT is an 8-bit readable/writable register that triggers the start of command transmission, representing the start of a command sequence. The following operations should have been completed before the command sequence starts.

- Analysis of prior command response, clearing the command response register write if necessary
- Analysis/transfer of receive data of prior command if necessary
- Preparation of transmit data of the next command if necessary
- Setting of CMDTYR, RSPTYR, TBCR and TBNCR
- Setting of CMDR0 to CMDR4

  The CMDR0 to CMDR4, CMDTYR, RSPTYR, TBCR and TBNCR registers should not be changed until command transmission has ended (during the CWRE flag in CSTR has been set to 1 or until command transmit end interrupt has occurred).

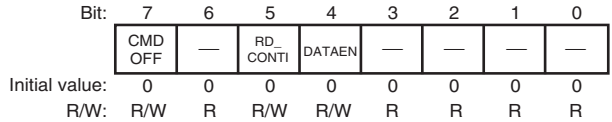
Command sequences are controlled by the sequencers in both the MMCIF side and the MMC card side. Normally, these operate synchronously. However, if an error occurs or a command is aborted, these may become temporarily unsynchronized. Be careful when setting the CMDOFF bit in OPCR, issuing the CMD12 command, or processing an error in MMC mode. A new command sequence should be started only after the end of the command sequence on both the MMCIF and 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 value should always be 0.
0	START	0	R/W	Starts command transmission when 1 is written. This bit is automatically cleared after the MMCIF received START command. When 0 is written to this bit, operation is not affected.

24.3.3 Operation Control Register (OPCR)

OPCR is an 8-bit readable/writable register that aborts command operation, and suspends or continues data transfer.



Bit:	7	6	5	4	3	2	1	0
	CMD OFF	—	RD_ CONTI	DATAEN	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CMDOFF	0	R/W	Command Off Aborts all command operations (MMCIF command sequence) when 1 is written after a command is transmitted. This bit is cleared automatically after the MMCIF received the CMDOFF command. Write enabled period: From command transmission completion to command sequence end Write of 0: Operation is not affected. Write of 1: Command sequence is forcibly aborted. Note: Do not write to this bit out of the write enable period.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	RD_CONTI	0	R/W	Read Continue Read data reception is resumed when 1 is written while the sequence has been halted by FIFO full or termination of block reading in multiple block read. This bit is cleared automatically when 1 is written and the MMCIF received the RD_CONTI command. Write enabled period: While read data reception is 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 enable period.

Bit	Bit Name	Initial Value	R/W	Description
4	DATAEN	0	R/W	Data Enable Starts a write data transmission by a command with write data. This bit is cleared automatically when 1 is written and the MMCIF received the DATAEN command. Resumes write data transmission while the sequence has been halted by FIFO empty or termination of block writing in multiple block write. Write enabled period: (1) after receiving a response to a command with write data, (2) while sequence is halted by FIFO empty, (3) when one block writing in multiple block write is terminated. Write of 0: Operation is not affected. Write of 1: Starts or resumes write data transmission. Note: Do not write to this bit out of the write enable period.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

In write data transmission, the contents of the command response and data response should be analyzed, and then transmission should be triggered. In addition, the data transmission should be temporarily halted by FIFO full/empty, and it should be resumed when the preparation has been completed.

In multiple block transfer, the transfer should be temporarily halted at every block break to select either to continue to the next block or to abort the multiple block transfer command by issuing the CMD12 command. To continue to the next block, the RD_CONTI and DATAEN bits should be set to 1. To issue the CMD12 command, the CMDOFF bit should be set to 1 to abort the command sequence on the MMCIF side. When using the auto-mode for a pre-defined multiple block transfer, the setting of the RD_CONTI bit or the DATAEN bit between blocks can be omitted.

24.3.4 Card Status Register (CSTR)

CSTR indicates the MMCIF status during command sequence execution.

Bit:	7	6	5	4	3	2	1	0
	BUSY	FIFO_ FULL	FIFO_ EMPTY	CWRE	DTBUSY	DTBUSY _TU	—	REQ
Initial value:	0	0	0	0	0	—	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUSY	0	R	Command Busy Indicates command execution status. When the CMDOFF bit in OPCR is set to 1, this bit is cleared to 0 because the MMCIF command sequence is aborted. 0: Idle state waiting for a command, or data busy state 1: Command sequence execution in progress
6	FIFO_FULL	0	R	FIFO Full This bit is set to 1 when the FIFO becomes full while data is being received from the card, and cleared to 0 when RD_CONTI is set to 1 or the command sequence 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 empty while data is being sent to the card, and cleared to 0 when DATA_EN is set to 1 or the command sequence is completed. Indicates whether the FIFO holds data or not. 0: The FIFO includes data. 1: The FIFO is empty.

Bit	Bit Name	Initial Value	R/W	Description
4	CWRE	0	R	Command Register Write Enable Indicates whether the CMDR command is being transmitted or has been transmitted. 0: The CMDR command has been transmitted, or the START bit in CMDSTRT has not been set yet, so the new command can be written. 1: The CMDR command is waiting for transmission or is being transmitted. If a new command is written, a malfunction will result.
3	DTBUSY	0	R	Data Busy Indicates command execution status. Indicates that the card is in the busy state after the command sequence of a command without data transfer which includes the busy state in the response, or a command with write data has been ended. 0: Idle state waiting for a command, or command sequence execution in progress 1: Card is in the data busy state after command sequence termination.
2	DTBUSY_TU	Undefined	R	Data Busy Pin Status Indicates the MCDAT pin level. By reading this bit, the 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 should always be 0.
0	REQ	0	R	Interrupt Request Indicates whether an interrupt is requested or not. An interrupt request is the logical OR of the INTSTR0, INTSTR1 and INTSTR2 flags. The INTSTR0, INTSTR1 and INTSTR2 flags set is controlled by the enable bits in INTCR0, INTSTR1 and INTCR2. 0: No interrupt requested. 1: Interrupt requested.

24.3.5 Interrupt Control Registers 0 to 2 (INTCR0 to INTCR2)

The INTCR registers enable or disable interrupts.

- INTCR0

Bit:	7	6	5	4	3	2	1	0
	FEIE	FFIE	DRPIE	DTIE	CRPIE	CMDIE	DBSYIE	BTIE
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	FEIE	0	R/W	FIFO Empty Interrupt Flag Setting Enable 0: Disables FIFO empty interrupt (disables FEI flag setting). 1. Enables FIFO empty interrupt (enables FEI flag setting).
6	FFIE	0	R/W	FIFO Full Interrupt Flag Setting Enable 0: Disables FIFO full interrupt (disables FFI flag setting). 1: Enables FIFO full interrupt (enables FFI flag setting).
5	DRPIE	0	R/W	Data Response Interrupt Flag Setting Enable 0: Disables data response interrupt (disables DPRI flag setting). 1: Enables data response interrupt (enables DPRI flag setting).
4	DTIE	0	R/W	Data Transfer End Interrupt Flag Setting Enable 0: Disables data transfer end interrupt (disables DTI flag setting). 1: Enables data transfer end interrupt (enables DTI flag setting).
3	CRPIE	0	R/W	Command Response Receive End Interrupt Flag Setting Enable 0: Disables command response receive end interrupt (disables CRPI flag setting). 1: Enables command response receive end interrupt (enables CRPI flag setting).

Bit	Bit Name	Initial Value	R/W	Description
2	CMDIE	0	R/W	Command Transmit End Interrupt Flag Setting Enable 0: Disables command transmit end interrupt (disables CMDI flag setting). 1: Enables command transmit end interrupt (enables CMDI flag setting).
1	DBSYIE	0	R/W	Data Busy End Interrupt Flag Setting Enable 0: Disables data busy end interrupt (disables DBSYI flag setting). 1: Enables data busy end interrupt (enables DBSYI flag setting).
0	BTIE	0	R/W	Multiple block Transfer End Flag Flag Setting Enable 0: Disables multiple block transfer end flag setting 1: Enables multiple block transfer end flag setting

- INTCR1

Bit:	7	6	5	4	3	2	1	0
	INTR Q2E	INTR Q1E	INTR Q0E	—	—	CRCE RIE	DTE RIE	CTE RIE
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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	CRCERIE	0	R/W	CRC Error Interrupt Flag Setting Enable 0: Disables CRC error interrupt (disables CRCERI flag setting). 1: Enables CRC error interrupt (enables CRCERI flag setting).
1	DTERIE	0	R/W	Data Timeout Error Interrupt Flag Setting Enable 0: Disables data timeout error interrupt (disables DTERI flag setting). 1: Enables data timeout error interrupt (enables DTERI flag setting).
0	CTERIE	0	R/W	Command Timeout Error Interrupt Flag Setting Enable 0: Disables command timeout error interrupt (disables CTERI flag setting). 1: Enables command timeout error interrupt (enables 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 value should always be 0.
0	FRDYIE	0	R/W	FIFO Ready Interrupt Enable 0: Disables FIFO ready interrupt (disables FRDY flag setting). 1: Enables FIFO ready interrupt (enables FRDY flag setting).

24.3.6 Interrupt Status Registers 0 to 2 (INTSTR0 to INTSTR2)

The INTSTR registers enable or disable MMCIF interrupts FSTAT, TRAN, ERR and FRDY.

- INTSTR0

Bit:	7	6	5	4	3	2	1	0
	FEI	FFI	DRPI	DTI	CRPI	CMDI	DBSYI	BTI
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	Interrupt output
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)	FSTAT
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)	FSTAT

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
5	DRPI	0	R/W	Data Response Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading DRPI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When the CRC status is received while DRPIE = 1.	TRAN
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.	TRAN
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.	TRAN
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.)	TRAN

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
1	DBSYI	0	R/W	Data Busy End Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading DBSYI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When data busy state is canceled while DBSYIE = 1. (When the DTBUSY bit in CSTR is cleared.)	TRAN
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.	TRAN

- INTSTR1

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRCERI	DTERI	CTERI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	—
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.	ERR
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.	ERR
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.	ERR

- INTSTR2

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FRDY_TU	FRDYI
Initial value:	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	Interrupt output
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	—
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.	FRDY

24.3.7 Transfer Clock Control Register (CLKON)

CLKON controls the transfer clock frequency and clock ON/OFF.

At this time, use a sufficiently slow clock for transfer through open-drain type output in MMC mode.

In a command sequence, do not perform clock ON/OFF or frequency modification.

Bit:	7	6	5	4	3	2	1	0
	CLKON	—	—	—	CSEL3	CSEL2	CSEL1	CSEL0
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 MCCLK pin to low level. 1: Outputs the transfer clock from the MCCLK pin.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value 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 transfer clock.
0	CSEL0	0	R/W	0010: Uses the 1/4-divided peripheral clock as a transfer clock. 0011: Uses the 1/8-divided peripheral clock as a transfer clock. 0100: Uses the 1/16-divided peripheral clock as a transfer clock. 0101: Uses the 1/32-divided peripheral clock as a transfer clock. 0110: Uses the 1/64-divided peripheral clock as a transfer clock. 0111: Uses the 1/128-divided peripheral clock as a transfer clock. 1000: Uses the 1/256-divided peripheral clock as a transfer clock. 1001 to 1111: Setting prohibited Note: To output transfer clock, it is necessary to set the CLKON bit to 1, and set the CSEL[3:0] bits other than 0000 and 1001 to 1111.

24.3.8 Command Timeout Control Register (CTOCR)

CTOCR specifies the period to generate a timeout for the command response.

The counter (CTOUTC), to which the peripheral bus does not have access, counts the transfer clock to monitor the command timeout. The initial value of CTOUTC is 0, and CTOUTC starts counting the transfer clock from the start of command transmission. CTOUTC is cleared and stops counting the transfer clock when command response reception has been completed, or when the command sequence has been aborted by setting the CMDOFF bit to 1.

When the command response cannot be received, CTOUTC continues counting the transfer clock, and enters the command timeout error state when the number of transfer clock cycles reaches the number specified in CTOCR. When the CTERIE bit in INTCR1 is set to 1, the CTERI flag in INTSTR1 is set. As CTOUTC continues counting transfer clock, the CTERI flag setting condition is repeatedly generated. To perform command timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the CTERI flag should be cleared to 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 value should always be 0.
0	CTSEL0	1	R/W	Command Timeout Select 0: 128 transfer clock cycles from command transmission completion to response reception completion 1: 256 transfer clock cycles from command transmission completion to response reception completion

Note: If R2 response (17-byte command response) is requested and CTSEL0 is cleared to 0, a timeout is generated during response reception. Therefore, set CTSEL0 to 1.

24.3.9 Transfer Byte Number Count Register (TBCR)

TBCR is an 8-bit readable/writable register that specifies the number of bytes to be transferred (block size) for each single block transfer command. TBCR specifies the number of data block bytes not including the start bit, end bit, and CRC.

The multiple block transfer command corresponds to the number of bytes of each data block. This setting is ignored by the stream transfer command.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	C3	C2	C1	C0
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 value should 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 executing a 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

24.3.10 Mode Register (MODER)

MODER is an 8-bit readable/writable register that specifies the MMCIF operating mode. The following sequence should be repeated when the MMCIF uses the multimedia card: Send a command, wait for the end of the command sequence and the end of the data busy state, and send a next command.

The series of operations from command sending, command response reception, data transmission/reception, and data response reception is called as the command sequence. The command sequence starts from sending a command by setting the START bit in CMDSTRT to 1, and ends when all necessary data transmission/reception and response reception have been completed. The multimedia card supports the data busy state such that only the specific command is accepted to write/erase data to/from the flash memory in the card during command sequence execution and after command sequence execution has ended. The data busy state is indicated by a 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 value should always be 0.
0	MODE	0	R/W	Operating Mode Specifies the MMCIF operating mode. 0: Operates in MMC mode 1: Setting prohibited

24.3.11 Command Type Register (CMDTYR)

CMDTYR is an 8-bit readable/writable register that specifies the command format in conjunction with RSPTYR. Bits TY1 and TY0 specify the existence and direction of transfer data, and bits TY6 to TY2 specify the additional settings. All of bits TY6 to TY2 should be cleared to 0 or only one of them should be set to 1. Bits TY6 to TY2 can only be set to 1 if the corresponding settings in TY1 and TY0 allow that setting. If these bits are not set correctly, the operation cannot be guaranteed. When executing a single block transaction, set TY1 and TY0 to 01 or 10, and TY6 to TY2 to all 0s.

Bit:	7	6	5	4	3	2	1	0
	—	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 value should always be 0.
6	TY6	0	R/W	Type 6 Specifies a predefined multiple block transaction. TY[1:0] should be set to 01 or 10. When using the command set to this bit, it is necessary to specify the transfer block size and the transfer block number in the TBCR and TBNCR respectively.
5	TY5	0	R/W	Type 5 Specifies a multiple block transaction when using secure MMC. TY[1:0] should be set to 01 or 10. Using the command to set to this bit, it is necessary to specify the transfer block size and the transfer block number in the TBCR and TBNCR respectively.
4	TY4	0	R/W	Type 4 Set this bit to 1 when specifying the CMD12 command. Bits TY1 and TY0 should be set to 00.

Bit	Bit Name	Initial Value	R/W	Description
3	TY3	0	R/W	Type 3 Set this bit to 1 when specifying stream transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the stream transfer specified by this bit ends when it is aborted by the CMD12 command.
2	TY2	0	R/W	Type 2 Set this bit to 1 when specifying a multiple block transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the multiple block transfer specified by this bit ends when it is aborted by the CMD12 command.
1	TY1	0	R/W	Types 1 and 0
0	TY0	0	R/W	These bits specify the existence and direction of 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 conjunction with CMDTYR. Bits RTY2 to RTY0 specify the number of response bytes, and bits RTY5 and RTY4 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

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	RTY5	0	R/W	Response Type 5 Sets data busy status from the MMC card. 0: A command without data busy 1: A command with data busy
4	RTY4	0	R/W	Response Type 4 Specifies that the command response CRC is checked through CRC7. Bits RTY2 to RTY0 should be set to 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 value should always be 0.
2	RTY2	0	R/W	Response Types 2 to 0 These bits specify the number of command response bytes. 000: A command needs no command response. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: A command needs 6-byte command responses. Specified by R1, R1b, R3, R4, and R5 responses. 101: A command needs a 17-byte command response. Specified by R2 response. 110: Setting prohibited 111: Setting prohibited
1	RTY1	0	R/W	
0	RTY0	0	R/W	

Note: When checking R2 response, read the value of RSPR0 to RSPR16 after receiving the response and check them by software.

Table 24.5 summarizes the correspondence between the commands described in the MultiMediaCard System Specification Version 3.1 and the settings of the CMDTYR and RSPTYR registers.

Table 24.5 Correspondence between Commands and Settings of CMDTYR and RSPTYR

CMD			CMDTYR						RSPTYR		
INDEX	Abbreviation	resp	TY6	TY5	TY4	TY3	TY2	TY[1:0]	RTY5	RTY4	RTY[2:0]
CMD0	GO_IDLE_STATE	—						00			000
CMD1	SEND_OP_COND	R3						00			100
CMD2	ALL_SEND_CID	R2						00			101
CMD3	SET_RELATIVE_ADDR	R1						00		*4	100
CMD4	SET_DSR	—						00			000
CMD7	SELECT/DESELECT_CARD	R1b						00	1	*4	100
CMD9	SEND_CSD	R2						00			101
CMD10	SEND_CID	R2						00			101
CMD11	READ_DAT_UNTIL_STOP	R1				1		01		*4	100
CMD12	STOP_TRANSMISSION	R1b			1			00	1	*4	100
CMD13	SEND_STATUS	R1						00		*4	100
CMD15	GO_INACTIVE_STATE	—						00			000
CMD16	SET_BLOCKLEN	R1						00		*4	100
CMD17	READ_SINGLE_BLOCK	R1		*3				01		*4	100
CMD18	READ_MULTIPLE_BLOCK	R1	*2				*2	01		*4	100
CMD20	WRITE_DAT_UNTIL_STOP	R1				1		10		*4	100
CMD23	SET_BLOCK_COUNT	R1						00		*4	100
CMD24	WRITE_BLOCK	R1		*3				10		*4	100
CMD25	WRITE_MULTIPLE_BLOCK	R1	*2				*2	10		*4	100
CMD26	PROGRAM_CID	R1						10		*4	100
CMD27	PROGRAM_CSD	R1						10		*4	100
CMD28	SET_WRITE_PROT	R1b						00	1	*4	100
CMD29	CLR_WRITE_PROT	R1b						00	1	*4	100
CMD30	SEND_WRITE_PROT	R1						01		*4	100

CMD			CMDTYR						RSPTYR		
INDEX	Abbreviation	resp	TY6	TY5	TY4	TY3	TY2	TY[1:0]	RTY5	RTY4	RTY[2:0]
CMD32*[1]	TAG_SECTOR_START	R1						00		*[4]	100
CMD33*[1]	TAG_SECTOR_END	R1						00		*[4]	100
CMD34*[1]	UNTAG_SECTOR	R1						00		*[4]	100
CMD35	TAG_ERASE_GROUP_ START	R1						00		*[4]	100
CMD36	TAG_ERASE_GROUP_END	R1						00		*[4]	100
CMD37*[1]	UNTAG_ERASE_GROUP	R1						00		*[4]	100
CMD38	ERASE	R1b						00	1	*[4]	100
CMD39	FAST_IO	R4						00		*[4]	100
CMD40	GO_IRQ_STATE	R5						00		*[4]	100
CMD42	LOCK_UNLOCK	R1b						10	1	*[4]	100
CMD55	APP_CMD	R1						00		*[4]	100
CMD56	GEN_CMD	R1b						*[5]	1	*[4]	100

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 advance, clear the TY6 bit and set the TY2 bit when the transfer block number is not set.
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. Note that it is impossible to check CRC of R2.
5. Set these bits to 01 in read and 10 in write access.

24.3.13 Transfer Block Number Counter (TBNCR)

A value other than 0 must be written to the TBNCR register if a multiple block transfer is selected through the TY5 and TY6 bits in the CMDTYR. Set the transfer block number in the TBNCR. The value of TBNCR is decremented by one as each block transfer is executed and the command sequence ends when the TBNCR value equals 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBNCR															
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/W

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 transferred or 0 is written to TBNCR.

24.3.14 Response Registers 0 to 16, D (RSPR0 to RSPR16, RSPRD)

RSPR0 to RSPR16 are command response registers, which are seventeen 8-bit registers. RSPRD is an 8-bit CRC status register.

The number of command response bytes differs according to the command. The number of command response bytes can be specified by RSPTYR in the MMCIF. The command response is shifted-in from bit 0 in RSPR16, and shifted to the number of command response bytes × 8 bits. Table 24.6 summarizes the correspondence between the number of command response bytes and valid RSPR register.

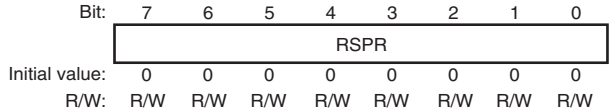
Table 24.6 Correspondence between Command Response Byte Number and RSPR

	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 in is not automatically cleared, and it is continuously shifted until it is shifted out from bit 7 in RSPR0. To clear unnecessary bytes to H'00, write an arbitrary value to each RSPR.

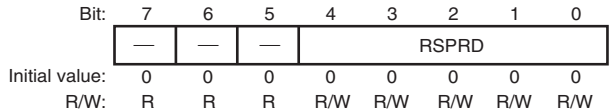
Clearing an RSPR is completed two transfer clock cycles after an arbitrary value is written to the RSPR.

- RSPR0 to RSPR16



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RSPR	H'00	R/W	These bits are cleared to H'00 by writing an arbitrary value. RSPR0 to RSPR16 comprise a continuous 17-byte shift 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 value should always be 0.
4 to 0	RSPRD	00000	R/W	CRC status [Clearing condition] When writing any value to these bits, cleared to 00000. CRC status is stored. CRC status is command response from the card when data is written into the MMC card.

24.3.15 Data Timeout Register (DTOUTR)

DTOUTR specifies the period to generate a data timeout. The 16-bit counter (DTOUTC) and a prescaler, to which the peripheral bus does not have access, count the peripheral clock to monitor the data timeout. The prescaler always counts the peripheral clock, and outputs a count pulse for every 10,000 peripheral clock cycles. The initial value of DTOUTC is 0, and DTOUTC starts counting the prescaler output from the start of the command sequence. DTOUTC is cleared when the command sequence has ended, or when the command sequence has been aborted by setting the CMDOFF bit to 1, after which the DTOUTC stops counting the prescaler output.

When the command sequence does not end, DTOUTC continues counting the prescaler output, and enters the data timeout error states when the number of prescaler outputs reaches the number specified in DTOUTR. When the DTERIE bit in INTCR1 is set to 1, the DTERI flag in INTSTR1 is set. As DTOUTC continues counting prescaler output, the DTERI flag setting condition is repeatedly generated. To perform data timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the DTERI flag should be cleared to prevent extra-interrupt generation.

For a command with data busy status, data timeout cannot be monitored since the command sequence is terminated before entering the data busy state. Timeout in the data busy state should be monitored by firmware.

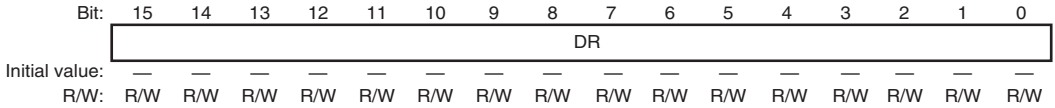
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTOUTR															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 × DTOUTR setting value × 10,000.

24.3.16 Data Register (DR)

DR is a register for reading/writing FIFO data.

Word/byte access is enabled to addresses of this register.



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 lower bytes are transmitted or received in that order. Word access and byte access can be done in random order. However, (DR address + 1) cannot be accessed in 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 FIFO as 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 data is started to be received while the FIFO is empty and data is received in the order of H'01, H'23, ..., 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.

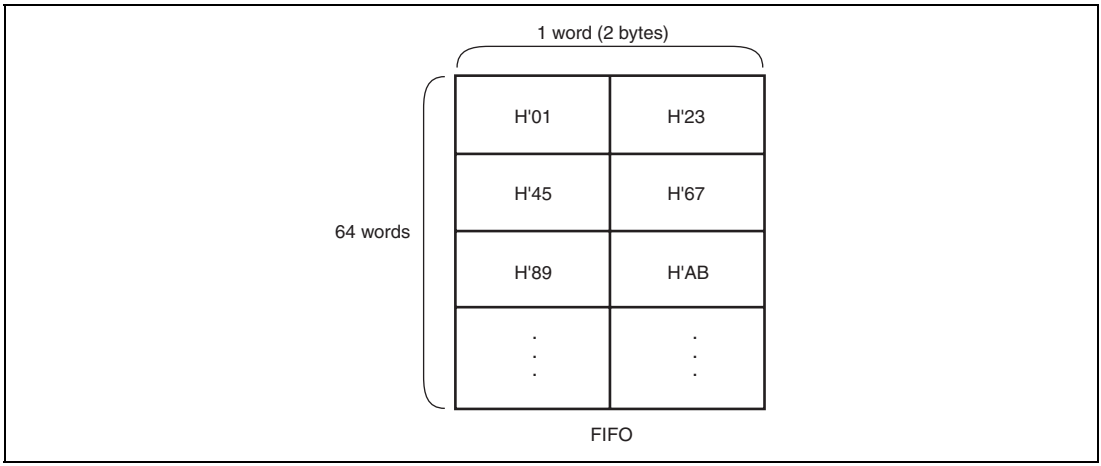


Figure 24.2 DR Access Example

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 arbitrary value to this register.

24.3.18 DMA Control Register (DMACR)

DMACR sets DMA request signal output. DMAEN enables or disables a DMA request signal. The DMA request signal is output based on a value that has been set to SET2 to SET0.

Bit:	7	6	5	4	3	2	1	0
	DMAEN	AUTO	—	—	—	SET2	SET1	SET0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial 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 transfer using 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 value should 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 capacity. 010: FIFO remained data is 1/2 or less of FIFO capacity. 011: FIFO remained data is 3/4 or less of FIFO capacity. 100: FIFO remained data is 1 byte or more. 101: FIFO remained data is 1/4 or more of FIFO capacity. 110: FIFO remained data is 1/2 or more of FIFO capacity. 111: FIFO remained data is 3/4 or more of FIFO capacity.

24.4 Operation

The multimedia card is an external storage media that can be easily connected or disconnected. The MMCIF operates in MMC mode.

Insert a card and supply power to it. Then operate the MMCIF by applying the transfer clock after setting an appropriate transfer clock frequency.

Do not connect or disconnect the card during command sequence execution or in the data busy state.

24.4.1 Operations in MMC Mode

MMC mode is an operating mode in which the transfer clock is output from the MCCLK pin, 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 while data is being transmitted/received.

This feature is efficient for multiple block or stream transfer. In this case, the next command is the CMD12 command, which aborts the current command sequence.

In MMC mode, broadcast commands that simultaneously issue commands to multiple cards are supported. After information of the inserted cards is recognized by a broadcast command, a relative address is given to each card. One card is selected by the relative address, other cards are deselected, and then various commands are issued to the selected card.

Commands in MMC mode are basically classified into three types: broadcast, relative address, and flash memory operation commands. The card can be operated by issuing these commands appropriately according to the card state.

(1) Operation of Broadcast Commands

CMD0, CMD1, CMD2, and CMD4 are broadcast commands. These commands and the CMD3 command comprise a sequence assigning relative addresses to individual cards. In this sequence, the CMD output format is open drain, and the command response is wired-OR. During the issuance of this command sequence, the transfer clock frequency should be set to a sufficiently low value.

- All cards are initialized to the idle state by CMD0.
- The operation condition registers (OCR) of all cards are read via wired-OR and cards that cannot operate are deactivated by CMD1.
  The cards that are not deactivated enter the ready state.
- The card identifications (CID) of all cards in the ready state are read via wired-OR by CMD2. Each card compares it's CID and data on the MCCMD, and if they are different, the card aborts the CID output. Only one card in which the CID can be entirely output enters the acknowledge state. When the R2 response is necessary, set CTOCR to H'01.
- A relative address (RCA) is given to the card in the acknowledge state by CMD3.
  The card to which the RCA is given enters the standby state.
- By repeating CMD2 and CMD3, RCAs are given to all cards in the ready state to make them enter the standby state.

(2) Operation of Relative Address Commands

CMD7, CMD9, CMD10, CMD13, CMD15, CMD39, and CMD55 are relative address commands that address the card by RCA. The relative address commands are used to read card administration 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 state. Only the card in the transfer state can execute flash-memory operation commands, other than broadcast 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 require a command response.

Figure 24.4 shows the operational flow for commands that do not require a command response.

- Make settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must be kept driven until the end bit output is completed.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or by the command transmit end interrupt (CMDI).

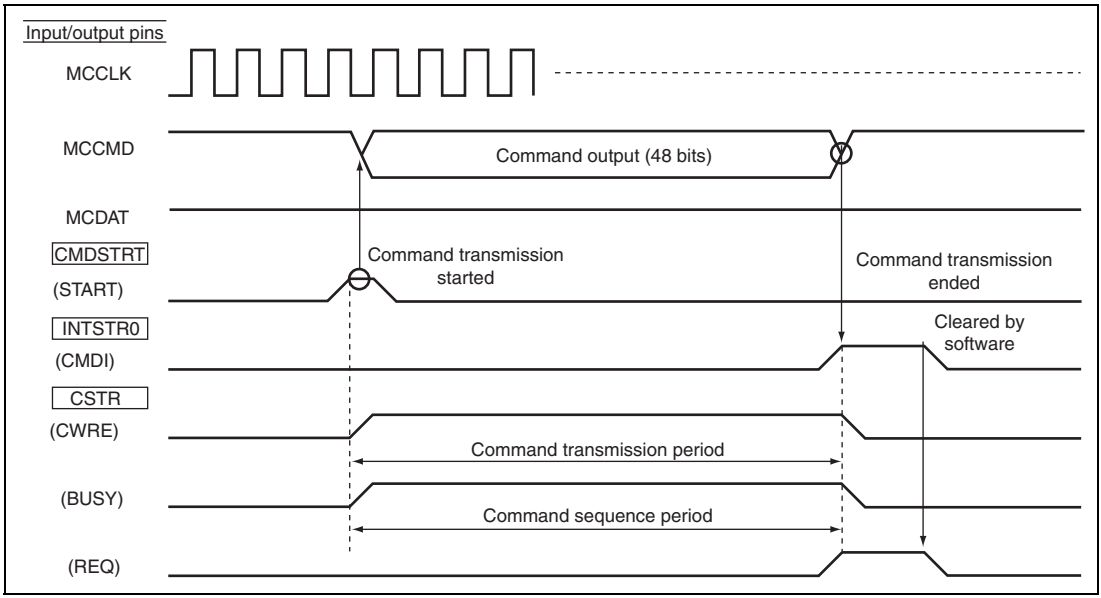


Figure 24.3 Example of Command Sequence for Commands Not Requiring Command Response

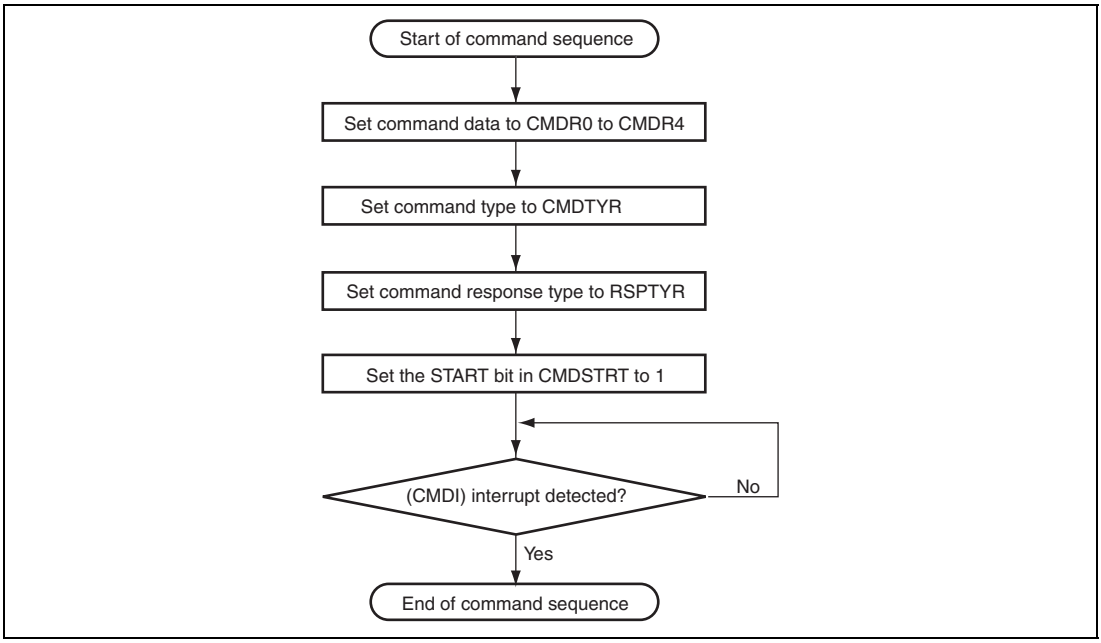


Figure 24.4 Example of Operational Flow for Commands Not Requiring Command Response

(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 transfer using command arguments and command responses. For a command that is related to time-consuming processing such as flash memory write/erase, the card indicates the data busy state via the MCDAT.

Figures 24.5 and 24.6 show examples of the command sequence for commands without data 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 interrupt (CMDI).

- The command response is received from the card.
  If the card returns no command response, the command response is detected by the command timeout error (CTERI).
- The end of the command sequence is detected by poling the BUSY flag in CSTR or by the command response receive end interrupt (CRPI).
- Check whether the state is data busy through the DTBUSY bit in CSTR. If data busy is detected, the end of the data busy state is then detected through the data busy end interrupt (DBSYI).
- Write the CMDOFF bit to 1, if a CRC error (CRCERI) or a command timeout error (CTERI) occurs.
- The MCCMD and MCDAT pins go to the high impedance state when the MMCIF and the MMC card do not drive the bus and the input level of these pins are high because they are pulled-up internally.

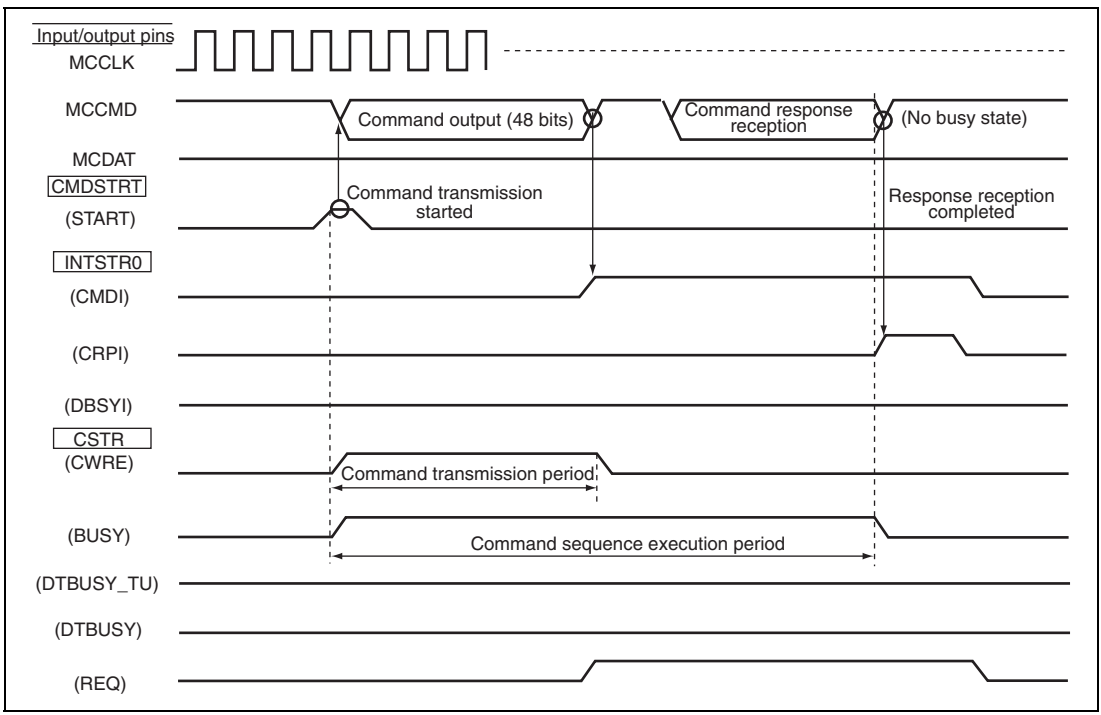


Figure 24.5 Example of Command Sequence for Commands without Data Transfer (No Data Busy State)

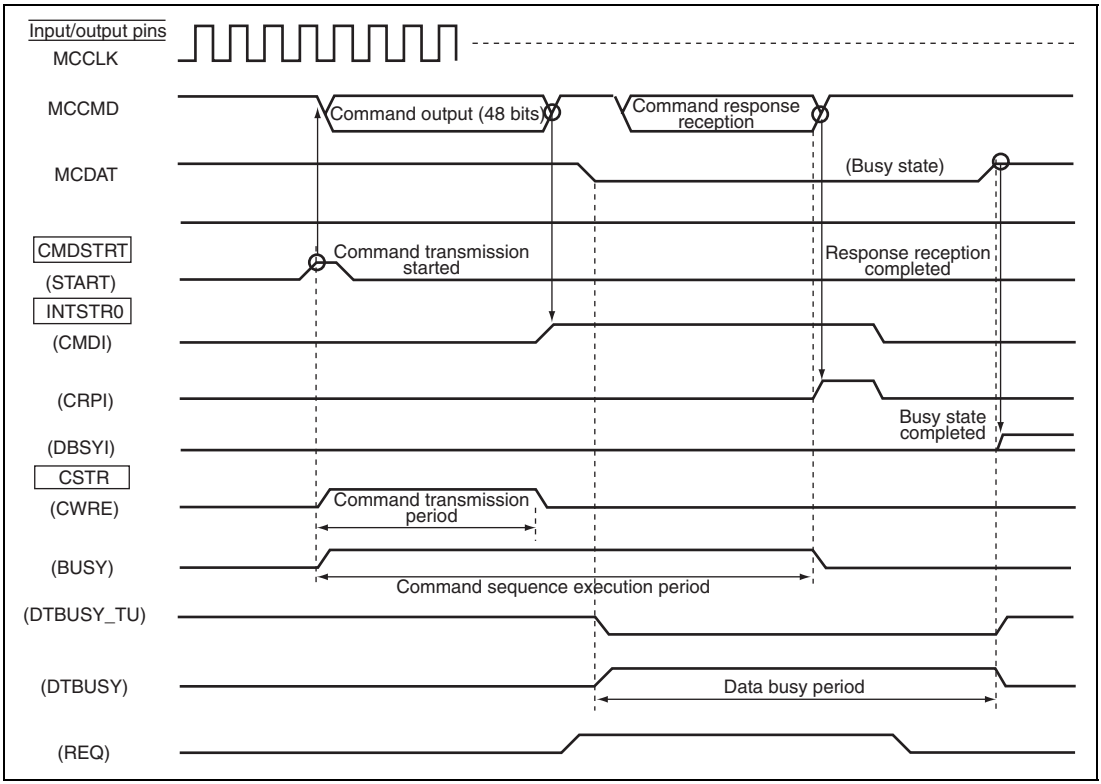


Figure 24.6 Example of Command Sequence for Commands without Data Transfer (with Data Busy State)

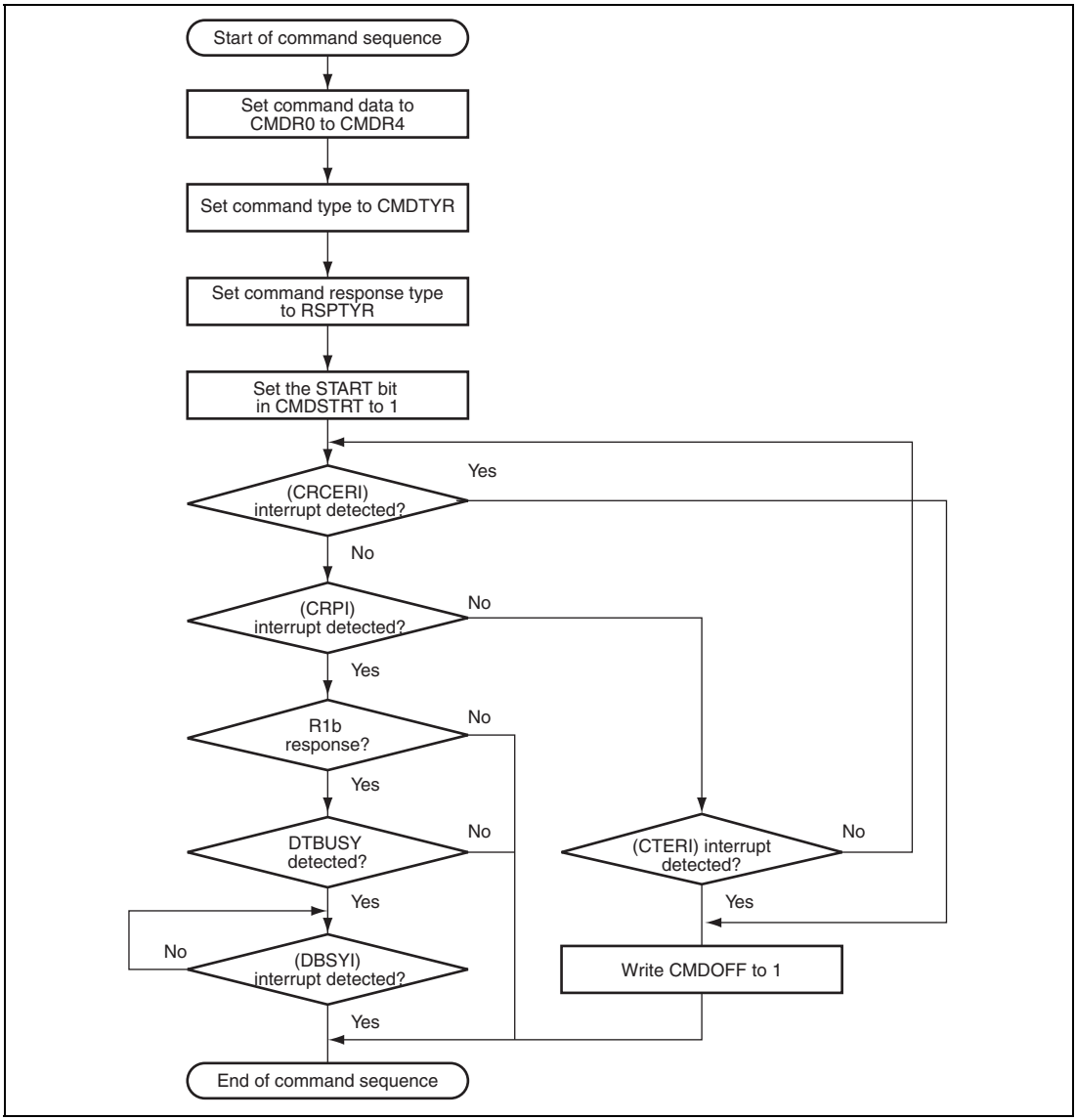


Figure 24.7 Example of Operational Flow for Commands without Data Transfer

(5) Commands with Read Data

Flash memory operation commands include a number of commands involving read data. Such commands confirm the card status by the command argument and command response, and receive card information and flash memory data from the MCDAT pin.

In multiple block transfer, two transfer methods can be used; one is open-ended and another one is pre-defined. Open-ended operation is suspended for each block transfer and an instruction to continue or end the command sequence is waited for. For pre-defined operation, the block number of the transmission is set before transfer.

When the FIFO is full between blocks in multiple block transfer, the command sequence is suspended. Once the command sequence is suspended, process the data in FIFO if necessary before allowing the command sequence to continue.

Note: In multiple block transfer, when the command sequence is ended (the CMDOFF bit is written to 1) before command response reception (CRPI), the command response may not be received correctly. Therefore, to receive the command response correctly, the command sequence must be continued (set the RD_CONT bit to 1) until the command response reception ends.

Figures 24.8 to 24.11 show examples of the command sequence for commands with read data.

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 be kept driven until the end bit output is completed.
  Command transmission completion can be confirmed by the command transmit end interrupt (CMDI).
- The command response is received from the card.
  If the card does not return the command response, the command response is detected by the 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 are detected by the data transfer end interrupt (DTI) and FIFO full interrupt (FFI), respectively. To continue the command sequence, the RD_CONTI bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and CMD12 should be issued. Unless the sequence is suspended in pre-defined multiple block transfer, CMD12 is not needed.

- The end of the command sequence is detected by poling the BUSY flag in CSTR, by the data transfer end interrupt (DTI) or pre-defined multiple block transfer end (BTI).
- Write the CMDOFF bit to 1 if a CRC error (CRCERI) or a command timeout error (CTERI) occurs in the command response reception.
- Clear the FIFO by writing the CMDOFF bit to 1, when CRC error (CRCERI) and data timeout error (DTERI) occurs in the read data reception.

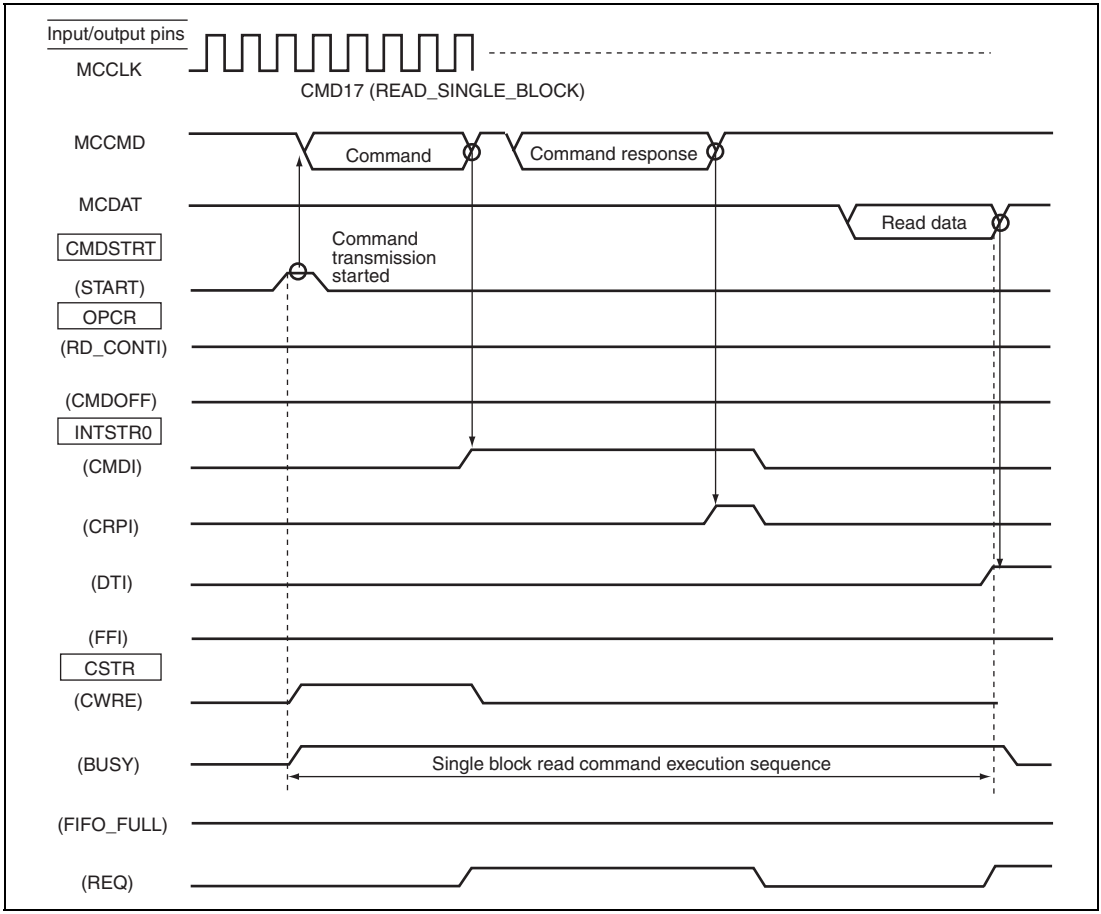


Figure 24.8 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size)

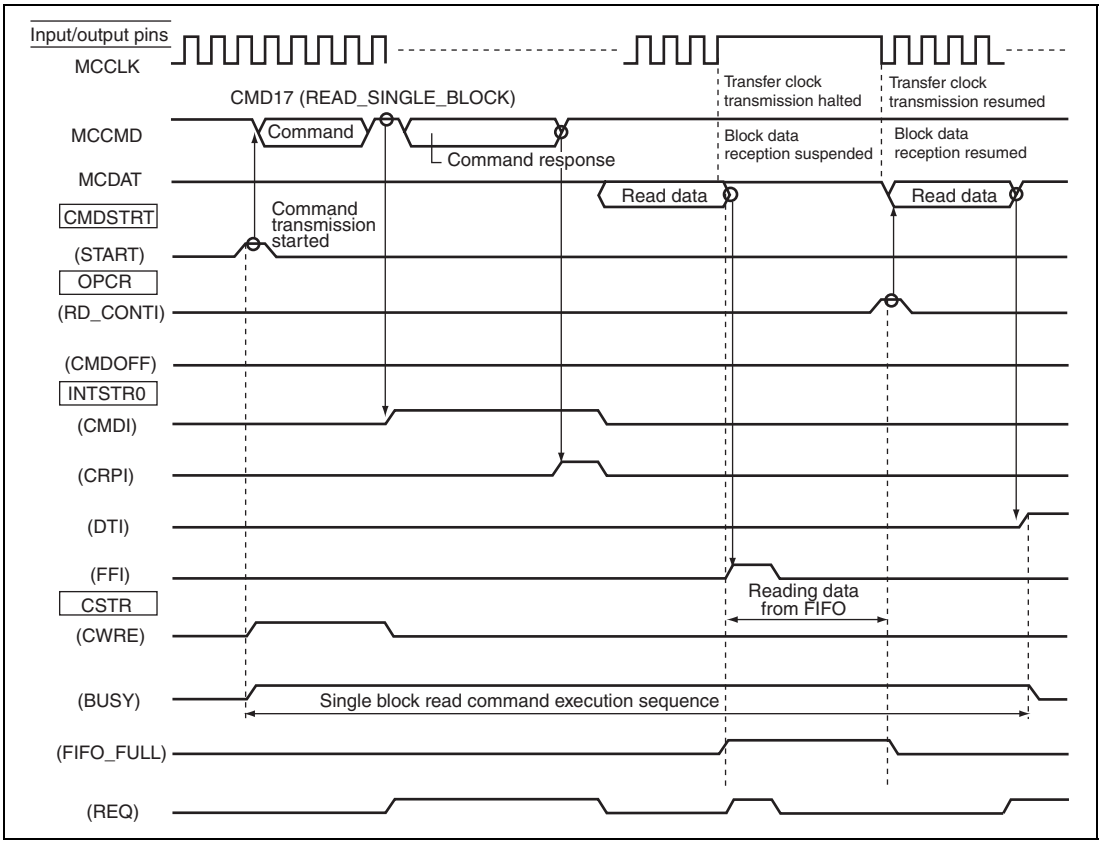


Figure 24.9 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size)

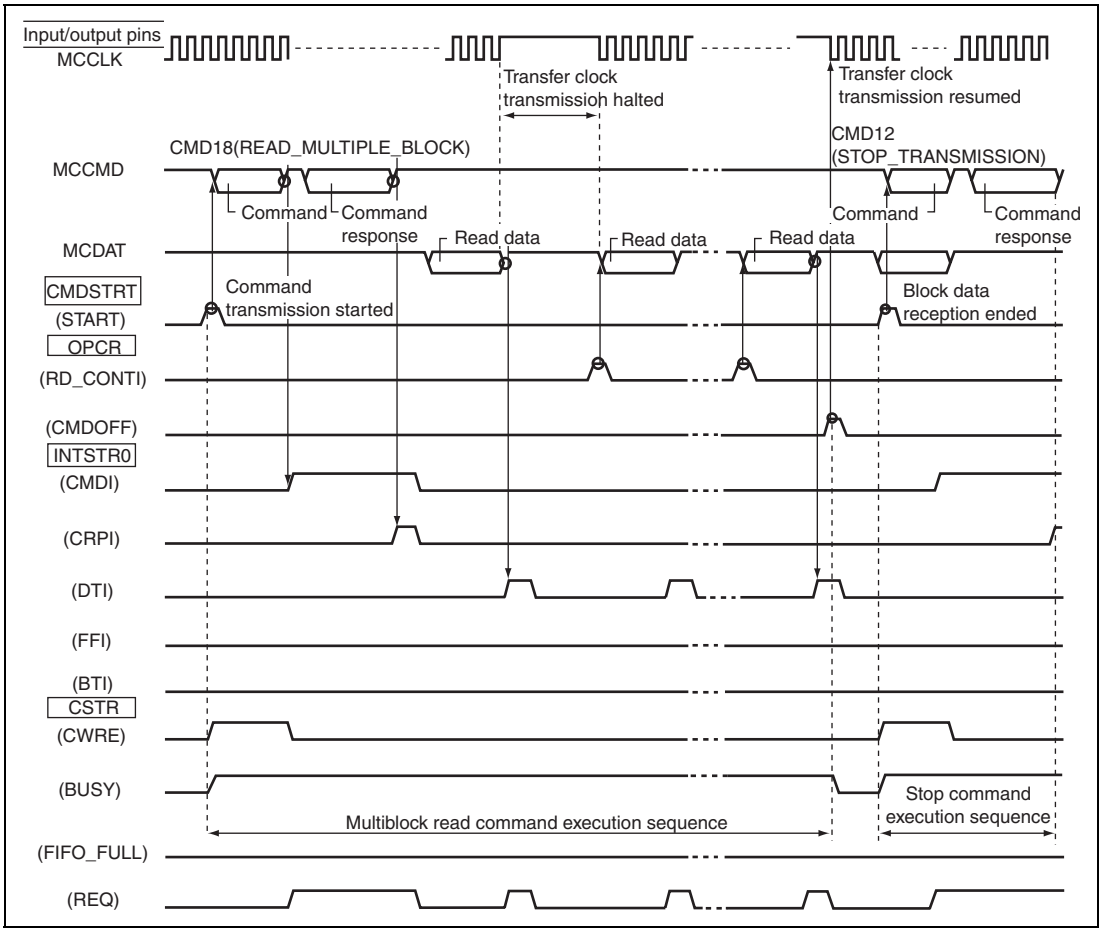


Figure 24.10 Example of Command Sequence for Commands with Read Data (Multiple Block Transfer)

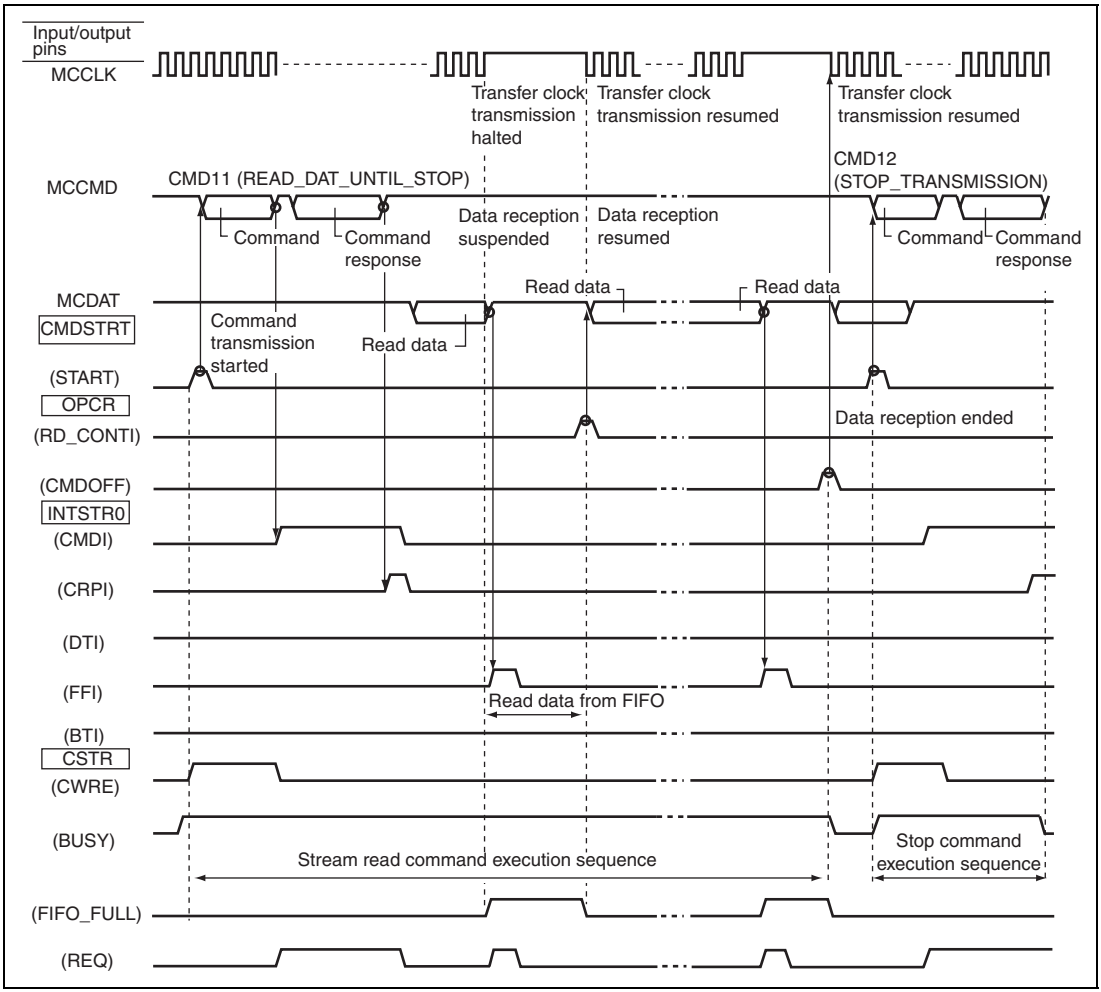


Figure 24.11 Example of Command Sequence for Commands with Read Data (Stream Transfer)

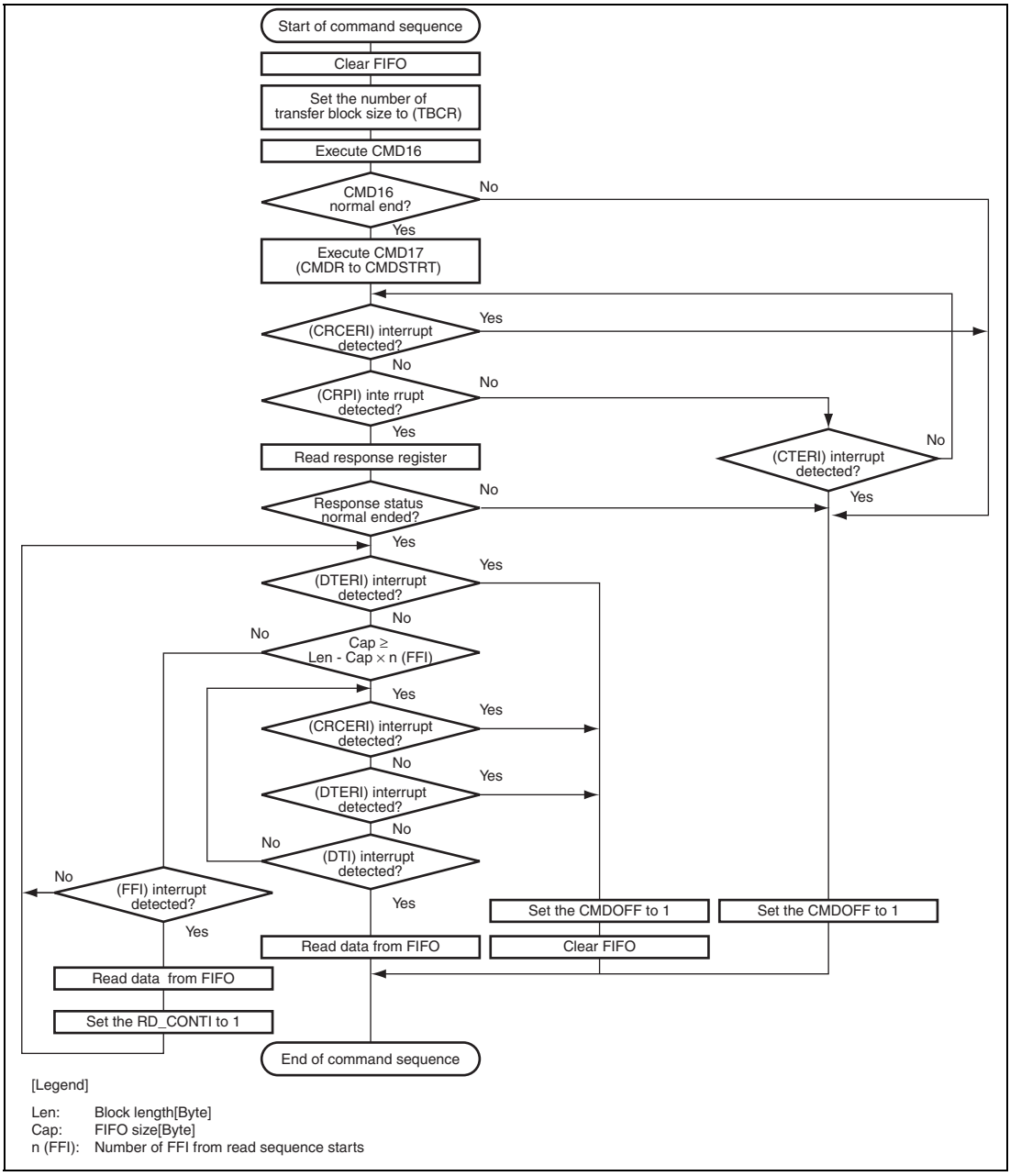


Figure 24.12 Example of Operational Flow for Commands with Read Data (Single Block Transfer)

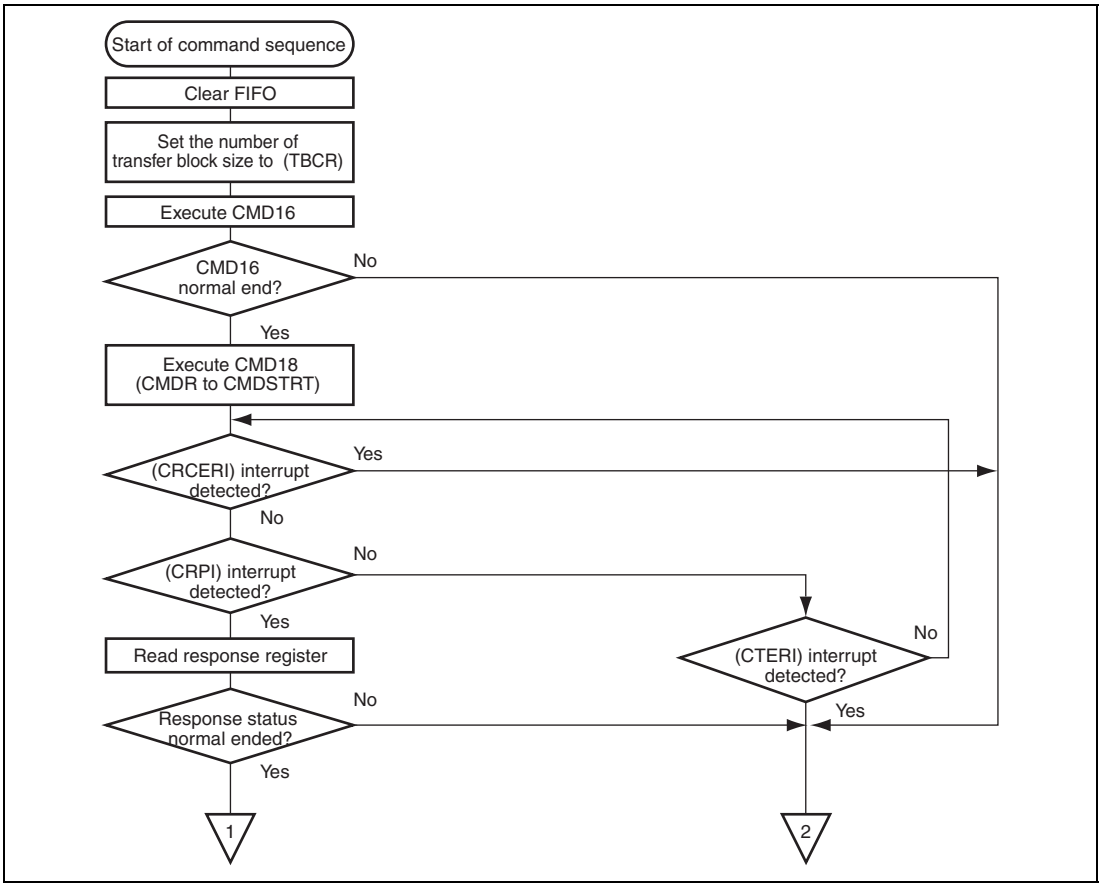


Figure 24.13 Example of Operational Flow for Commands with Read Data (1) (Open-ended Multiple Block Transfer)

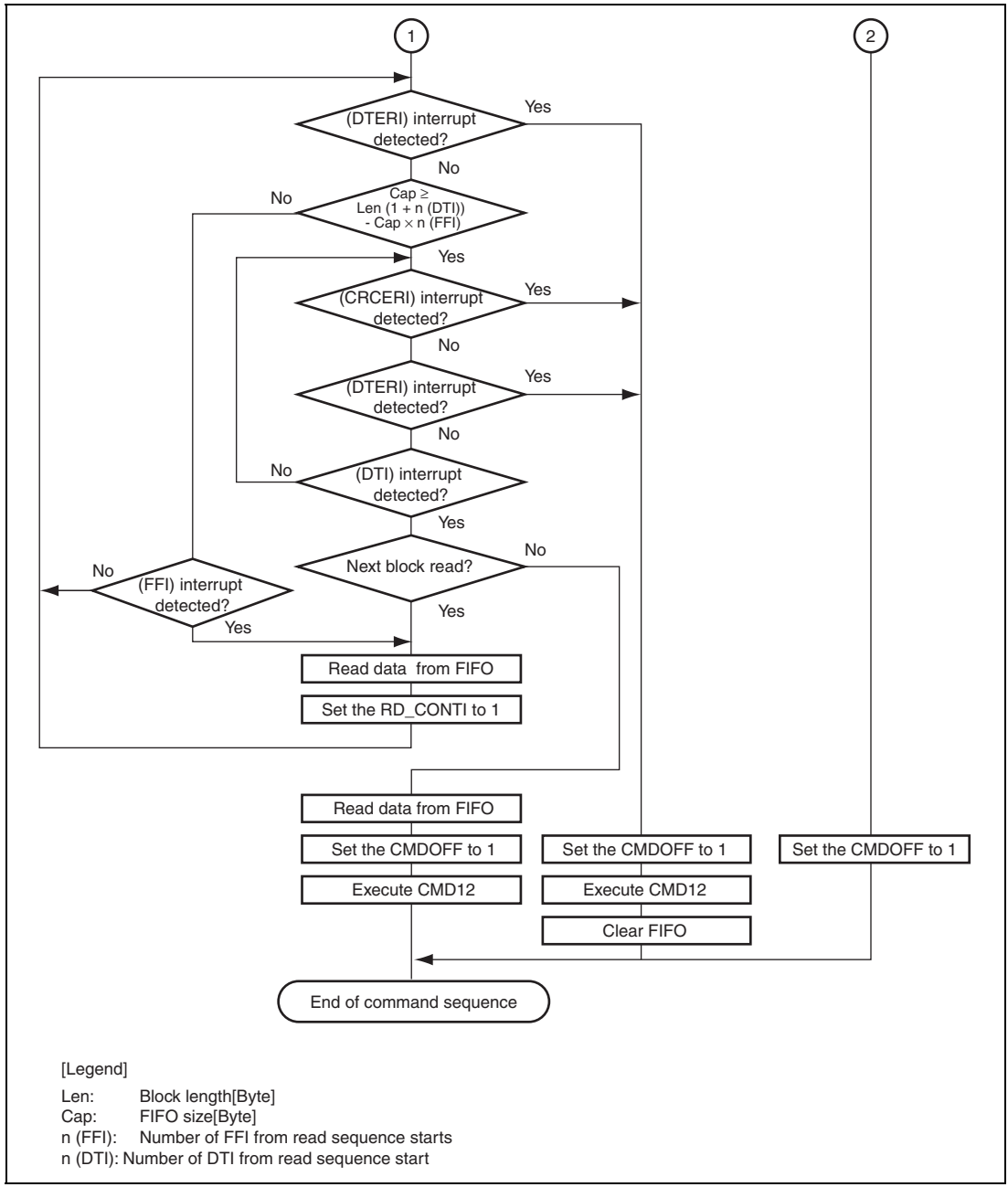


Figure 24.13 Example of Operational Flow for Commands with Read Data (2) (Open-ended Multiple Block Transfer)

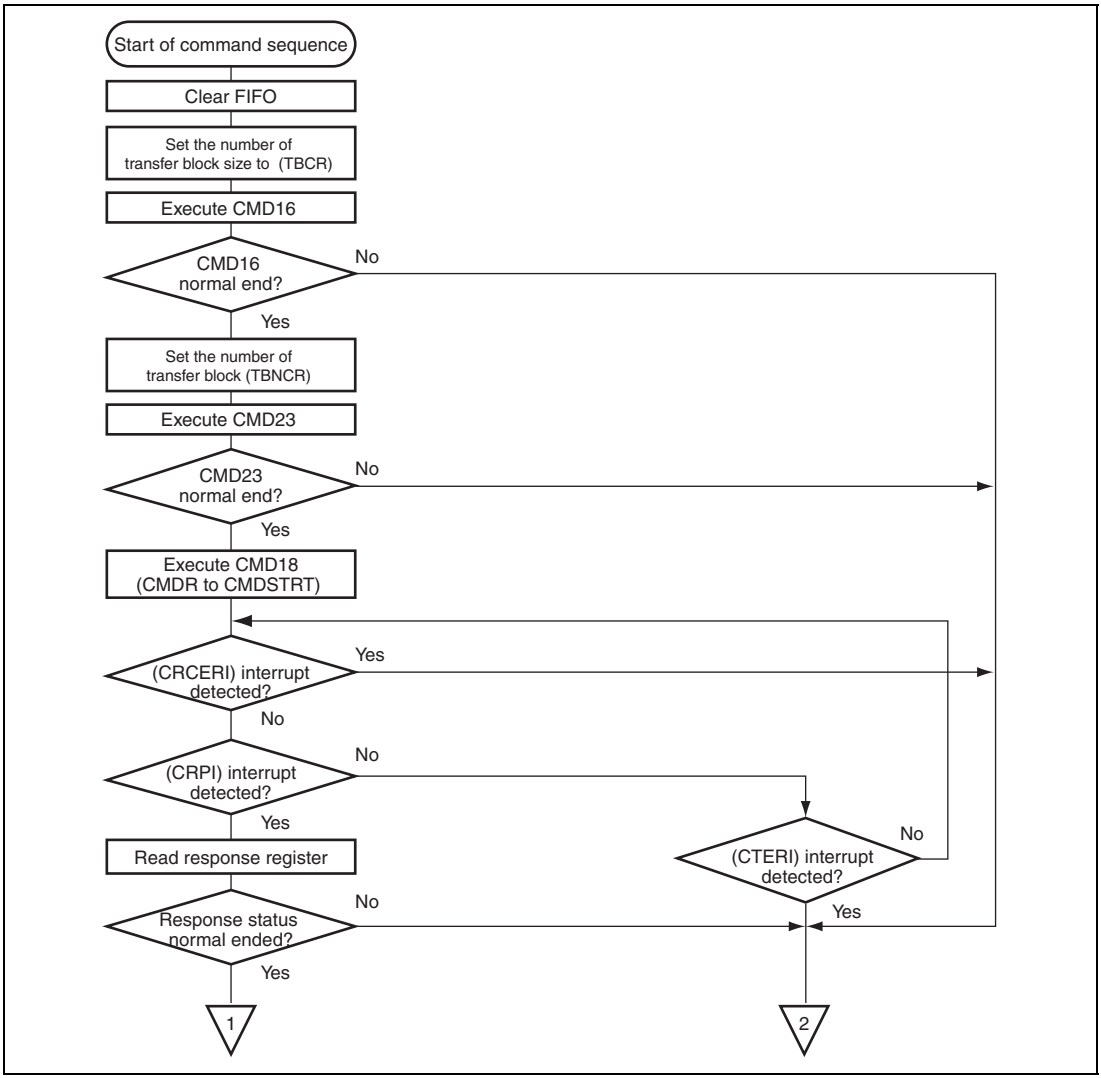


Figure 24.13 Example of Operational Flow for Commands with Read Data (3) (Pre-defined Multiple Block Transfer)

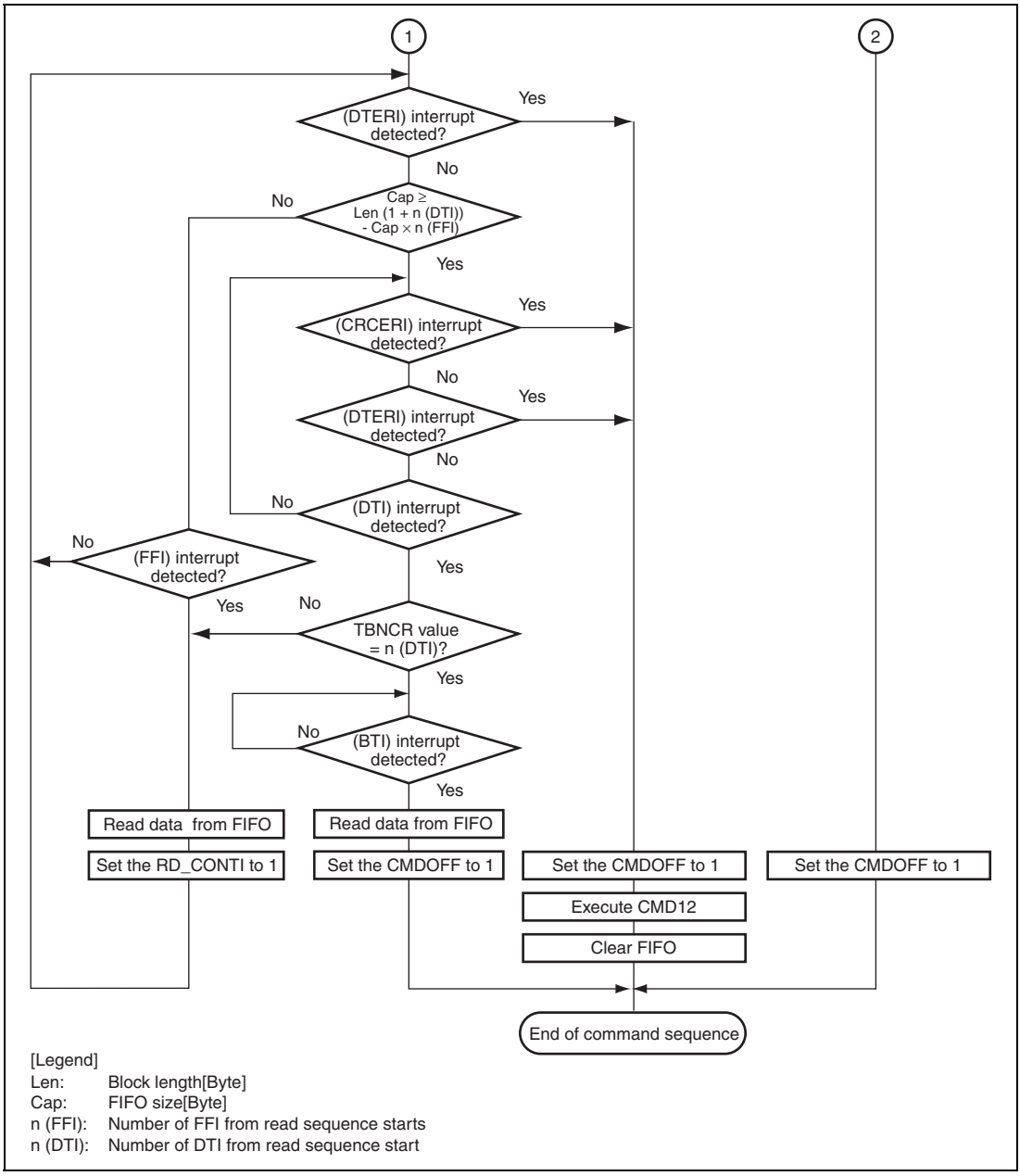


Figure 24.13 Example of Operational Flow for Commands with Read Data (4) (Pre-defined Multiple Block Transfer)

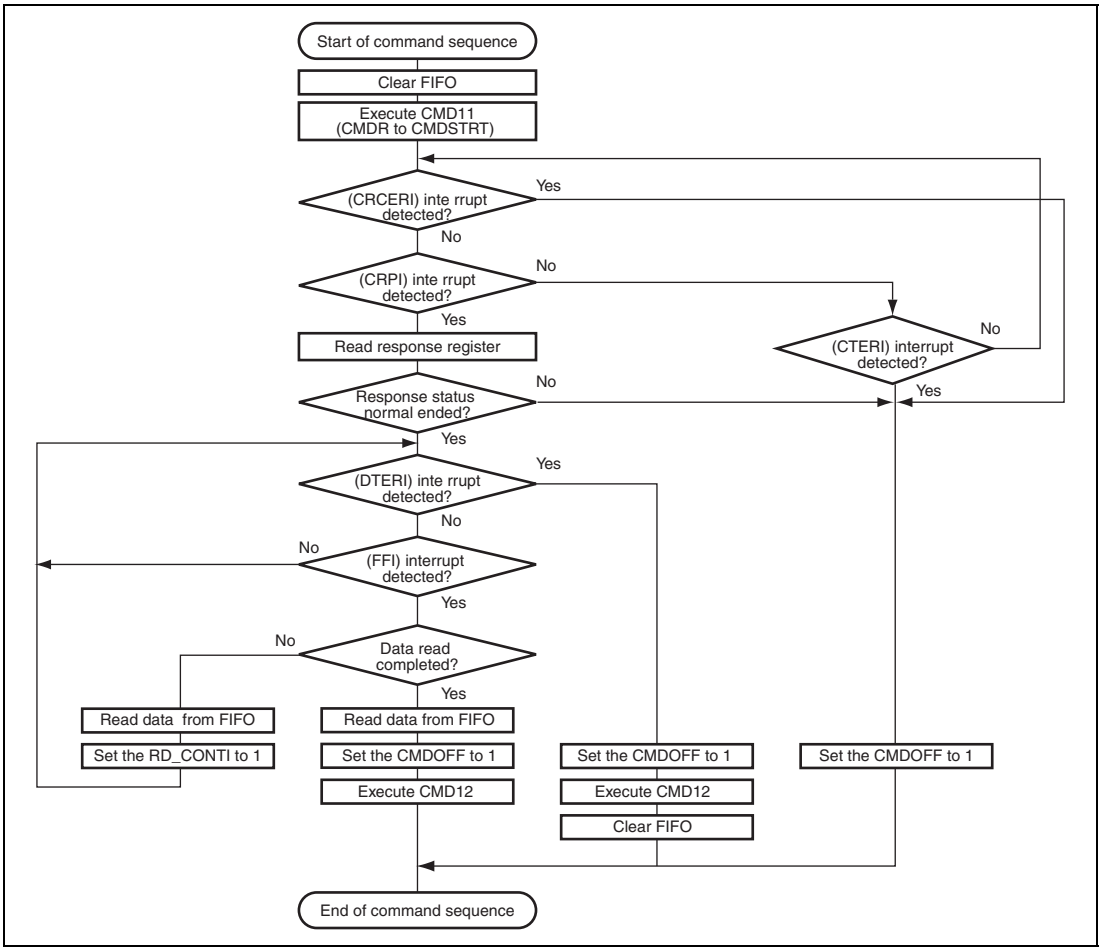


Figure 24.14 Example of Operational Flow for Commands with Read Data (Stream Transfer)

(6) Commands with Write Data

Flash memory operation commands include a number of commands involving write data. Such commands confirm the card status by the command argument and command response, and transmit card information and flash memory data via the MCDAT pin. For a command that is related to time-consuming processing such as flash memory write, the card indicates the data busy state via the MCDAT pin.

In multiple block transfer, two transfer methods can be used; one is open-ended and the other is pre-defined. Open-ended operation is suspended for each block transfer and an instruction to continue or end the command sequence is waited for. For pre-defined operation, the block number of the transmission is set before transfer.

When the FIFO is full between blocks in multiple block transfer, the command sequence is suspended. Once the command sequence is suspended, process the data in FIFO if necessary before allowing the command sequence to continue.

Figures 24.15 to 24.18 show examples of the command sequence for commands with write data.

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 must be kept driven until the end bit output is completed.
- Command transmission completion can be confirmed by the command transmit end interrupt (CMDI).
- The command response is received from the card.
- If the card returns no command response, the command response is detected by the command 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 kept driven until the end bit output is completed.
- Inter-block suspension in multiple block transfer and suspension according to the FIFO empty 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 the DATAEN bit in OPCR to 1. To end the command sequence, set the CMDOFF bit in OPCR to 1 and issue CMD12. Unless the sequence is suspended in pre-defined multiple block transfer, CMD12 is not needed.

- The end of the command sequence is detected by poling the BUSY flag in CSTR, data transfer end interrupt (DTI), data response interrupt (DRPI), or pre-defined multiple block transfer end (BTI).
- The data busy state is checked through DTBUSY in CSTR. If the card is in data busy state, the end of the data busy state is detected by the data busy end interrupt (DBSYI).
- Write the CMDOFF bit to 1 if a CRC error (CRCERI) or a command timeout error (CTERI) occurs in the command response reception.
- Write the CMDOFF bit to 1 if a CRC error (CRCERI) or a data timeout error (DTERI) occurs in the write data transmission.

Note: In a write to the card by stream transfer, the MMCIF continues data transfer to the card even after a FIFO empty interrupt is detected. In this case, complete the command sequence after at least 24 transfer clock cycles.

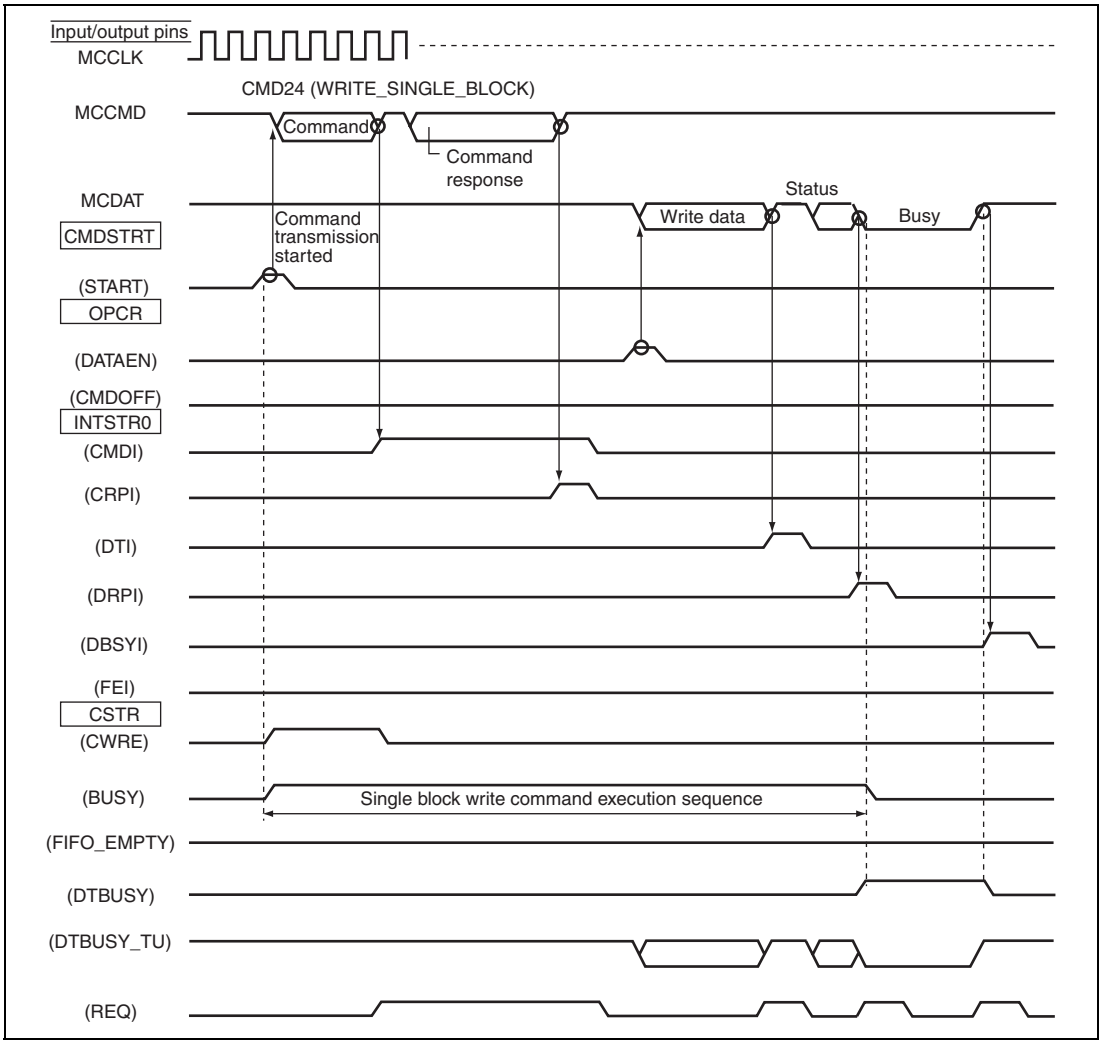


Figure 24.15 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size)

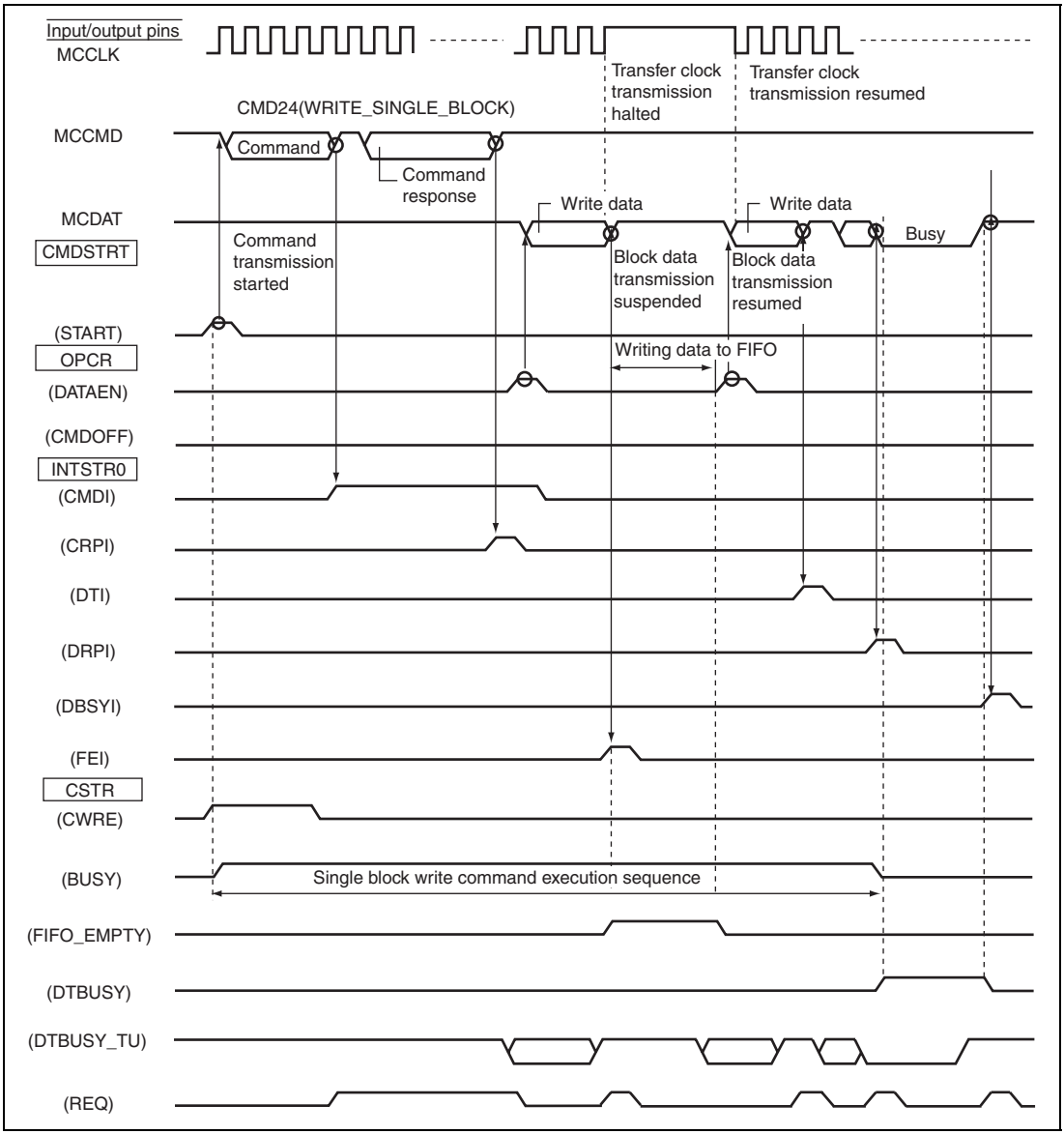


Figure 24.16 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size)

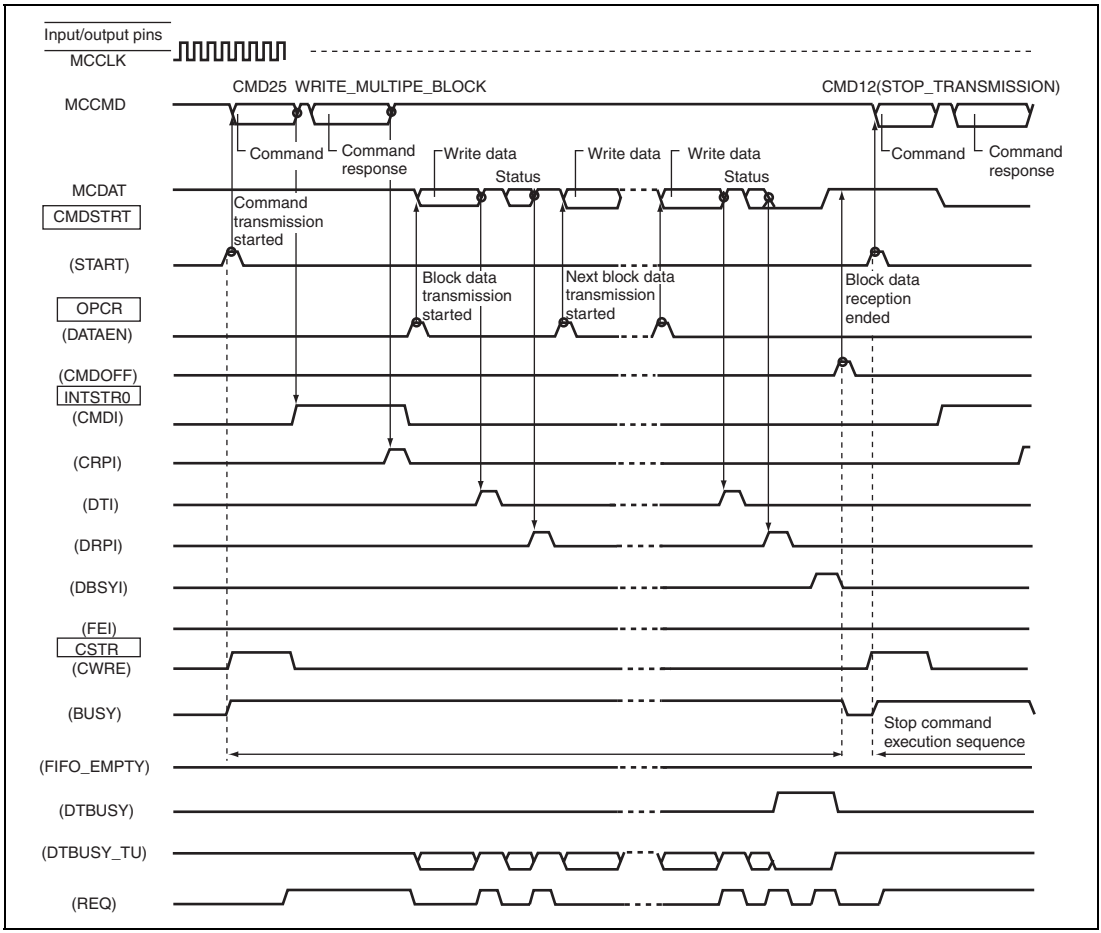


Figure 24.17 Example of Command Sequence for Commands with Write Data (Multiple Block Transfer)

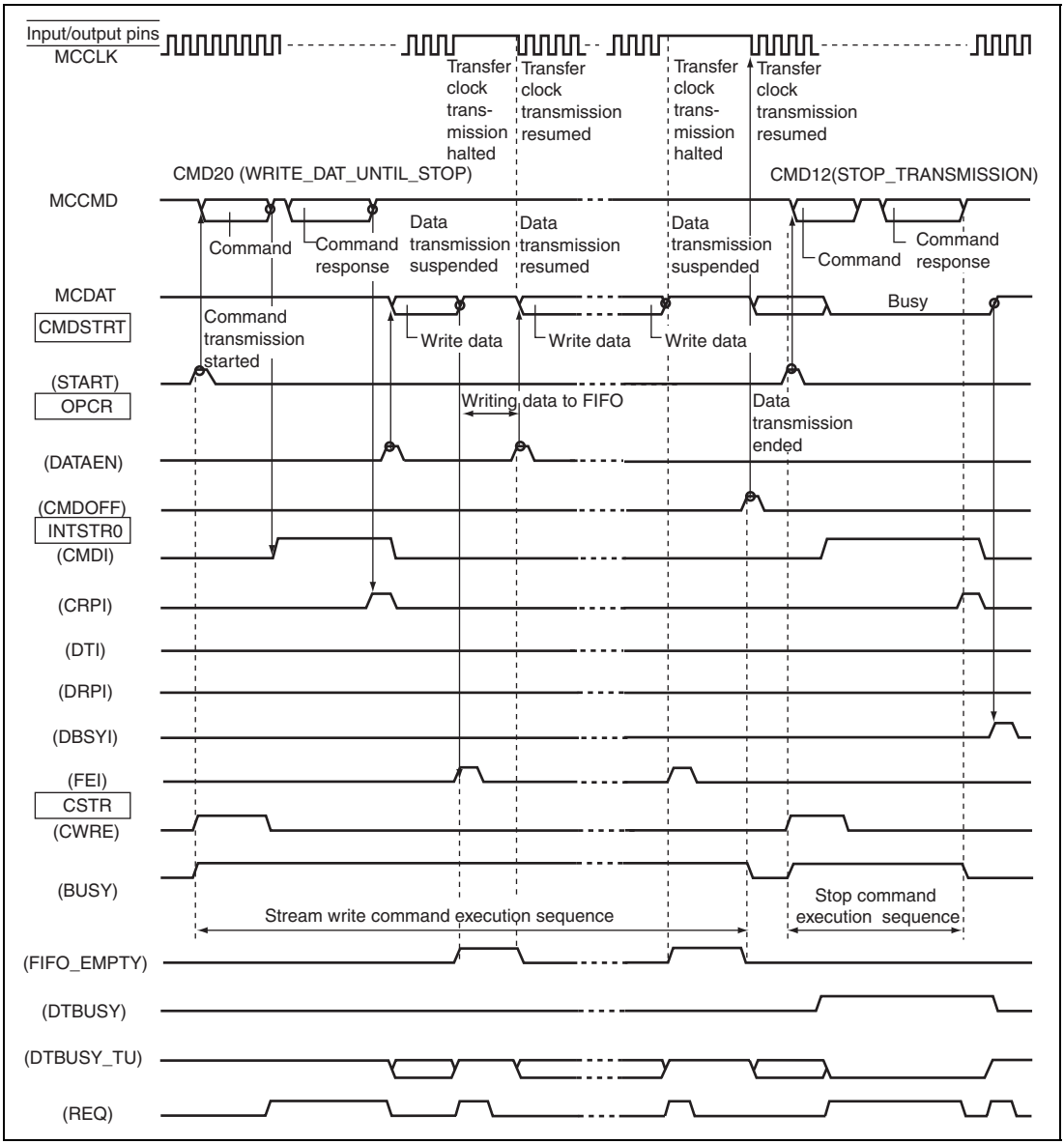


Figure 24.18 Example of Command Sequence for Commands with Write Data (Stream Transfer)

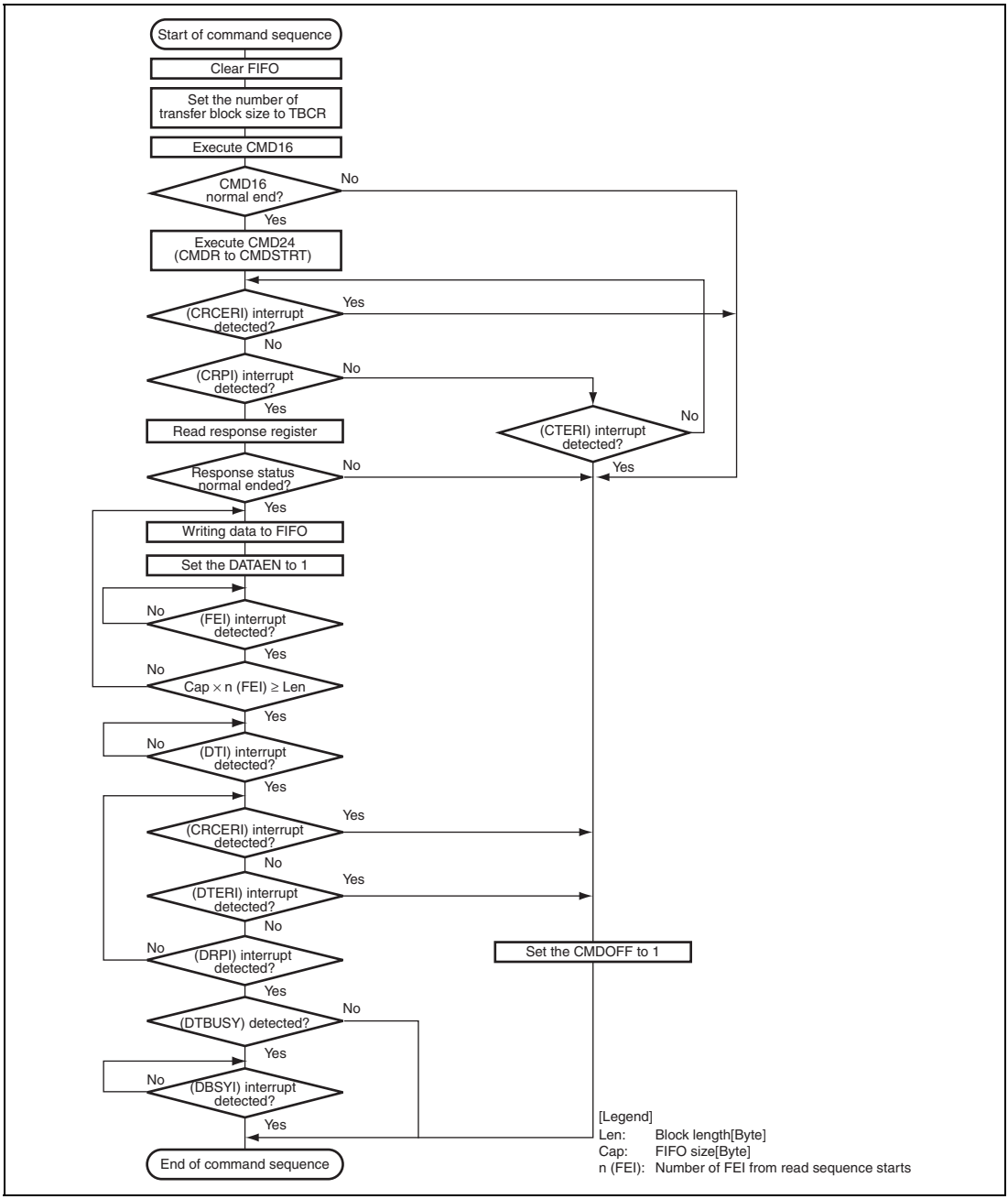


Figure 24.19 Example of Operational Flow for Commands with Write Data (Single Block Transfer)

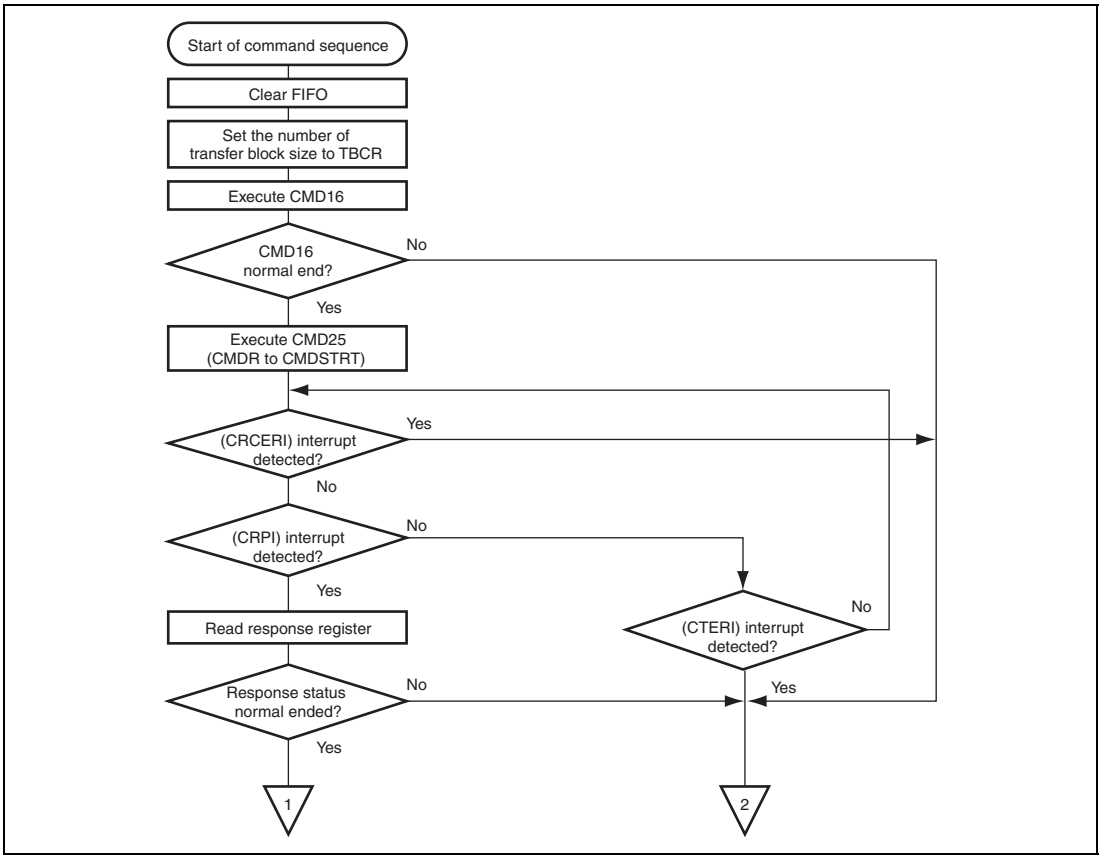
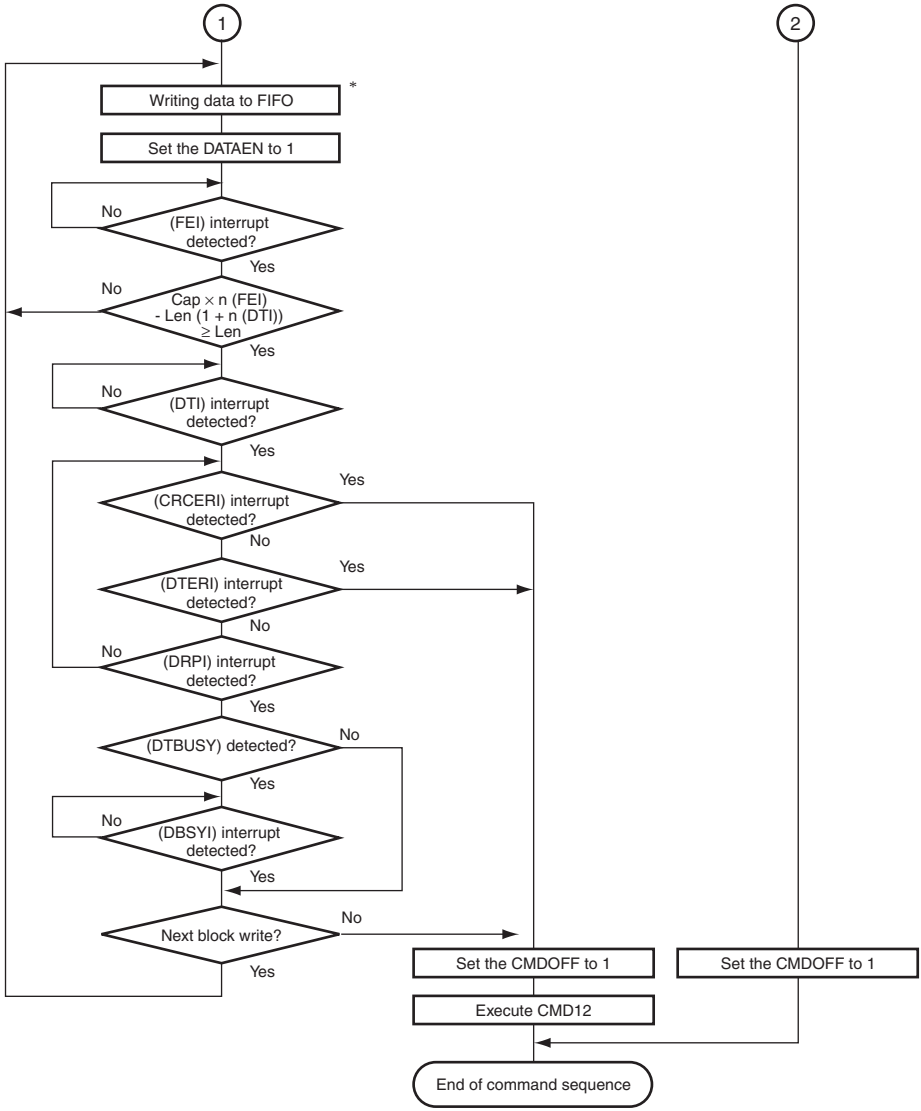


Figure 24.20 Example of Operational Flow for Commands with Write Data (1) (Open-ended Multiple Block Transfer)



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 (2) (Open-ended Multiple Block Transfer)

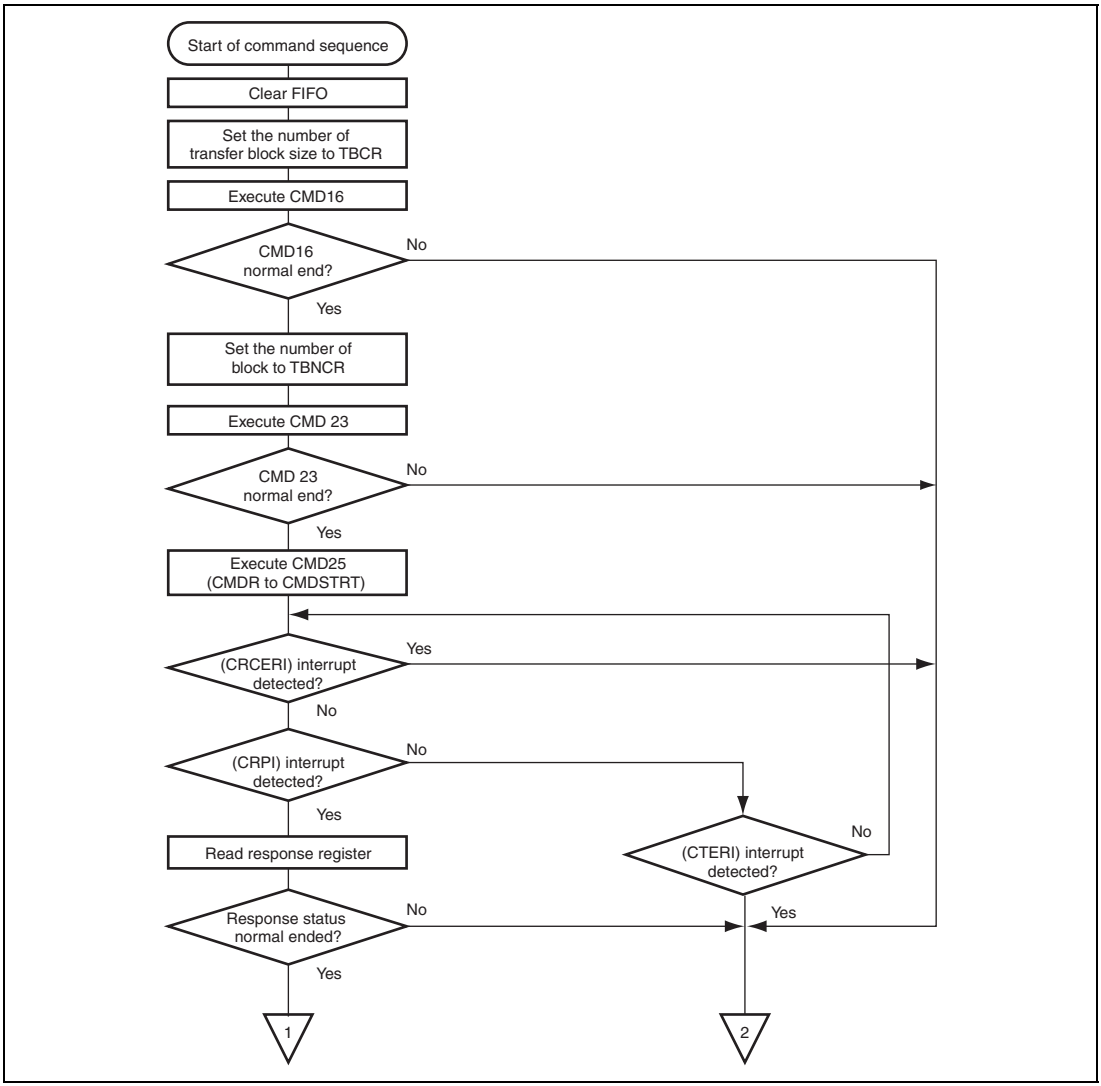
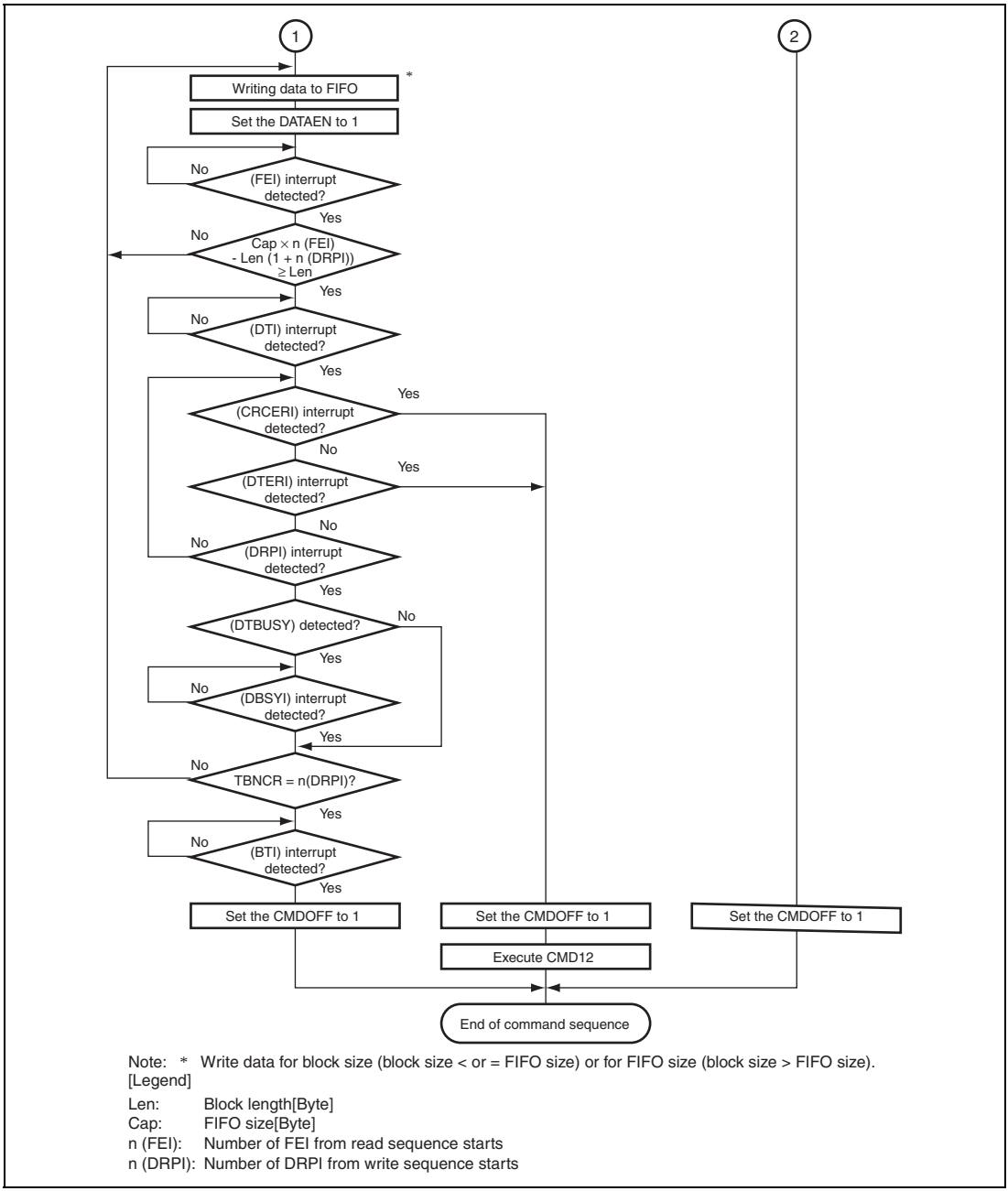


Figure 24.20 Example of Operational Flow for Commands with Write Data (3) (Pre-defined Multiple Block Transfer)



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 (4) (Pre-defined Multiple Block Transfer)

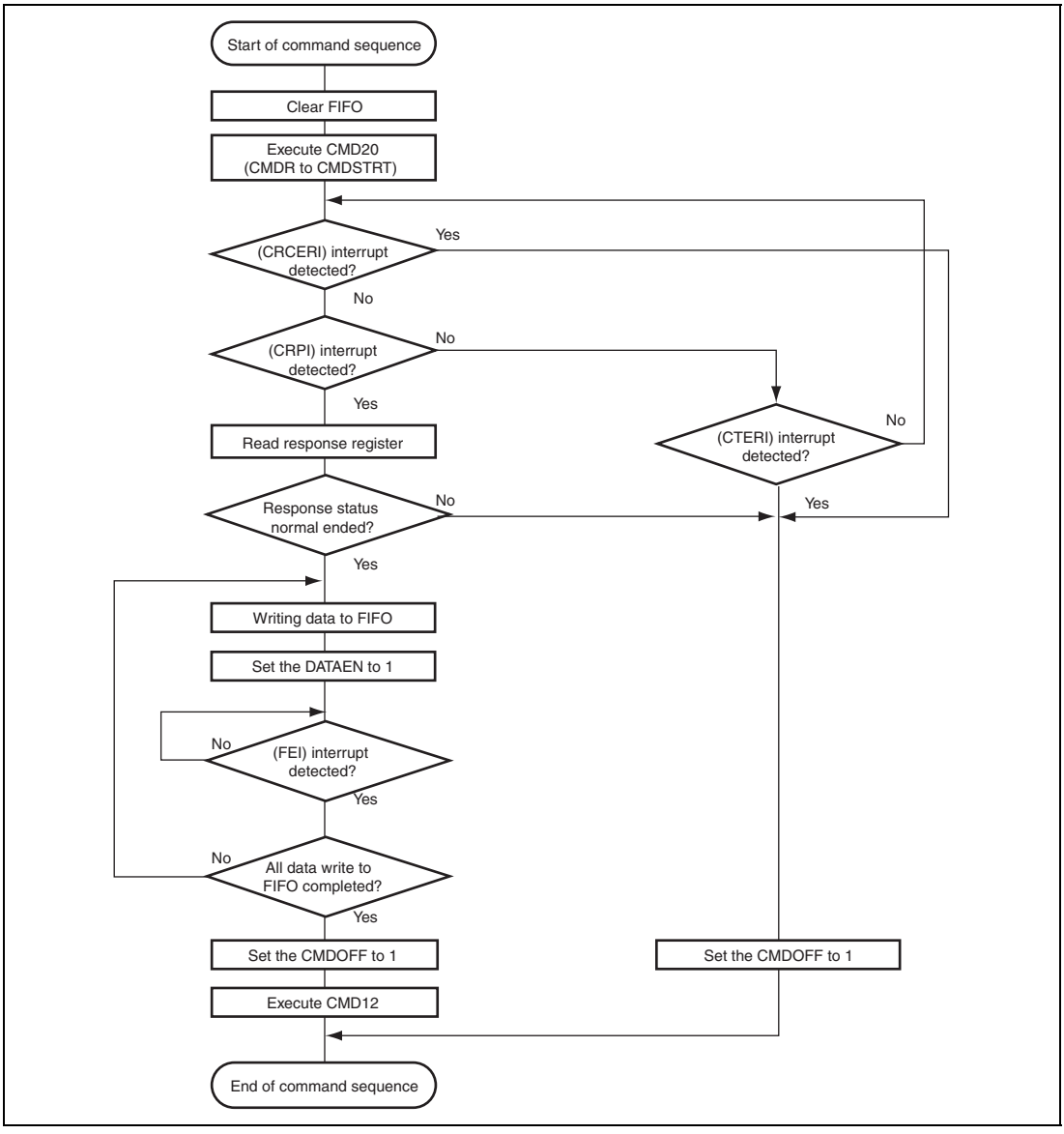


Figure 24.21 Example of Operational Flow for Commands with Write Data (Stream Transfer)

24.5 MMCIF Interrupt Sources

Table 24.7 lists the MMCIF interrupt sources. The interrupt sources are classified into four groups, and four interrupt vectors are assigned. Each interrupt source can be individually enabled by the enable bits in INTCR0 to INTCR2. Disabled interrupt sources do not set the flag.

Table 24.7 MMCIF Interrupt Sources

Name	Interrupt source	Interrupt flag
FSTAT	FIFO empty	FEI
	FIFO full	FFI
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.

24.6 Operations when Using DMA

24.6.1 Operation in Read Sequence

In order to transfer data in FIFO with the DMAC, set MMCIF (DMACR) after setting the DMAC*. Transmit the read command after setting DMACR.

Figure 24.22 to 24.24 shows the operational flow for a read sequence.

- Clear FIFO and make settings in DMACR.
- 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 DMACR to read all data from FIFO.
- Confirm that the DMAC transfer is completed and set the DMAEN bit in DMACR to 0.
- Set the CMDOFF bit to 1 and clear DMACR to H'00 if a CRC error (CRCERI) or a command 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 (CRCERI) or 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 DMACR is set to 1 and normal read is detected after the block transfer end of a pre-defined multiple block transfer. Figure 24.25 shows the operational flow for a pre-defined multiple block read using auto-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 from the card.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or through the pre-defined multiple block transfer end flag (BTI).

- An error in a command sequence (during data reception) is detected through the CRC error flag or data timeout flag. When these flags are detected, set the CMDOFF bit in OPCR to 1, issue CMD12 and suspend the command sequence.
- The data remains in FIFO after the read sequence end. Set the SET[2:0] bits in DMACR to 100 to read all data in FIFO if necessary.
- Confirm the DMA transfer end and clear the DMAEN bit in DMACR to 0.
- Set the CMDOFF bit to 1 and clear DMACR to H'00 if a CRC error (CRCERI) or a command 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 (CRCERI) or a data timeout error (DTERI) occurs in the read data reception.

Note: * In multiple block transfer, when the command sequence is ended (the CMDOFF bit is written to 1) before command response reception (CRPI), the command response may not be received correctly. Therefore, to receive the command response correctly, the command sequence must be continued (set the RD_CONT bit to 1) until the command response reception ends.
Access from the DMAC to FIFO must be done in bytes or words.

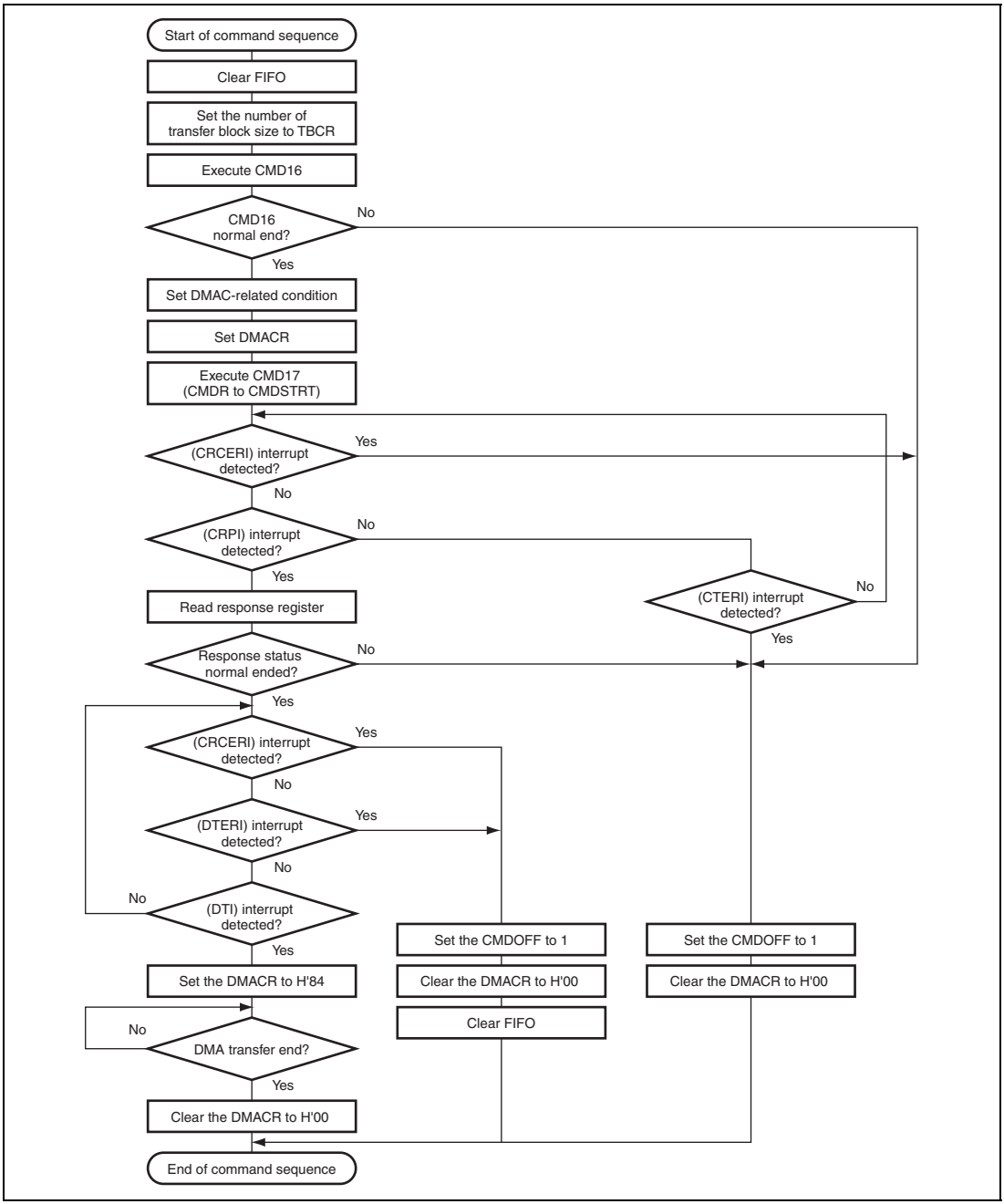


Figure 24.22 Example of Read Sequence Flow (Single Block Transfer)

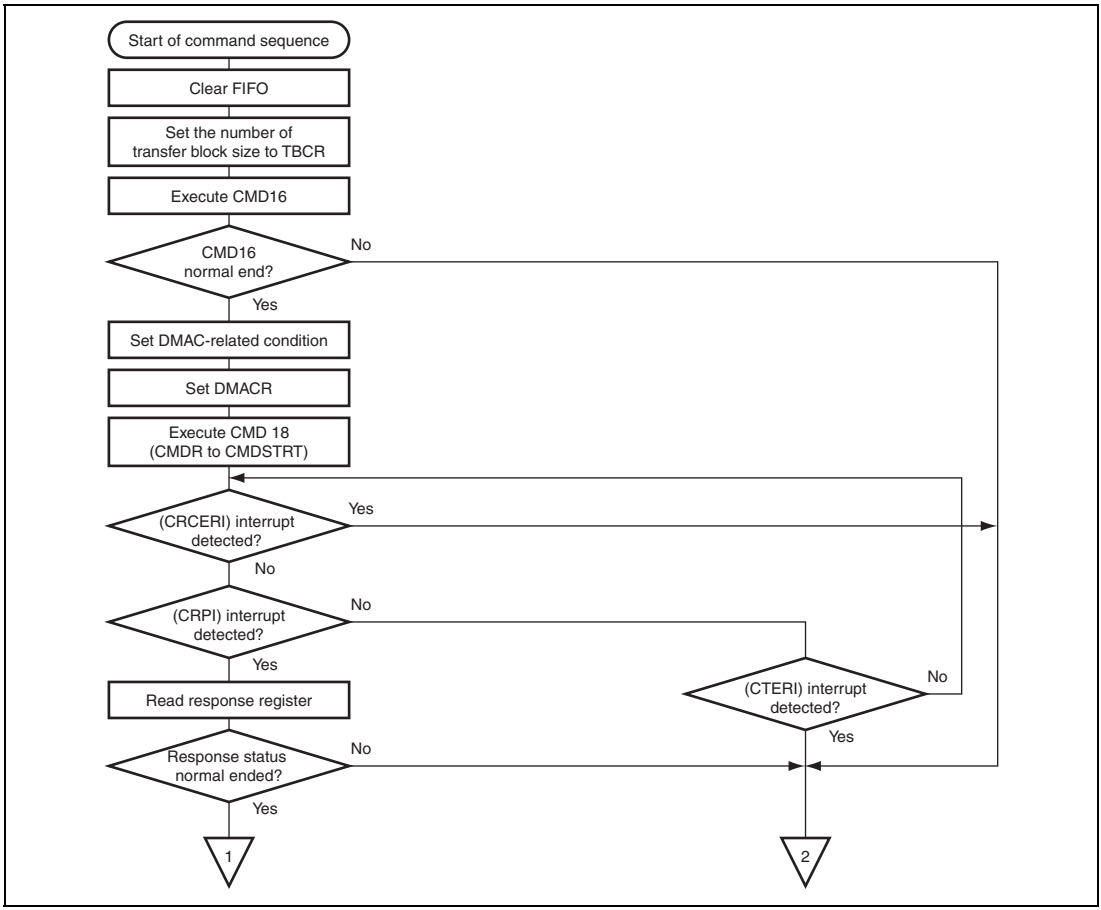


Figure 24.23 Example of Read Sequence Flow (1) (Open-ended Multiple Block Transfer)

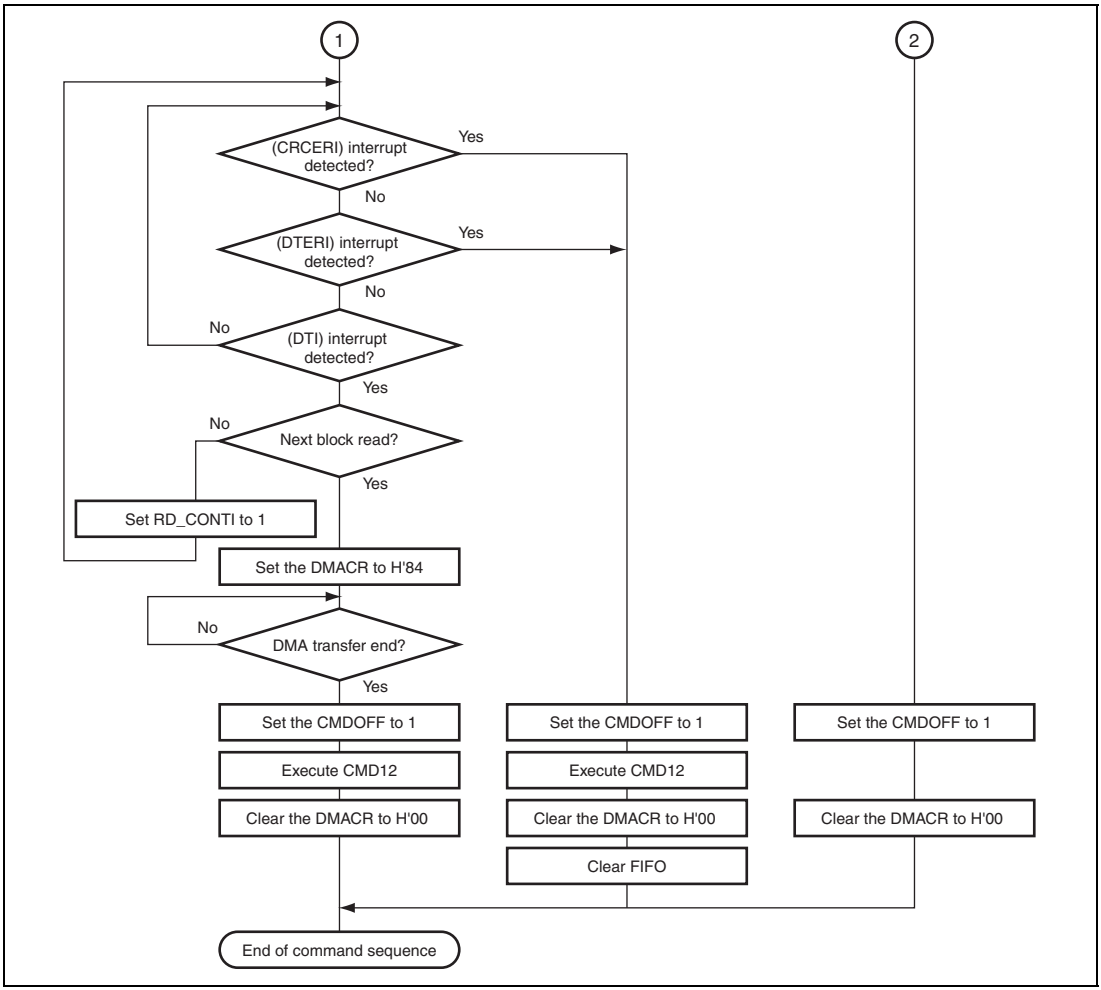


Figure 24.23 Example of Read Sequence Flow (2) (Open-ended Multiple Block Transfer)

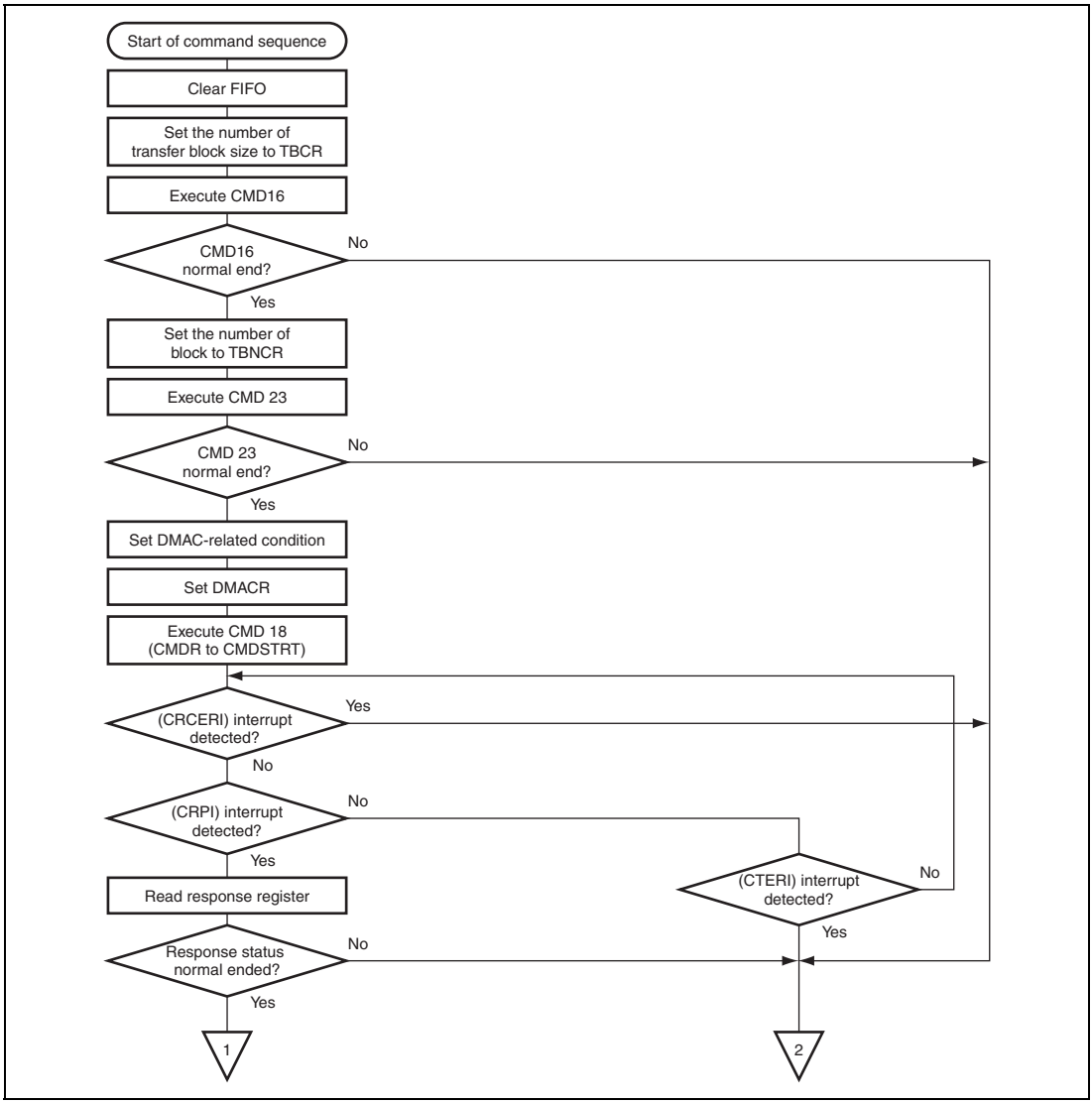


Figure 24.23 Example of Read Sequence Flow (3) (Pre-defined Multiple Block Transfer)

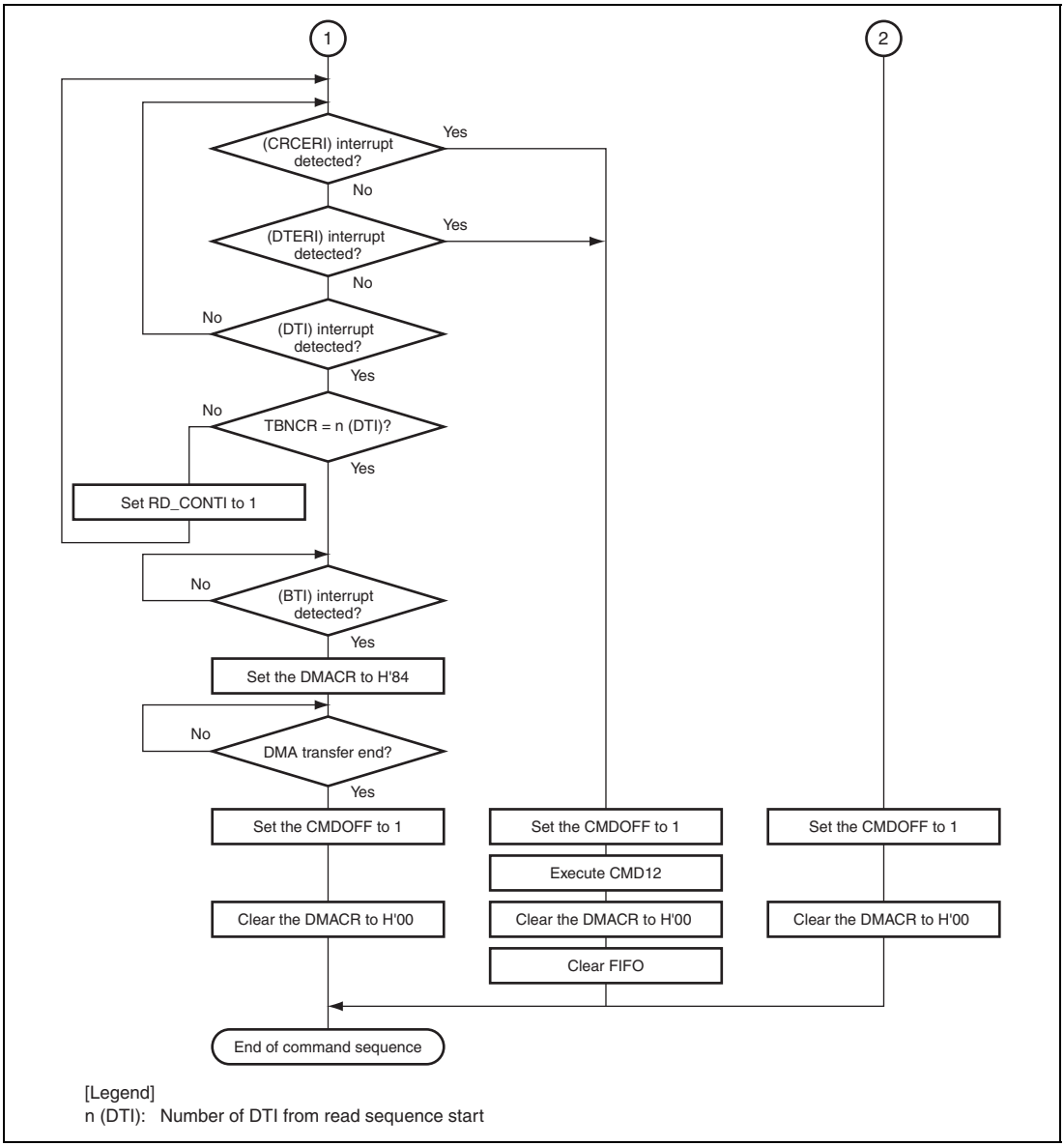


Figure 24.23 Example of Read Sequence Flow (4) (Pre-defined Multiple Block Transfer)

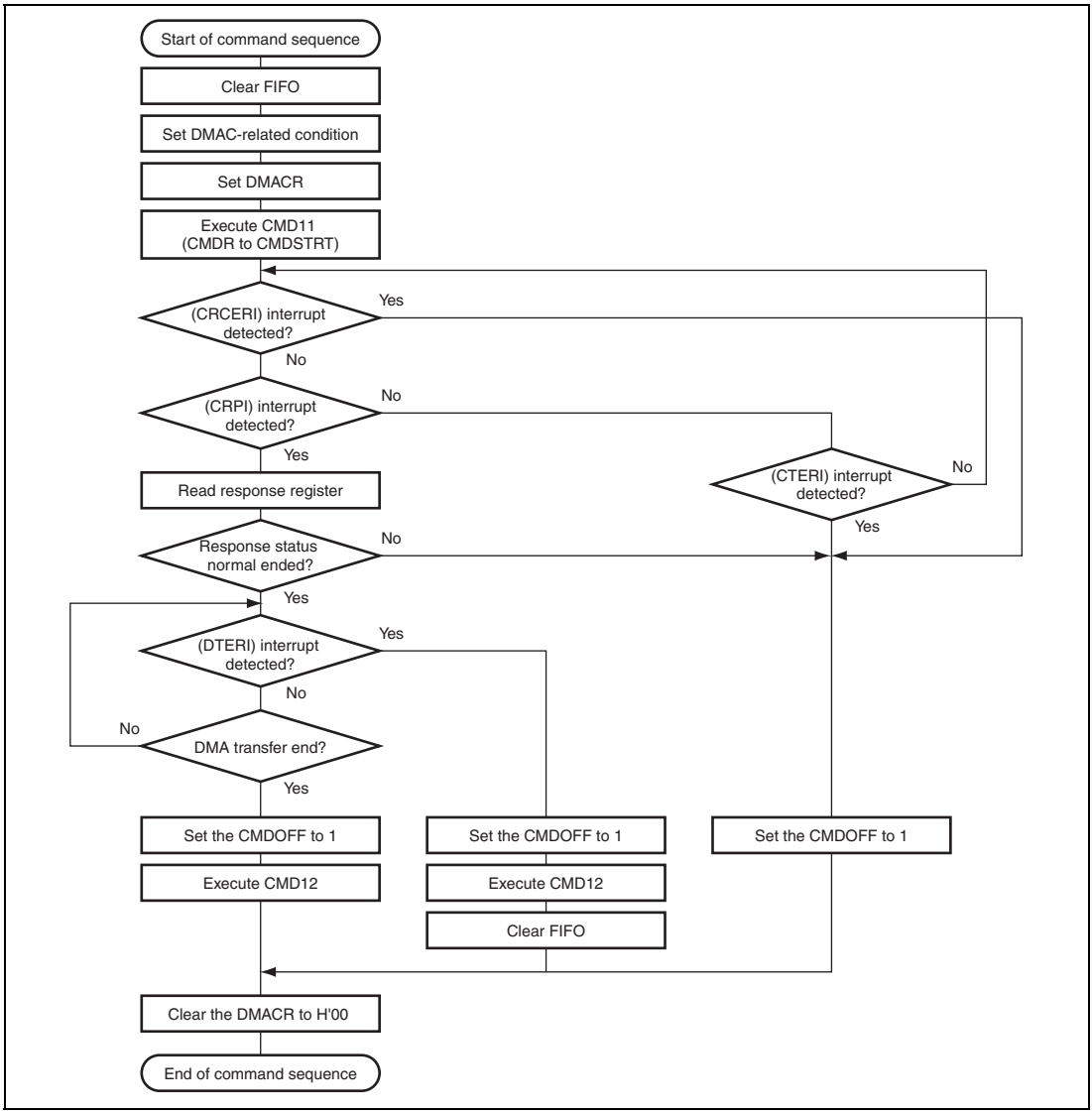


Figure 24.24 Example of Operational Flow for Stream Read Transfer

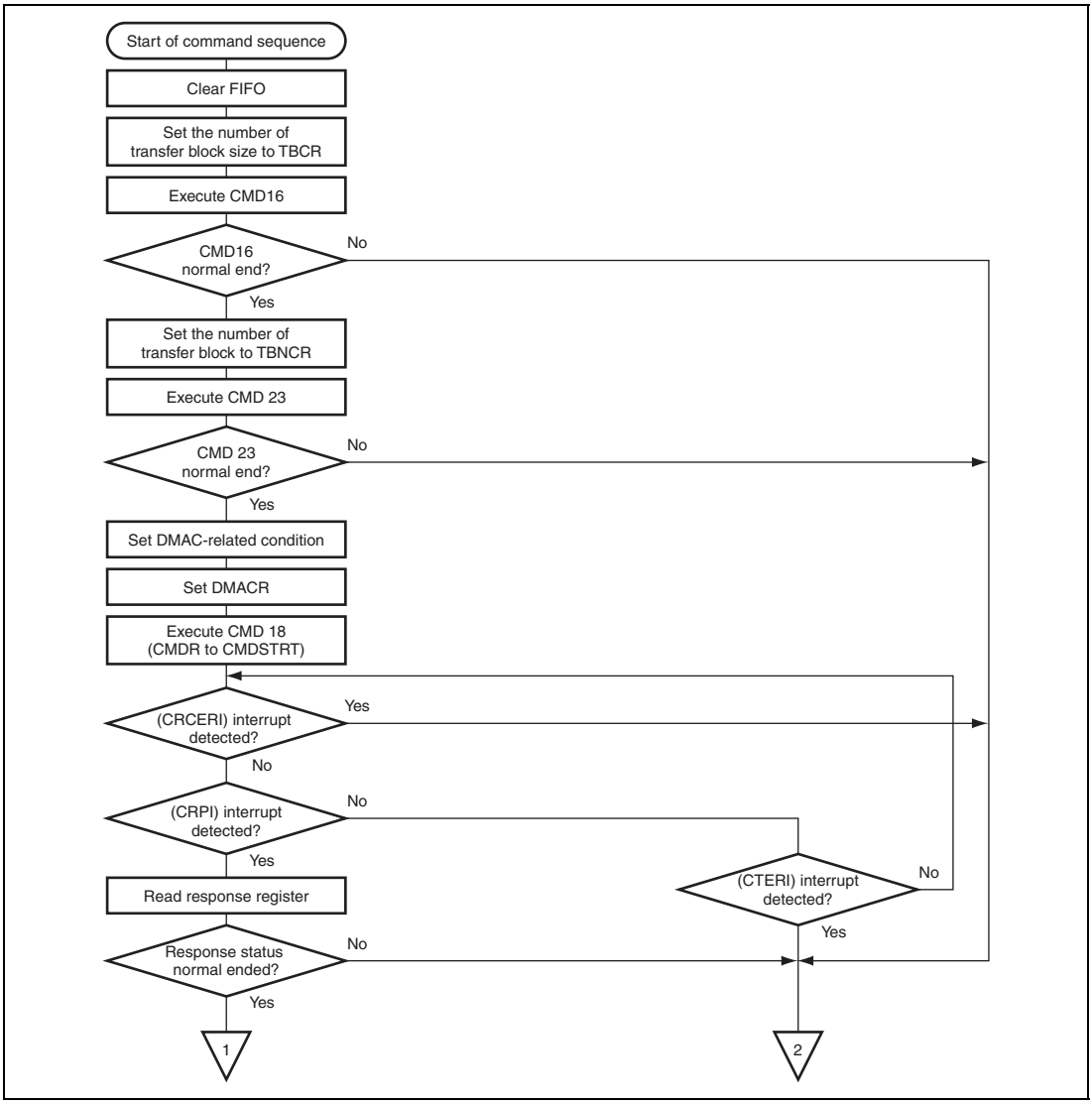


Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (1)

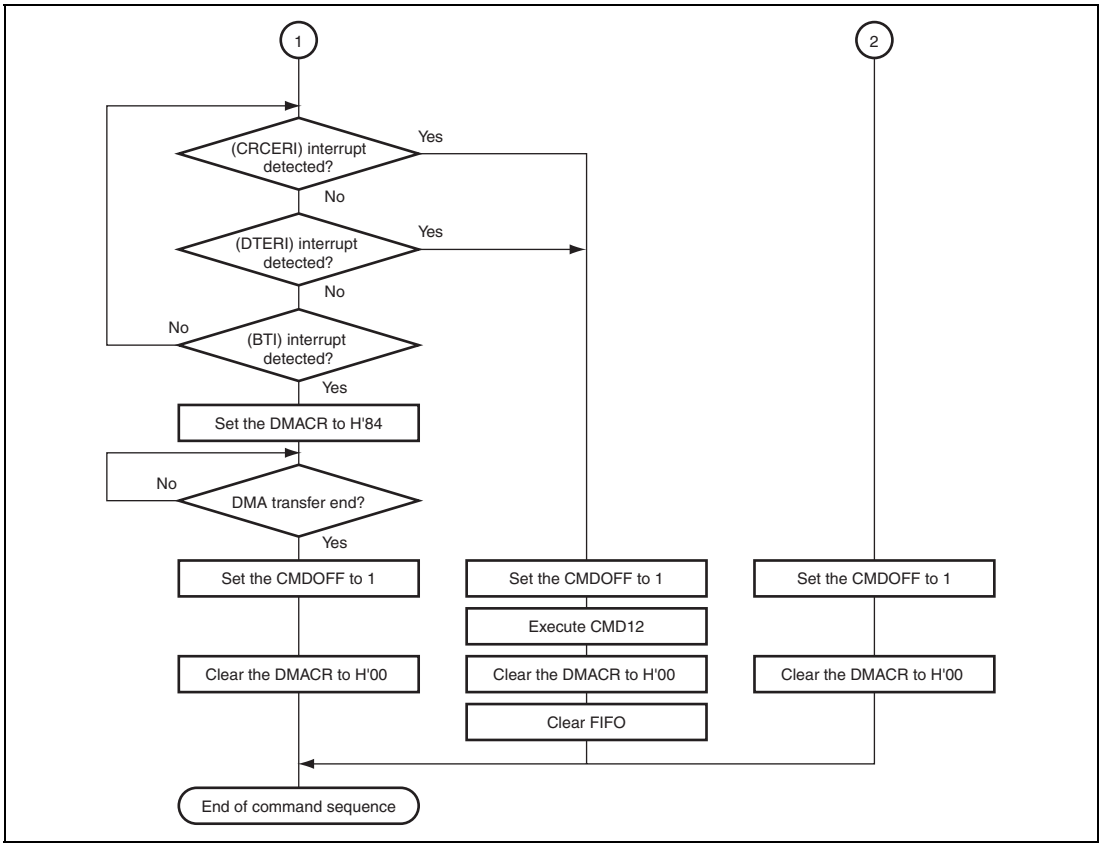


Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (2)

24.6.2 Operation in Write Sequence

To transfer data to FIFO with the DMAC, set MMCIF (DMACR) after setting the DMAC. Then, start transfer to the card after a FIFO ready interrupt. Figure 24.26 to 24.28 shows the operational flow for a write sequence.

- Clear FIFO.
- Transmit write command.
- Make settings in DMACR, and set write data to FIFO.
- Check whether data exceeding the DMACR setting condition is written to FIFO by a FIFO 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 card even after a FIFO empty interrupt is detected. Therefore, complete the write sequence after at least 24 card clock cycles.
- Confirm that the DMAC transfer is completed and be sure to clear the DMAEN bit in DMACR to 0.
- Set the CMDOFF bit to 1 if a CRC error (CRCERI) or a data timeout error (DTERI) occurs in the command response reception.
- Set the CMDOFF bit to 1, clear FIFO, and clear DMACR to H'00 if a CRC error (CRCERI) or 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 multiple block transfer by setting the AUTO bit in DMACR to 1. Figure 24.29 shows the operational flow 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 from the card.
- Set DMACR and write data in FIFO.
- Confirm that the DMA transfer has been completed and clear the DMAEN bit in DMACR to 0.
- Detect the end of the command sequence by poling the BUSY flag in CSTR or through the pre-defined multiple block transfer end flag (BTI).

- An error in a command sequence (during data transmission) is detected through the CRC error flag (CRCERI) or data timeout error flag (DTERI). When these flags are detected, set the CMDOFF bit in OPCR to 1, issue CMD12 (Stop Tran in SPI mode), and suspend the command sequence.
- Confirm there is no data busy state. Detect the data busy end flag (DBSYI) in the data busy state.
- Detect whether in the data busy state through the DTBUSY bit in CSTR after data transfer end (after DRPI is detected). If still in the data busy state, wait for the DBSY flag to confirm that the data busy state has ended.
- Set the CMDOFF bit to 1 and end the command sequence.
- Set the CMDOFF bit to 1 and clear DMACR to H'00 if a CRC error (CRCERI) or a command 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 (CRCERI) or 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.

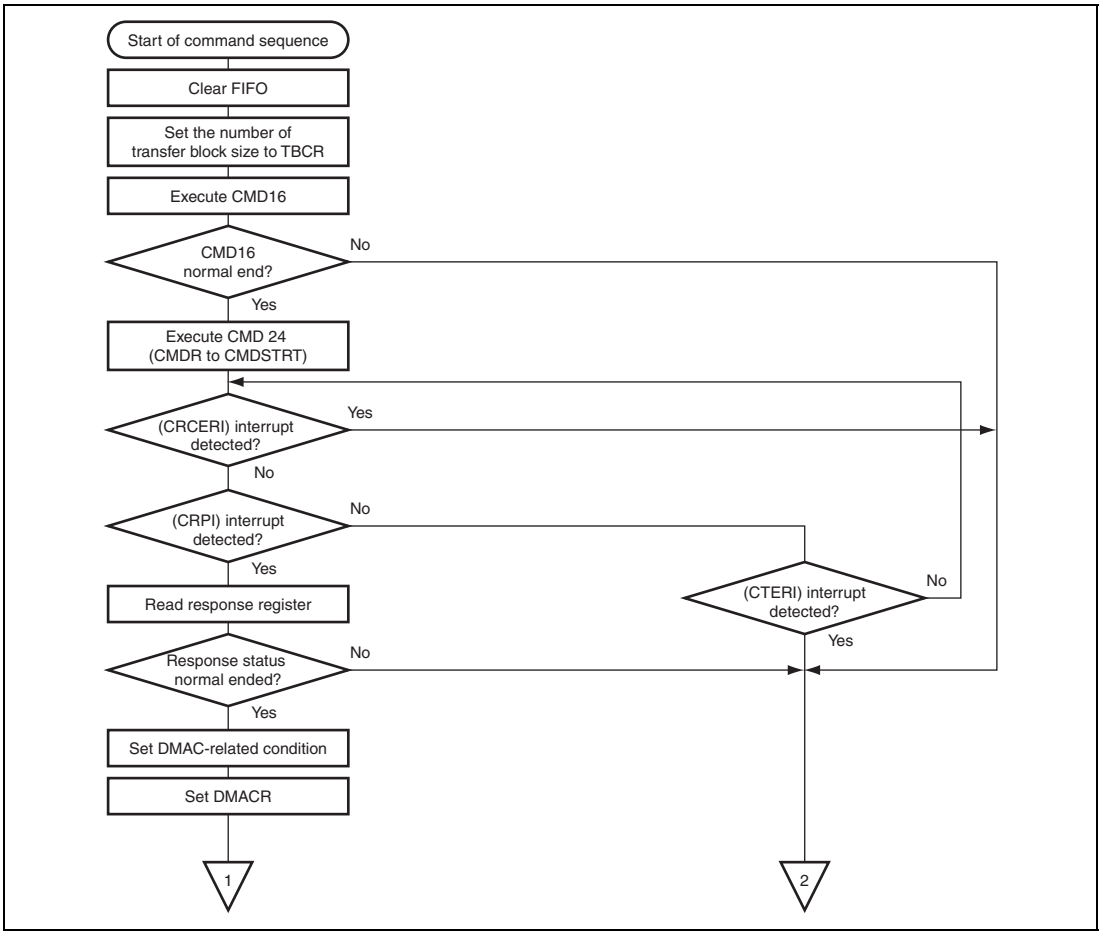


Figure 24.26 Example of Write Sequence Flow (1) (Single Block Transfer)

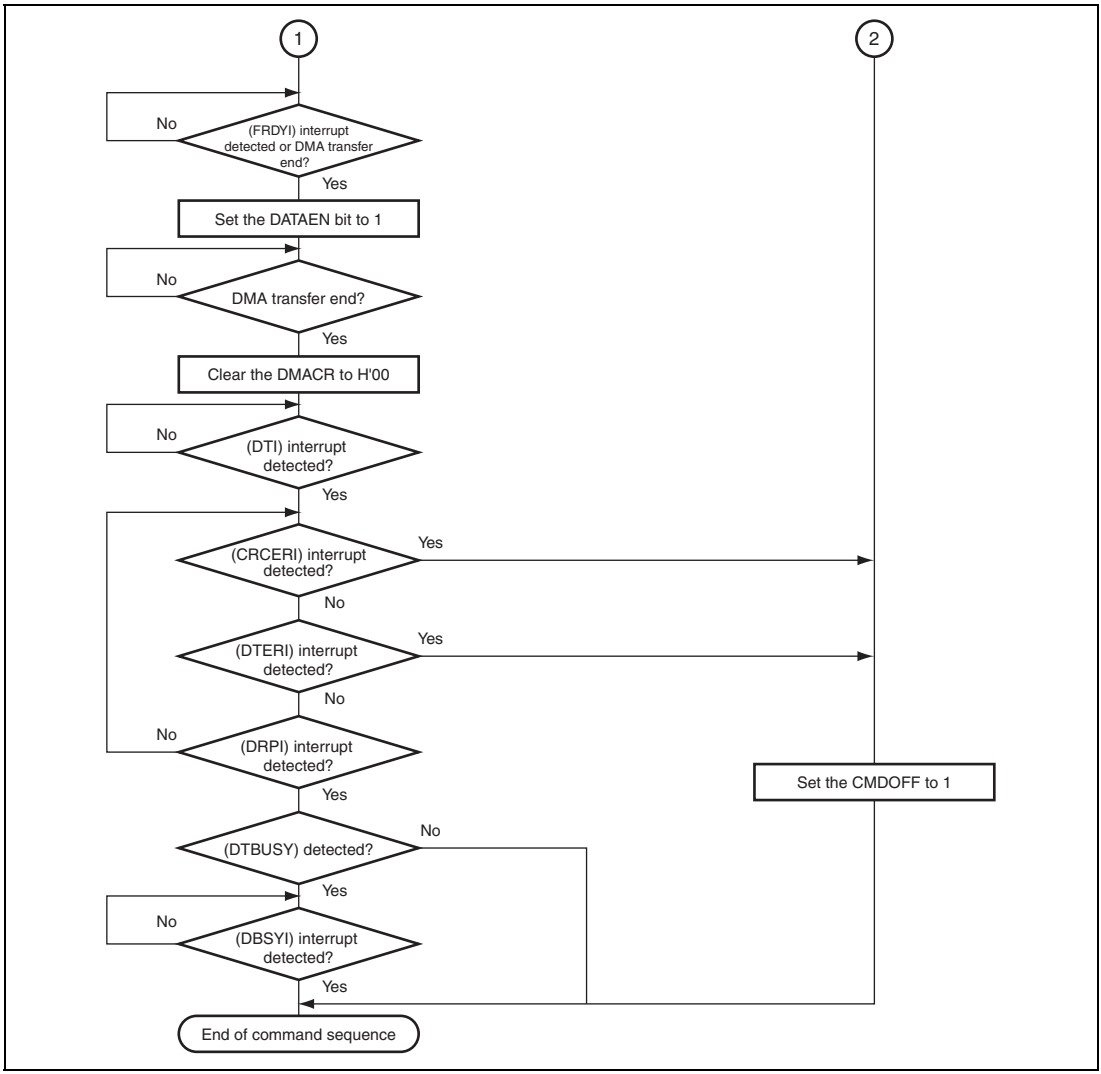


Figure 24.26 Example of Write Sequence Flow (2) (Single Block Transfer)

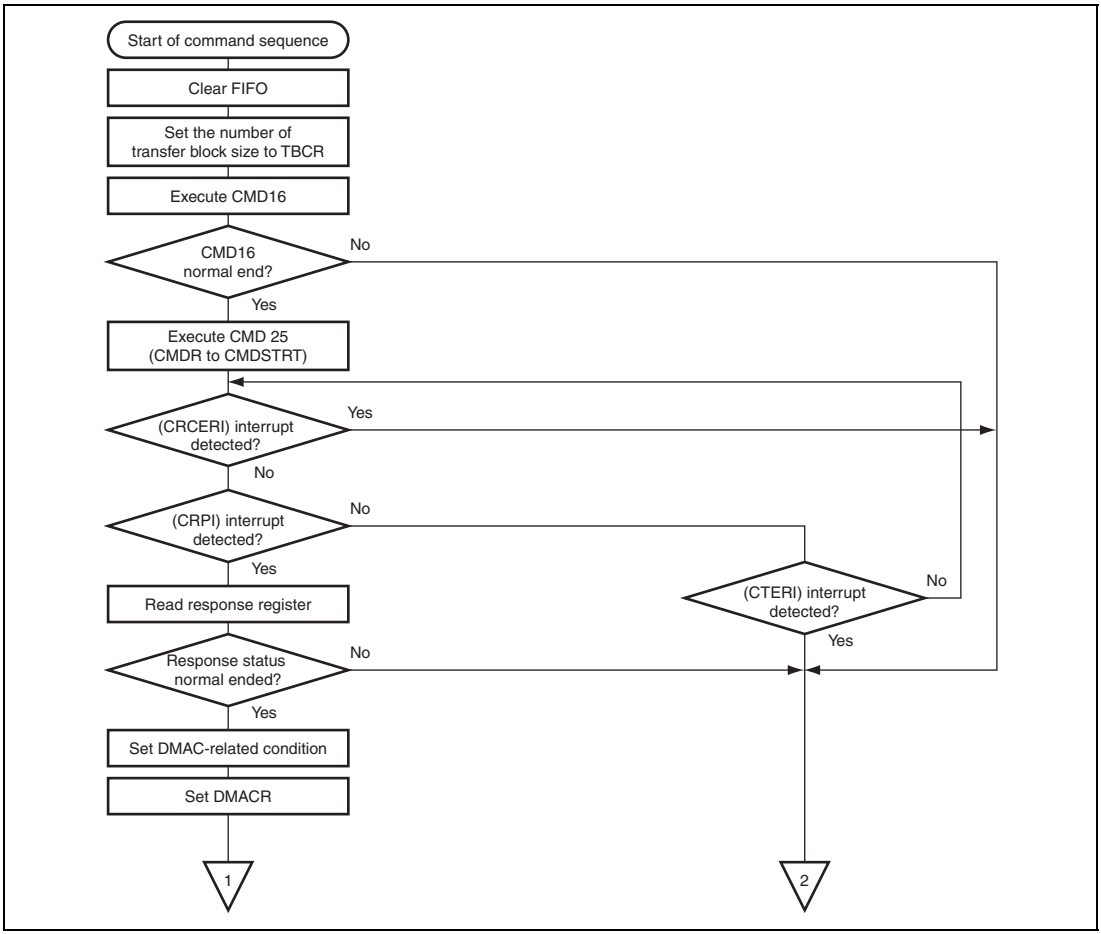


Figure 24.27 Example of Write Sequence Flow (1) (Open-ended Multiple Block Transfer)

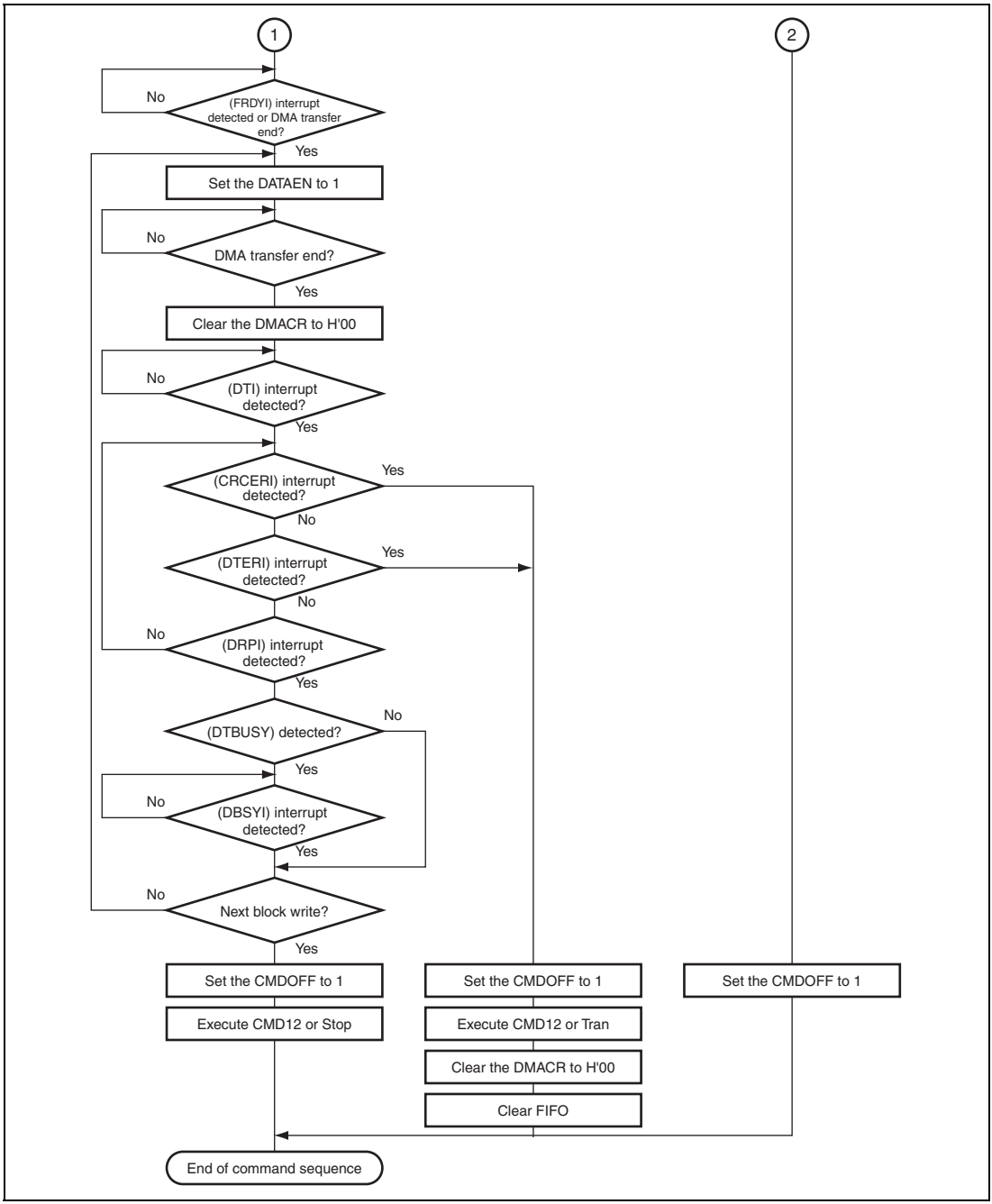


Figure 24.27 Example of Write Sequence Flow (2) (Open-ended Multiple Block Transfer)

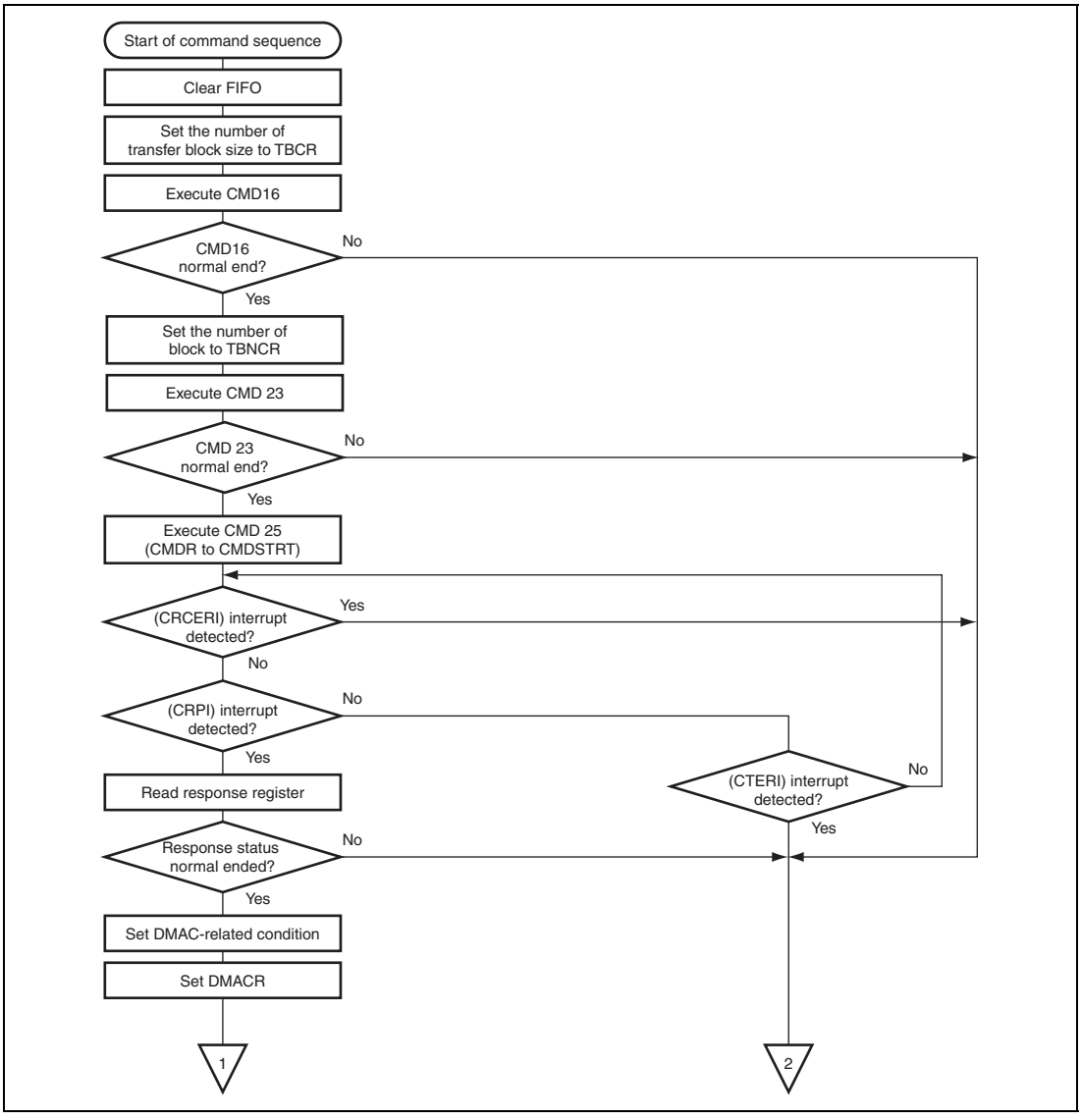


Figure 24.27 Example of Write Sequence Flow (3) (Pre-defined Multiple Block Transfer)

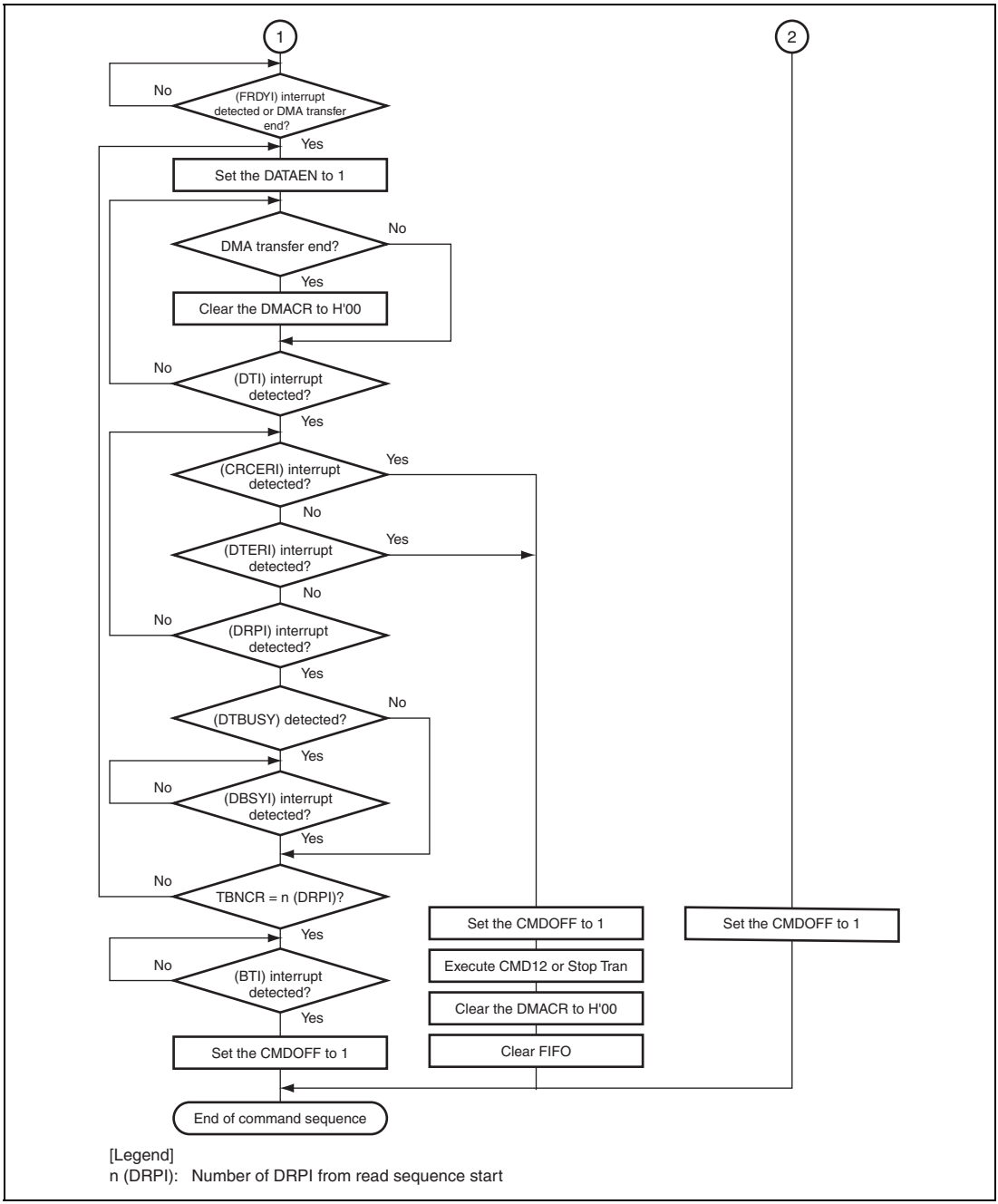


Figure 24.27 Example of Write Sequence Flow (4) (Pre-defined Multiple Block Transfer)

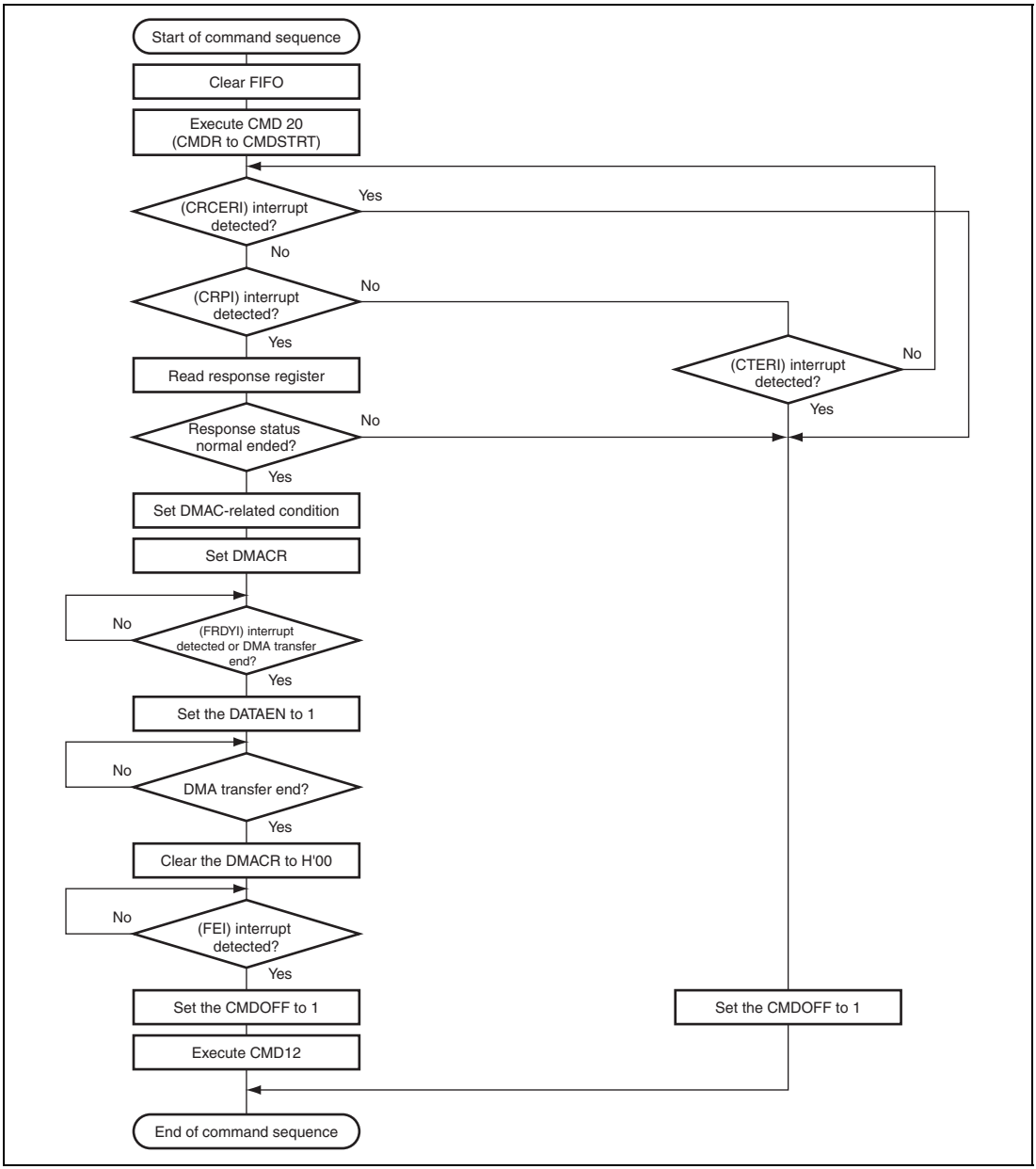


Figure 24.28 Example of Operational Flow for Stream Write Transfer

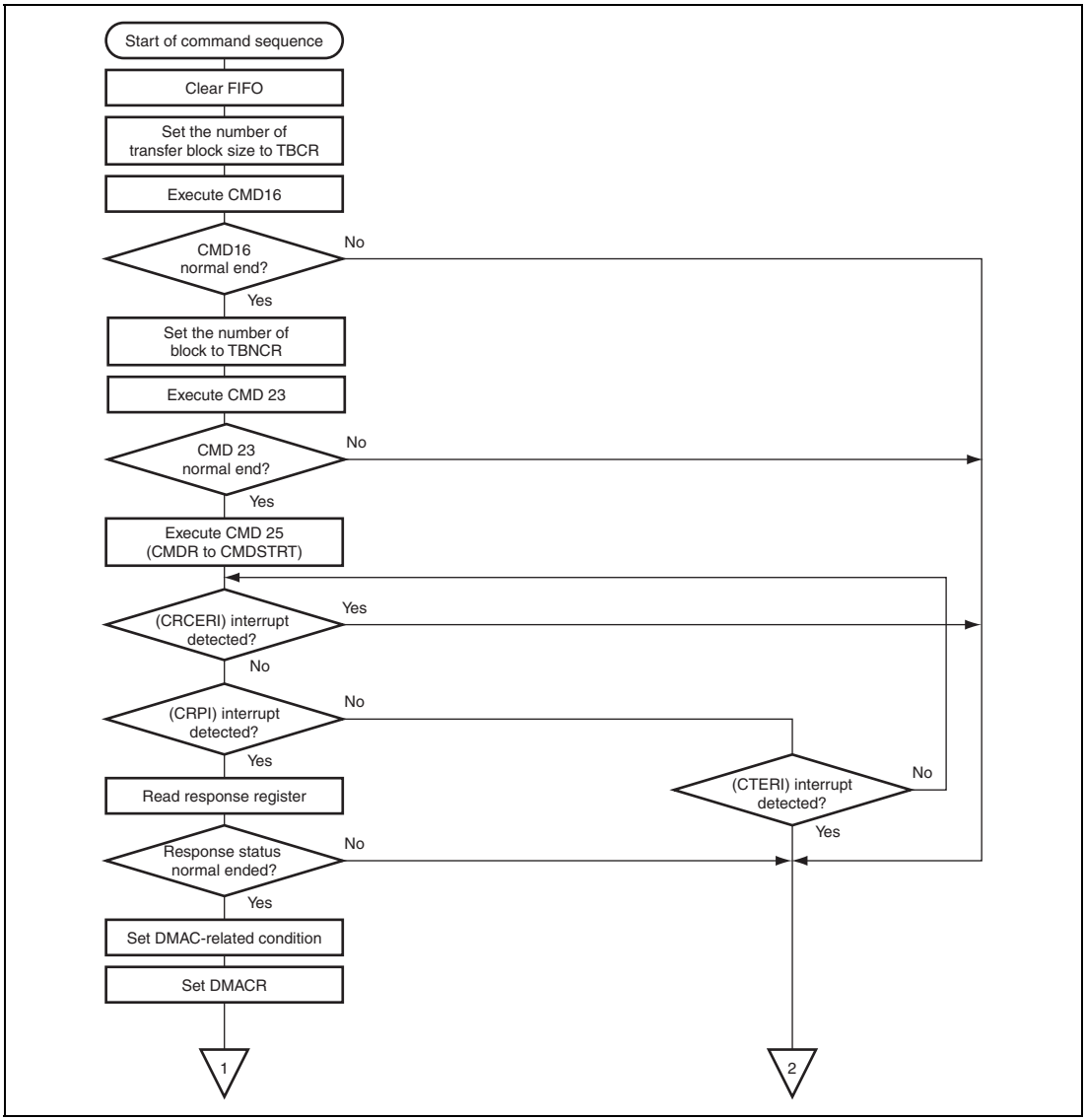


Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (1)

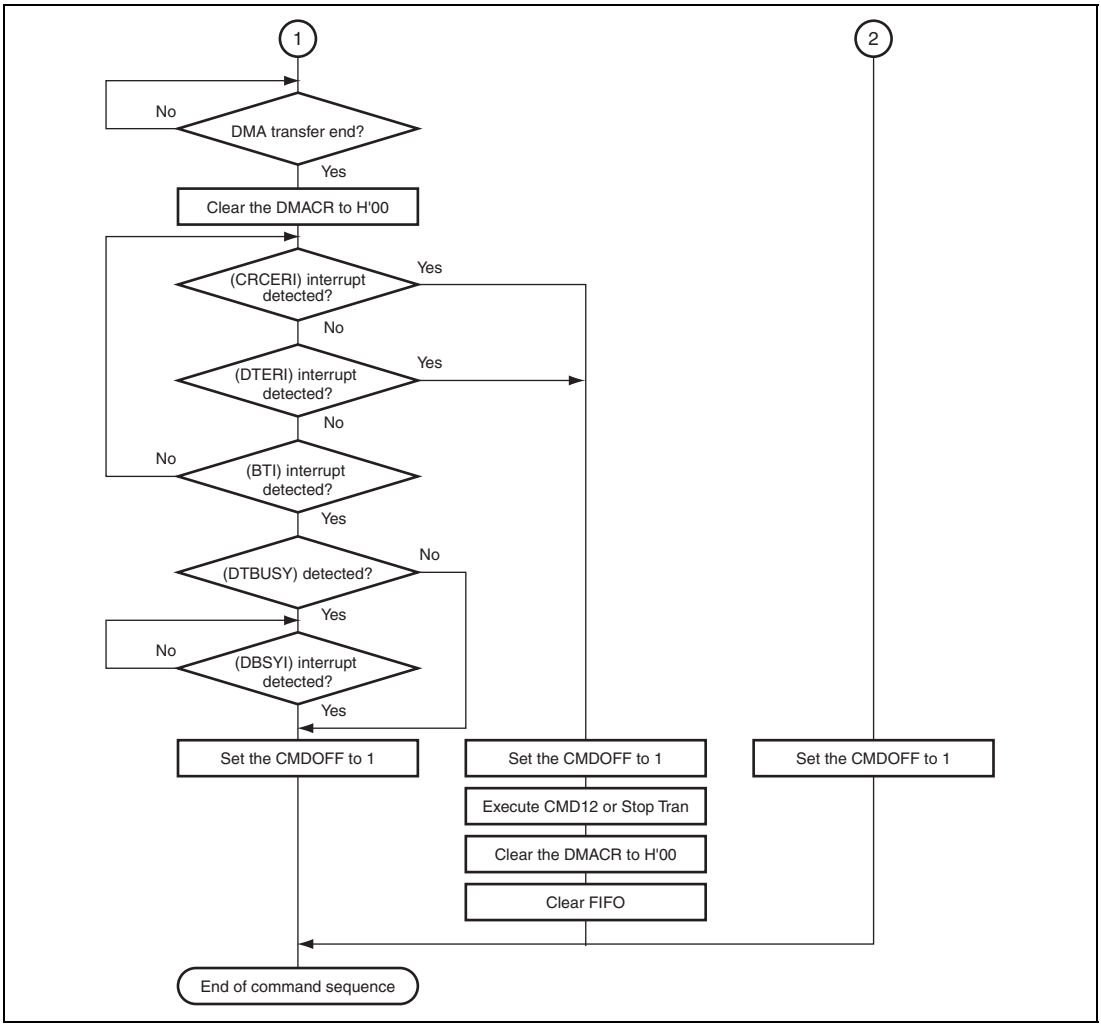


Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (2)

24.7 Register Accesses with Little Endian Specification

When the little endian is specified, the access size for registers or that for memory where the corresponding data is stored should be fixed. For example, if data read from the MMCIF with the word size is written to memory and then it is read from memory with the byte size, data misalignment occurs.

Section 25 Audio Codec Interface (HAC)

The HAC, the audio codec digital controller interface, supports bidirectional data transfer which is a subset of Audio Codec 97 (AC'97) revision 2.1. The HAC provides serial transmission to /reception from the AC97 codec. Each channel of the HAC can be connected to a single audio codec device.

The HAC carries out data extraction from/insertion into audio frames. For data slots within both receive and transmit frames, the PIO transfer by the CPU or the DMA transfer by the DMAC can be used.

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 monitoring the 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 shows a block diagram of the HAC.

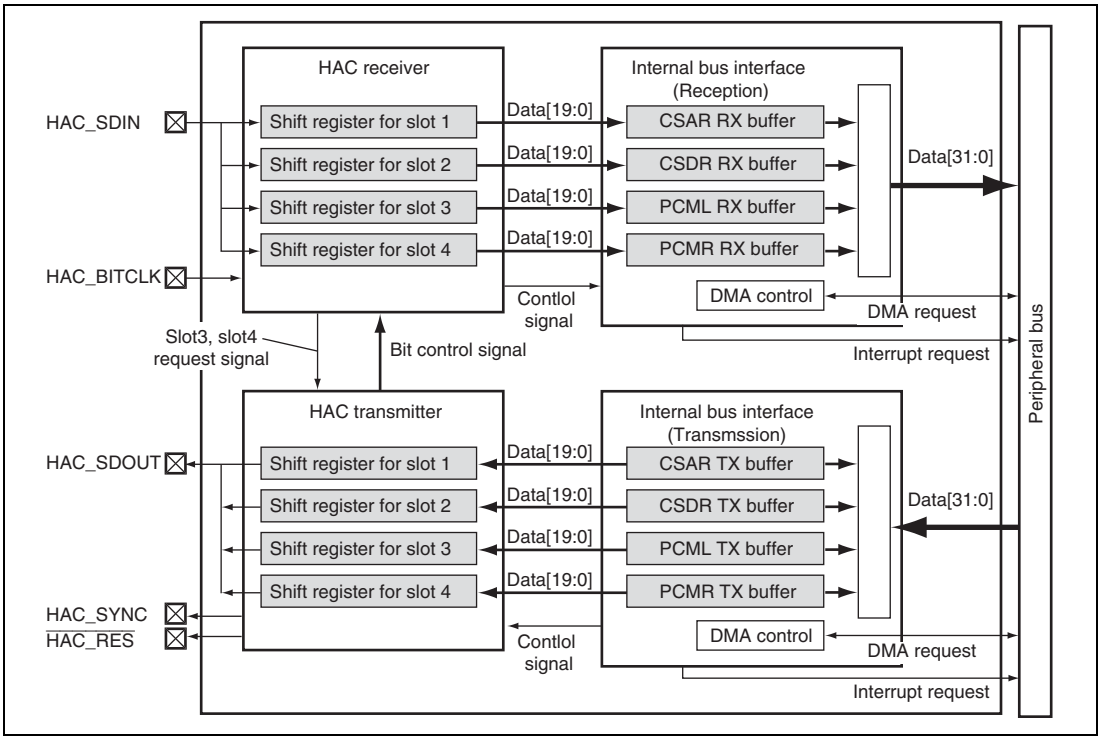


Figure 25.1 Block Diagram

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.

25.3 Register Descriptions

Table 25.2 shows the HAC register configuration. Table 25.3 shows the register states in each processing mode.

Table 25.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Control and status register	HACCR	R/W	H'FFE4 0008	H'1FE4 0008	32	Pck
Command/status address register	HACCSAR	R/W	H'FFE4 0020	H'1FE4 0020	32	Pck
Command/status data register	HACCSDR	R/W	H'FFE4 0024	H'1FE4 0024	32	Pck
PCM left channel register	HACPCML	R/W	H'FFE4 0028	H'1FE4 0028	32	Pck
PCM right channel register	HACPCMR	R/W	H'FFE4 0002C	H'1FE4 002C	32	Pck
TX interrupt enable register	HACTIER	R/W	H'FFE4 0050	H'1FE4 0050	32	Pck
TX status register	HACTSR	R/W	H'FFE4 0054	H'1FE4 0054	32	Pck
RX interrupt enable register	HACRIER	R/W	H'FFE4 0058	H'1FE4 0058	32	Pck
RX status register	HACRSR	R/W	H'FFE4 005C	H'1FE4 005C	32	Pck
HAC control register	HACACR	R/W	H'FFE4 0060	H'1FE4 0060	32	Pck

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	Module Standby
Control and status register	HACCR	H'0000 0200	H'0000 0200	Retained	Retained
Command/status address register	HACCSAR	H'0000 0000	H'0000 0000	Retained	Retained
Command/status data register	HACCSDR	H'0000 0000	H'0000 0000	Retained	Retained
PCM left channel register	HACPCML	H'0000 0000	H'0000 0000	Retained	Retained
PCM right channel register	HACPCMR	H'0000 0000	H'0000 0000	Retained	Retained
TX interrupt enable register	HACTIER	H'0000 0000	H'0000 0000	Retained	Retained
TX status register	HACTSR	H'F000 0000	H'F000 0000	Retained	Retained
RX interrupt enable register	HACRIER	H'0000 0000	H'0000 0000	Retained	Retained
RX status register	HACRSR	H'0000 0000	H'0000 0000	Retained	Retained
HAC control register	HACACR	H'8400 0000	H'8400 0000	Retained	Retained

25.3.1 Control and Status Register (HACCR)

HACCR is a 32-bit read/write register for controlling input/output and monitoring the interface status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CR	—	—	—	CDRT	WMRT	—	—	—	—	ST	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	W	W	R	R	R	R	W	R	R	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 exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a cold reset on the HAC. [Read] Always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10	WMRT	0	W	HAC Warm Reset Use a warm reset only after power-up, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a warm reset on the HAC. [Read] 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 end of the current frame. Do not take this action to terminate 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 register index 26 in the off-chip codec via the HAC. When entering the power-down mode, the off-chip codec stops HAC_BITCLK and suspends the normal operation. The off-chip codec acts in the same manner at power-on. To resume the normal operation, perform a cold reset or a warm reset on the off-chip codec.

25.3.2 Command/Status Address Register (HACCSAR)

HACCSAR is a 32-bit read/write register that specifies the address of the codec register to be read /written. When requesting a write to/read from a codec register, write the command register address to HACCSAR. Then the HAC transmits this register address to the codec via slot 1.

After the codec has responded to a read request (HACRSR.STARY = 1), the status address received via slot 1 can be read out from HACCSAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RW	CA6/SA6	CA5/SA5	CA4/SA4
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
	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	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	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 access to the register specified in the address field (CA6/SA6 to CA0/SA0). Write the data to HACCSDR in advance. When HACACR.TX12_ATOMIC is 1, the HAC transmits HACCSAR and HACCSDR as a pair in the same Tx frame. When HACACR.TX12_ATOMIC is 0, transmission of HACCSAR and HACCSDR in the same Tx frame is not guaranteed. 1: Notifies the off-chip codec device of a read access to the register specified in the address field (CA6/SA6 to CA0/SA0).

Bit	Bit Name	Initial Value	R/W	Description
18	CA6/SA6	0	R/W	Codec Control Register Addresses 6 to 0
17	CA5/SA5	0	R/W	/Codec Status Register Addresses 6 to 0
16	CA4/SA4	0	R/W	[Write]
15	CA3/SA3	0	R/W	Specify the address of the codec register to be written.
14	CA2/SA2	0	R/W	[Read]
13	CA1/SA1	0	R/W	Indicate the status address received via slot 1,
12	CA0/SA0	0	R/W	corresponding to the codec register whose data has been returned in HACCSDR.
11	SLREQ3	0	R	Slot Requests 3 to 12
10	SLREQ4	0	R	Valid only in the Rx frame. Indicate whether the codec
9	SLREQ5	0	R	is requesting slot data in the next Tx frame.
8	SLREQ6	0	R	Automatically set by hardware, and correspond to bits 11 to 2 of slot 1 in the Rx frame.
7	SLREQ7	0	R	0: Slot data is requested.
6	SLREQ8	0	R	1: Slot data is not requested.
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.

25.3.3 Command/Status Data Register (HACCSDR)

HACCSDR is a 32-bit read/write data register used for accessing the codec register. Write the command data to HACCSDR. The HAC then transmits the data to the codec via slot 2.

After the codec has responded to a read request (HACRSR.STDRY = 1), the status data received via slot 2 can be read out from HACCSDR. In both read and write, HACCSAR stores the related codec register address.

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 Write data to these bits and then write the codec register address in HACCSAR. The HAC then transmits the data to the codec. Read these bits to get the contents of the codec register indicated by HACCSAR.
18	CD14/SD14	0	R/W	
17	CD13/SD13	0	R/W	
16	CD12/SD12	0	R/W	
15	CD11/SD11	0	R/W	
14	CD10/SD10	0	R/W	
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.

25.3.4 PCM Left Channel Register (HACPCML)

HACPCML is a 32-bit read/write data register used for accessing the left channel of the codec in digital audio recording or stream playback. To transmit the PCM playback left channel data to the codec, write the data to HACPCML. To receive the PCM record left channel data from the codec, read HACPCML. The data is left justified to accommodate a codec with ADC/DAC resolution of 20 bits or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	D19	D18	D17	D16
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
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
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/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 these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.

In 16-bit packed DMA mode, HACPCML is defined as follows:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
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/W

Bit	Bit Name	Initial 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 these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data 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 these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

25.3.5 PCM Right Channel Register (HACPCMR)

HACPCMR is a 32-bit read/write register used for accessing the right channel of the codec in digital audio recording or stream playback. To transmit the PCM playback right channel data to the codec, write the data to HACPCMR. To receive the PCM record right channel data from the codec, read HACPCMR. The data is left justified to accommodate a codec with ADC/DAC resolution of 20-bit or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	D19	D18	D17	D16
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
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
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/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 these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

25.3.6 TX Interrupt Enable Register (HACTIER)

HACTIER is a 32-bit read/write register that enables or disables HAC TX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PLTF RQIE	PRTF RQIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	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	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PLTF UNIE	PRTF UNIE	—	—	—	—	—	—	—	—
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/W	R/W	R	R	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.

25.3.7 TX Status Register (HACTSR)

HACTSR is a 32-bit read/write register that indicates the status of the HAC TX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD AMT	CMD DMT	PLT FRQ	PRT FRQ	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	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	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PLT FUN	PRT FUN	—	—	—	—	—	—	—	—
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/W	R/W	R	R	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 data. 1: CSAR Tx buffer is empty and ready to store data.*[1]
30	CMDDMT	1	R/W*[2]	Command Data Empty 0: CSDR Tx buffer contains untransmitted data. 1: CSDR Tx buffer is empty and ready to store data. *[1]
29	PLTFRQ	1	R/W*[2]	PCML TX Request 0: PCML Tx buffer contains untransmitted data. 1: PCML TX buffer is empty and needs to store data. In DMA mode, writing to HACPCML will automatically clear this bit to 0.
28	PRTFRQ	1	R/W*[2]	PCMR TX Request 0: PCMR Tx buffer contains untransmitted data. 1: PCMR TX buffer is empty and needs to store data. In DMA mode, writing to HACPCMR will automatically clear this bit to 0.
27 to 10	—	All 0	R	Reserved Always 0 for read and write.

Bit	Bit Name	Initial Value	R/W	Description
9	PLTFUN	0	R/W*[2]	PCML TX Underrun 0: No PCML TX underrun has occurred. 1: PCML TX underrun has occurred because the codec has requested slot 3 data but new data is not written to PCML.
8	PRTFUN	0	R/W*[2]	PCMR TX Underrun 0: No PCMR TX underrun has occurred. 1: PCMR TX underrun has occurred because the codec has requested slot 4 data but new data is not written 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 they are read as 1 before writing a new command to HACCSAR/HACCSDR. When bit 19 (RW) of HACCSAR is 0 and TX12_ATOMIC is 1, take the following steps:
   1. Initialize CMDDMT and CMDAMT before first accessing a codec register after HAC initialization by any reset event.
   2. After making the settings in HACCSDR and HACCSAR, poll CMDDMT and 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 effect.

25.3.8 RX Interrupt Enable Register (HACRIER)

HACRIER is a 32-bit read/write register that enables or disables HAC RX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	—	—	—
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/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	1	0
	—	—	PLRFOVIE	PRRFOVIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	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	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.

25.3.9 RX Status Register (HACRSR)

HACRSR is a 32-bit read/write register that indicates the status of the HAC RX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARY	STDRY	PLR FRQ	PRR FRQ	—	—	—
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/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	1	0
	—	—	PLR FOV	PRR FOV	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	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	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. In DMA mode, reading HACPCML automatically clears this 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. In DMA mode, reading HACPCMR automatically clears this bit to 0.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.

Bit	Bit Name	Initial Value	R/W	Description
13	PLRFOV	0	R/W	PCML RX Overrun 0: No PCML RX data overrun has occurred. 1: PCML RX data overrun has occurred because the HAC has received new data from slot 3 before PCML data is not read out.
12	PRRFOV	0	R/W	PCMR RX Overrun 0: No PCMR RX data overrun has occurred. 1: PCMR RX data overrun has occurred because the HAC has received new data from slot 4 before PCMR 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 effect.

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	17	16
	—	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	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	R	R

Bit:	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
R/W:	R	R	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. Enables the RXDMAL_EN and RXDMAR_EN settings. 1: Enables 16-bit packed RX DMA mode. Disables the RXDMAL_EN and RXDMAR_EN settings.

Bit	Bit Name	Initial Value	R/W	Description
29	DMATX16	0	R/W	16-bit TX DMA Enable 0: Disables 16-bit packed TX DMA mode. Enables the TXDMAL_EN and TXDMAR_EN settings. 1: Enables 16-bit packed TX DMA mode. Disables the TXDMAL_EN and TXDMAR_EN settings.
28, 27	—	All 0	R	Reserved Always 0 for read and write.
26	TX12_ATOMIC	1	R/W	TX Slot 1 and 2 Atomic Control 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 HACCSAR 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.

25.4 AC 97 Frame Slot Structure

Figure 25.2 shows the AC97 frame slot structure. This LSI supports slots 0 to 4 only. Slots 5 to 12 are out of scope.

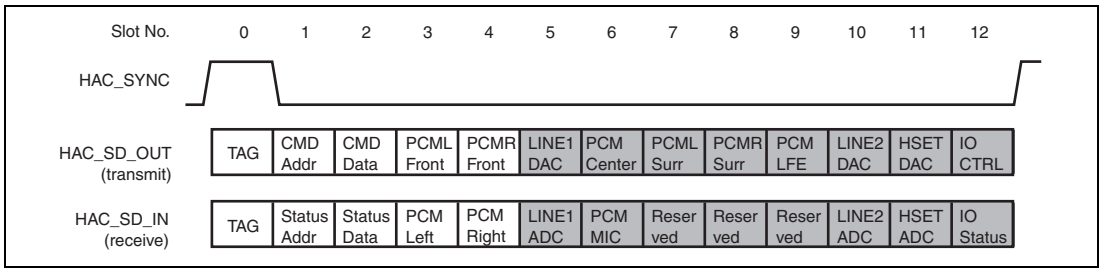


Figure 25.2 AC97 Frame Slot Structure

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 (unsupported)*
8	PCM Surround R	Surround right channel PCM data (unsupported)*
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.

Table 25.5 AC97 Receive Frame Structure

Slot	Name	Description
0	SDATA_IN TAG	Tags indicating valid data
1	Status ADDR read port	Register address and slot request
2	Status DATA read port	Register read data
3	PCM L ADC record	Left channel PCM input data
4	PCM R ADC record	Right channel PCM input data
5	Modem Line 1 ADC	Modem 1 input data (unsupported)*
6	Dedicated Microphone ADC	Optional PCM data (unsupported)*
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 to HAC_BITCLK. From slot 0, the receiver extracts tag bits that indicate which other slots contain valid data. It will update the receive data only when receiving valid slot data indicated by the tag bits.

Supporting data only in slots 1 to 4, the receiver ignores tag bits and data related to slots 5 to 12. It loads valid slot data to the corresponding shift register to hold the data for PIO or DMA transfer, and sets the corresponding status bits. It is possible to read 20-bit data within a 32-bit register using PIO.

In the case of RX overrun, the new data will overwrite the current data in the RX buffer of the HAC.

25.5.2 Transmitter

The HAC transmitter outputs serial audio data on the HAC_SDOUT pin, synchronous to HAC_BIT_CLK. The transmitter sets the tag bits in slot 0 to indicate which slots in the current frame contain valid data. It loads data slots to the current TX frame in response to the corresponding slot request bits from the previous RX frame.

The transmitter supports data only in slots 1 to 4. The TX buffer holds data that has been transferred using PIO or DMA, and sets the corresponding status bit. It is possible to write 20-bit data within a 32-bit register using PIO.

In the case of a TX underrun, the HAC will transmit the current TX buffer data until the next data arrives.

25.5.3 DMA

The HAC supports DMA transfer for slots 3 and 4 of both the RX and TX frames. Specify the slot data size for DMA transfer, 16 or 20 bits, with the DMARX16 and DMATX16 bits in HACACR.

When the data size is 20 bits, transfer of data slots 3 and 4 requires two local bus access cycles. Since each of the receiver and transmitter has its DMA request, the stereo mode generates a DMA request for slots 3 and 4 separately. The mono mode generates a DMA request for just one slot.

When the data size is 16 bits, data from slots 3 and 4 are packed into a single 32-bit quantity (left data and right data are in PCML), which requires only one local bus access cycle.

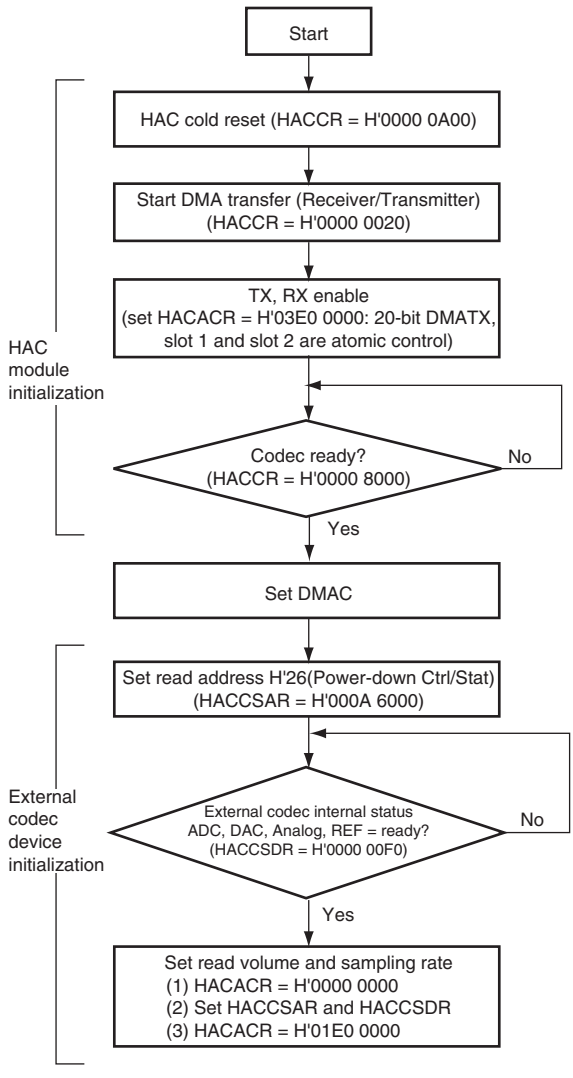
It may be necessary to halt a DMA transfer before the end count is reached, depending on system applications. If so, clear the corresponding DMA bit in HACACR to 0 (DMA disabled). To resume a DMA transfer, reprogram the DMAC and then set the corresponding DMA bit to 1 (DMA enabled).

25.5.4 Interrupts

Interrupts can be used for flag events from the receiver and transmitter. Make the setting for each interrupt in the corresponding interrupt enable register. Interrupts include a request to the CPU to read/write slot data, overrun and underrun. To get the interrupt source, read the status register. Writing 0 to the bit will clear the corresponding interrupt.

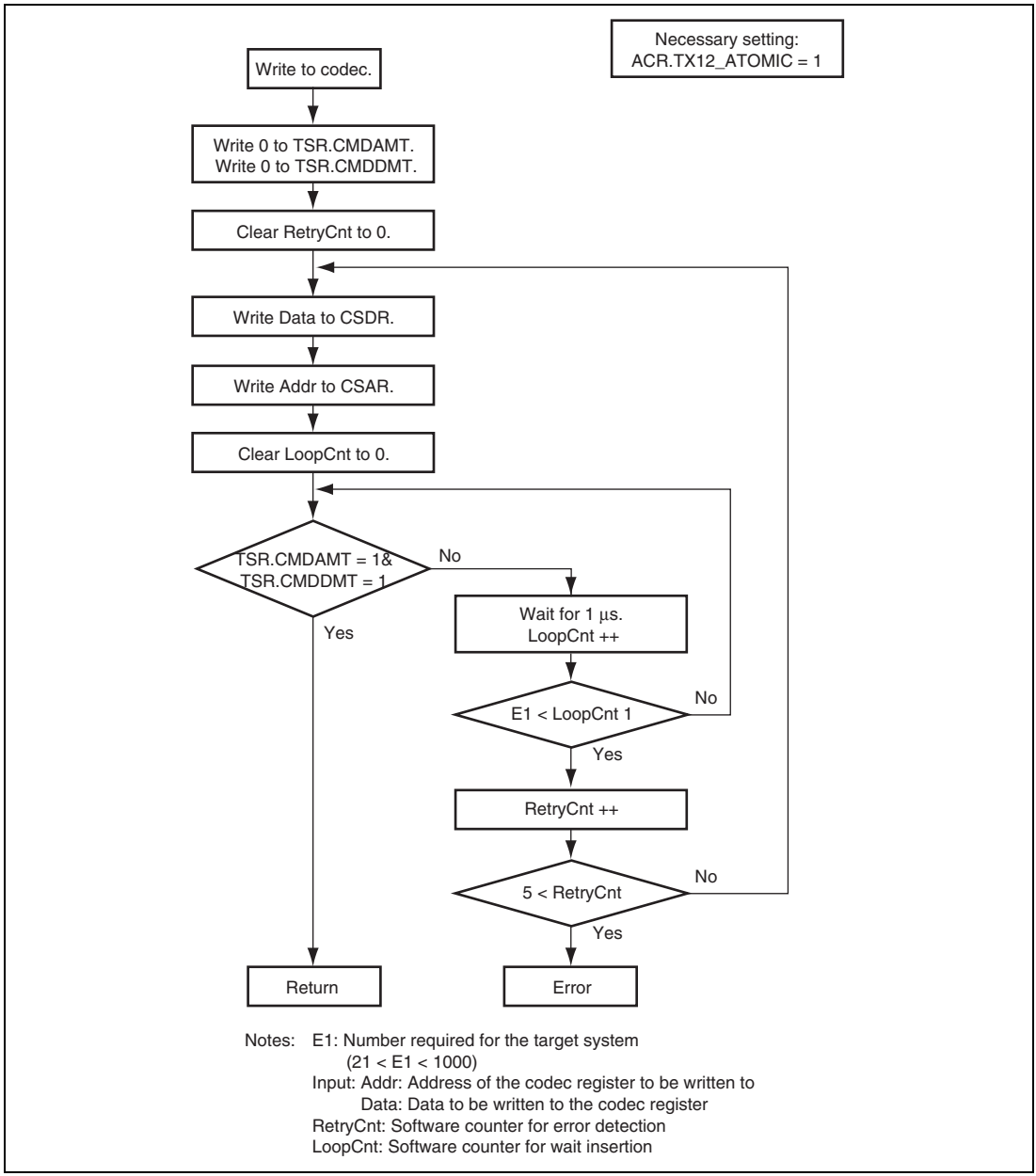
25.5.5 Initialization Sequence

Figure 25.3 shows an example of the initialization sequence.



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

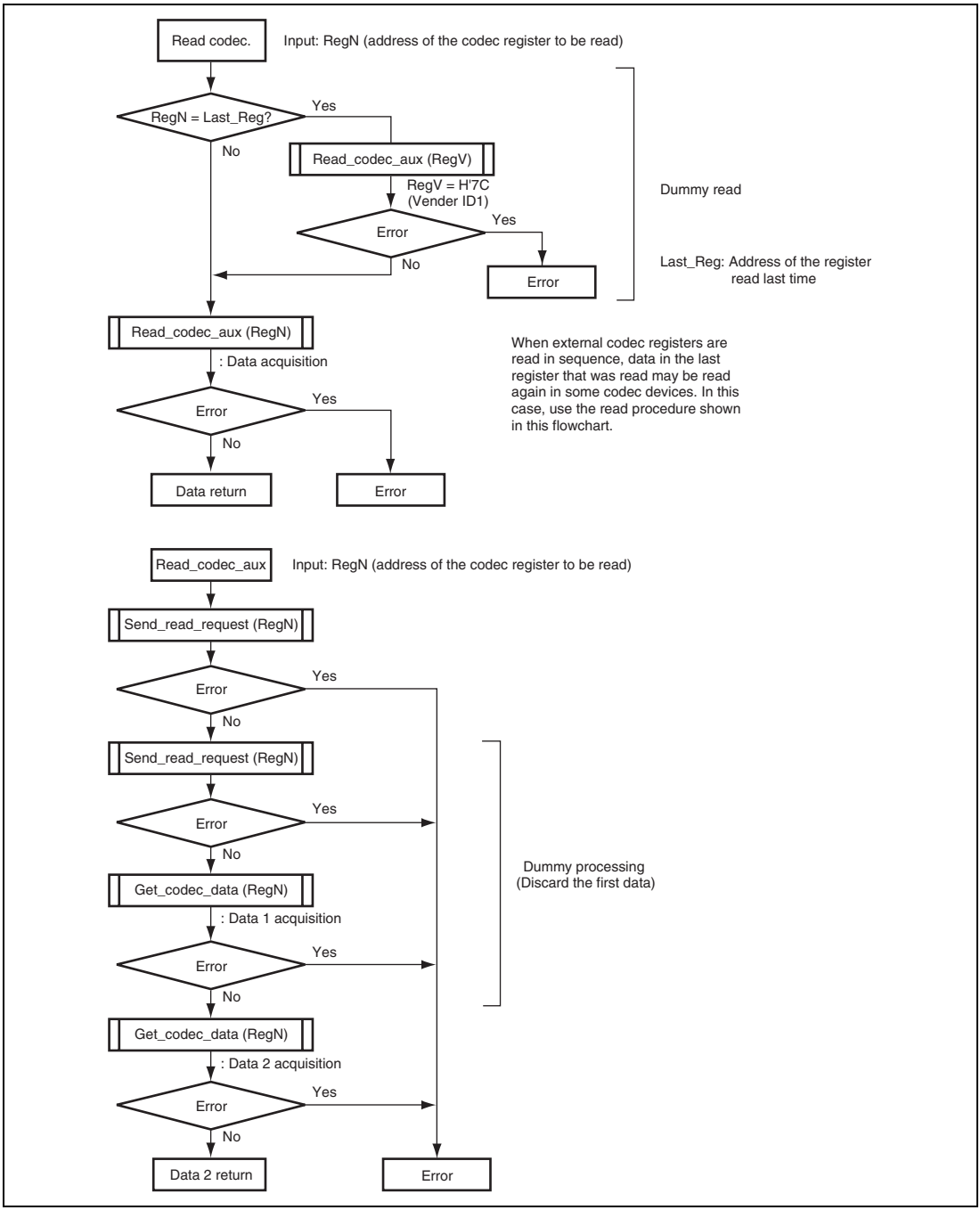


Figure 25.5 Sample Flowchart for Off-Chip Codec Register Read (1)

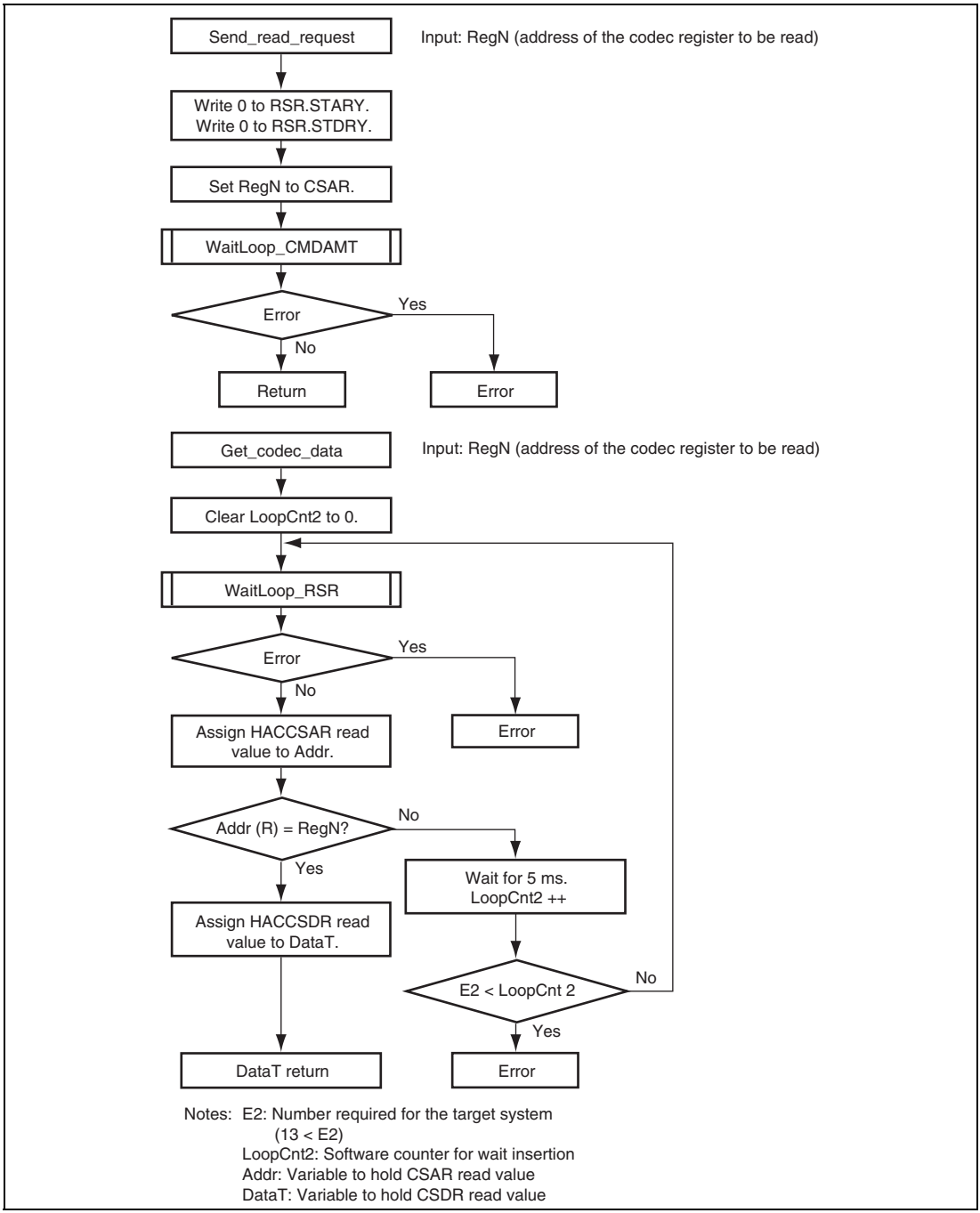
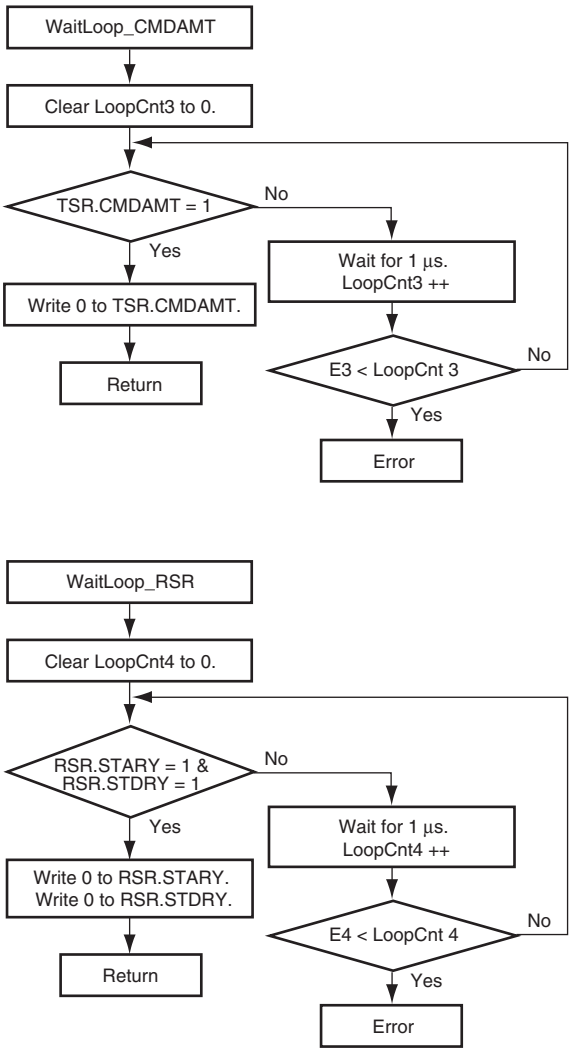


Figure 25.6 Sample Flowchart for Off-Chip Codec Register Read (2)



Notes: E3 and E4: Numbers required for the target system
(21 < E3, 21 < E4 < 1000)
LoopCnt3: Software counter for wait insertion
LoopCnt4: Software counter for wait insertion

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. For details of MSTPCR, refer to section 17, Power-Down Mode.

To cancel module standby mode and resume the supply of clock to the HAC, write 1 to the MSTP0 bit in MSTPCR. After that, it enables all accesses to the HAC.

An example of the procedure to enter the module standby mode is shown below.

1. Check that all data transactions have ended. Also check that the transmit buffer of the HAC is empty and the receive buffer of the HAC has been read out to be empty.
2. Disable all DMA requests and interrupt requests.
3. Place the codec into power-down mode.
4. Write 1 to the MSTP0 bit in MSTPCR.

25.5.6 Notes

The HAC_SYNC signal is generated by the HAC to indicate the position of slot 0 within a frame.

25.5.7 Reference

AC'97 Component Specification, Revision 2.1

Section 26 Serial Sound Interface (SSI) Module

The serial sound interface (SSI) module is a module designed to send or receive audio data interface with a variety of devices offering Philips format compatibility. It also provides additional modes for other common formats, as well as support for a burst and multi-channel mode.

26.1 Features

The SSI has the following features.

- 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 format can be 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.

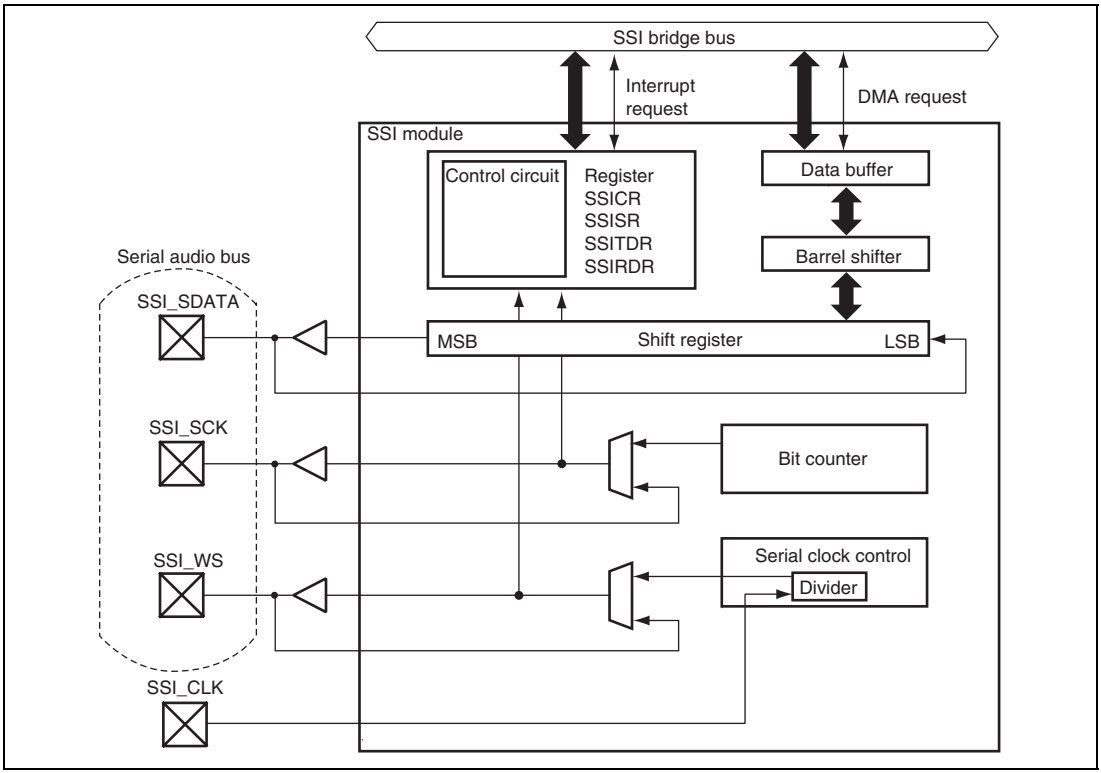


Figure 26.1 Block Diagram of SSI Module

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 input)

Note: These pins are multiplexed with the SIOF, HAC and GPIO pins.

26.3 Register Descriptions

Table 26.2 shows the SSI register configuration. Table 26.3 shows the register states in each processing mode.

Table 26.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Control register	SSICR	R/W	H'FFE7 0000	H'1FE7 0000	32	Pck
Status register	SSISR	R/W*[1]	H'FFE7 0004	H'1FE7 0004	32	Pck
Transmit data register	SSITDR	R/W	H'FFE7 0008	H'1FE7 0008	32	Pck
Receive data register	SSIRDR	R	H'FFE7 000C	H'1FE7 000C	32	Pck

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	Module Standby
Control register	SSICR	H'0000 0000	H'0000 0000	Retained	Retained
Status register	SSISR	H'0200 0003	H'0200 0003	Retained	Retained
Transmit data register	SSITDR	H'0000 0000	H'0000 0000	Retained	Retained
Receive data register	SSIRDR	H'0000 0000	H'0000 0000	Retained	Retained

26.3.1 Control Register (SSICR)

SSICR is a 32-bit readable/writable register that controls the IRQ, selects each polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0
Initial value:	—	—	—	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/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	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	BREN	CKDV			MUEN	CPEN	TRMD	EN
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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined value. The 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

Bit	Bit Name	Initial Value	R/W	Description
23	CHNL1	0	R/W	Channels
22	CHNL0	0	R/W	These bits indicate the number of channels in each system word. These bits are ignored if CPEN = 1. 00: 1 channel per system word 01: 2 channels per system word 10: 3 channels per system word 11: 4 channels per system word
21	DWL2	0	R/W	Data Word Length
20	DWL1	0	R/W	These bits indicate the encoded number of bits in a data word. These bits are ignored if CPEN = 1.
19	DWL0	0	R/W	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 bits in a system word. These bits are ignored if CPEN = 1.
16	SWL0	0	R/W	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

Bit	Bit Name	Initial Value	R/W	Description
13	SCKP	0	R/W	Serial Clock Polarity 0: SSI_WS and SSI_SDATA change on falling edge of SSI_SCK (sampled on rising edge of SCK) 1: SSI_WS and SSI_SDATA change on rising edge of SSI_SCK (sampled on falling edge of SCK)

	SCKP = 0	SCKP = 1
SSI_SDATA input sampling timing in receive mode (TRMD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_SDATA output change timing in transmit mode (TRMD = 1)	SSI_SCK falling edge	SSI_SCK rising edge
SSI_WS input sampling in slave mode (SWSD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_WS output change timing in master mode (SWSD = 1)	SSI_SCK falling edge	SSI_SCK rising edge

12	SWSP	0	R/W	Serial WS Polarity The function of this bit depends on whether the SSI module is in non-compressed mode or compressed mode. CPEN = 0 (Non compressed mode): 0: SSI_WS is low for the first channel, high for the second channel 1: SSI_WS is high for the first channel, low for the second channel CPEN = 1 (Compressed mode): 0: SSI_WS is active high flow control. WS = high means data should be transferred, low means data should not be transferred. 1: SSI_WS is active low flow control. WS = low means data should be transferred, high means data should not be transferred.

Bit	Bit Name	Initial Value	R/W	Description
11	SPDP	0	R/W	Serial Padding Polarity This bit is ignored if CPEN = 1. 0: Padding bits are low 1: Padding bits are high
10	SDTA	0	R/W	Serial Data Alignment This bit is ignored if CPEN = 1. 0: Serial data is transmitted/ received first, followed by padding bits. 1: Padding bits are transmitted/ received first, followed 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 bit has no meaning. This bit is applied to SSIRDR in receive mode and to SSITDR in transmit mode. 0: Parallel data (SSITDR or SSIRDR) is left aligned 1: Parallel data (SSITDR or SSIRDR) is right aligned • DWL = 000 (data word length: 8 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted/received in each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is stored in bits 31 to 24. • DWL = 001 (data word length: 16 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted/received in each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is stored in bits 31 to 16.

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 0 (left aligned) • The data bits which are used in SSIRDR or SSITDR are the following: Bits 31 to (32 – number of bits having data word length specified by DWL). If DWL = 011 then data word length is 20 bits and bits 31 to 12 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits (number of bits having data word length specified by DWL - 1) to 0. If DWL = 011 then data word length is 20 bits and bits 19 to 0 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (data word length: 32 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the 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 compressed mode (CPEN = 1). When burst mode is enabled the SSI_SCK signal is gated. Clock pulses are output only when there is valid serial data being output on SSI_SDATA.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV	All 0	R/W	Serial Oversampling Clock Division Ratio These bits define the division ratio between oversampling Clock (HAC_BIT_CLK) and the serial bit clock. These bits are ignored if SCKD = 0. The Serial Bit Clock is used for the shift register and is provided on the SSI_SCK pin. 000: (Serial bit clock frequency = oversampling clock frequency/1) 001: (Serial bit clock frequency = oversampling clock frequency/2) 010: (Serial bit clock frequency = oversampling clock frequency/4) 011: (Serial bit clock frequency = oversampling clock frequency/8) 100: (Serial bit clock frequency = oversampling clock frequency/16) 101: (Serial bit clock frequency = oversampling clock frequency/6) 110: (Serial bit clock frequency = oversampling clock frequency/12) 111: Setting prohibited
3	MUEN	0	R/W	Mute Enable When in transmit mode (TRMD = 1), by making MUEN = 1, the output of SSI_SDATA will be in low level. 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 operation mode except slave transmitter (SWSD=0 and 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

26.3.2 Status Register (SSISR)

SSISR is configured by status flags that indicate the operating status of the SSI module and bits that indicate the current channel number and word number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	—	—	—	—	—	—	—	—
Initial value:	—	—	—	0	0	0	1	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W*	R/W*	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	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO1	CHNO0	SWNO	IDST
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	1	1
R/W:	R	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	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the status of the 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 there is new unread data. TRMD = 1 (Transmit Mode): • If DMRQ = 1, SSITDR requests data to be written to continue the transmission onto the audio serial bus. • Once data is written to SSITDR then DMRQ = 0 until further transmit data is requested.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/W*	Underflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a lower rate than the required rate. This bit is set to 1 regardless of the setting of UIEN bit. In order to clear it to 0, write 0 in it. If UIRQ = 1 and UIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If UIRQ = 1, it indicates that SSIRDR was read out before DMRQ and DIRQ bits would indicate the existence of new unread data. In this instance, the same received data may be stored twice by the host, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If UIRQ = 1, it indicates that the transmitted data was not written in SSITDR. By this, the same data may be transmitted one time too often, which can lead to destruction of multi-channel data. Consequently, erroneous SSI data will be output, which makes this error more serious than underflow in the receive mode. Note: When underflow error occurs, the data in the data buffer will be transmitted until the next data is written in.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W*	Overflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a higher rate than the required rate. This bit is set to 1 regardless of the setting of OIEN bit. In order to clear it to 0, write 0 in it. If OIRQ = 1 and OIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If OIRQ = 1, it indicates that the previous unread data had not been read out before new unread data was written in SSIRDR. This may cause the loss of data, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If OIRQ = 1, it indicates that SSITDR had data written in before the data in SSITDR was transferred to the shift register. This may cause the loss of data, which can lead to destruction of multi-channel data. Note: When overflow error occurs, the data in the data buffer will be overwritten by the next data sent from the SSI interface.
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This status flag indicates whether the SSI module is in the idle status. This bit is set to 1 regardless of the setting of IIEN bit, so that polling will be possible. The interrupt can be masked by clearing IIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If IIRQ = 1 and IIEN = 1, then an interrupt will be generated. 0: The SSI module is not in the idle status. 1: The SSI module is in the idle status.

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the SSI module requires that data be either read out or written in. This bit is set to 1 regardless of the setting of DIEN bit, so that polling will be possible. The interrupt can be masked by clearing DIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If DIRQ = 1 and DIEN = 1, then an interrupt will be generated. 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 that data be written in SSITDR.
23 to 4	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
3	CHNO1	0	R	Channel Number
2	CHNO0	0	R	The number indicates the current channel. When TRMD = 0 (Receive Mode): This bit indicates to which channel the current data in SSIRDR belongs. When the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates the data of which channel should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	Serial Word Number The number indicates the current word number. When TRMD = 0 (Receive Mode): This bit indicates which system word the current data in SSIRDR is. Regardless whether the data has been read out from SSIRDR, when the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates which system word should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this 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 is currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 and TRMD = 1): This bit is set to 1 if no more data has been written to SSITDR and the current system word has been completed. It can also be set to 1 by clearing the EN bit after sufficient data has been written to SSITDR to complete the system word currently being output. SSI = Serial bus master receiver (SWSD = 1 and TRMD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. SSI = slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external device stops the serial bus clock before the current system word is completed then this bit will never be set.

Note: * These bits are readable/writable bits. If writing 0, these bits are initialized, although writing 1 is ignored.

26.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to SSITDR is transferred to the shift register as it is required for transmission. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit.

Reading this register will return the data in the buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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/W

Bit:	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
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

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 the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit in SSICR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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	R	R	R

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

26.4 Operation

26.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus formats can be one of eight major modes as shown in table 26.4.

Table 26.4 Bus Formats of SSI Module

Bus Format	TPMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
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		Ignored		
Compressed Slave Transmitter	1	1	0/1	0															
Compressed Master Receiver	0	1	0/1	1															
Compressed Master Transmitter	1	1	0/1	1															

26.4.2 Non-Compressed Modes

The non-compressed mode is designed to support all serial audio streams which are split into channels. It can support Philips, Sony and Matsushita modes as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(2) Slave Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(3) Master Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these signals is as defined in the SSI module. If the incoming data does not conform to the defined format then operation is not guaranteed.

(4) Master Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these 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 modes.

There are many configurations that the SSI module can support and it is not sensible to show all of the Serial Data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, and Matsushita.

1. Philips Format

   Figures 26.2 and 26.3 show the supported Philips protocol both with padding and without. Padding occurs when the data word length is smaller than the system word length.

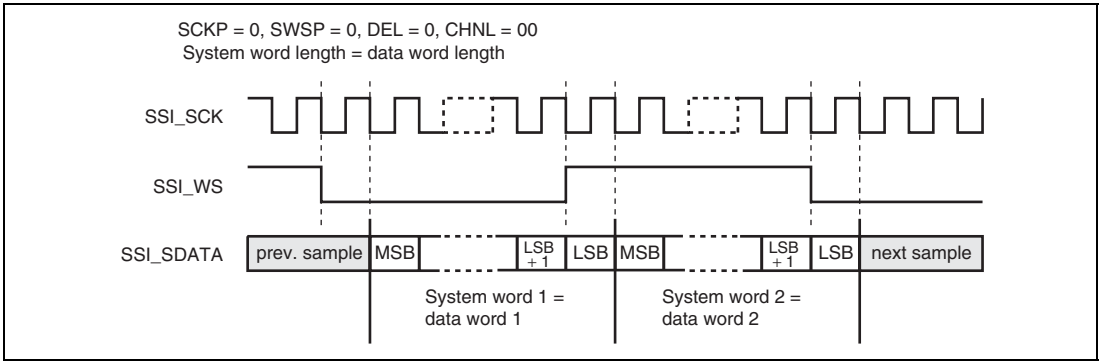


Figure 26.2 Philips Format (with no Padding)

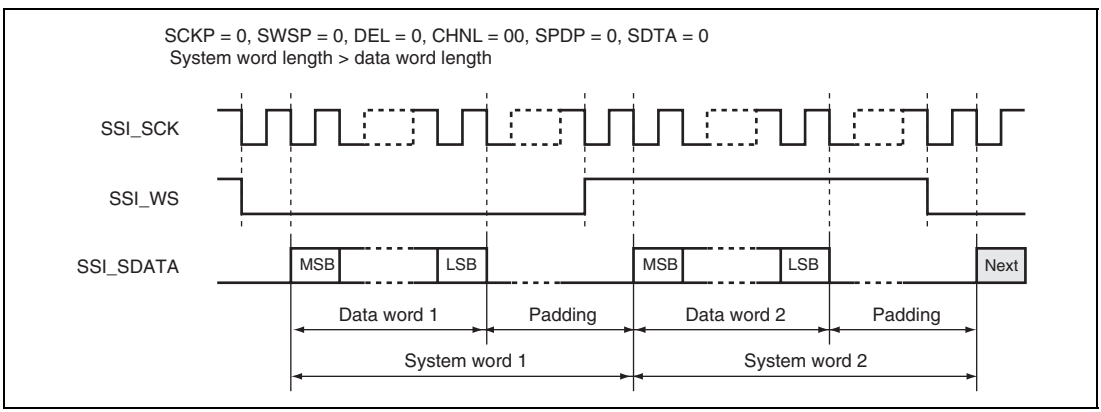


Figure 26.3 Philips Format (with Padding)

Figure 26.4 shows the format used by Sony. Figure 26.5 shows the format used by Matsushita. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

2. Sony Format

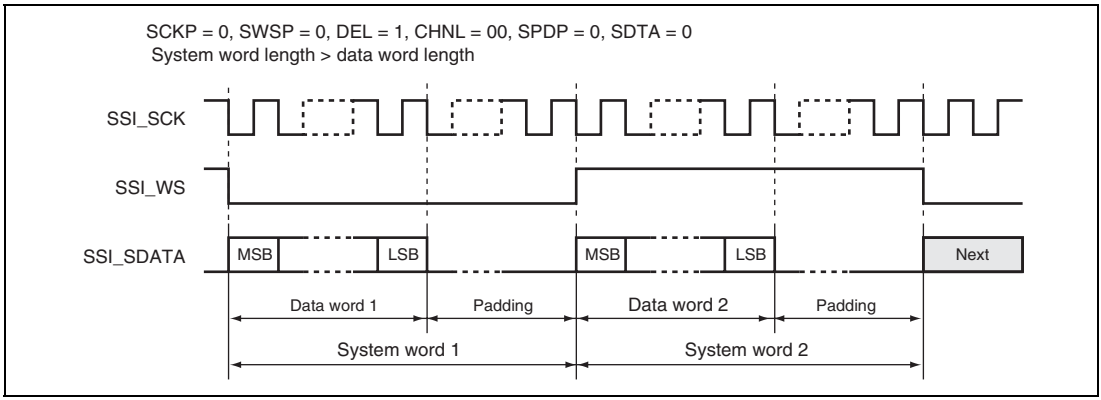


Figure 26.4 Sony Format (with Serial Data First, Followed by Padding Bits)

3. Matsushita Format

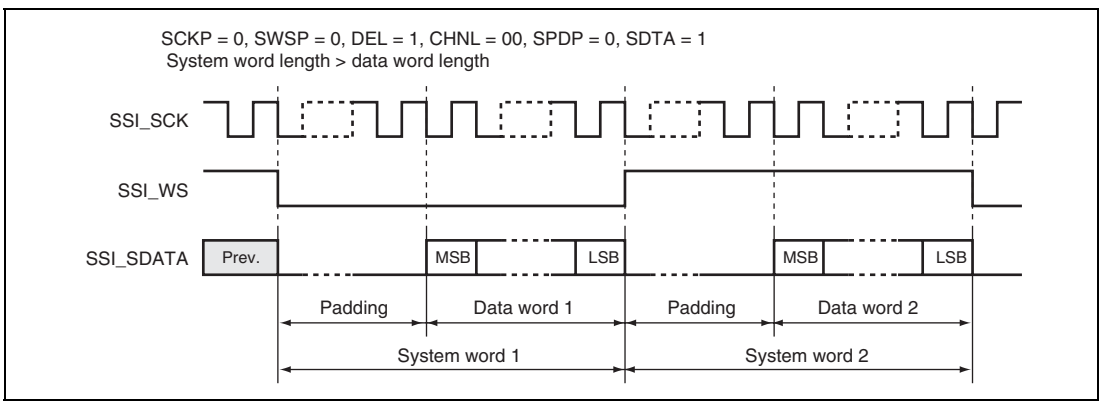


Figure 26.5 Matsushita Format (with Padding Bits First, Followed by Serial Data)

(6) Multi-Channel Formats

There are some extend format of the Philips' specification that allows more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by the use of the CHNL, SWL and DWL bits. It is important that the system word length (SWL) is greater than or equal to the 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 setup is not valid it does not have a number in the following table and has instead a dash.

Table 26.5 Number of Padding Bits for Each Valid Configuration

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
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	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
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	32
		111	256	232	208	202	196	190	184	160
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	0
		111	256	224	192	184	176	168	160	128

In the case of the SSI module configured as a transmitter then each word that is written to SSITDR is transmitted in order on the serial audio bus.

In the case of the SSI module configured as a receiver each word received on the Serial Audio Bus is presented for reading in order by SSIRDR.

Figures 26.6 to 26.8 show how 4, 6 and 8 channels are transferred on the serial audio bus.

Note that there are no padding bits in the first example, serial data is transmitted/received first and followed by padding bits in the second example, and padding bits are transmitted/received first and followed by serial data in the third example. This selection is purely arbitrary.

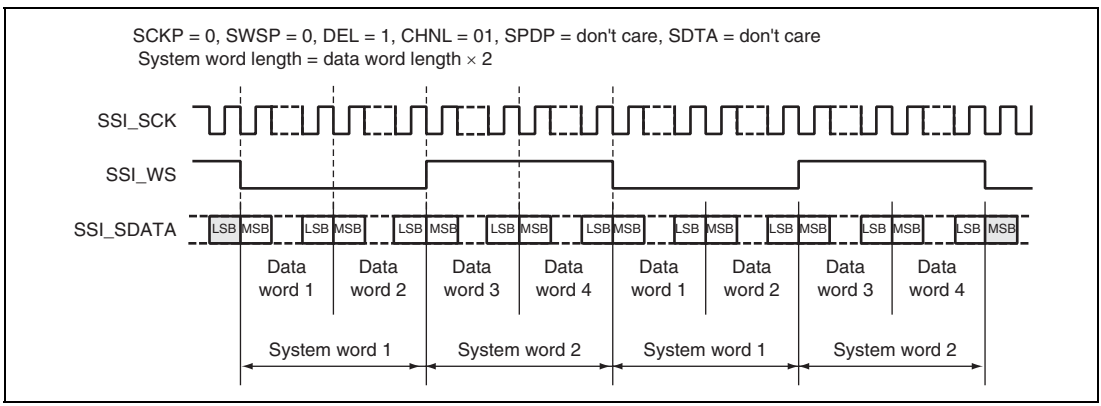


Figure 26.6 Multi-channel Format (4 Channels, No Padding)

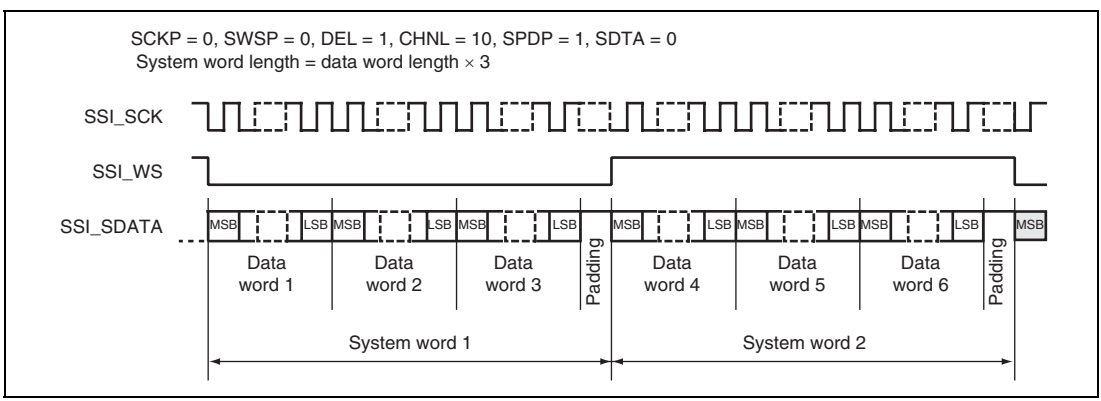


Figure 26.7 Multi-channel Format (6 Channels with High Padding)

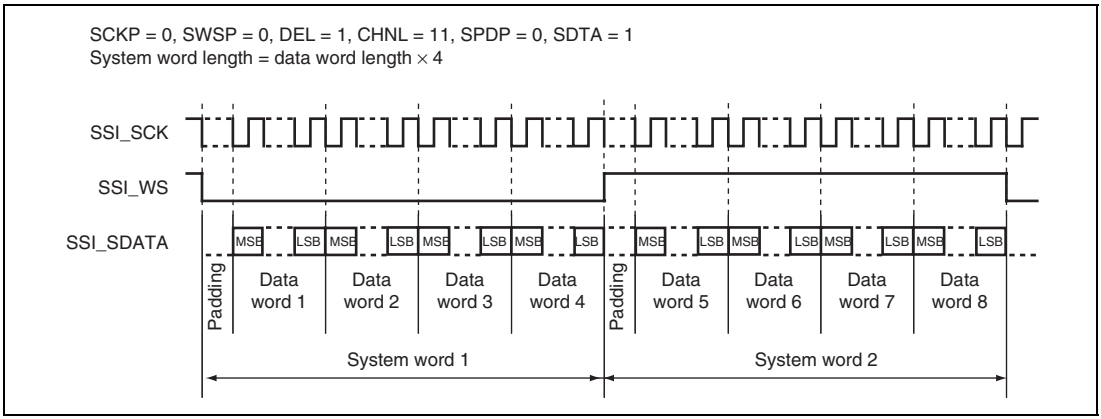


Figure 26.8 Multi-channel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding)

(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 probably not be useful for any other device.

They are demonstrated by referring to the following basic sample format shown in figure 26.9.

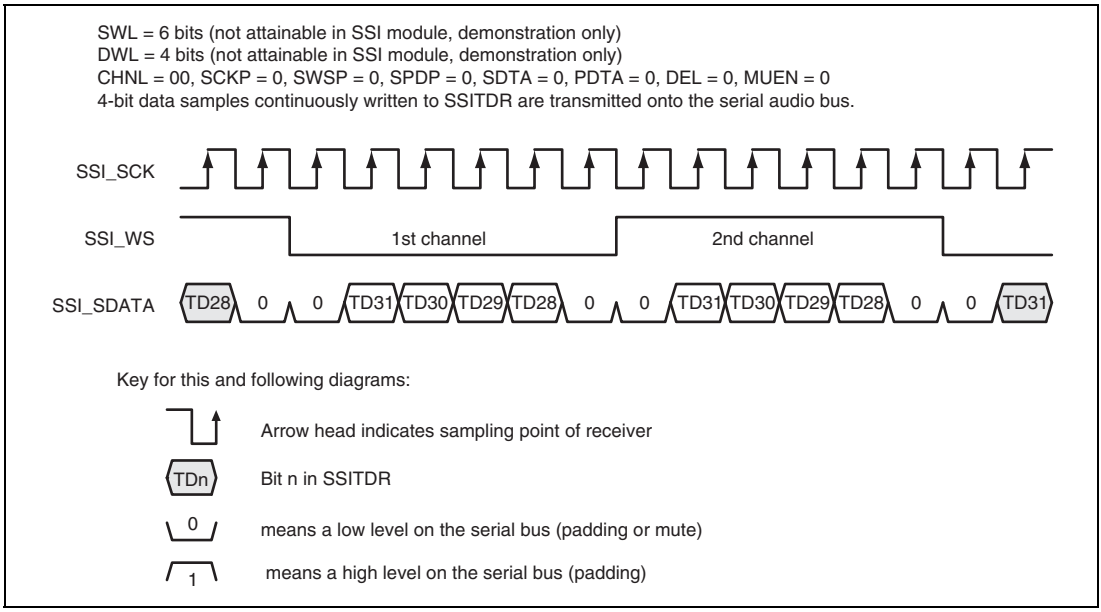


Figure 26.9 Basic Sample Format
(Transmit Mode with Example System/Data Word Length)

In figure 26.9, system word length of 6 bits and a data word length of 4 bits are used. Neither of these are possible with the SSI module but are used only for clarification of the other configuration bits.

1. Inverted Clock

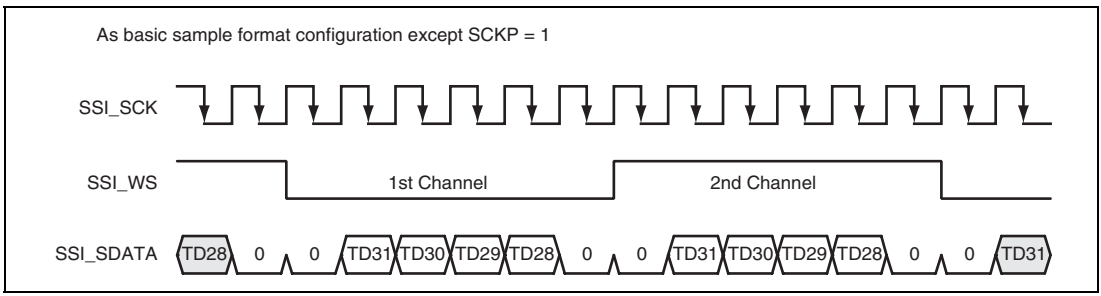


Figure 26.10 Inverted Clock

2. Inverted Word Select

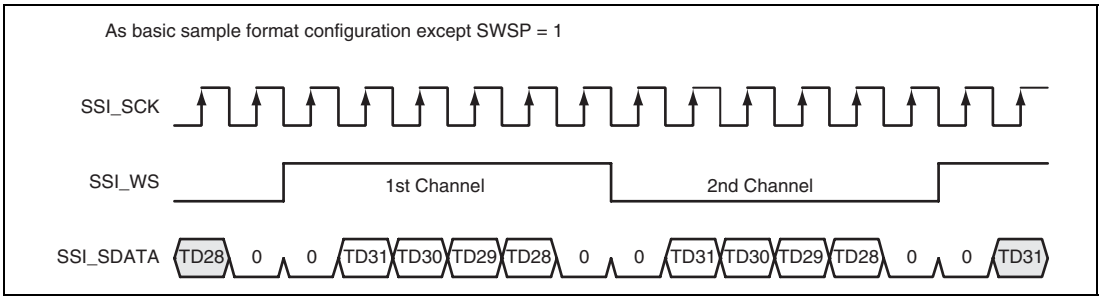


Figure 26.11 Inverted Word Select

3. Inverted Padding Polarity

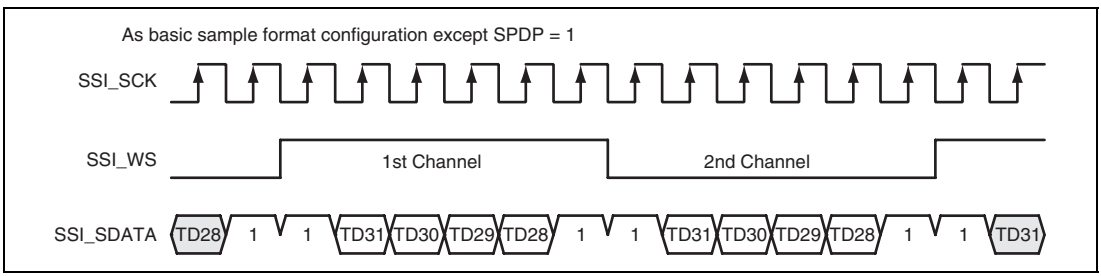


Figure 26.12 Inverted Padding Polarity

4. Padding Bits First, Followed by Serial Data, with Delay

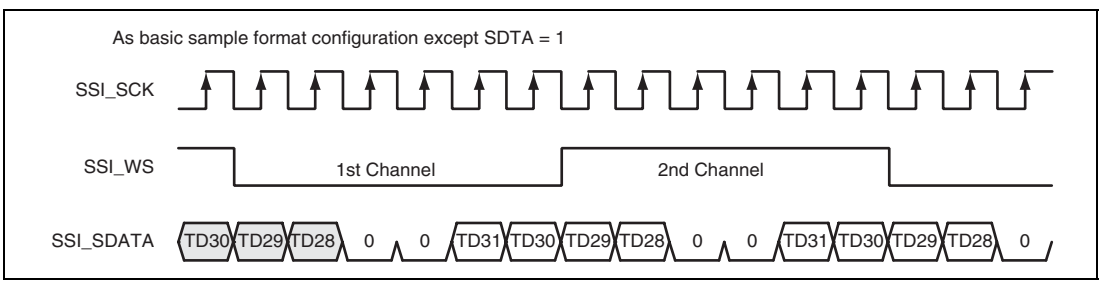


Figure 26.13 Padding Bits First, Followed by Serial Data, with Delay

5. Padding Bits First, Followed by Serial Data, without Delay

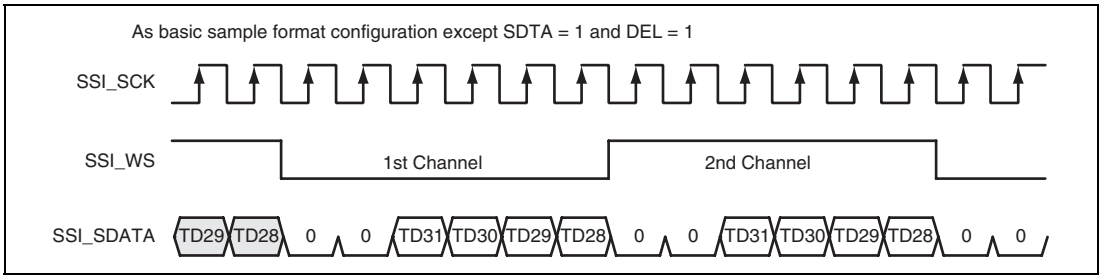


Figure 26.14 Padding Bits First, Followed by Serial Data, without Delay

6. Serial Data First, Followed by Padding Bits, without Delay

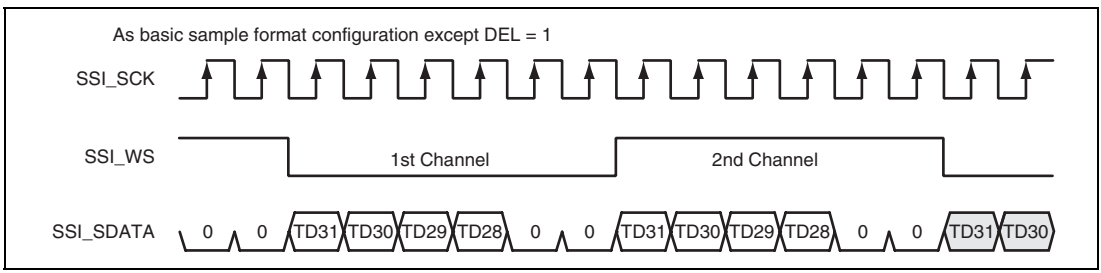


Figure 26.15 Serial Data First, Followed by Padding Bits, without Delay

7. Parallel Right Aligned with Delay

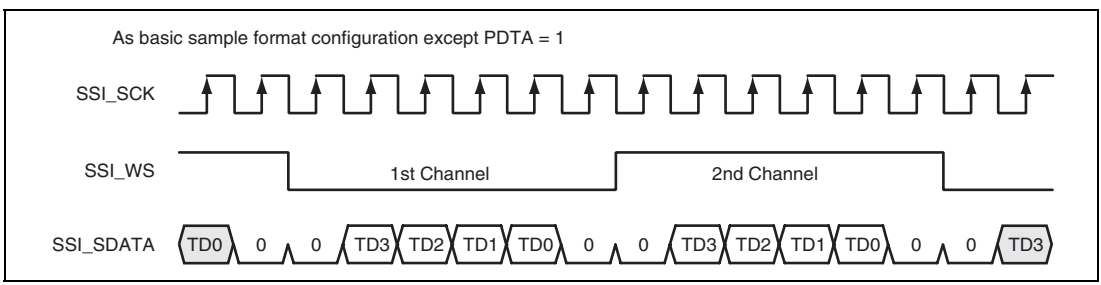


Figure 26.16 Parallel Right Aligned with Delay

8. Mute Enabled

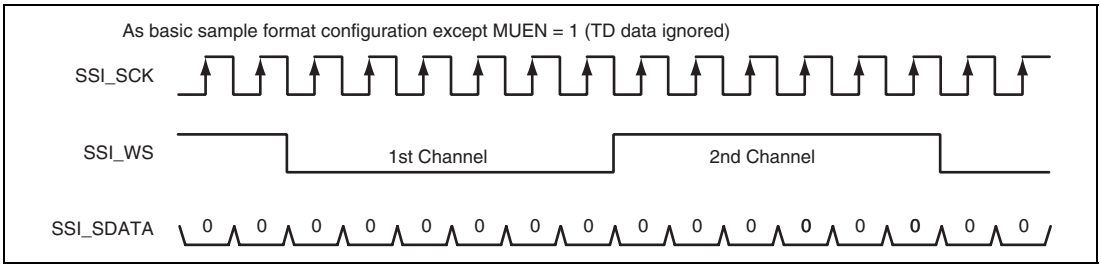


Figure 26.17 Mute Enabled

26.4.3 Compressed Modes

The compressed mode is used to transfer a continuous bit stream. This would typically be a compressed bit stream which requires downstream decoding.

In streaming transfer (burst mode not enabled) there is no concept of a data word. However in order to receive and transmit it is necessary to transfer between the serial bus and word formatted memory. Therefore the word boundary selection is arbitrary during receive/transmit and must be dealt with by another module. When burst mode is enabled then data bits being transmitted can be identified by virtue of the fact that the serial clock output is only activated when there is a word to be output and only the required number of clock pulses necessary to clock out each 32-bit word are generated. The serial bit clock stops at a low level when SSICR.SCKP = 0, and at a high level when SSICR.SCKP = 1. Note burst mode is only valid in the context of the SSI module being a 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 bit is bit 31 when stored in memory.

The word select pin in this mode does not act as a system word start signal as in non-compressed mode, but instead is used to indicate that the receiver can receive another data burst, or the transmitter can transmit another data burst.

Figures 26.18 and 26.19 show the compressed mode data transfer, with burst mode disabled, and enabled, respectively.

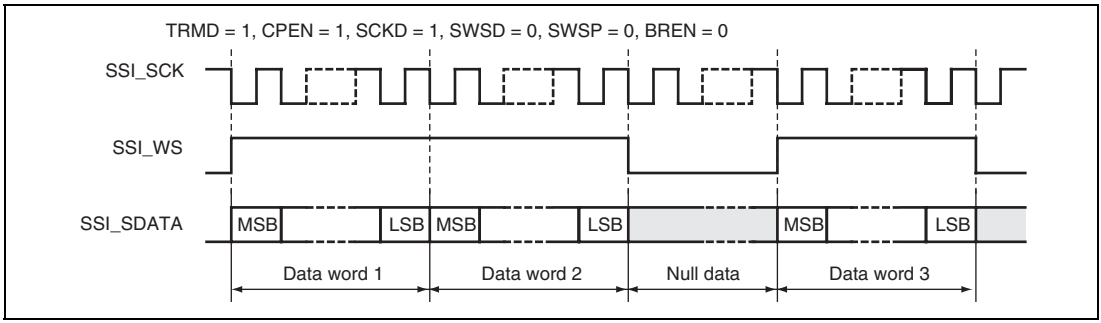


Figure 26.18 Compressed Data Format, Slave Transmitter, Burst Mode Disabled

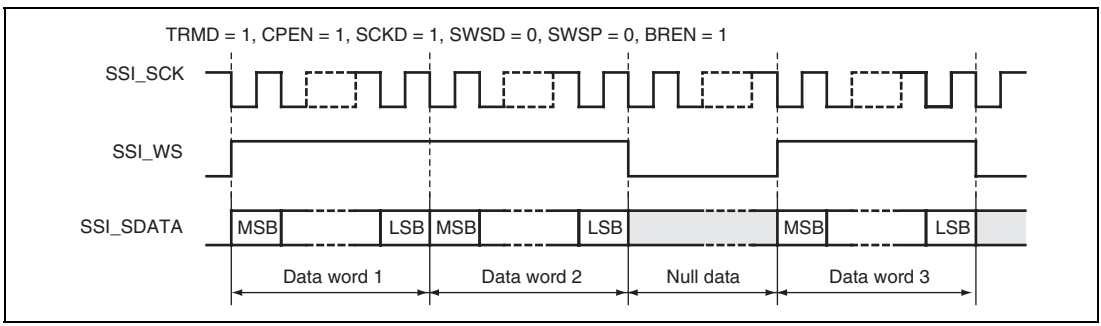


Figure 26.19 Compressed Data Format, Slave Transmitter, and Burst Mode Enabled

(1) Slave Receiver

This mode allows the module to receive a serial bit stream from another device and store it in memory.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an input flow control. Assuming that SWSP = 0 if SSI_WS is high then the module will receive the bit stream in blocks of 32 bits, one data bit per clock. If SSI_WS goes low then the module will complete the current 32-bit block and then stop any further reception, until SSI_WS goes high again.

(2) Slave Transmitter

This mode cannot be used.

(3) Master Receiver

This mode allows the SSI module to receive a serial bit stream from another device and store it in memory.

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 word select signal to indicate it can receive more data continuously. It is the responsibility of the host CPU to ensure it can transmit data to the SSI module in time to ensure no data is lost.

(4) Master Transmitter

This mode allows the module to transmit a serial bit stream from internal memory to another 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 word select signal to indicate it will transmit more data continuously. Word select signal is not asserted until the first word is ready to transmit however. It is the responsibility of the receiving device to ensure 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 DMAC then handles overflow/ underflow interrupts if required.

26.4.4 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 26.20 shows the transition diagram between these operation modes.

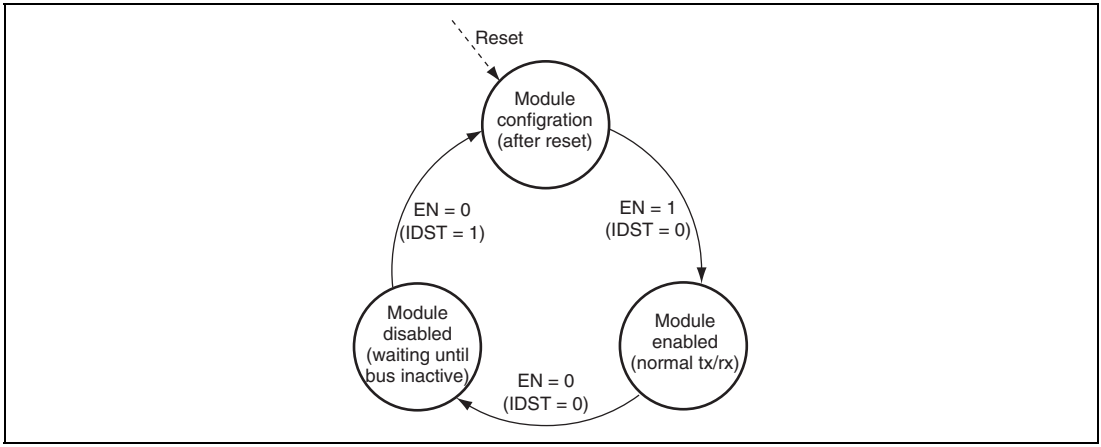


Figure 26.20 Transition Diagram between Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required settings in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

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 details, see section 26.4.5, Transmit Operation and section 26.4.6, Receive Operation.

26.4.5 Transmit Operation

Transmission can be controlled in one of two ways: either DMA or an interrupt driven.

DMA driven is preferred to reduce the CPU load. In DMA control mode, an underflow or overflow of data or DMAC transfer end is notified by using an interrupt.

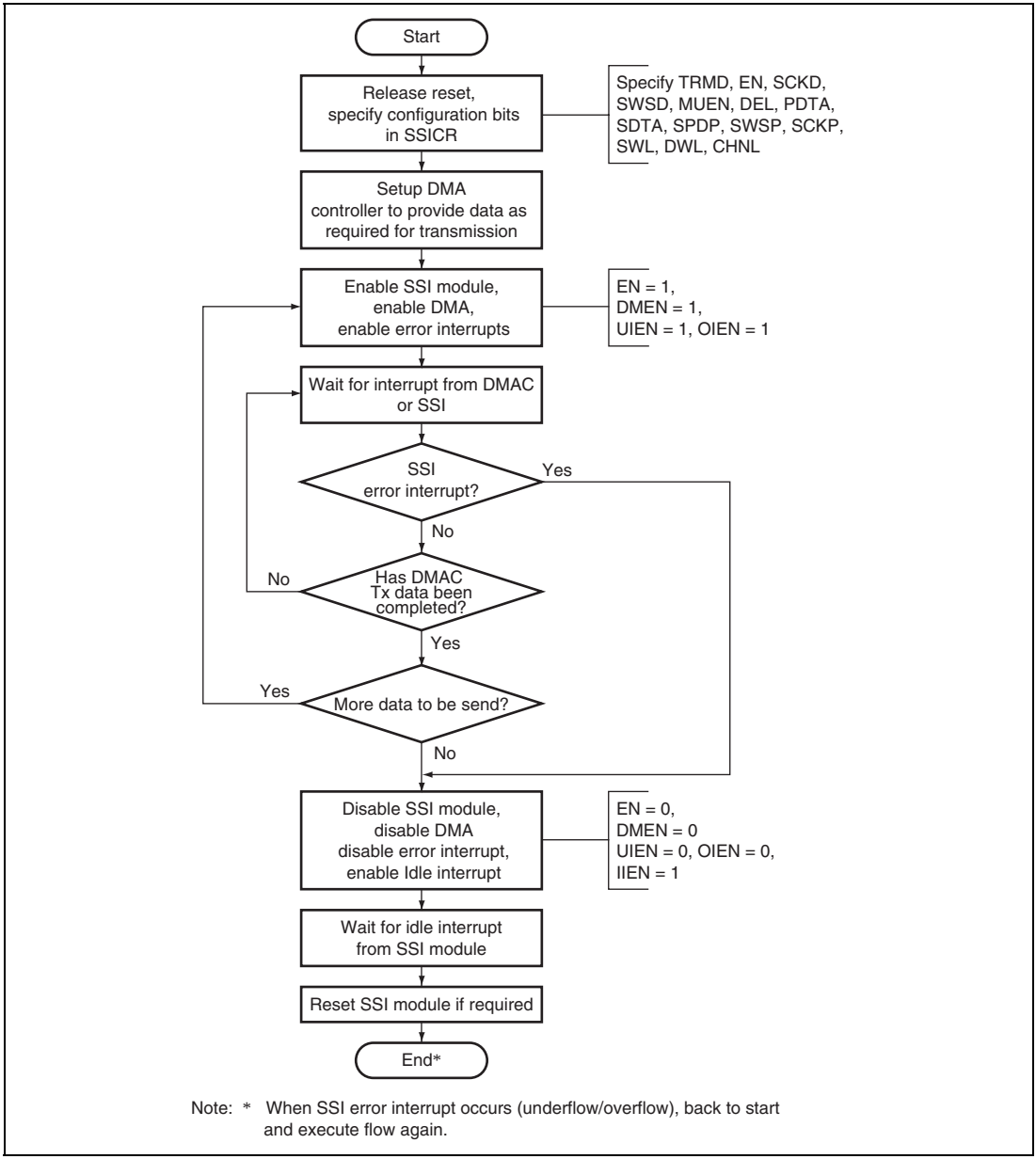
The alternative is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the SSI module is only double buffered and will require data to be written at least every system word period.

When disabling the SSI module, the SSI clock* must be supplied continuously until the module 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 shows the 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

(1) Transmission Using DMA Controller



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 26.21 Transmission Using DMA Controller

(2) Transmission using Interrupt Data Flow Control

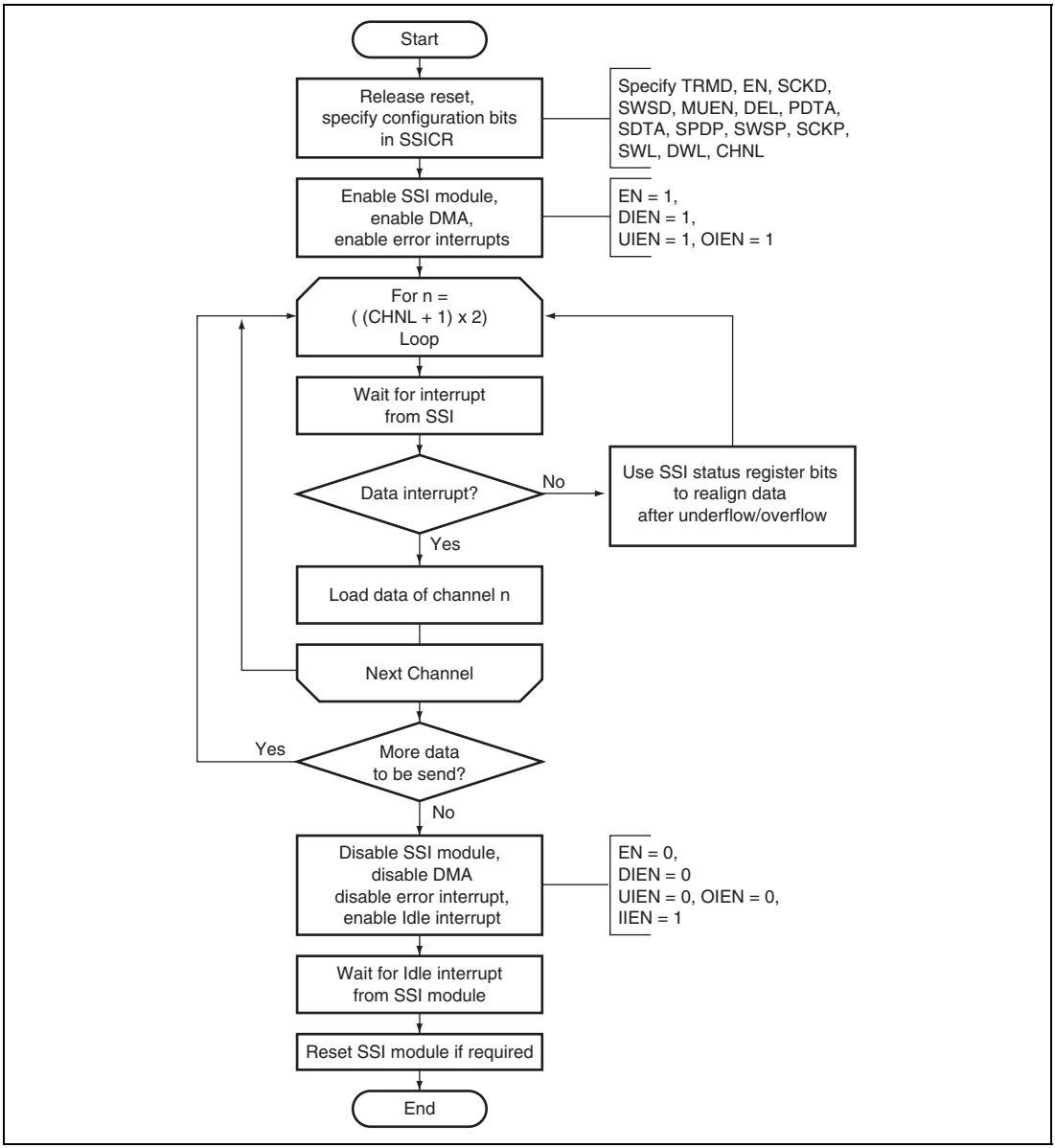


Figure 26.22 Transmission using Interrupt Data Flow Control

26.4.6 Receive Operation

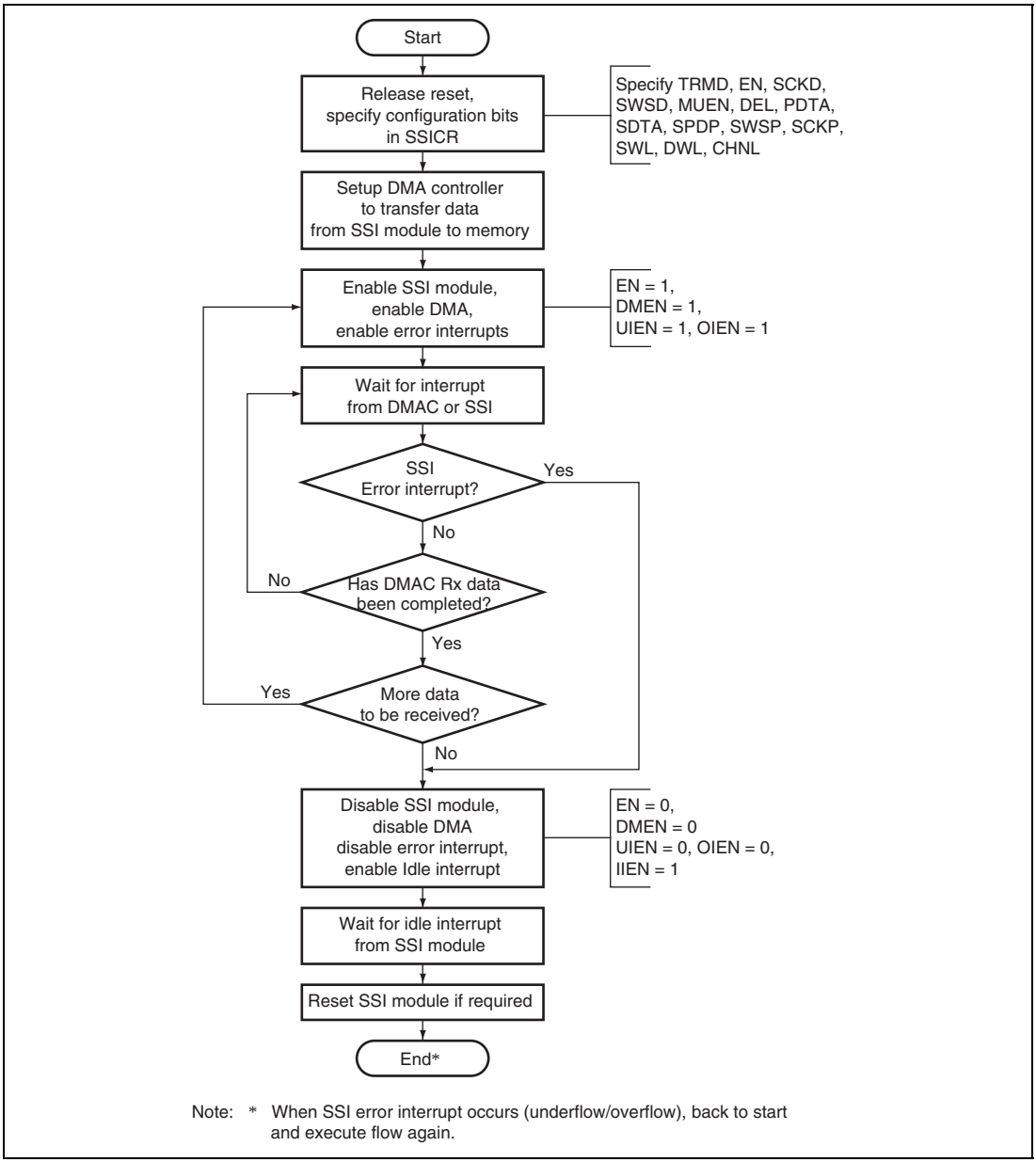
As with transmission the reception can be controlled in one of two ways: either DMA or an interrupt driven.

Figures 26.23 and 26.24 show the flow of operation.

When disabling the SSI module, the SSI clock must be supplied continuously until the module enters in the idle state, which is indicated by the IIRQ bit.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin

(1) Reception Using DMA Controller



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 26.23 Reception using DMA Controller

(2) Reception using Interrupt Data Flow Control

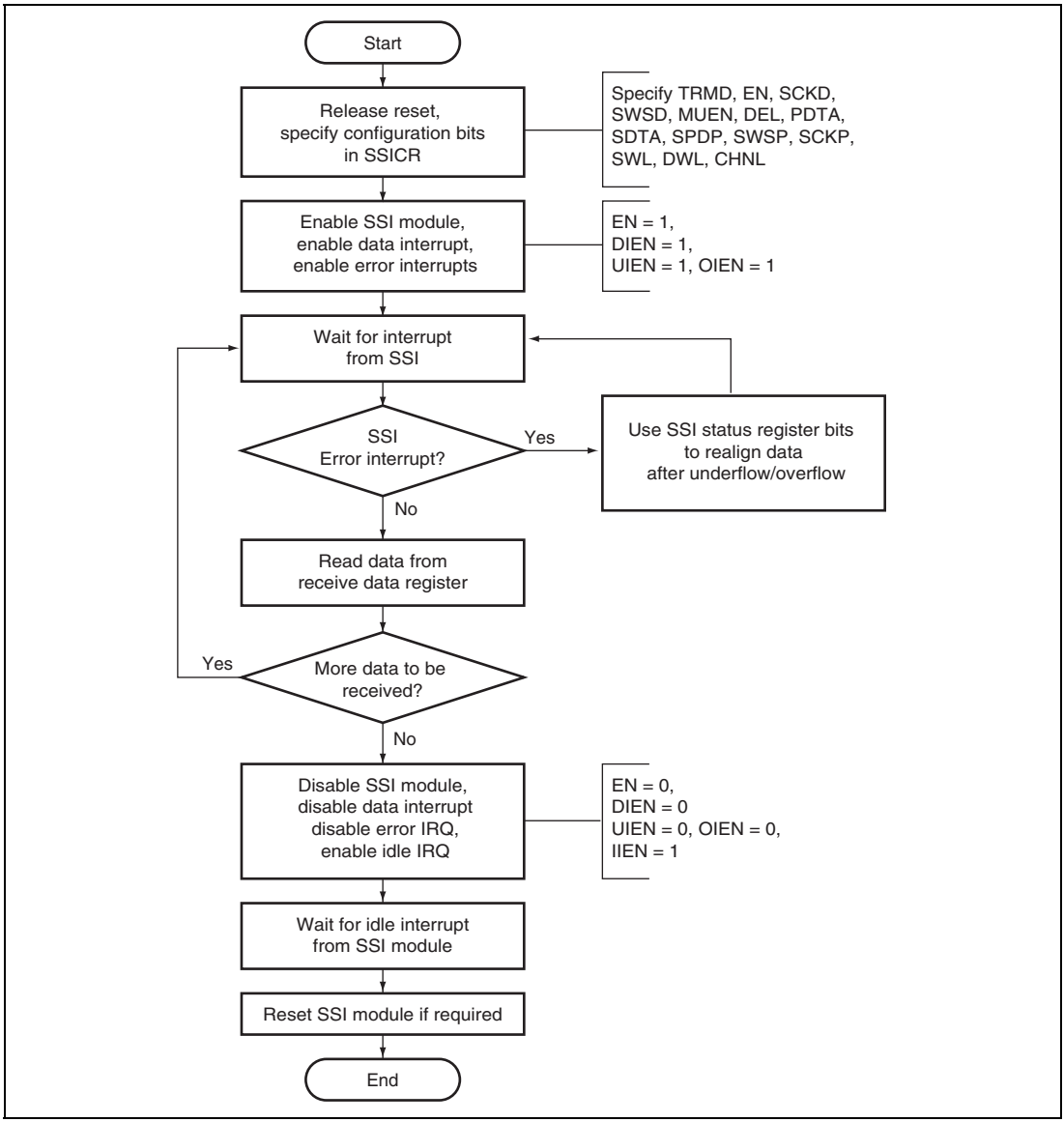


Figure 26.24 Reception using Interrupt Data Flow Control

When an underflow or overflow error condition is met, the CHNO[1:0] and SWNO bits can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the host CPU can read the number of channels and the number of system words to determine what point the serial audio stream has reached. In the transmitter case, the host CPU can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case, the host CPU can skip forward storing null sample data until it is ready to store the sample data that the SSI module is indicating that it will receive next to ensure consistency of the number of received data, and so resynchronize with the audio data stream.

26.4.7 Serial Clock 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 mode, then 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 master mode, and the shift register uses the bit clock derived from the HAC_BIT_CLK input pin or its clock divided. This input clock is then divided by the ratio in the serial oversampling clock division ratio (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.

26.5 Usage Note

26.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation

If an overflow occurs during receive DMA operation, the module must be reactivated.

The receive buffer of SSI has 32-bit common register both left channel and right channel. If an overflow occurs under the condition of control register (SSICR) data-word length (DWL2 to DWL0) is 32-bit and system-word length (SWL2 to SWL0) is 32-bit, SSI has received the data at right channel that should be received at left channel.

If an overflow occurs through an overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR), disable the DMA transfer of the SSI to halt its operation by writing 0 to the EN bit and DMEN bit in SSICR (then terminate the DMA setting). And clear the overflow status flag by writing 0 to the OIRQ bit, set the DMA again and transfer restart.

Section 27 NAND Flash Memory Controller (FLCTL)

The NAND flash memory controller (FLCTL) provides interfaces for an external NAND-type flash memory.

27.1 Features

NAND-Type Flash Memory Interface:

- Read or write in sector* units (512 + 16 bytes)
- 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 NAND-type flash memory. In this manual, an access unit of 512 + 16 bytes is always referred to as 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 from the FLCTL to flash memory, address, and data size to be input or output. Read, write, or erasure of data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in physical sector units by specifying a physical sector. By specifying the number of sectors, the continuous physical sectors can be read or 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 control code 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.

Data Error:

- When a program error or erase error occurs, the error is reflected on the error source flags. Interrupts for each source can be specified.
- When an ECC error is detected by software, perform an error correction, specify another sector to be replaced, and copy the contents of the block to another sector as required.

Data Transfer FIFO:

- The 224-byte FLDTFIFO is incorporated for data transfer of flash memory.
- The 32-byte FLECFIFO is incorporated for data transfer of a control code.
- 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 to the 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 specified access size.
- FIFOs are accessed in 32 bits (4 bytes). Set the byte number for read to a multiple of four, and 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 the QTSEL bit in the common control register (FLCMNCR), regardless of the operating frequency of the peripheral bus.
- The operating clock FCLK on the pins for the NAND-type flash memory is generated by dividing a peripheral clock (Pck).
- In NAND-type flash memory, the $\overline{\text{FRE}}$ and $\overline{\text{FWE}}$ pins operate with the frequency (FCLK) on the pins which common control register (FLCMNCR) designated. To ensure the setup time, this operating frequencies must be specified within the maximum operating frequency of memory to be connected.

Figure 27.1 shows a block diagram of the FLCTL.

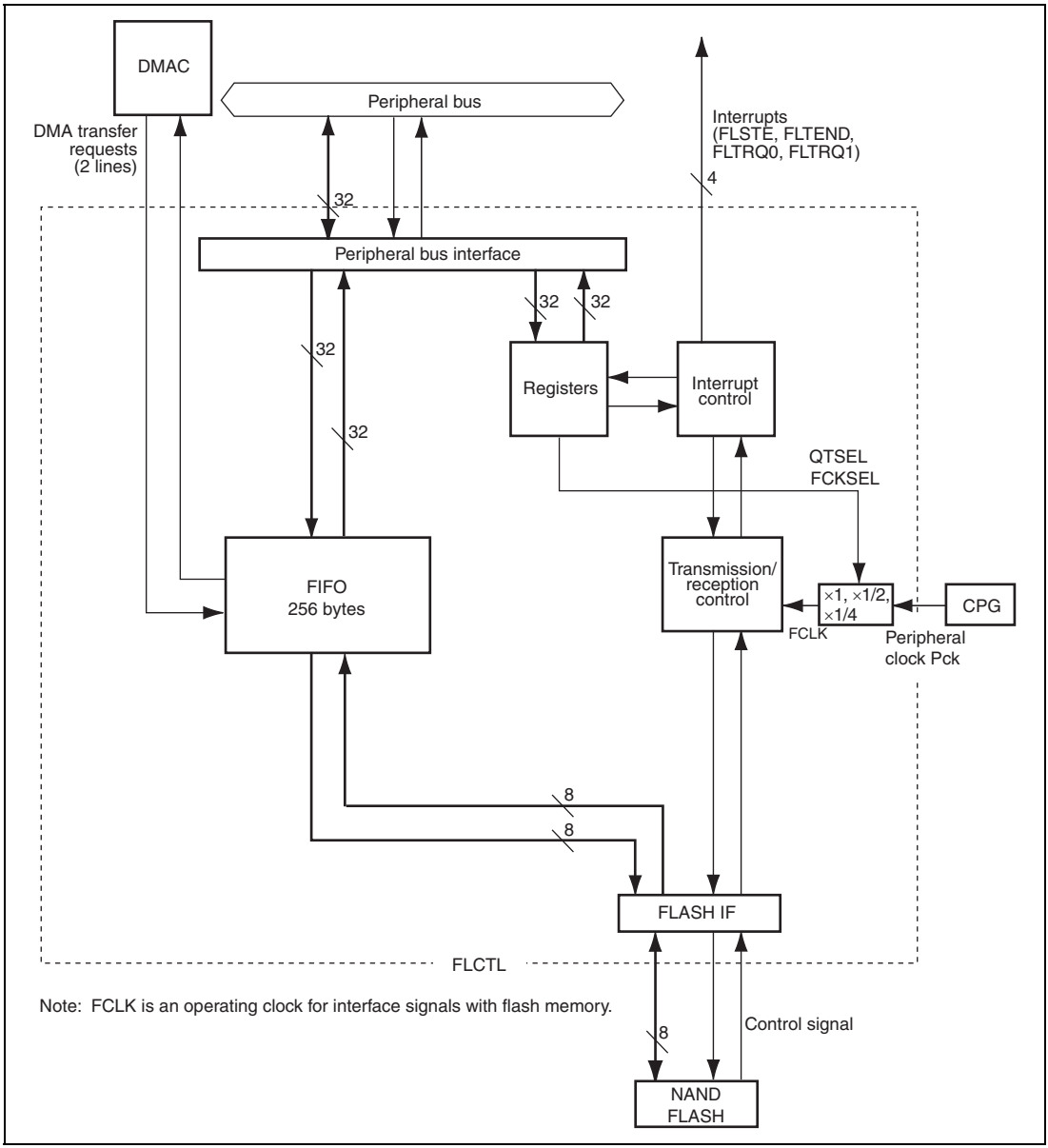


Figure 27.1 FLCTL Block Diagram

27.2 Input/Output Pins

The pin configuration of the FLCTL is listed in table 27.1.

Table 27.1 Pin Configuration

			Corresponding Flash Memory Pin	
Pin Name	**Function**	**I/O**	**NAND Type**	**Description**
$\overline{\text{FCE}}$*1	Chip enable	Output	$\overline{\text{CE}}$	Enables flash memory connected to this LSI.
FD7 to FD0*2	Data I/O pins	I/O	I/O7 to I/O0	I/O pins for command, address, and data.
FCLE*3	Command latch enable	Output	CLE	Command Latch Enable (CLE) Asserted when a command is output.
FALE*1	Output enable	Output	ALE	Address Latch Enable (ALE) Asserted when an address is output and negated when data is input or output.
$\overline{\text{FRE}}$*4	Read Enable	Output	$\overline{\text{RE}}$	Read Enable ($\overline{\text{RE}}$) Reads data at the falling edge of $\overline{\text{RE}}$.
$\overline{\text{FWE}}$*5	Write enable	Output	$\overline{\text{WE}}$	Write Enable Flash memory latches a command, address, and data at the rising edge of $\overline{\text{WE}}$.
FRB *4	Ready/busy	Input	R/$\overline{\text{B}}$	Ready/Busy Indicates ready state at high level; indicates busy state at low level.
—	—	—	$\overline{\text{WP}}$	Write Protect/Reset (Not supported) When this pin goes low, erroneous erasure or programming at power on or off can be prevented.
$\overline{\text{FSE}}$*4	Spare area enable	Output	$\overline{\text{SE}}$	Spare Area Enable Used to access spare area. This pin must be fixed at low in sector access mode.

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 pins.
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 control pin.

27.3 Register Descriptions

Table 27.2 shows the FLCTL register configuration. Table 27.3 shows the register states in each processing mode.

Table 27.2 Register Configuration of FLCTL

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Common control register	FLCMNCR	R/W	H'FFE9 0000	H'1FE9 0000	32
Command control register	FLCMDCR	R/W	H'FFE9 0004	H'1FE9 0004	32
Command code register	FLCMCDR	R/W	H'FFE9 0008	H'1FE9 0008	32
Address register	FLADR	R/W	H'FFE9 000C	H'1FE9 000C	32
Data register	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010	32
Data counter register	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014	32
Interrupt DMA control register	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018	32
Ready busy timeout setting register	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C	32
Ready busy timeout counter	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020	32
Data FIFO register	FLDTFIFO	R/W	H'FFE9 0024	H'1FE9 0024	32
Control code FIFO register	FLECFIFO	R/W	H'FFE9 0028	H'1FE9 0028	32
Transfer control register	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C	8

Table 27.3 Register States of FLCTL in Each Processing Mode

Register Abbreviation	Power-On Reset	Manual Reset	Module Standby	Sleep
FLCMNCR	H'0000 0000	H'0000 0000	Retained	Retained
FLCMDCR	H'0000 0000	H'0000 0000	Retained	Retained
FLCMCDR	H'0000 0000	H'0000 0000	Retained	Retained
FLADR	H'0000 0000	H'0000 0000	Retained	Retained
FLDATAR	H'0000 0000	H'0000 0000	Retained	Retained
FLDTCNTR	H'0000 0000	H'0000 0000	Retained	Retained
FLINTDMACR	H'0000 0000	H'0000 0000	Retained	Retained
FLBSYTMR	H'0000 0000	H'0000 0000	Retained	Retained
FLBSYCNT	H'0000 0000	H'0000 0000	Retained	Retained
FLDTFIFO	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
FLECFIFO	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
FLTRCR	H'00	H'00	Retained	Retained

27.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies the type (NAND) of flash memory, access mode, and $\overline{\text{FCE}}$ pin output.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QTSEL	—
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	R	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCK SEL	—	ECCPOS[1:0]		ACM[1:0]		NAND WF	SE	—	—	—	—	CE0	—	—	TYPE SEL
Initial value:	0	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	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 from the CPG by four and uses it as FCLK when FCKSEL = 0 Note: When FCKSEL = 1, setting to 1 to this bit is prohibited.
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
15	FCKSEL	0	R/W	Flash Clock Select 0: Divides a peripheral clock (Pck) provided from the CPG by two and uses it as FCLK 1: Uses a peripheral clock (Pck) provided from the CPG as FCLK
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13, 12	ECCPOS [1:0]	00	R/W	ECC Position Specification 1 and 0 Specify the position (0/4th/8th byte) to place the ECC in the control code area. 00: Places the ECC at the 0 to 7th byte of control code area 01: Places the ECC at the 4th to 11th byte of control code area 10: Places the ECC at the 8th to 15th byte of control code area 11: Setting prohibited
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 one FCLK cycle 1: Performs address or data input/output in two FCLK cycles
8	SE	0	R/W	Spare Area (control code area) Enable bit 0: Spare area access enable (can be access the data 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 value should always be 0.
3	CE0	0	R/W	Chip Enable 0 0: Disables the chip (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enables the chip (Outputs low level to the $\overline{\text{FCE}}$ pin)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should 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.

27.3.2 Command Control Register (FLCMDCR)

FLCMDCR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies source or destination of data transfer. In sector access mode, FLCMDCR specifies the number of sector transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADRMD	CDSRC	DOSR	—	—	SELRW	DOADR	ADRCNT[1:0]		DOCMD2	DOCMD1
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/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	1	0
	SCTCNT[15:0]															
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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. This bit is valid only in sector access mode. 0: The value of the address register is handled as a physical sector number. Use this value usually in sector access. 1: The value of the address register is output as the address of flash memory. Note: Clear this bit to 0 in continuous sector access.
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or written to in 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 performed after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read

Bit	Bit Name	Initial Value	R/W	Description
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SELRW	0	R/W	Data Read/Write Specification Specifies the direction of read or write in data stage. 0: Read 1: Write
20	DOADR	0	R/W	Address Stage Execution Specification Specifies whether or not the address stage is executed 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 data to be 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 Specification Specifies whether or not the second command stage* is executed in command access mode. 0: Does not execute the second command stage 1: Executes the second command stage
16	DOCMD1	0	R/W	First Command Stage Execution Specification Specifies whether or not the first command stage* is 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 continuously in sector access mode. These bits are counted down for each sector transfer end and stop when they reach 0. In command access mode, these bits become H'0001. When accessing one sector, set H'0001 to the SCTCNT.

Note: * Refer to figure 27.2 for command stage, address stage and data stage.

27.3.3 Command Code Register (FLCMCDR)

FLCMCDR is a 32-bit readable/writable register that specifies a command to be issued in command access or sector access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD[15:8]								CMD[7:0]							
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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	CMD[15:8]	H'00	R/W	Specify a command code to be issued in the second command stage.
7 to 0	CMD[7:0]	H'00	R/W	Specify a command code to be issued in the first command stage.

27.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies an address to be output in command access mode. In sector access mode, a physical sector number specified in the physical sector 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	17	16
	ADR[31:24]								ADR[23:16]							
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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:8]								ADR[7:0]							
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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR[31:24]	H'00	R/W	Fourth Address Data Specify 4th data to be output to flash memory as an address in command access mode.
23 to 16	ADR[23:16]	H'00	R/W	Third Address Data Specify 3rd data to be output to flash memory as an address in command access mode.
15 to 8	ADR[15:8]	H'00	R/W	Second Address Data Specify 2nd data to be output to flash memory as an 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 as an address in command access mode.

- Sector Access Mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADR[17:16]	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0
R/W:	R	R	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	1	0
	ADR[15:0]															
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/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	Undefined	R	Reserved These bits are always read as an undefined value (depends on the FLCTL operation mode). The write 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 accessed in sector access mode. The physical sector number is converted into an address and is output to flash memory.

27.3.5 Data Counter Register (FLDTCNTR)

FLDTCNTR is a 32-bit readable/writable register that specifies the number of bytes to be read or written in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[7:0]							
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DTCNT[11:0]											
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/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	ECFLW[7:0]	H'00	R	FLECFIFO Access Count Specify the number of longwords (4-byte) in FLECFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLECFIFO. In FLECFIFO read, these bits specify the number of longwords of the data that can be read from FLECFIFO. In FLECFIFO write, these bits specify the number of 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 FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO. In FLDTFIFO read, these bits specify the number of longwords of the data that can be read from FLDTFIFO. In FLDTFIFO write, these bits specify the number of 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 should 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 or written in command access mode. (Up to 2048 + 64 bytes can be specified.)

27.3.6 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores input/output data used when 0 is written to the CDSRC bit in FLCMDCR in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT[31:24]								DT[23:16]							
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/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT[15:8]								DT[7:0]							
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/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 the FD7 to 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 the FD7 to 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 the FD7 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 the FD7 to FD0 pins. In write: Specify write data In read: Store read data

27.3.7 Interrupt DMA Control Register (FLINTDMACR)

FLINTDMACR is a 32-bit readable/writable register that enables or disables DMA transfer requests or interrupts. A transfer request from the FLCTL to the DMAC is issued after each access mode has been started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FIFOTRG[1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN
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/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	1	0
	—	—	—	—	—	—	—	STE RB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
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/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 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Change the condition for the FIFO transfer request. In flash-memory read: 00: Issue an interrupt to the CPU or a DMA transfer request to the DMAC when FLDTFIFO stores 4 bytes of data. 01: Issue an interrupt to the CPU or a DMA transfer request to the DMAC when FLDTFIFO stores 16 bytes of data. 10: Issue an interrupt to the CPU or a DMA transfer request to the DMAC when FLDTFIFO stores 128 bytes of data. 11: Issue an interrupt to the CPU when FLDTFIFO stores 128 bytes of data, or issue a DMA transfer request to the DMAC when FLDTFIFO stores 16 bytes of data. In flash-memory programming: 00: Issue an interrupt to the CPU when FLDTFIFO has empty area of 4 bytes or more (do not set DMA transfer). 01: Issue an interrupt or a DMA transfer request to the CPU when FLDTFIFO has empty area of 16 bytes or more. 10: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more (do not set DMA transfer). 11: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more, or issue a DMA transfer request to the CPU when FLDTFIFO has 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 FLECFIFO. In flash-memory access, this bit should be cleared to 0. 1: Clears the address counter of FLECFIFO. After clearing the counter, this bit should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
18	AC0CLR	0	R/W	FLDTFIFO Clear Clears the address counter of FLDTFIFO. 0: Retains the address counter value of FLDTFIFO. In flash-memory access, this bit should be cleared to 0. 1: Clears the address counter of FLDTFIFO. After clearing the counter, this bit should be cleared to 0.
17	DREQ1EN	0	R/W	FLECFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLECFIFO. 0: Disables the DMA transfer request issued from FLECFIFO 1: Enables the DMA transfer request issued from FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLDTFIFO. 0: Disables the DMA transfer request issued from the FLDTFIFO 1: Enables the DMA transfer request issued from the FLDTFIFO
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	—	0	R	Reserved Although the initial value is 0, this bit will be read as an undefined value. The write value should always be 0.
8	STERB	0	R/W	Status Error Indicates the result of status read. This bit is set to 1 if the specific bit in the bits STAT[7:0] in FLBSYCNT is set to 1 in status read. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no status error occurs (the specific bit 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 STAT0 bits, see section 27.4.5, Status Read.

Bit	Bit Name	Initial Value	R/W	Description
7	BTOERB	0	R/W	Timeout Error This bit is set to 1 if a timeout error occurs (the bits RBTIMCNT[19:0] in FLBSYCNT are decremented to 0). This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no timeout error occurs 1: Indicates that a timeout error occurs
6	TRREQF1	0	R/W	FLECFIFO Transfer Request Flag Indicates that a transfer request is issued from FLECFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLECFIFO 1: Indicates that a transfer request is issued from FLECFIFO
5	TRREQF0	0	R/W	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued from FLDTFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLDTFIFO 1: Indicates that a transfer request is issued from FLDTFIFO
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to the CPU when a status error has occurred. 0: Disables the interrupt request to the CPU by a status error 1: Enables the interrupt request to the CPU by a status error

Bit	Bit Name	Initial Value	R/W	Description
3	BTOINTE	0	RW	Interrupt Enable at Timeout Error Enables or disables an interrupt request to the CPU when a timeout error has occurred. 0: Disables the interrupt request to the CPU by a timeout error 1: Enables the interrupt request to the CPU by a timeout error
2	TEINTE	0	R/W	Transfer End Interrupt Enable Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR). 0: Disables the transfer end interrupt request to the CPU 1: Enables the transfer end interrupt request to the CPU
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLECFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO. When the DMA transfer is enabled, this bit should be cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLDTFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO When the DMA transfer is enabled, this bit should be cleared to 0.

27.3.8 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the FRB pin is busy.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RBTMOUT[19:16]			
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
	RBTMOUT[15:0]															
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/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 value 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 Pck cycles

27.3.9 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory obtained by the status read is stored in the bits STAT[7:0].

The timeout time set in the bits RBTMOUT[19:0] in FLBSYTMR is copied to the bits RBTIMCNT[19:0] and counting down is started when the FRB pin is placed in a busy state. When values in the RBTIMCNT[19:0] become 0, 1 is set to the BTOERB bit in FLINTDMACR, thus notifying that a timeout error has occurred. In this case, an FLSTE interrupt request can be issued if an interrupt is enabled by the RBERINTE bit in FLINTDMACR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								—	—	—	—	RBTIMCNT[19:16]			
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	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTIMCNT[15:0]															
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	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 by the 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, the values of the bits RBTMOUT[19:0] in FLBSYTMR are copied to these bits. These bits are counted down while the FRB pin is busy. A timeout error occurs when these bits are decremented to 0.

27.3.10 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read or write the data FIFO area.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTFO[31:24]								DTFO[23:16]							
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

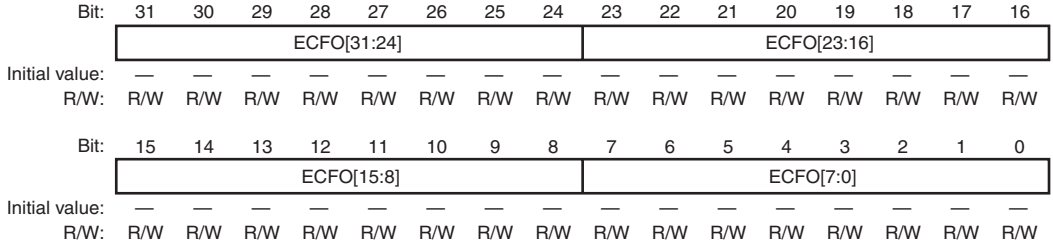
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTFO[15:8]								DTFO[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 FD7 to 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 FD7 to 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 FD7 to 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 FD7 to FD0 pins. In write: Specify write data In read: Store read data

27.3.11 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read or write the control code FIFO area.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register.



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 FD7 to 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 FD7 to 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 FD7 to 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 FD7 to FD0 pins. In write: Specify write data In read: Store read data

27.3.12 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 initiates access to flash memory. Access completion can be checked by the TREND bit.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TREND	TRSTRT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TREND	0	R/W	Processing End Flag Bit Indicates that the processing performed in the specified access mode has been completed. The write value should always be 0.
0	TRSTRT	0	R/W	Transfer Start By setting this bit from 0 to 1 when the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

27.4 Operation

27.4.1 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

27.4.2 Command Access Mode

Command access mode accesses flash memory by specifying a command to be issued to flash memory, address, data, read or write direction, and number of times to the registers. In this mode, 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 for NAND-type flash memory. In this example, the first command is specified as H'00, address data length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the data counter.

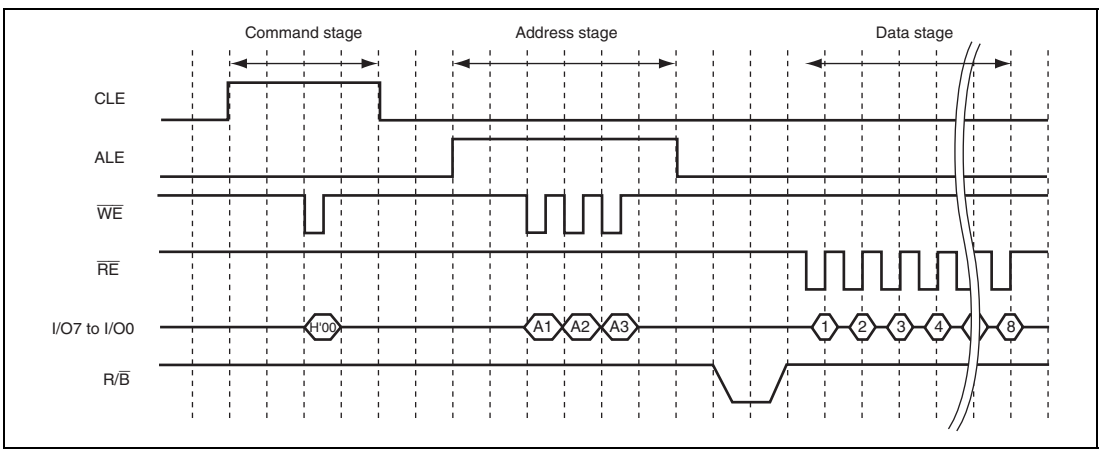


Figure 27.2 Read Operation Timing for NAND-Type Flash Memory (1)

Figures 27.3 and 27.4 show examples of programming operation for NAND-type flash memory.

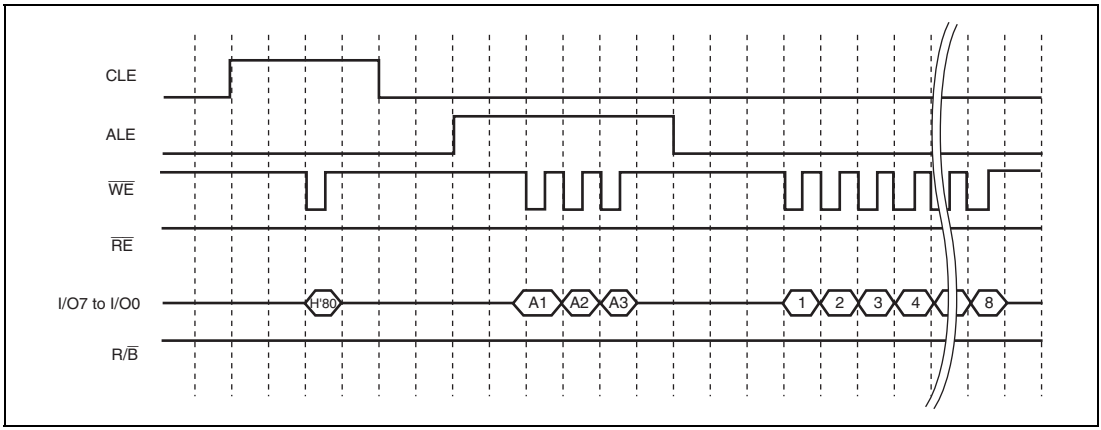


Figure 27.3 Programming Operation Timing for NAND-Type Flash Memory (1)

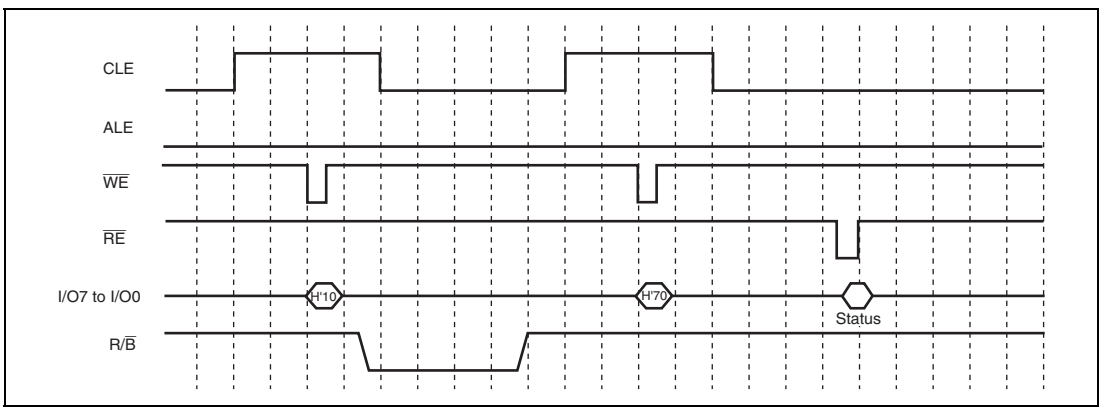


Figure 27.4 Programming Operation Timing for NAND-Type Flash Memory (2)

27.4.3 Sector Access Mode

In sector access mode, flash memory can be read or programmed in sector units by specifying the number of physical sectors to be accessed.

Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, the DREQ1EN and DREQ0EN bits in FLINTDMACR can be set to transfer by the DMA.

Figure 27.5 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory on the address space.

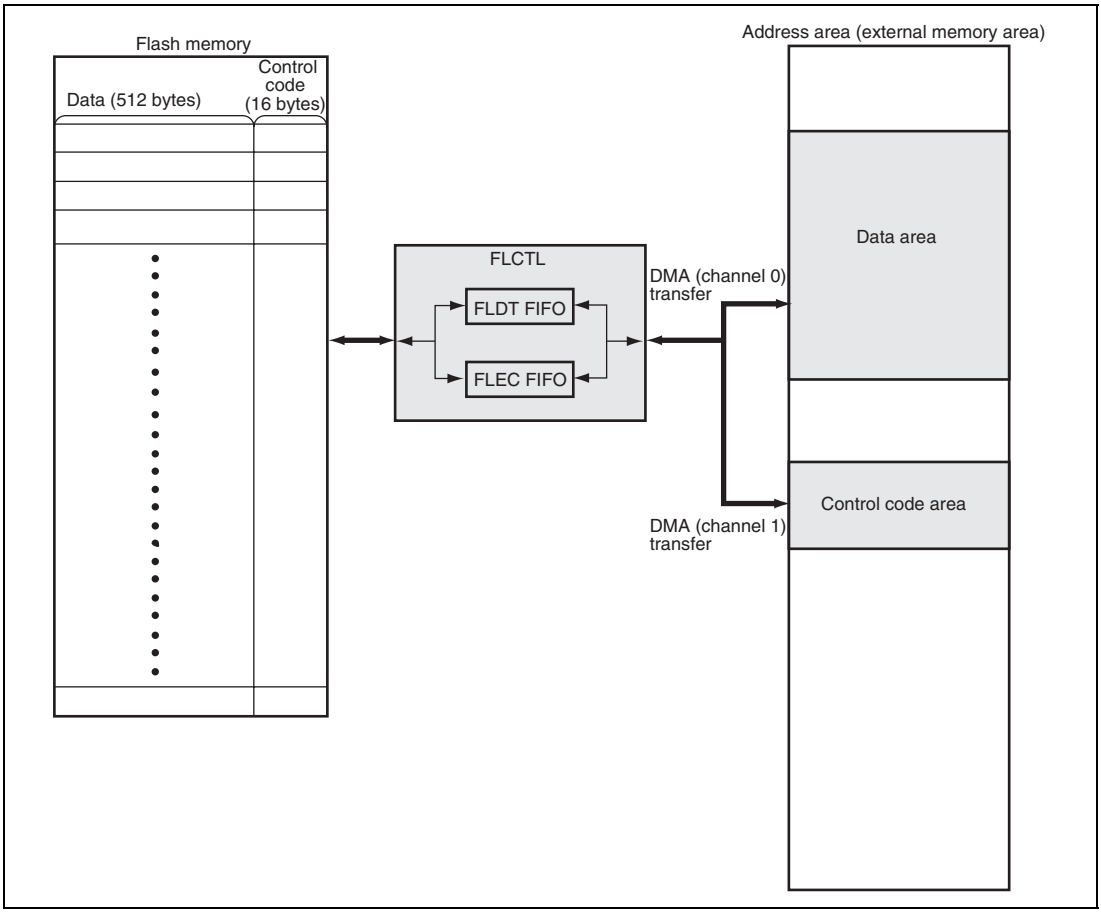


Figure 27.5 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer

Physical Sector: Figure 27.6 shows the relationship between the physical sector address of NAND-type flash memory and the address of flash memory.

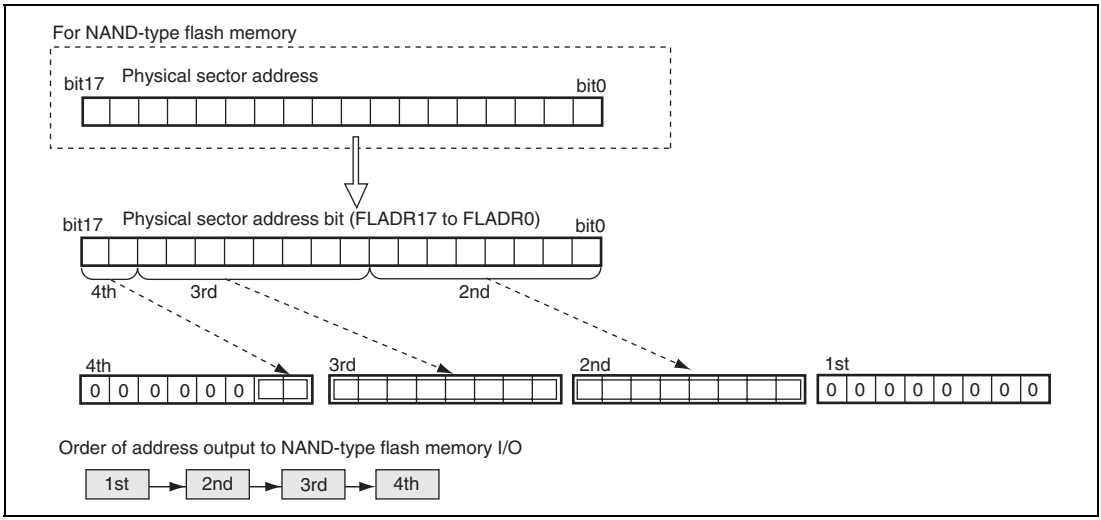


Figure 27.6 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory

Continuous Sector Access: Continuous physical sectors can be read or written by specifying the start physical sector of NAND-type flash memory and the number of sectors to be transferred. Figure 27.7 shows an example of physical sector specification register and transfer count specification register settings when transferring logical sectors 0 to 40, which are not contiguous because of an unusable sector in NAND-type flash memory.

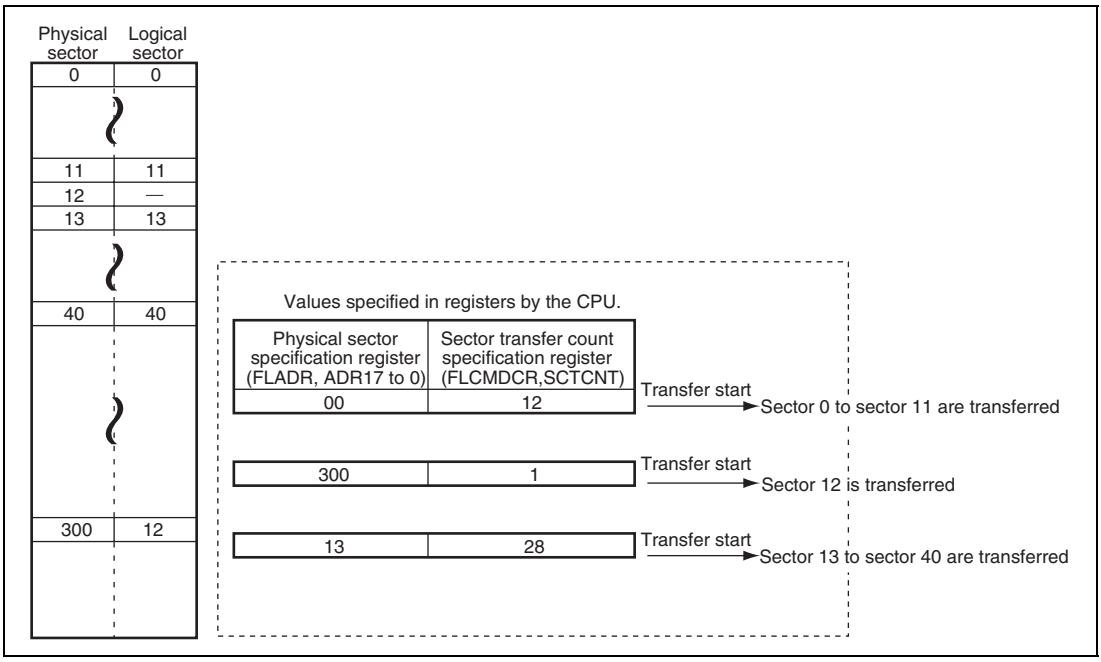


Figure 27.7 Sector Access when Unusable Sector Exists in Continuous Sectors

27.4.4 ECC Error Correction

The FLCTL does not perform ECC processing. An ECC generation, error detection and correction must be performed by software.

27.4.5 Status Read

The FLCTL can read the status register of a NAND-type flash memory. The data in the status register of a NAND-type flash memory is input through the I/O7 to I/O0 pins and stored in the bits STAT[7:0] in FLBSYCNT. The bits STAT[7:0] in FLBSYCNT can be read by the CPU. If a program error or erase error is detected when the status register value is stored in the bits STAT[7:0] in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled.

Status Read of NAND-Type Flash Memory: The status register of NAND-type flash memory can be read by inputting command H'70 to NAND-type flash memory. If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically inputs command H'70 to NAND-type flash memory and reads the status register of NAND-type flash memory. When the status register of NAND-type flash memory is 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

27.5 Example of Register Setting

Figure 27.8 to 28.10 show examples of register setting and processing flow in each access mode.

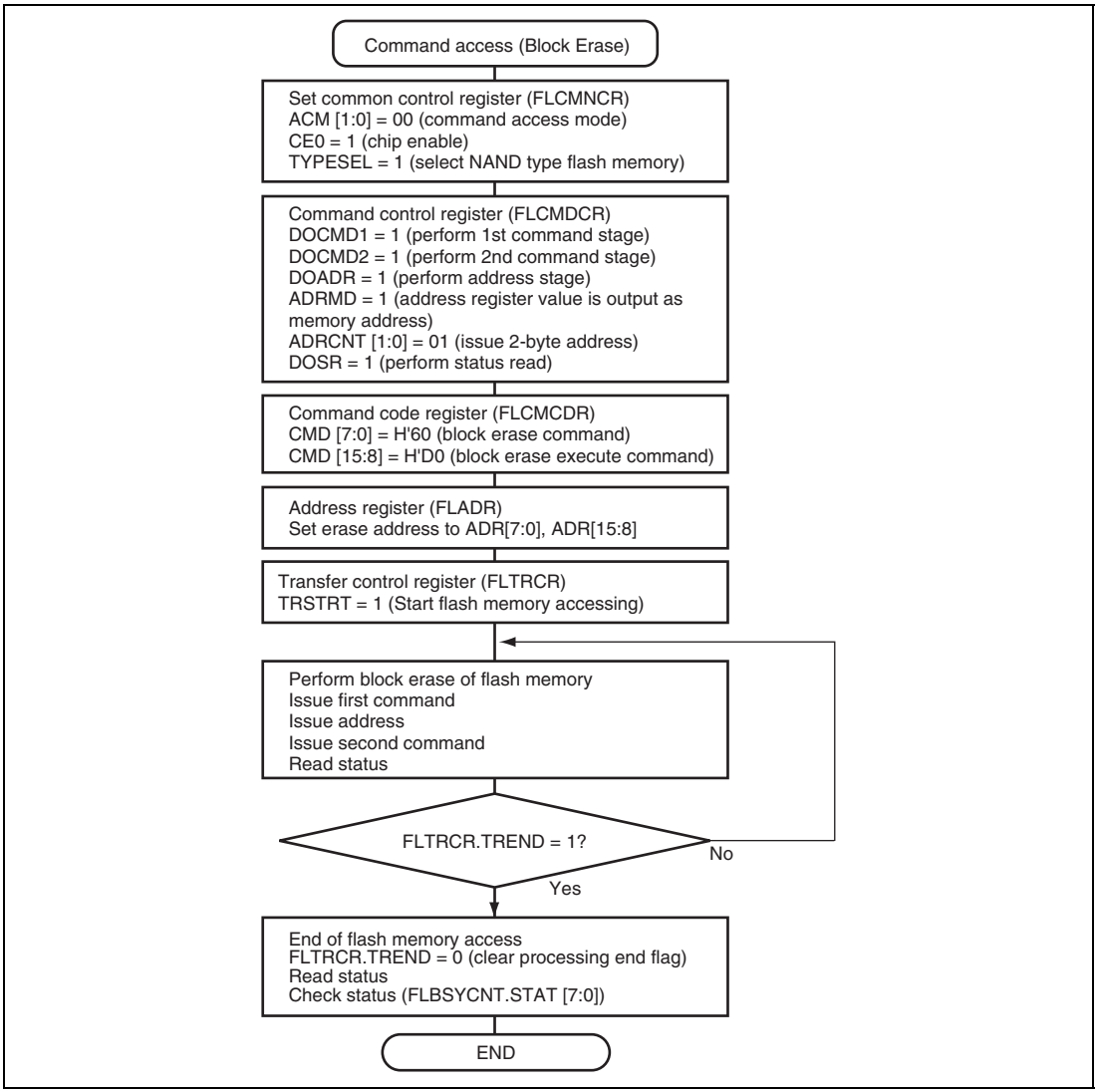


Figure 27.8 NAND Flash Command Access (Block Erase)

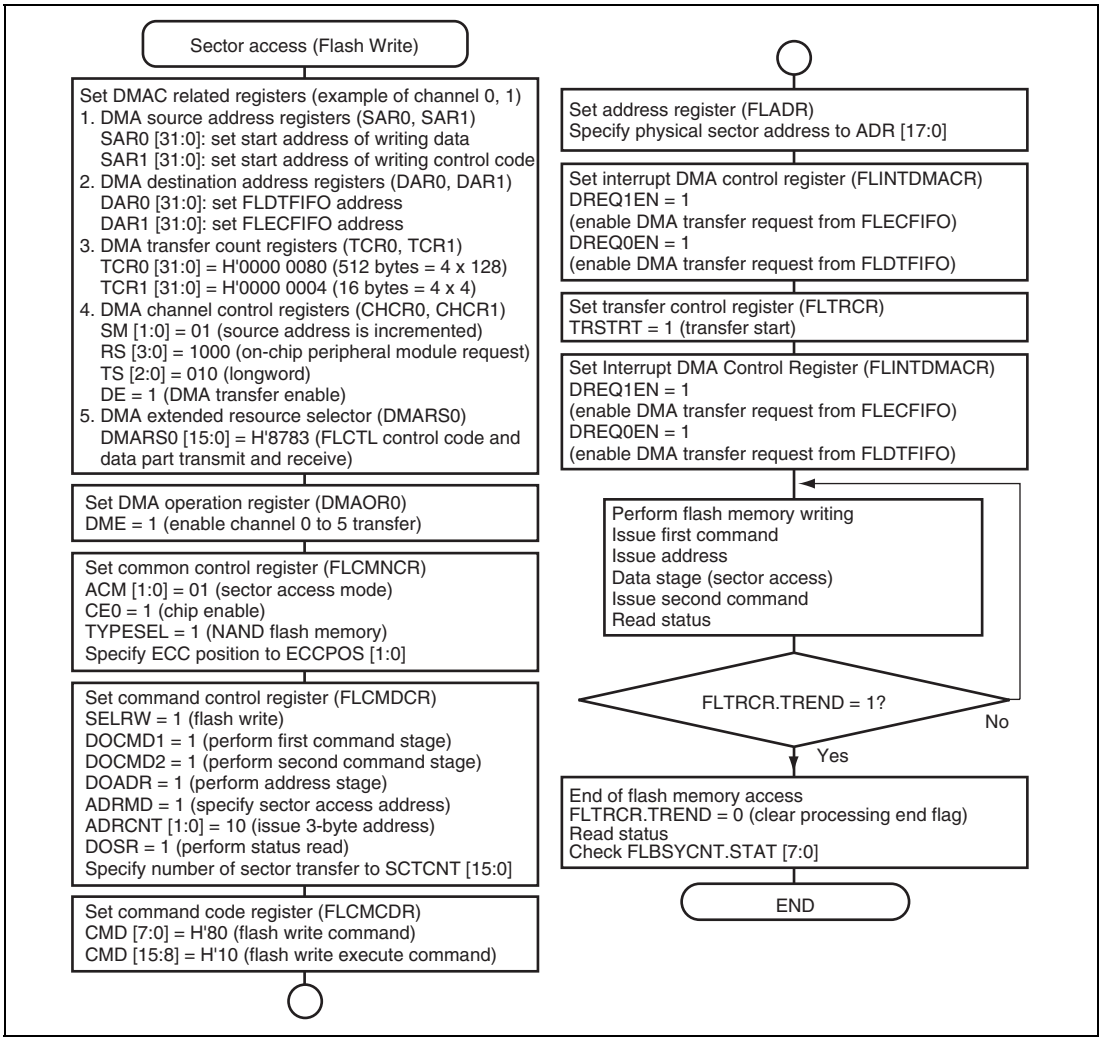


Figure 27.9 NAND Flash Sector Access (Flash Write) Using DMA

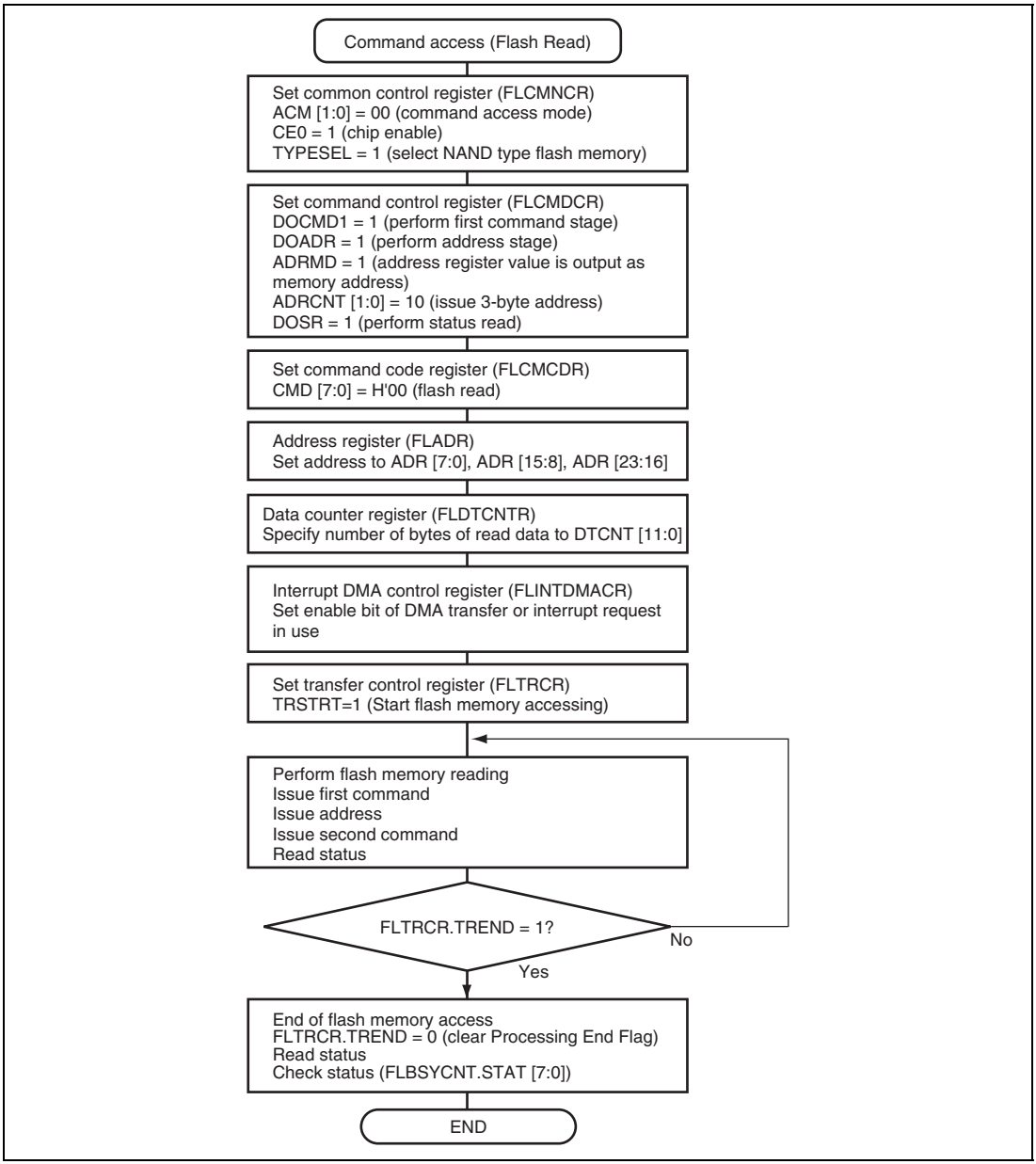


Figure 27.10 NAND Flash Command Access (Flash Read)

27.6 Interrupt Sources

The FLCTL has six interrupt sources: Status error, ready/busy timeout error, ECC error, transfer end, FIFO0 transfer request, and FIFO1 transfer request. Each of the interrupt sources has its corresponding interrupt flag and the interrupt can be requested independently to the CPU if the interrupt is enabled by the interrupt enable bit. Note that the status error and ready/busy timeout error use the common FLSTE interrupt to the CPU.

Table 27.5 FLCTL Interrupt Requests

Interrupt Source	Interrupt Flag	Enable Bit	Description
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/underrun error 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 control code area FLECFIFO. Table 27.6 summarizes DMA transfer enable or disable states in each access 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 (DMAC).

Section 28 General Purpose I/O (GPIO)

28.1 Features

This LSI has twelve general ports (A to H, J to M), which provide 75 input/output pins and 8 output pins in total.

Each port pins is multiplexed pin with on-chip modules, selected of use whether General Purpose I/Os (GPIO) or on-chip modules by port control register (PACR to PHCR and PJCR to PMCR) and on-chip module select register (OMSELR).

The GPIO has the following features.

- Each port pin is multiplexed pin, for which the port control register can set the pin function and 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 section 10, Interrupt Controller (INTC).

Table 28.1 Multiplexed Pins Controlled by Port Control Registers

Pin Name	Port	GPIO	Multiplexed Module	GPIO Interrupt
AD31	A	PA7 input/output	PCIC	
AD30	A	PA6 input/output	PCIC	
AD29	A	PA5 input/output	PCIC	
AD28	A	PA4 input/output	PCIC	
AD27	A	PA3 input/output	PCIC	
AD26	A	PA2 input/output	PCIC	
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	

Pin Name	Port	GPIO	Multiplexed Module	GPIO Interrupt
AD1	D	PD1 input/output	PCIC	
AD0	D	PD0 input/output	PCIC	
IRQ/$\overline{\text{IRL7}}$/FD7	E	PE6 input/output	INTC/FLCTL	Available
$\overline{\text{REQ1}}$	E	PE5 input/output	PCIC	Available
$\overline{\text{REQ2}}$	E	PE4 input/output	PCIC	Available
$\overline{\text{REQ3}}$	E	PE3 input/output	PCIC	Available
$\overline{\text{GNT1}}$	E	PE2 input/output	PCIC	Available
$\overline{\text{GNT2}}$	E	PE1 input/output	PCIC	Available
$\overline{\text{GNT3}}$	E	PE0 input/output	PCIC	Available
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/$\overline{\text{FRE}}$*	H	PH4 input/output	SCIF0/HSPI/FLCTL	
SCIF0_TXD/HSPI_TX/$\overline{\text{FWE}}$/MODE8*	H	PH3 output	SCIF0/HSPI/FLCTL/—	
SCIF0_RXD/HSPI_RX/FRB*	H	PH2 input/output	SCIF0/HSPI/FLCTL	

Pin Name	Port	GPIO	Multiplexed Module	GPIO Interrupt
SCIF0_CTS/INTD/FCLE*	H	PH1 input/output	SCIF0/PCIC/FLCTL	Available
SCIF0_RTS/HSPI_CS/FSE*	H	PH0 input/output	SCIF0/HSPI/FLCTL	Available
SIOF_TXD/HAC_SDOUT/SSI_SDATA*	J	PJ5 input/output	SIOF/HAC/SSI	
SIOF_RXD/HAC_SDIN/SSI_SCK*	J	PJ4 input/output	SIOF/HAC/SSI	
SIOF_SYNC/HAC_SYNC/SSI_WS*	J	PJ3 input/output	SIOF/HAC/SSI	
SIOF_MCLK/HAC_RES*	J	PJ2 input/output	SIOF/HAC	
SIOF_SCK/HAC_BITCLK/SSI_CLK*	J	PJ1 input/output	SIOF/HAC/SSI	
TCLK/IOIS16*	J	PJ0 input/output	TMU/LBSC	Available
DREQ0	K	PK7 input/output	DMAC	
DREQ1	K	PK6 input/output	DMAC	
DREQ2/INTB/AUDATA0*	K	PK5 input/output	DMAC/LBSC/H-UDI	Available
DREQ3/INTC/AUDATA1*	K	PK4 input/output	DMAC/LBSC/H-UDI	Available
DACK2/MRESETOUT/AUDATA2*	K	PK3 input/output	DMAC/LBSC/H-UDI	
DACK3/IRQOUT/AUDATA3*	K	PK2 input/output	DMAC/LBSC/H-UDI	
DRAK2/CE2A*	K	PK1 output	DMAC/LBSC/H-UDI	
DRAK3/CE2B/AUDSYNC*	K	PK0 output	DMAC/LBSC/H-UDI	
DACK0/MODE0	L	PL3 output	DMAC/—	
DACK1/MODE1	L	PL2 output	DMAC/—	
DRAK0/MODE2	L	PL1 output	DMAC/—	
DRAK1/MODE7	L	PL0 output	DMAC/—	
BREQ	M	PM1 input/output	LBSC	
BACK	M	PM0 input/output	LBSC	
IRQ/IRL4/FD4/MODE3*	—	—	INTC/FLCTL/—	
IRQ/IRL5/FD5/MODE4*	—	—	INTC/FLCTL/—	
IRQ/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/FCE*	—	—	H-UDI/FLCTL	

Pin Name	Port	GPIO	Multiplexed Module	GPIO Interrupt
STATUS0/CMT_CTR0*	—	—	Power Down Modes/CMT	
STATUS1/CMT_CTR1*	—	—	Power Down Modes/ CMT	

Note: * A module that uses this pin is selected by OMSELR.

28.2 Register Descriptions

Table 28.2 shows the GPIO register configuration. Table 28.3 shows the register states in each processing mode.

Table 28.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*	Sync Clock
Port A control register	PACR	R/W	H'FFEA 0000	H'1FEA 0000	16	Pck
Port B control register	PBCR	R/W	H'FFEA 0002	H'1FEA 0002	16	Pck
Port C control register	PCCR	R/W	H'FFEA 0004	H'1FEA 0004	16	Pck
Port D control register	PDCR	R/W	H'FFEA 0006	H'1FEA 0006	16	Pck
Port E control register	PECR	R/W	H'FFEA 0008	H'1FEA 0008	16	Pck
Port F control register	PFCR	R/W	H'FFEA 000A	H'1FEA 000A	16	Pck
Port G control register	PGCR	R/W	H'FFEA 000C	H'1FEA 000C	16	Pck
Port H control register	PHCR	R/W	H'FFEA 000E	H'1FEA 000E	16	Pck
Port J control register	PJCR	R/W	H'FFEA 0010	H'1FEA 0010	16	Pck
Port K control register	PKCR	R/W	H'FFEA 0012	H'1FEA 0012	16	Pck
Port L control register	PLCR	R/W	H'FFEA 0014	H'1FEA 0014	16	Pck
Port M control register	PMCR	R/W	H'FFEA 0016	H'1FEA 0016	16	Pck
Port A data register	PADR	R/W	H'FFEA 0020	H'1FEA 0020	8	Pck
Port B data register	PBDR	R/W	H'FFEA 0022	H'1FEA 0022	8	Pck
Port C data register	PCDR	R/W	H'FFEA 0024	H'1FEA 0024	8	Pck
Port D data register	PDDR	R/W	H'FFEA 0026	H'1FEA 0026	8	Pck
Port E data register	PEDR	R/W	H'FFEA 0028	H'1FEA 0028	8	Pck
Port F data register	PFDR	R/W	H'FFEA 002A	H'1FEA 002A	8	Pck
Port G data register	PGDR	R/W	H'FFEA 002C	H'1FEA 002C	8	Pck
Port H data register	PHDR	R/W	H'FFEA 002E	H'1FEA 002E	8	Pck
Port J data register	PJDR	R/W	H'FFEA 0030	H'1FEA 0030	8	Pck
Port K data register	PKDR	R/W	H'FFEA 0032	H'1FEA 0032	8	Pck
Port L data register	PLDR	R/W	H'FFEA 0034	H'1FEA 0034	8	Pck
Port M data register	PMDR	R/W	H'FFEA 0036	H'1FEA 0036	8	Pck
Port E pull-up control register	PEPUPR	R/W	H'FFEA 0048	H'1FEA 0048	8	Pck
Port H pull-up control register	PHPUPR	R/W	H'FFEA 004E	H'1FEA 004E	8	Pck

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*	Sync Clock
Port J pull-up control register	PJPUPR	R/W	H'FFEA 0050	H'1FEA 0050	8	Pck
Port K pull-up control register	PKPUPR	R/W	H'FFEA 0052	H'1FEA 0052	8	Pck
Port M pull-up control register	PMPUPR	R/W	H'FFEA 0056	H'1FEA 0056	8	Pck
Input pin pull-up control register 1	PPUPR1	R/W	H'FFEA 0060	H'1FEA 0060	16	Pck
Input pin pull-up control register 2	PPUPR2	R/W	H'FFEA 0062	H'1FEA 0062	16	Pck
On-chip module select register	OMSELR	R/W	H'FFEA 0080	H'1FEA 0080	16	Pck

Note: * There are 8-bit and 16-bit registers and access registers in designate size.

Table 28.3 Register States of GPIO 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
Port A control register	PACR	H'0000	Retained	Retained
Port B control register	PBCR	H'0000	Retained	Retained
Port C control register	PCCR	H'0000	Retained	Retained
Port D control register	PDCR	H'0000	Retained	Retained
Port E control register	PECR	H'3000	Retained	Retained
Port F control register	PFCR	H'0000	Retained	Retained
Port G control register	PGCR	H'0000	Retained	Retained
Port H control register	PHCR	H'FFFF	Retained	Retained
Port J control register	PJCR	H'FFFF	Retained	Retained
Port K control register	PKCR	H'FFFF	Retained	Retained
Port L control register	PLCR	H'FFFF	Retained	Retained
Port M control register	PMCR	H'FFFF	Retained	Retained
Port A data register	PADR	H'00	Retained	Retained
Port B data register	PBDR	H'00	Retained	Retained
Port C data register	PCDR	H'00	Retained	Retained
Port D data register	PDDR	H'00	Retained	Retained
Port E data register	PEDR	H'x0	Retained	Retained
Port F data register	PFDR	H'00	Retained	Retained
Port G data register	PGDR	H'00	Retained	Retained
Port H data register	PHDR	H'xx	Retained	Retained
Port J data register	PJDR	H'xx	Retained	Retained
Port K data register	PKDR	H'xx	Retained	Retained
Port L data register	PLDR	H'00	Retained	Retained
Port M data register	PMDR	H'0x	Retained	Retained
Port E pull-up control register	PEPUPR	H'FF	Retained	Retained
Port H pull-up control register	PHPUPR	H'FF	Retained	Retained
Port J pull-up control register	PJPUPR	H'FF	Retained	Retained
Port K pull-up control register	PKPUPR	H'FF	Retained	Retained
Port M pull-up control register	PMPUPR	H'FF	Retained	Retained
Input pin pull-up control register 1	PPUPR1	H'FFFF	Retained	Retained
Input pin pull-up control register 2	PPUPR2	H'FFFF	Retained	Retained
On-chip module select register	OMSELR	H'0000	Retained	Retained

28.2.1 Port A Control Register (PACR)

PACR 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	1	0
	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	PA4 MD1	PA4 MD0	PA3 MD1	PA3 MD0	PA2 MD1	PA2 MD0	PA1 MD1	PA1 MD0	PA0 MD1	PA0 MD0
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/W

Bit	Bit Name	Initial value	R/W	Description
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

Bit	Bit Name	Initial value	R/W	Description
5	PA2MD1	0	R/W	PA2 Mode
4	PA2MD0	0	R/W	00: PCIC module 01: Port output 10: Port input (pull-up MOS: Off) 11: Setting prohibited
3	PA1MD1	0	R/W	PA1 Mode
2	PA1MD0	0	R/W	00: PCIC module 01: Port output 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	1	0
	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	PB0 MD1	PB0 MD0
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/W

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

Bit	Bit Name	Initial value	R/W	Description
13	PB6MD1	0	R/W	PB6 Mode
12	PB6MD0	0	R/W	00: PCIC module 01: Port output 10: Port input (pull-up MOS: Off) 11: Setting prohibited
11	PB5MD1	0	R/W	PB5 Mode
10	PB5MD0	0	R/W	00: PCIC module 01: Port output 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

28.2.3 Port C Control Register (PCCR)

PCCR 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	1	0
	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	PC5 MD1	PC5 MD0	PC4 MD1	PC4 MD0	PC3 MD1	PC3 MD0	PC2 MD1	PC2 MD0	PC1 MD1	PC1 MD0	PC0 MD1	PC0 MD0
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/W

Bit	Bit Name	Initial value	R/W	Description
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

Bit	Bit Name	Initial value	R/W	Description
5	PC2MD1	0	R/W	PC2 Mode
4	PC2MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
3	PC1MD1	0	R/W	PC1 Mode
2	PC1MD0	0	R/W	00: PCIC module
				01: Port output
				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	1	0
	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	PD0 MD1	PD0 MD0
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/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

Bit	Bit Name	Initial value	R/W	Description
13	PD6MD1	0	R/W	PD6 Mode
12	PD6MD0	0	R/W	00: PCIC module 01: Port output 10: Port input (pull-up MOS: Off) 11: Setting prohibited
11	PD5MD1	0	R/W	PD5 Mode
10	PD5MD0	0	R/W	00: PCIC module 01: Port output 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

28.2.5 Port E Control Register (PECR)

PECR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PE6 MD1	PE6 MD0	PE5 MD1	PE5 MD0	PE4 MD1	PE4 MD0	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1	PE0 MD0
Initial value:	0	0	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15	—	All 0	R/W	Reserved
14				These bits are always read as 0, and the write value 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)

Bit	Bit Name	Initial value	R/W	Description
5	PE2MD1	0	R/W	PE2 Mode
4	PE2MD0	0	R/W	00: PCIC module
				01: Port output
				10: Setting prohibited
				11: Port input (pull-up MOS: On)
3	PE1MD1	0	R/W	PE1 Mode
2	PE1MD0	0	R/W	00: PCIC module
				01: Port output
				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 register.

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 MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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	PF0 MD1	PF0 MD0
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/W

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)

Bit	Bit Name	Initial value	R/W	Description
7	PF6MD1	0	R/W	PF6 Mode
6	PF6MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	PF5MD1	0	R/W	PF5 Mode
6	PF5MD0	0	R/W	00: LBSC module
				01: Port output
				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)

28.2.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7 MD1	PG7 MD0	PG6 MD1	PG6 MD0	PG5 MD1	PG5 MD0	PG4 MD1	PG4 MD0	PG3 MD1	PG3 MD0	PG2 MD1	PG2 MD0	PG1 MD1	PG1 MD0	PG0 MD1	PG0 MD0
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/W

Bit	Bit Name	Initial 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)

Bit	Bit Name	Initial value	R/W	Description
7	PG3MD1	0	R/W	PG3 Mode
6	PG3MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
5	PG2MD1	0	R/W	PG2 Mode
4	PG2MD0	0	R/W	00: LBSC module
				01: Port output
				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)

28.2.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7 MD1	PH7 MD0	—	PH6 MD0	PH5 MD1	PH5 MD0	PH4 MD1	PH4 MD0	—	PH3 MD0	PH2 MD1	PH2 MD0	PH1 MD1	PH1 MD0	PH0 MD1	PH0 MD0
Initial value:	1	1	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/W

Bit	Bit Name	Initial 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 value should 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 value should always be 1.

Bit	Bit Name	Initial value	R/W	Description
6	PH3MD	1	R/W	PH3 Mode 0: SCIF0/HSPI/FLCTL module* 1: Port output
5	PH2MD1	1	R/W	PH2 Mode
4	PH2MD0	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)
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 register.

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 MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PJ5 MD1	PJ5 MD0	PJ4 MD1	PJ4 MD0	PJ3 MD1	PJ3 MD0	PJ2 MD1	PJ2 MD0	PJ1 MD1	PJ1 MD0	PJ0 MD1	PJ0 MD0
Initial value:	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15 to 12	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.
11	PJ5MD1	1	R/W	PJ5 Mode
10	PJ5MD0	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)
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 register.

28.2.10 Port K Control Register (PKCR)

PKCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PK7 MD1	PK7 MD0	PK6 MD1	PK6 MD0	PK5 MD1	PK5 MD0	PK4 MD1	PK4 MD0	PK3 MD1	PK3 MD0	PK2 MD1	PK2 MD0	—	PK1 MD0	—	PK0 MD0
Initial value:	1	1	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/W

Bit	Bit Name	Initial 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)

Bit	Bit Name	Initial value	R/W	Description
7	PK3MD1	1	R/W	PK3 Mode
6	PK3MD0	1	R/W	00: DMAC/RESET/H-UDI module 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PK2MD1	1	R/W	PK2 Mode
4	PK2MD0	1	R/W	00: DMAC/INTC/H-UDI module 01: Port output 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 value should 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 value should 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.

28.2.11 Port L Control Register (PLCR)

PLCR 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	1	0
	—	—	—	—	—	—	—	—	—	PL3 MD0	—	PL2 MD0	—	PL1 MD0	—	PL0 MD0
Initial value:	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15 to 7	—	All 1	R/W	Reserved These bits are always read as 1, and the write value 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 value should 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 value should 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 value should always be 1.
0	PL0MD	1	R/W	PL0 Mode 0: DMAC module 1: Port output

28.2.12 Port M Control Register (PMCR)

PMCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PL1 MD1	PL1 MD0	PL0 MD1	PL0 MD0
Initial value:	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15 to 4	—	All 1	R/W	Reserved These bits are always read as 1, and the write value 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)

28.2.13 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores port A data.

Bit:	7	6	5	4	3	2	1	0
	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
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	PA7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

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 used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

28.2.15 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores port C data.

Bit:	7	6	5	4	3	2	1	0
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
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	PC7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the 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 used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the 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	

28.2.17 Port E Data Register (PEDR)

PEDR is an 8-bit readable/writable register that stores port E data.

Bit:	7	6	5	4	3	2	1	0
	—	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
Initial value:	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	—	0	R/W	Reserved This bit is always read as 0, and the write value should always be 0.
6	PE6DT	Pin input	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
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	

28.2.18 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores port F data.

Bit:	7	6	5	4	3	2	1	0
	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
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	PF7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the 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 used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the 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	

28.2.20 Port H Data Register (PHDR)

PHDR is an 8-bit readable/writable register that stores port H data.

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
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	PH7DT	Pin input	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out. However, Bit 6 and 3 are exclusively used as output ports.
6	PH6DT	0	R/W	
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 value should always be 0.
5	PJ5DT	Pin input	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
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	

28.2.22 Port K Data Register (PKDR)

PKDR is an 8-bit readable/writable register that stores port K data.

Bit:	7	6	5	4	3	2	1	0
	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT
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	PK7DT	Pin input	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the corresponding value of this register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out. However, Bit 0 is exclusively used as an 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 value should always be 0.
3	PL3DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the corresponding value of this register will be read out.
2	PL2DT	0	R/W	
1	PL1DT	0	R/W	
0	PL0DT	0	R/W	

28.2.24 Port M Data Register (PMDR)

PMDR is an 8-bit readable/writable register that stores port M data.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PM1DT	PM0DT
Initial value:	0	0	0	0	0	0	x	x
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 2	—	All 0	R/W	Reserved These bits are always read as 0, and the write value should always be 0.
1	PM1DT	Pin input	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the corresponding value of this register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be 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 this port. Bit 6 of this register corresponds to port E6 (PE6), and when the pin is set to the on-chip module, the pull-up control is performed. However, if the pin is set to the GPIO in the PECR, the setting 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

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R/W	Reserved This bit is always read as 1, and the write value should always be 1.
6	PE6PUPR	1	R/W	Pull-up control of the pin of Port E can be set. 0: PE6 pull-up off 1: PE6 pull-up on
5 to 0	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.

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 this port. Each bit of this register corresponds to port H (PH7 to PH0), and when these pins are set to the on-chip modules, the pull-up control is performed individually. However, if these pins are set to the 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

28.2.27 Port J Pull-Up Control Register (PJPUPR)

PJPUPR is an 8-bit readable/writable register that individually controls the pull-up for this port. Each bit of this register corresponds to port J (PJ5 to PJ0), and when these pins are set to the on-chip modules, the pull-up control is performed individually. However, if these pins are set to the GPIO in the PJCR, the setting in this register is invalid.

Bit:	7	6	5	4	3	2	1	0
	—	—	PJ5 PUPR	PJ4 PUPR	PJ3 PUPR	PJ2 PUPR	PJ1 PUPR	PJ0 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, 6	—	All 1	R/W	Reserved These bits are always read as 1, and the write value 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

28.2.28 Port K Pull-Up Control Register (PKPUPR)

PKPUPR is an 8-bit readable/writable register that individually controls the pull-up for this port. Each bit of this register corresponds to port K (PK7 to PK0), and when these pins are set to the on-chip modules, the pull-up control is performed individually. However, if these pins are set to the GPIO in the PKCR, the setting in this register is invalid.

Bit:	7	6	5	4	3	2	1	0
	PK7 PUPR	PK6 PUPR	PK5 PUPR	PK4 PUPR	PK3 PUPR	PK2 PUPR	PK1 PUPR	PK0 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	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

28.2.29 Port M Pull-Up Control Register (PMPUPR)

PMPUPR is an 8-bit readable/writable register that individually controls the pull-up for this port. Each bit of this register corresponds to port M (PM7 to PM0), and when these pins are set to the on-chip modules, the pull-up control is performed individually. However, if these pins are set to the GPIO in the PMCR, the setting in this register is invalid.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PM1 PUPR	PM0 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 to 2	—	All 1	R/W	Reserved These bits are always read as 1, and the write value should always be 1.
1	PM1PUPR	1	R/W	Pull-up control of the pins of Port M can be set individually. 0: PMn pull-up off 1: PMn pull-up on
0	PM0PUPR	1	R/W	

Note: n = 1 to 0

28.2.30 Input-Pin Pull-Up Control Register 1 (PPUPR1)

PPUPR1 is a 16-bit readable/writable register that individually controls the pull-up for the pin corresponding to each bit of the register field.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RDY PUP	CTR1 PUP	CTR0 PUP
Initial value:	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15 to 3	—	All 1	R/W	Reserved These bits are always read as 1, and the write value 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

28.2.31 Input-Pin Pull-Up Control Register 2 (PPUPR2)

PPUPR2 is a 16-bit readable/writable register that individually controls the pull-up for the pin corresponding to each bit of the register field.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FD3 PUP	FD2 PUP	FD1 PUP	FD0 PUP	NMI PUP	—	—	—	IRL3 PUP	IRL2 PUP	IRL1 PUP	IRL0 PUP
Initial value:	1	1	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/W

Bit	Bit Name	Initial value	R/W	Description
15 to 12	—	All 1	R/W	Reserved These bits are always read as 1, and the write value 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 value 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

Bit	Bit Name	Initial value	R/W	Description
2	IRL2PUP	1	R/W	Controls pull-up of IRQ/$\overline{\text{IRL2}}$ 0: IRQ/$\overline{\text{IRL2}}$ pull-up off 1: IRQ/$\overline{\text{IRL2}}$ pull-up on
1	IRL1PUP	1	R/W	Controls pull-up of IRQ/$\overline{\text{IRL1}}$ 0: IRQ/$\overline{\text{IRL1}}$ pull-up off 1: $\overline{\text{IRL1}}$ pull-up on
0	IRL0PUP	1	R/W	Controls pull-up of IRQ/$\overline{\text{IRL0}}$ 0: IRQ/$\overline{\text{IRL0}}$ pull-up off 1: IRQ/$\overline{\text{IRL0}}$ 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 specified by this register. For details of pin multiplexing, see table 28.1, Multiplexed Pins Controlled by Port Control Registers.

This register is valid only when on-chip modules are selected by PECR (PE6), PHCR, PJCR, or PKCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	OMSEL14	OMSEL13	OMSEL12	OMSEL11	OMSEL10	OMSEL9	OMSEL8	—	OMSEL6	OMSEL5	OMSEL4	OMSEL3	OMSEL2	OMSEL1	OMSEL0
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/W

Bit	Bit Name	Initial value	R/W	Description
15	—	0	R/W	Reserved This bit is always read as 0, and the write value should always be 0.
14	OMSEL14	0	R/W	Out of the modules RESET (STATUS) and CMT, select the one using the pins STATUS0/CMT_CTR0 and STATUS1/CMT_CTR1. 0: Selects RESET (STATUS) 1: Selects CMT

Bit	Bit Name	Initial value	R/W	Description
13	OMSEL13	0	R/W	Out of the modules SCIF1 and MMCIF, select the one using the pins SCIF1_SCK/MCCMD, SCIF1_TXD/MCCLK, and SCIF1_RXD/MCDAT. 0: Selects SCIF1 1: Selects MMCIF
12	OMSEL12	0	R/W	Out of the modules H-UDI, INTC, and FLCTL, select the one using the pins AUDATA[3:0]/FD[3:0], AUDCK/FALE, AUDSYNC/$\overline{\text{FCE}}$, IRQ/$\overline{\text{IRL4}}$/FD4, IRQ/$\overline{\text{IRL5}}$/FD5, IRQ/$\overline{\text{IRL6}}$/FD6, and IRQ/$\overline{\text{IRL7}}$/FD7. 0: H-UDI, INTC 1: FLCTL
11 10	OMSEL11 OMSEL10	0 0	R/W R/W	Out of the modules SCIF0, HSPI, PCIC, and FLCTL, 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, $\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE, and $\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CE}}$/$\overline{\text{FSE}}$ 00: SCIF0 01: HSPI, PCIC 10: FLCTL 11: SCIF0*, PCIC Note: * Cannot use modem control pin $\overline{\text{SCIF0\_CTS}}$ (this pin is used by PCIC), and $\overline{\text{SCIF0\_RTS}}$ pin should be pulled-up by PHPUPR PH0 bit.
9 8	OMSEL9 OMSEL8	0 0	R/W R/W	Out of the modules SIOF, HAC, and SSI, select the one using the pins SIOF_TXD/HAC_SDOUT/SSI_SDATA, 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

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0, and the write value should always be 0.
6	OMSEL6	0	R/W	Out of the modules TMU and LBSC, select the one using the pin TCLK/$\overline{\text{IOIS16}}$. 0: TMU 1: LBSC
5	OMSEL5	0	R/W	Out of the modules DMAC, PCIC, and H-UDI, select the one using the pins $\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1, and $\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0 00: DMAC 01: PCIC 10: H-UDI 11: Setting prohibited
4	OMSEL4	0	R/W	
3	OMSEL3	0	R/W	Out of the modules DMAC, INTC, RESET, and H-UDI, select the one using the pins $\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3 and $\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2 00: DMAC 01: INTC, RESET 10: H-UDI 11: Setting prohibited
2	OMSEL2	0	R/W	
1	OMSEL1	1	R/W	Out of the modules DMAC, LBSC, and H-UDI, select the one using the pins $\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC and $\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK 00: DMAC 01: LBSC 10: H-UDI 11: Setting prohibited
0	OMSEL0	1	R/W	

28.3 Usage Example

28.3.1 Port Output Function

To use the GPIO as an output port, set the corresponding port control register.

To output the data of port data register (PADR to PMDR) from the GPIO output port, write B'01 to the corresponding two bits in port control register (PACR to PMCR).

Then for each output port, the settings of port pull-up control register (PEPUPR, PHPUPR, PJPUPR, PKPUPR and PMPUPR) and on-chip module select register (OMSELR) are invalid.

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 peripheral clock (Pck).

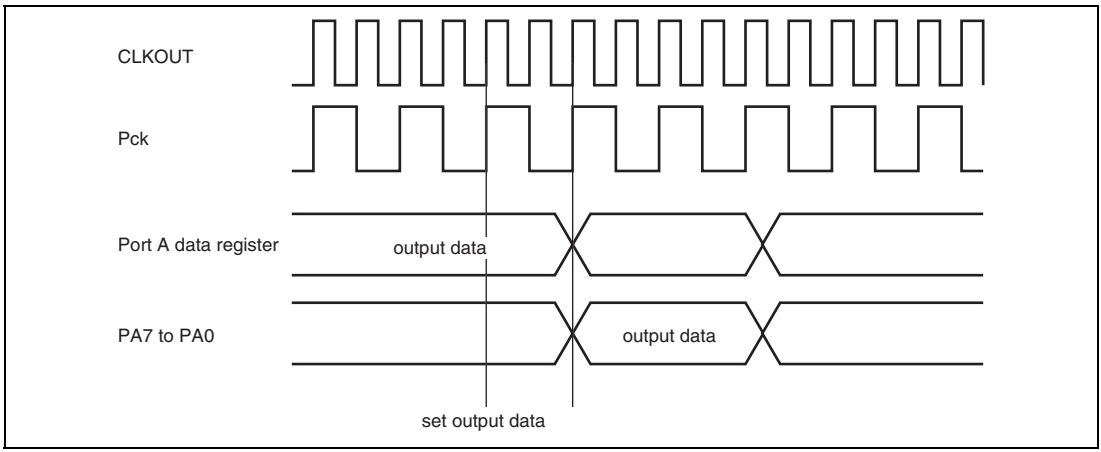


Figure 28.1 Port Data Output Timing (Example of Port A)

28.3.2 Port Input function

To use the GPIO as an input port, set the corresponding port control register.

To input the data from the GPIO port, write B'10 or B'11 to the corresponding two bits in port control register (PACR to PKCR and PMCR). Then the pull-up MOS is off by B'10 and on by B'11. The input data to each port can be read out from the corresponding bit in port data register.

Then for each input port, the settings of port pull-up control register (PEPUPR, PHPUPR, PJPUPR, PKPUPR and PMPUPR) and on-chip module select register (OMSELR) are invalid.

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 the 2nd rising edge of the peripheral clock (Pck).

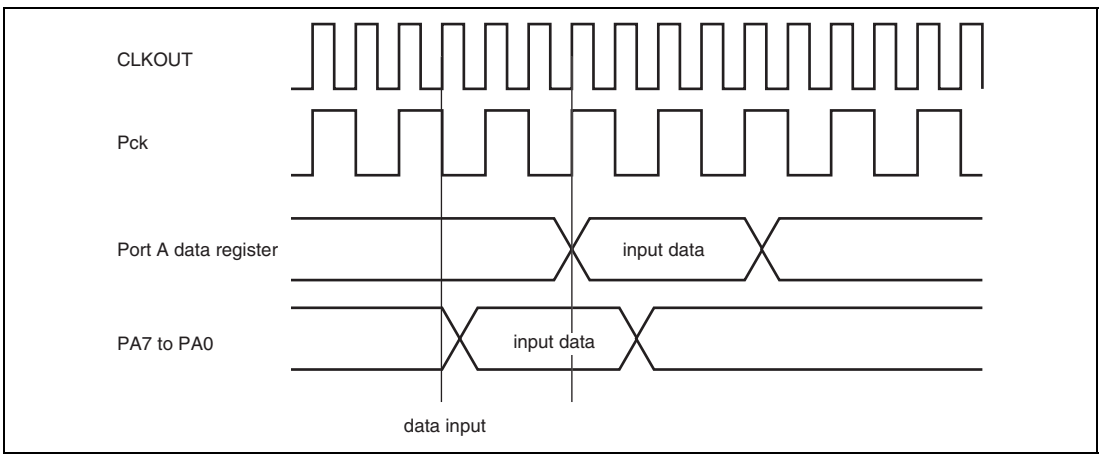


Figure 28.2 Port Data input Timing (Example of Port A)

28.3.3 On-chip Module Function

To use the peripheral modules, first select the on-chip module by setting corresponding bit in on-chip module select register (OMSELR).

When the corresponding port is input or input/output port, it is necessary for each port to set the pull-up MOS by setting port pull-up control register (PEPUPR, PHPUPR, PJPUPR, PKPUPR and PMPUPR). Write B'0 when pull-up MOS is off or B'1 when pull-up MOS is on to the corresponding bit. For an output port, the pull-up MOS is off regardless of the settings of the port pull-up control register.

After that write B'00 to the corresponding two bits in port control register (PACR to PMCR).

Section 29 User Break Controller (UBC)

The user break controller (UBC) provides versatile functions to facilitate program debugging. These functions help to ease creation of a self-monitor/debugger, which allows easy program debugging using this LSI alone, without using the in-circuit emulator. Various break conditions can be set in the UBC: instruction fetch or read/write access of an operand, operand size, data contents, address value, and program stop timing for instruction fetch.

29.1 Features

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 set as a single sequential condition for the two channels, that is, a sequential break. (Sequential break involves two cases such that the channel 0 break condition is satisfied in a certain bus cycle and then the channel 1 break condition is satisfied in a different bus cycle, and vice versa.)

- Address

  When 40 bits containing ASID and 32-bit address are compared with the specified value, all 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 desired page 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 executed.
3. Pre-instruction-execution or post-instruction-execution can be selected as the PC break timing.
4. A maximum of $2^{12} - 1$ repetition counts can be specified as the break condition (available only for channel 1).

Figure 29.1 shows the UBC block diagram.

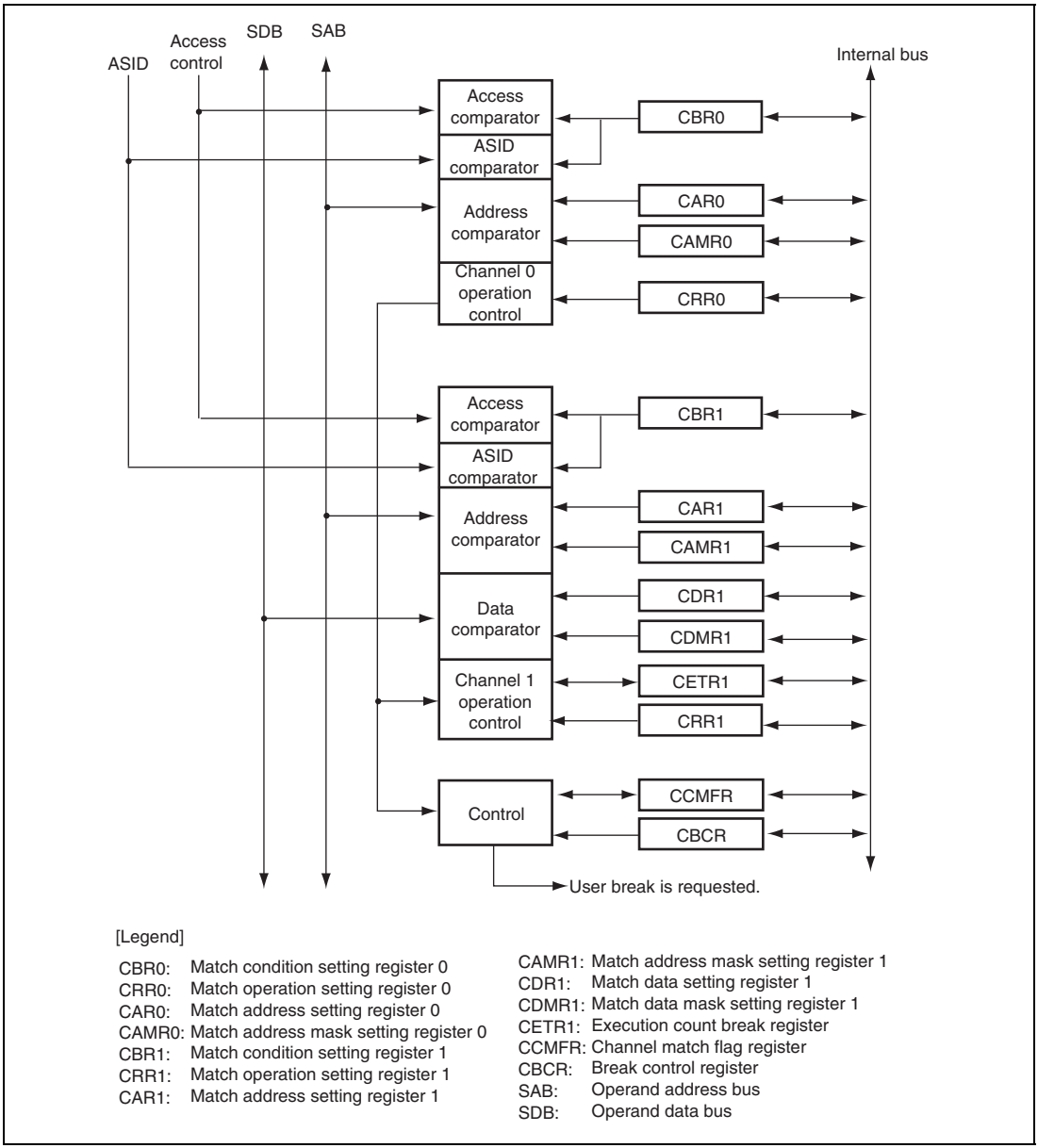


Figure 29.1 Block Diagram of UBC

29.2 Register Descriptions

The UBC has the following registers.

Table 29.1 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Match condition setting register 0	CBR0	R/W	H'FF200000	H'1F200000	32
Match operation setting register 0	CRR0	R/W	H'FF200004	H'1F200004	32
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008	32
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C	32
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020	32
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024	32
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028	32
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C	32
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030	32
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034	32
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038	32
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600	32
Break control register	CBCR	R/W	H'FF200620	H'1F200620	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 29.2 Register Status in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep
Match condition setting register 0	CBR0	H'20000000	Retained	Retained
Match operation setting register 0	CRR0	H'00002000	Retained	Retained
Match address setting register 0	CAR0	Undefined	Retained	Retained
Match address mask setting register 0	CAMR0	Undefined	Retained	Retained
Match condition setting register 1	CBR1	H'20000000	Retained	Retained
Match operation setting register 1	CRR1	H'00002000	Retained	Retained
Match address setting register 1	CAR1	Undefined	Retained	Retained
Match address mask setting register 1	CAMR1	Undefined	Retained	Retained
Match data setting register 1	CDR1	Undefined	Retained	Retained
Match data mask setting register 1	CDMR1	Undefined	Retained	Retained
Execution count break register 1	CETR1	Undefined	Retained	Retained
Channel match flag register	CCMFR	H'00000000	Retained	Retained
Break control register	CBCR	H'00000000	Retained	Retained

The access size must be the same as the control register size. If the size is different, the register is not written to if attempted, and reading the register returns the undefined value. A desired break may not occur between the time when the instruction for rewriting the control register is executed and the time when the written value is actually reflected on the register. In order to confirm the exact timing when the control register is updated, read the data which has been written most recently. The subsequent instructions are valid for the most recently written register value.

29.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)

CBR0 and CBR1 are readable/writable 32-bit registers which specify the break conditions for channels 0 and 1, respectively. The following break conditions can be set in the CBR0 and CBR1: (1) whether or not to include the match flag in the conditions, (2) whether or not to include the ASID, and the ASID value when included, (3) whether or not to include the data value, (4) operand size, (5) whether or not to include the execution count, (6) bus type, (7) instruction fetch cycle or operand access cycle, and (8) read or write access cycle.

- CBR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	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/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SZ			—	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, 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 conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; thus, not checked. 1: The ASID is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: MF0 bit of the CCMFR register 000001: MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR0[0], MFI must be set to 000000 or 000001. And note that the channel 0 is not hit when MFE bit of this register is 1 and MFI bits are 000000 in the condition of CCRMF.MF0 = 0.
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match 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 value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
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 operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is 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 0, all the 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 operand 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 the contents of both the match data setting register and the match data mask setting register.

- CBR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	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/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBE	SZ			ETBE	—	—	—	CD		ID		—	RW		CE
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	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, 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 conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; 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 match 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, but when 1 is written into CBR1[0], MFI must be set to 000000 or 000001. And note that the channel 1 is not hit when MFE bit of this register is 1 and MFI bits are 000001 in the condition of CCRMF.MF1 = 0.
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
15	DBE	0	R/W	Data Value Enable*[3] Specifies whether or not to include the data value in the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 0: The data value is not included in the match conditions; thus, not checked. 1: The data value is included in the match conditions.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match condition; 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	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execution count value in the match conditions. If this bit is 1 and the match condition satisfaction count matches the value specified by the CETR1 register, the operation specified by the CRR1 register is performed. 0: The execution count value is not included in the match conditions; thus, not checked. 1: The execution count value is included in the match conditions.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should 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 operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is 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 0, all the 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 operand 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 the contents of both the match data setting register and the match data mask setting register.

3. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

29.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)

CRR0 and CRR1 are readable/writable 32-bit registers which specify the operation to be executed when channels 0 and 1 satisfy the match condition, respectively. The following operations can be set in the CRR0 and CRR1 registers: (1) breaking at a desired timing for the instruction fetch cycle and (2) requesting a break.

- CRR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than the ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

- CRR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

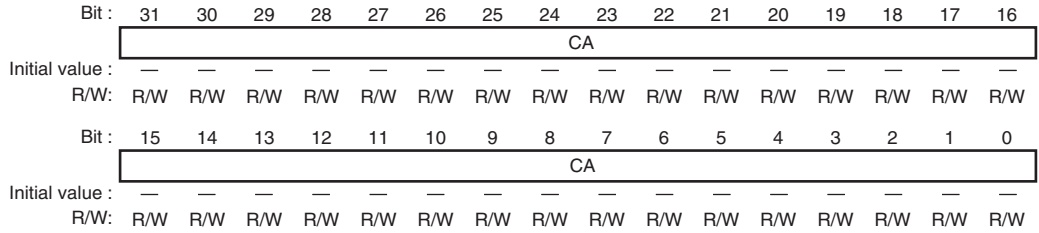
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

29.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)

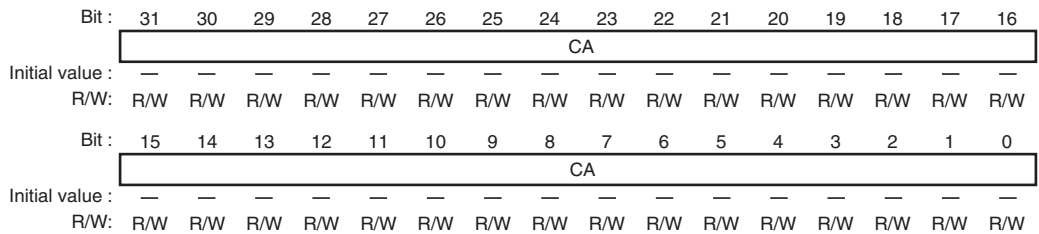
CAR0 and CAR1 are readable/writable 32-bit registers specifying the virtual address to be included in the break conditions for channels 0 and 1, respectively.

- CAR0



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 break conditions. When the operand bus has been specified using the CBR0 register, specify the SAB address in CA[31:0].

- CAR1

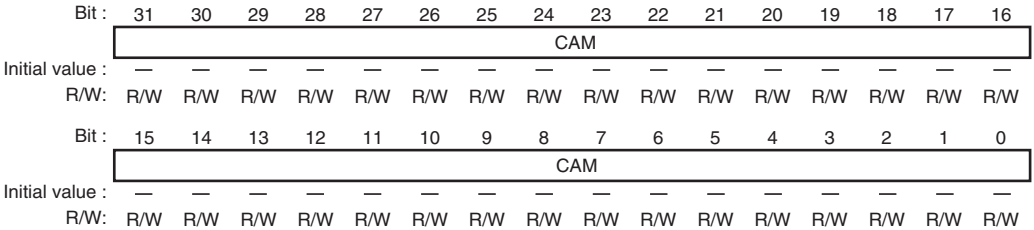


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SAB address in CA[31:0].

29.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)

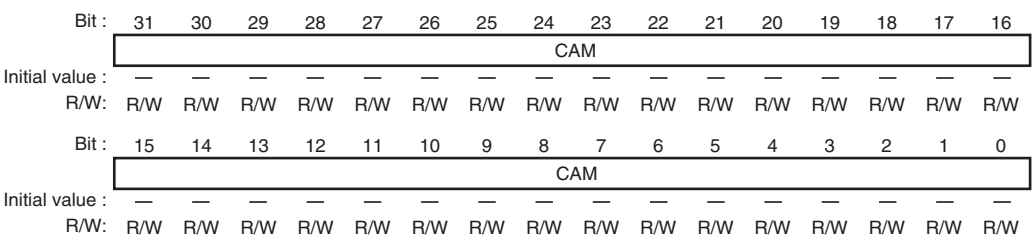
CMAR0 and CMAR1 are readable/writable 32-bit registers which specify the bits to be masked among the address bits specified by using the match address setting register of the corresponding channel. (Set the bits to be masked to 1.)

- CAMR0



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 address bits which are specified using the CAR0 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

- CAMR1



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 address bits which are specified using the CAR1 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

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 in the break conditions for channel 1.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CD															
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

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD															
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

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 break conditions. When the operand bus has been specified using the CBR1 register, specify the SDB data value in CD[31:0].

Table 29.3 Settings for Match Data Setting Register

Bus and Size Selected Using CBR1	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
Operand bus (byte)	Don't care	Don't care	Don't care	SDB7 to SDB0
Operand bus (word)	Don't care	Don't care	SDB15 to SDB8	SDB7 to SDB0
Operand bus (longword)	SDB31 to SDB24	SDB23 to SDB16	SDB15 to SDB8	SDB7 to SDB0

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.
2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions 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 the contents of both the match data setting register and match data mask setting register.

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 among the data value bits specified using the match data setting register. (Set the bits to be masked to 1.)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDM															
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

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDM															
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

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 data value bits specified using the CDR1 register. (Set the bits to be masked to 1.) 0: Data value bits CD[n] are included in the break condition. 1: Data value bits CD[n] are masked and not included in the break condition. [n] = any values from 31 to 0

29.2.7 Execution Count Break Register 1 (CETR1)

CETR1 is a readable/writable 32-bit register which specifies the number of the channel hits before a break occurs. A maximum value of $2^{12}-1$ can be specified. When the execution count value is included in the match conditions by using the match condition setting register, the value of this register is decremented by one every time the channel is hit. When the channel is hit after the register value reaches H'001, a break occurs.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CET											
Initial value :	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	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 value should always be 0.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in the break conditions.

29.2.8 Channel Match Flag Register (CCMFR)

CCMFR is a readable/writable 32-bit register which indicates whether or not the match conditions have been satisfied for each channel. When a channel match condition has been satisfied, the corresponding flag bit is set to 1. To clear the flags, write the data containing value 0 for the bits to be cleared and value 1 for the other bits to this register. (The logical AND between the value which has been written and the current register value is actually written to the register.) Sequential operation using multiple channels is available by using these match flags.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
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	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

29.2.9 Break Control Register (CBCR)

CBCR is a readable/writable 32-bit register which specifies whether or not to use the user break debugging support function. For details on the user break debugging support function, refer to section 29.4, User Break Debugging Support Function.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	UBDE	0	R/W	User Break Debugging Support Function Enable Specifies whether or not to use the user break debugging support function. 0: Does not use the user break debugging support function. 1: Uses the user break debugging support function.

29.3 Operation Description

29.3.1 Definition of Words Related to Accesses

"Instruction fetch" refers to an access in which an instruction is fetched. For example, fetching the instruction located at the branch destination after executing a branch instruction is an instruction access. "Operand access" refers to any memory access accompanying execution of an instruction. For example, accessing an address (PC + disp × 2 + 4) in the instruction MOV.W@(disp,PC),Rn is an operand access. "Data" is used in contrast to "address".

All types of operand access are classified into read or write access. Special care must be taken in 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 access without the data value; therefore, do not include the data value in the match conditions for these instructions.

The operand size should be defined for all types of operand access. Available operand sizes are byte, word, longword, and quadword. For operand access accompanying the PREF, OCBP, OCBWB, MOVCA.L, and OCBI instructions, the operand size is defined as longword.

29.3.2 User Break Operation Sequence

The following describes the sequence from when the break condition is set until the user break exception handling is initiated.

1. Specify the operand size, bus, instruction fetch/operand access, and read/write as the match conditions using the match condition setting register (CBR0 or CBR1). Specify the break address using the match address setting register (CAR0 or CAR1), and specify the address mask condition using the match address mask setting register (CAMR0 or CAMR1). To include the ASID in the match conditions, set the AIE bit in the match condition setting register and specify the ASID value by the AIV bit in the same register. To include the data value in the match conditions, set the DBE bit in the match condition setting register; specify the break data using the match data setting register (CDR1); and specify the data mask condition using the match data mask setting register (CDMR1). To include the execution count in the match conditions, set the ETBE bit of the match condition setting register; and specify the execution count using the execution count break register (CETR1). To use the sequential break, set the MFE bit of the match condition setting register; and specify the 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 the break timing when the match condition is satisfied as a result of fetching the instruction using the match operation setting register (CRR0 or CRR1). After having set all the bits in the match condition setting register except the CE bit and the other necessary registers, set the CE bit and read the match condition setting register again. This ensures that the set values in the control registers are valid for the subsequent instructions immediately after reading the register. Setting the CE bit of the match condition setting register in the initial state after reset via the control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag (MF1 or MF0) in the channel match flag register (CCMFR) is set. A break is also requested to the CPU according to the set values in the match operation setting register (CRR0 or CRR1). The CPU operates differently according to the BL bit value of the SR register: when the BL bit is 0, the CPU accepts the break request and executes the specified exception handling; and when the 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 corresponding match condition has been satisfied. Although the flag is set when the condition is satisfied, it is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory store 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 one break request is sent to the CPU; however, the two condition match flags corresponding to these breaks may be set.

6. While the BL bit in the SR register is 1, no break requests are accepted. However, whether or not the condition has been satisfied is determined. When the condition is determined to be satisfied, the corresponding condition match flag is set.
7. If the sequential break conditions are set, the condition match flag is set every time the match conditions are satisfied for each channel. When the conditions have been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.

29.3.3 Instruction Fetch Cycle Break

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CBR1), the instruction fetch cycle is handled as a match condition. To request a break upon satisfying the match condition, set the BIE bit in the match operation setting register (CRR0 or CRR1) of the corresponding channel. Either before or after executing the instruction can be selected as the break timing according to the PCB bit value. If the instruction fetch cycle is specified as a match condition, be sure to clear the LSB to 0 in the match address setting register (CAR0 or CAR1); otherwise, no break occurs.
2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break is requested when the instruction is fetched and determined to be executed. Therefore, this function cannot be used for the instructions which are fetched through overrun (i.e., the instructions fetched during branching or making transition to the interrupt routine but not executed). For priorities of pre-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If pre-instruction-execution break is specified for the delayed slot of the delayed branch instruction, the break is requested before the delayed branch instruction is executed. However, do not specify pre-instruction-execution break for the delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the break is requested after the instruction which satisfied the match condition has been executed and before the next instruction is executed. Similar to pre-instruction-execution break, this function cannot be used for the instructions which are fetched through overrun. For priorities of post-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If post-instruction-execution break is specified for the delayed branch instruction 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 bit of match condition setting register CBR1 becomes invalid, the settings of match data setting register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, the data value cannot be specified for the instruction fetch cycle break.

29.3.4 Operand Access Cycle Break

1. Table 29.4 shows the relation between the operand sizes specified using the match condition setting register (CBR0 or CBR1) and the address bits to be compared for the operand access cycle break.

Table 29.4 Relation between Operand Sizes and Address Bits to be Compared

Selected Operand Size	Address Bits to be Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
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 register (CAR0 or CAR1), for example, the match condition is satisfied for the following access cycles (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, longword, word, or byte as the operand size using the operand size select bit (SZ) of the match condition setting register (CBR1), and also set the match data setting register (CDR1) and the match data mask setting register (CDMR1). With these settings, the match condition is satisfied when both of the address and data conditions are satisfied. The data value and mask control for byte access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 to 0 in the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divided into the upper and lower 32-bit data units, and each unit is independently compared with the specified condition. When either the upper or lower 32-bit data unit satisfies the match condition, the match condition for the 64-bit data is determined to be satisfied.

3. The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions are access without the data value; therefore, if the data value is included in the match conditions for these instructions, the match conditions will never be satisfied.
4. If the operand bus is selected, a break occurs after executing the instruction which has satisfied the conditions and immediately before executing the next instruction. However, if the data value is included in the match conditions, a break may occur after executing several instructions after the instruction which has satisfied the conditions; therefore, it is impossible to identify the instruction causing the break. If such a break has occurred for the delayed branch instruction or its delayed slot, the break does not occur until the first instruction at the branch destination.

   However, do not specify the operand break for the delayed slot of the RTE instruction. And if the data value is included in the match conditions, it is not allowed to set the break for the 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 match condition setting registers (CBR0 and CBR1). (Sequential break involves two cases such that channel 0 break condition is satisfied then channel 1 break condition is satisfied, and vice versa.) To use the sequential break function, clear the MFE bit of the match condition setting register and the BIE bit of the match operation setting register of the first channel in the sequence, and set the MFE bit and specify the number of the second channel in the sequence using the MFI bit in the match condition setting register of the second channel in the sequence. If the sequential break condition is set, the condition match flag is set every time the match condition is satisfied for each channel. When the condition has been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequential break conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied within a significantly short time, sequential operation may not be guaranteed in some cases, as shown below.

- When the Match Condition is Satisfied at the Instruction Fetch Cycle for Both the First and Second Channels in the Sequence:

Instruction B is 0 instruction after instruction A	Equivalent to setting the same addresses; do not use this setting.
Instruction B is 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 instruction fetch cycle for the first channel in the sequence whereas the match condition is satisfied at the operand access cycle for the second 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 channel in the sequence whereas the match condition is satisfied at the instruction fetch cycle for the second 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 and second 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.

29.3.6 Program Counter Value to be Saved

When a break has occurred, the address of the instruction to be executed when the program restarts is saved in the SPC then the exception handling state is initiated. A unique instruction causing a break can be identified unless the data value is included in the match conditions.

- When the instruction fetch cycle (before instruction execution) is specified as the match condition:

  The address of the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is not executed, but a break occurs instead. However, if the match conditions are satisfied for the delayed slot instruction, the 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 match conditions is saved in the SPC. The instruction which has satisfied the match conditions is executed, then a break occurs before the next instruction. If the match conditions are satisfied for the delayed branch instruction or its delayed slot, these instructions are executed and the 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 break conditions is saved in the SPC. The instruction which has satisfied the match conditions are executed, then a break occurs before the next instruction. However, if the conditions are 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 the match conditions is executed. A user break occurs before executing an instruction that is one through six instructions after the instruction which has satisfied the match conditions. The address of the instruction is saved in the SPC; thus, it is impossible to identify exactly where a break will occur. If the conditions are satisfied for the delayed slot instruction, the address of the branch destination is saved in the SPC. If a branch instruction follows the instruction which has satisfied the match conditions, a break may occur after the delayed instruction and delayed slot are executed. In this case, the address of the branch destination is also saved in the SPC.

29.4 User Break Debugging Support Function

By using the user break debugging support function, the branch destination address can be modified when the CPU accepts the user break request. Specifically, setting the UBDE bit of break control register CBCR to 1 allows branching to the address indicated by DBR instead of branching to the address indicated by the [VBR + offset]. Figure 29.2 shows the flowchart of the user break debugging support function.

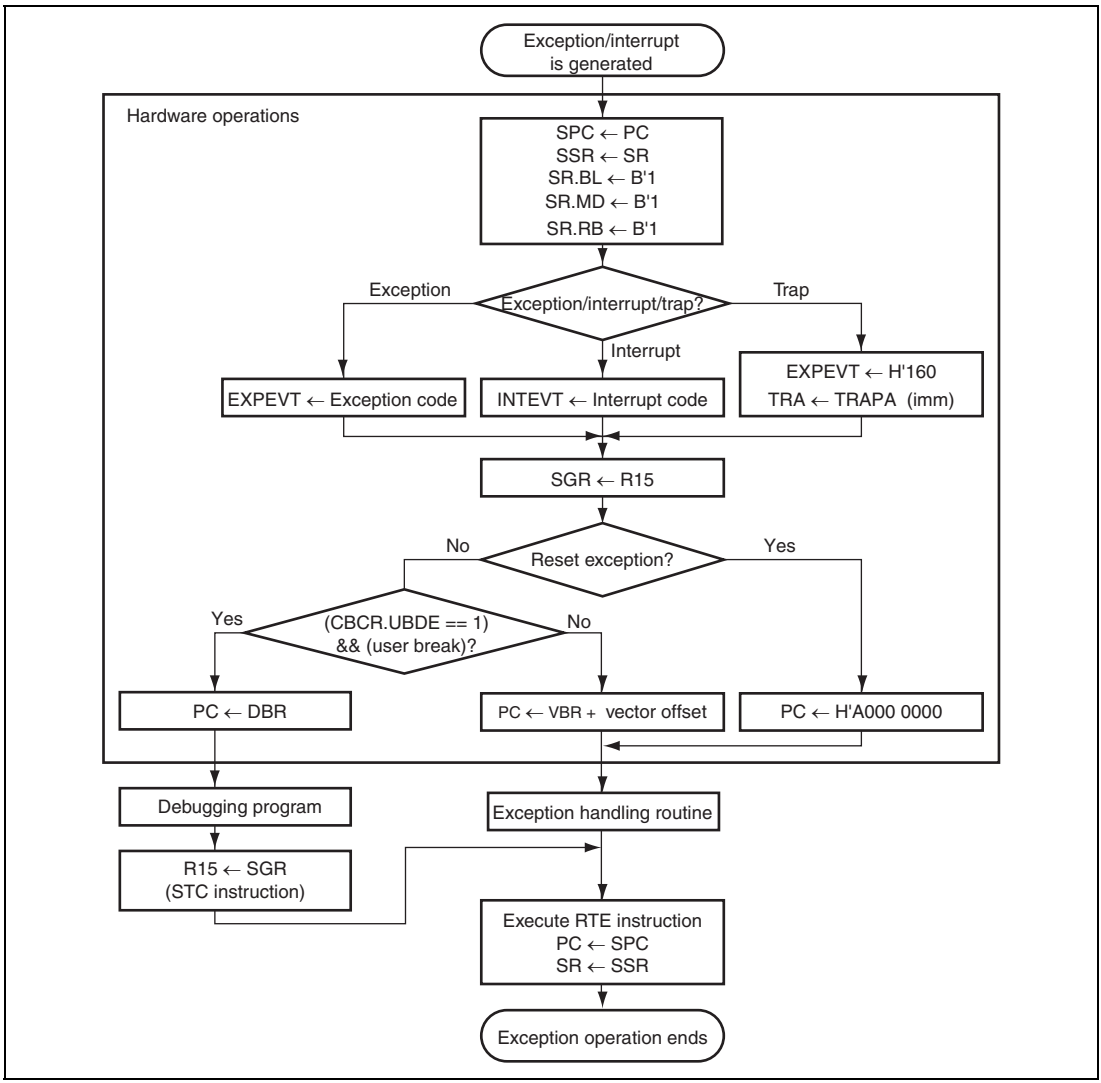


Figure 29.2 Flowchart of User Break Debugging Support Function

29.5 User Break Examples

Match Conditions are Specified for an Instruction Fetch Cycle:

- Example 1-1

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002003 / CAR0 = H'00000404 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00008010 / 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'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 address H'00000404 or before executing the instruction at address H'00008010 to H'00008016.

- 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'0003722E / 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.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-3

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00027128 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00031415 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    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 instruction at address H'00027128. No user break occurs for channel 1 since the instruction fetch is executed only at even addresses.

- Example 1-4

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / 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.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 and before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-5

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00000500 / CAMR0 = H'00000000 / CBR1 = H'00000813 / CRR1 = H'00002001 / CAR1 = H'00001000 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000005 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  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 instruction at address H'00000500. The user break occurs for channel 1 after executing the instruction at 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'00008010 / 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)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00008000 to H'00008FFE where ASID is H'80 or before executing the instruction at address H'00008010 to H'00008016 where ASID is H'70.

Match Conditions are Specified for an Operand Access Cycle:

- Example 2-1

  Register settings: CBR0 = H'40800023 / CRR0 = H'00002001 / CAR0 = H'00123456 / CAMR0 = H'00000000 / CBR1 = H'4070A025 / CRR1 = H'00002001 / CAR1 = H'000ABCDE / CAMR1 = H'000000FF / CDR1 = H'0000A512 / CDMR1 = H'00000000 / 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 in the 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: longword read access to address H'000123454, word read access to address H'000123456, byte read access to address H'000123456 where ASID is H'80. The user break occurs for channel 1 when word H'A512 is written to address H'000ABC00 to H'000ABCFE where ASID is H'70.

29.6 Usage Notes

- A desired break may not occur between the time when the instruction for rewriting the UBC register is executed and the time when the written value is actually reflected on the register.

  After the UBC register is updated, execute one of the following three methods.

  A. Read the updated UBC register, and execute a branch using the RTE instruction.
  (It is not necessary that a branch using the RTE instruction is next to a reading UBC register.)

  B. Execute the ICBI instruction for any address (including non-cacheable area).
  (It is not necessary that the ICBI instruction is next to a reading UBC register.)

  C. Set 0(initial value) to IRMCR.R1 before updating the UBC register and update with 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 updating the UBC registers is not necessary. At only last updating the UBC register, execute one of these methods.

- The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch is specified as the match condition.
- If the sequential break conditions are set, the sequential break conditions are satisfied when the 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 be satisfied simultaneously for the same bus cycle, the sequential break conditions will not be satisfied, causing no break.
- For the SLEEP instruction, do not allow the post-instruction-execution break where the instruction fetch cycle is the match condition. For the instructions preceding the SLEEP instruction by one to five instructions, do not allow the break where the operand access is the match condition.
- If the user break and other exceptions occur for the same instruction, they are determined according to the specified priority. For the priority, refer to section 5, Exception Handling. If 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.

— If the post-instruction-execution break and data access break have occurred simultaneously with the re-execution type exception (including the pre-instruction-execution break) having a higher priority, only the re-execution type exception is accepted, and no condition match flags are set. When the exception handling has finished thus clearing the exception source, and when the same instruction has been executed again, the break occurs setting the corresponding flag.

— If the post-instruction-execution break or operand access break has occurred simultaneously with the completion-type exception (TRAPA) having a higher priority, then no user break occurs; however, the condition match flag is set.

- When conditions have been satisfied simultaneously and independently for channels 0 and 1, resulting in identical SPC values for both of the breaks, the user break occurs only once. However, the condition match flags are set for both channels. For example,

  Instruction at address 110 (post-instruction-execution break for instruction fetch for channel 0) → SPC = 112, CCMFR.MF0 = 1

  Instruction at address 112 (pre-instruction-execution break for instruction fetch for channel 1) → SPC = 112, CCMFR.MF1 = 1
- It is not allowed to set the pre-instruction-execution break or the operand break in the delayed 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 instruction by one to six instructions.
- If the re-execution type exception and the post-instruction-execution break are in conflict for the instruction requiring two or more execution states, then the re-execution type exception occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 when the break conditions have been satisfied.

Section 30 User Debugging Interface (H-UDI)

The H-UDI is a serial interface which conforms to the JTAG (IEEE 1149.1: IEEE Standard Test Access Port and Boundary-Scan Architecture) standard. The H-UDI is also used for emulator connection.

30.1 Features

The H-UDI is a serial interface which conforms to the JTAG standard. The H-UDI is also used for emulator connection. When using an emulator, H-UDI functions should not be used. Refer to the appropriate emulator users manual for the method of connecting the emulator.

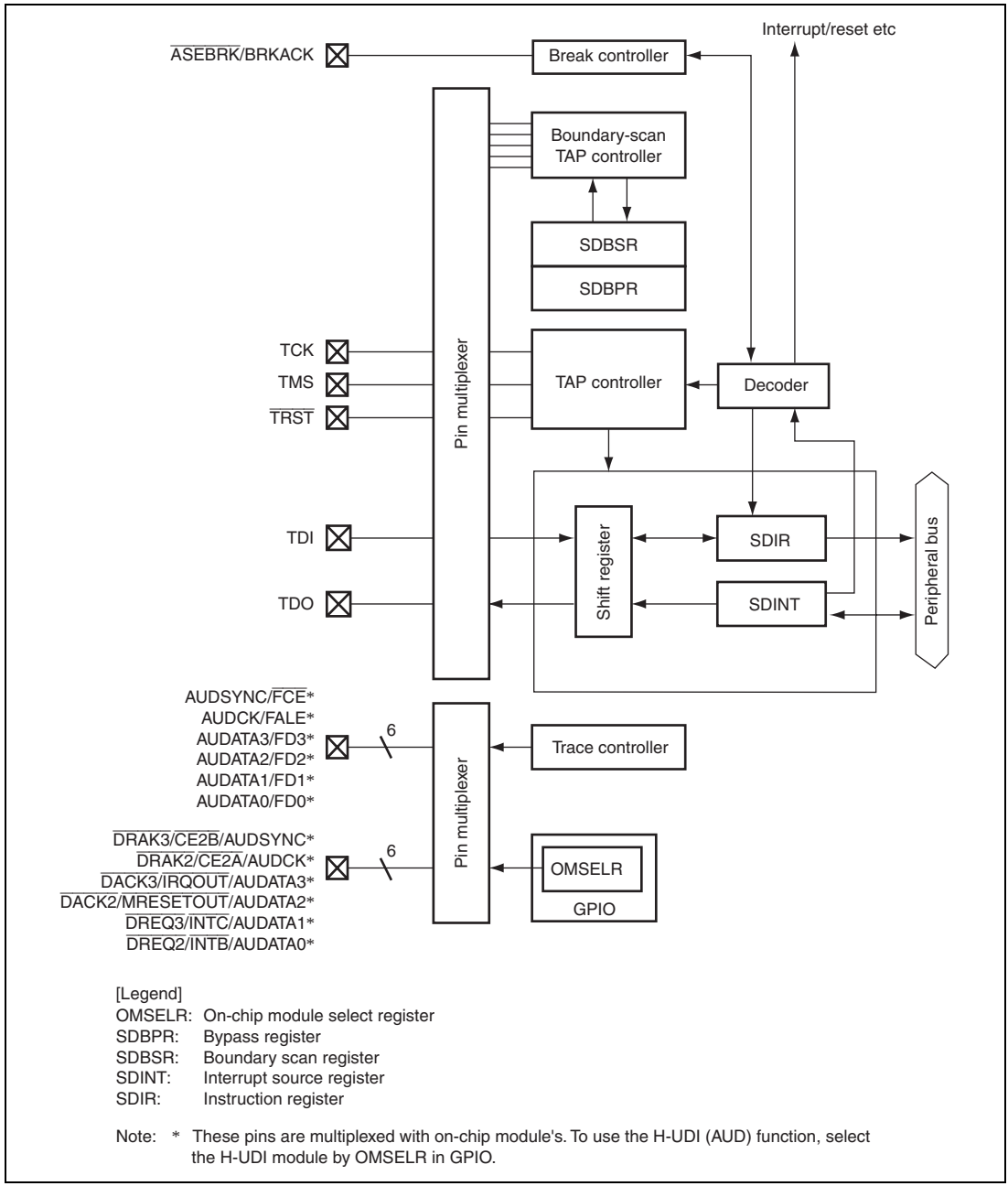
The H-UDI has six pins: TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK. The pin functions except $\overline{\text{ASEBRK}}$/BRKACK and serial communications protocol conform to the JTAG standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDCK, and AUDATA3 to AUDATA0). These six pins for emulator are multiplexed with on-chip modules. 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 another is H-UDI function except the boundary-scan test. The H-UDI initial state is for the boundary scan after power-on or $\overline{\text{TRST}}$ asserted. It is necessary to set H-UDI switchover command to use the H-UDI 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, SDBSR, SDIR, and SDINT). SDBPR supports the JTAG bypass mode, SDBSR supports the JTAG boundary scan mode, SDIR is used for commands, and SDINT is used for H-UDI interrupts. SDIR is directly accessed from the TDI and TDO pins.

The TAP controller, control registers and boundary scan TAP controller are initialized by driving the $\overline{\text{TRST}}$ pin low or by applying the TCK signal for five or more clock cycles with the TMS pin set to 1. This initialization sequence is independent of the reset pin for this LSI. Other circuits are 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

30.2 Input/Output Pins

Table 30.1 shows the pin configuration for the H-UDI.

Table 30.1 Pin Configuration

Pin Name	Function	I/O	Description	When Not in Use
TCK	Clock	Input	Functions as the serial clock input pin stipulated in the JTAG standard. Data input to the H-UDI via the TDI pin or data Output via the TDO pin is performed in synchronization with this signal.	Open*[1]
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).	Open*[1]
$\overline{\text{TRST}}$*[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.	Fixed to ground or connected to the $\overline{\text{PRESET}}$ pin*[3]
TDI	Data input	Input	Data Input Data is sent to the H-UDI by changing this signal in synchronization with the TCK signal.	Open*[1]
TDO	Data output	Output	Data Output Data is read from the H-UDI in synchronization with the TCK signal.	Open
$\overline{\text{ASEBRK}}$/ BRKACK	Emulator	I/O	Pins for an emulator	Open*[1]
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for an emulator	Open
MPMD	Chip-mode	Input	Selects the operation mode of this LSI, whether emulation support mode (Low level) or LSI operation mode (High level).	Open

Notes: 1. This pin is pulled up in this LSI. When using interrupts or resets via the H-UDI or 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 should be designed so that it can be controlled independently and can be controlled to retain low level while the $\overline{\text{PRESET}}$ pin is asserted at a power-on reset.

3. This pin should be connected to ground, the $\overline{\text{PRESET}}$, or another pin which operates in the same manner as the $\overline{\text{PRESET}}$ pin. However, when connected to a ground pin, the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground pin. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

The TCK clock or the CPG of this LSI should be set to ensure that the frequency of the TCK clock is less than the peripheral-clock frequency of this LSI.

30.3 Boundary Scan TAP Controllers (IDCODE, EXTEST, SAMPLE/PRELOAD, and BYPASS)

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan function 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 H-UDI 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 the 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 H-UDI 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-UDI.

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-UDI.

Table 30.2 Commands Supported by Boundary-Scan TAP Controller

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
0	1	0	1	0	1	0	1	IDCODE
1	1	1	1	1	1	1	1	BYPASS
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	0	0	0	1	0	0	0	H-UDI (switchover command)
Other than above								Setting prohibited

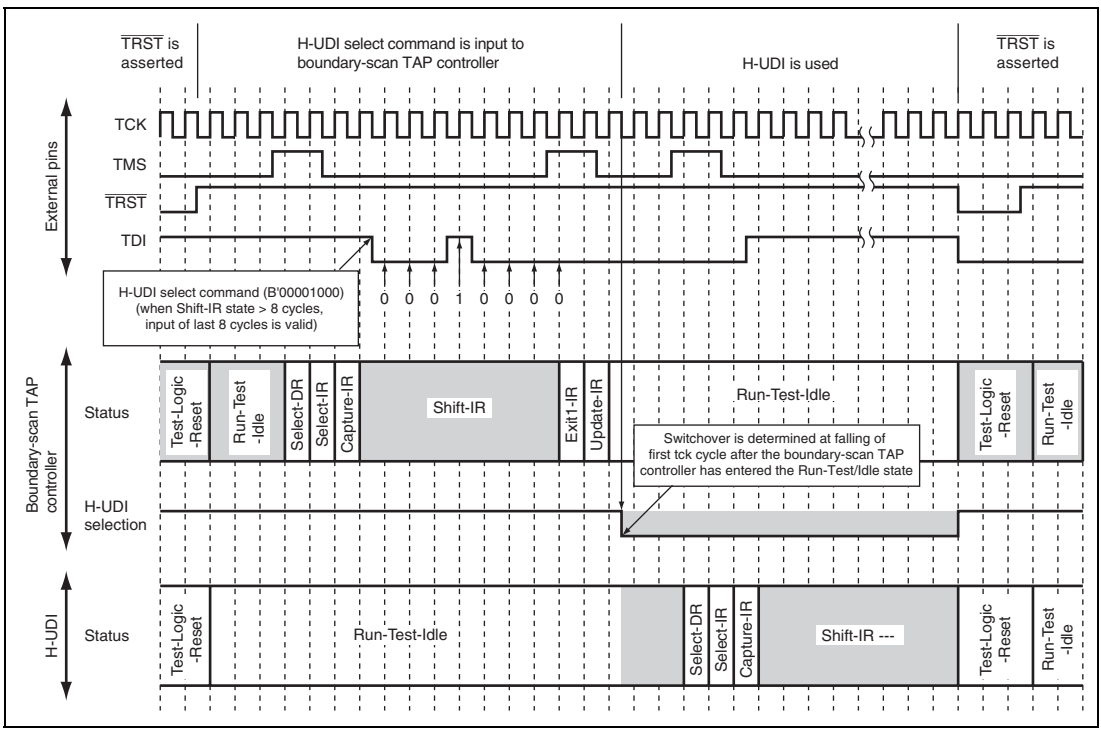


Figure 30.2 Sequence for Switching from Boundary-Scan TAP Controller to H-UDI

30.4 Register Descriptions

The H-UDI has the following registers.

Table 30.3 Register Configuration (1)

		CPU Side				
Register Name	**Abbrev.**	**R/W**	**P4 Address*[1]**	**Area 7 Address*[1]**	**Size**	**Initial Value*[2]**
Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
Boundary scan register	SDBSR	—	—	—	—	—
Bypass register	SDBPR	—	—	—	—	—

Notes: 1. The P4 address is an address when accessing through P4 area in a virtual address space. The area 7 address is an address when accessing through area 7 in a physical space using the TLB.
2. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP controller 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 controller 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	Retained
Interrupt source register	SDINT	H'0000	Retained	Retained

30.4.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register that can be read from the CPU. Commands are set via the serial input (TDI). SDIR is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

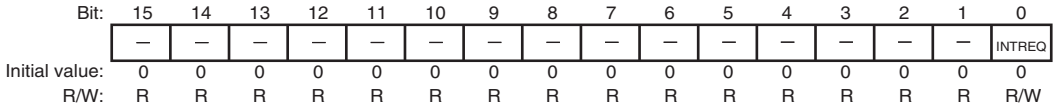
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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 registers are not initialized.
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

30.4.2 Interrupt Source Register (SDINT)

SDINT is a 16-bit register that can be read from or written to by the CPU. Specifying an H-UDI interrupt command in SDIR via H-UDI pin (Update-IR) sets the INTREQ bit to 1. While an H-UDI interrupt command is set in SDIR, SDINT which is connected between the TDI and TDO pins can be read as a 32-bit register. In this case, the upper 16 bits will be 0 and the lower 16 bits represent the SDINT value.

Only 0 can be written to the INTREQ bit by the CPU. While this bit is set to 1, an interrupt request will continue to be generated. This bit, therefore, should be cleared by the interrupt handling routine. It is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state.



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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. When writing 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 command is set to the boundary scan TAP controller, the TDI and TDO are connected by way of SDBPR. This register cannot be accessed from the CPU regardless of the LSI mode. Though this register is not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted, initialized to 0 in the Capture-DR state.

30.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/Output pins, which supports the boundary scan mode of the JTAG standard. Using the EXTEST and SAMPLE/PRELOAD commands, a boundary-scan test complying with the JTAG standards (IEEE1149.1) can be carried out. Table 30.6 shows the correspondence between pins of this LSI and the SDBSR values.

This register cannot be accessed from the CPU regardless of the LSI mode. This register is not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted.

Table 30.6 SDBSR Configuration

Number	Pin Name	I/O*	Number	Pin Name	I/O*
	From TDI		503	$\overline{\text{DREQ0}}$	Output
524	$\overline{\text{DACK3}}$/IRQOUT/AUDATA3	Output	502	$\overline{\text{DREQ0}}$	Control
523	$\overline{\text{DACK3}}$/IRQOUT/AUDATA3	Control	501	$\overline{\text{DREQ0}}$	Input
522	$\overline{\text{DACK3}}$/IRQOUT/AUDATA3	Input	500	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC	Output
521	$\overline{\text{DACK2}}$/MRESETOUT/AUDATA2	Output	499	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC	Control
520	$\overline{\text{DACK2}}$/MRESETOUT/AUDATA2	Control	498	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC	Input
519	$\overline{\text{DACK2}}$/MRESETOUT/AUDATA2	Input	497	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK	Output
518	$\overline{\text{DACK1}}$/MODE1	Output	496	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK	Control
517	$\overline{\text{DACK1}}$/MODE1	Control	495	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK	Input
516	$\overline{\text{DACK1}}$/MODE1	Input	494	$\overline{\text{DRAK1}}$/MODE7	Output
515	$\overline{\text{DACK0}}$/MODE0	Output	493	$\overline{\text{DRAK1}}$/MODE7	Control
514	$\overline{\text{DACK0}}$/MODE0	Control	492	$\overline{\text{DRAK1}}$/MODE7	Input
513	$\overline{\text{DACK0}}$/MODE0	Input	491	$\overline{\text{DRAK0}}$/MODE2	Output
512	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Output	490	$\overline{\text{DRAK0}}$/MODE2	Control
511	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Control	489	$\overline{\text{DRAK0}}$/MODE2	Input
510	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Input	488	A25	Output
509	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Output	487	A25	Control
508	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Control	486	A25	Input
507	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Input	485	STATUS0/CMT_CTR0	Output
506	$\overline{\text{DREQ1}}$	Output	484	STATUS0/CMT_CTR0	Control
505	$\overline{\text{DREQ1}}$	Control	483	STATUS0/CMT_CTR0	Input
504	$\overline{\text{DREQ1}}$	Input	482	STATUS1/CMT_CTR1	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
481	STATUS1/CMT_CTR1	Control	448	A15	Control
480	STATUS1/CMT_CTR1	Input	447	A15	Input
479	A22	Output	446	A13	Output
478	A22	Control	445	A13	Control
477	A22	Input	444	A13	Input
476	A23	Output	443	A12	Output
475	A23	Control	442	A12	Control
474	A23	Input	441	A12	Input
473	A24	Output	440	A11	Output
472	A24	Control	439	A11	Control
471	A24	Input	438	A11	Input
470	A19	Output	437	A10	Output
469	A19	Control	436	A10	Control
468	A19	Input	435	A10	Input
467	A20	Output	434	A9	Output
466	A20	Control	433	A9	Control
465	A20	Input	432	A9	Input
464	A21	Output	431	A8	Output
463	A21	Control	430	A8	Control
462	A21	Input	429	A8	Input
461	A16	Output	428	A7	Output
460	A16	Control	427	A7	Control
459	A16	Input	426	A7	Input
458	A17	Output	425	A6	Output
457	A17	Control	424	A6	Control
456	A17	Input	423	A6	Input
455	A18	Output	422	A5	Output
454	A18	Control	421	A5	Control
453	A18	Input	420	A5	Input
452	A14	Output	419	A4	Output
451	A14	Control	418	A4	Control
450	A14	Input	417	A4	Input
449	A15	Output	416	A3	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
415	A3	Control	382	D24	Control
414	A3	Input	381	D24	Input
413	A2	Output	380	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	Output
412	A2	Control	379	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	Control
411	A2	Input	378	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	Input
410	A1	Output	377	D23	Output
409	A1	Control	376	D23	Control
408	A1	Input	375	D23	Input
407	A0	Output	374	D22	Output
406	A0	Control	373	D22	Control
405	A0	Input	372	D22	Input
404	D31	Output	371	D21	Output
403	D31	Control	370	D21	Control
402	D31	Input	369	D21	Input
401	D30	Output	368	D20	Output
400	D30	Control	367	D20	Control
399	D30	Input	366	D20	Input
398	D29	Output	365	D19	Output
397	D29	Control	364	D19	Control
396	D29	Input	363	D19	Input
395	D28	Output	362	D18	Output
394	D28	Control	361	D18	Control
393	D28	Input	360	D18	Input
392	D27	Output	359	D17	Output
391	D27	Control	358	D17	Control
390	D27	Input	357	D17	Input
389	D26	Output	356	D15	Output
388	D26	Control	355	D15	Control
387	D26	Input	354	D15	Input
386	D25	Output	353	D16	Output
385	D25	Control	352	D16	Control
384	D25	Input	351	D16	Input
383	D24	Output	350	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
349	$\overline{\text{WE2/IORD}}$	Control	316	D5	Control
348	$\overline{\text{WE2/IORD}}$	Input	315	D5	Input
347	D14	Output	314	D6	Output
346	D14	Control	313	D6	Control
345	D14	Input	312	D6	Input
344	D13	Output	311	D1	Output
343	D13	Control	310	D1	Control
342	D13	Input	309	D1	Input
341	D12	Output	308	D2	Output
340	D12	Control	307	D2	Control
339	D12	Input	306	D2	Input
338	D11	Output	305	D3	Output
337	D11	Control	304	D3	Control
336	D11	Input	303	D3	Input
335	D10	Output	302	$\overline{\text{WE0/REG}}$	Output
334	D10	Control	301	$\overline{\text{WE0/REG}}$	Control
333	D10	Input	300	$\overline{\text{WE0/REG}}$	Input
332	D9	Output	299	D0	Output
331	D9	Control	298	D0	Control
330	D9	Input	297	D0	Input
329	$\overline{\text{WE1}}$	Output	296	$\overline{\text{BREQ}}$	Output
328	$\overline{\text{WE1}}$	Control	295	$\overline{\text{BREQ}}$	Control
327	$\overline{\text{WE1}}$	Input	294	$\overline{\text{BREQ}}$	Input
326	D7	Output	293	$\overline{\text{BACK}}$	Output
325	D7	Control	292	$\overline{\text{BACK}}$	Control
324	D7	Input	291	$\overline{\text{BACK}}$	Input
323	D8	Output	290	R/$\overline{\text{W}}$	Output
322	D8	Control	289	R/$\overline{\text{W}}$	Control
321	D8	Input	288	R/$\overline{\text{W}}$	Input
320	D4	Output	287	$\overline{\text{RD/FRAME}}$	Output
319	D4	Control	286	$\overline{\text{RD/FRAME}}$	Control
318	D4	Input	285	$\overline{\text{RD/FRAME}}$	Input
317	D5	Output	284	$\overline{\text{BS}}$	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
283	$\overline{BS}$	Control	250	PCICLK	Input
282	$\overline{BS}$	Input	249	$\overline{GNT0/GNTIN}$	Output
281	$\overline{CS4}$	Output	248	$\overline{GNT0/GNTIN}$	Control
280	$\overline{CS4}$	Control	247	$\overline{GNT0/GNTIN}$	Input
279	$\overline{CS4}$	Input	246	$\overline{PCIRESET}$	Output
278	$\overline{RDY}$	Output	245	$\overline{REQ2}$	Output
277	$\overline{RDY}$	Control	244	$\overline{REQ2}$	Control
276	$\overline{RDY}$	Input	243	$\overline{REQ2}$	Input
275	$\overline{CS2}$	Output	242	$\overline{REQ1}$	Output
274	$\overline{CS2}$	Control	241	$\overline{REQ1}$	Control
273	$\overline{CS2}$	Input	240	$\overline{REQ1}$	Input
272	$\overline{CS5}$	Output	239	$\overline{GNT2}$	Output
271	$\overline{CS5}$	Control	238	$\overline{GNT2}$	Control
270	$\overline{CS5}$	Input	237	$\overline{GNT2}$	Input
269	$\overline{CS6}$	Output	236	$\overline{GNT1}$	Output
268	$\overline{CS6}$	Control	235	$\overline{GNT1}$	Control
267	$\overline{CS6}$	Input	234	$\overline{GNT1}$	Input
266	CLKOUT	Control	233	AD31	Output
265	CLKOUT	Output	232	AD31	Control
264	$\overline{CS0}$	Output	231	AD31	Input
263	$\overline{CS0}$	Control	230	$\overline{REQ3}$	Output
262	$\overline{CS0}$	Input	229	$\overline{REQ3}$	Control
261	$\overline{CS1}$	Output	228	$\overline{REQ3}$	Input
260	$\overline{CS1}$	Control	227	AD30	Output
259	$\overline{CS1}$	Input	226	AD30	Control
258	$\overline{INTA}$	Output	225	AD30	Input
257	$\overline{INTA}$	Control	224	$\overline{GNT3}$	Output
256	$\overline{INTA}$	Input	223	$\overline{GNT3}$	Control
255	$\overline{REQ0/REQOUT}$	Output	222	$\overline{GNT3}$	Input
254	$\overline{REQ0/REQOUT}$	Control	221	AD27	Output
253	$\overline{REQ0/REQOUT}$	Input	220	AD27	Control
252	PCICLK	Output	219	AD27	Input
251	PCICLK	Control	218	AD29	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
217	AD29	Control	184	AD17	Control
216	AD29	Input	183	AD17	Input
215	AD26	Output	182	AD19	Output
214	AD26	Control	181	AD19	Control
213	AD26	Input	180	AD19	Input
212	AD28	Output	179	AD16	Output
211	AD28	Control	178	AD16	Control
210	AD28	Input	177	AD16	Input
209	CBE3	Output	176	AD18	Output
208	CBE3	Control	175	AD18	Control
207	CBE3	Input	174	AD18	Input
206	AD25	Output	173	$\overline{\text{IRDY}}$	Output
205	AD25	Control	172	$\overline{\text{IRDY}}$	Control
204	AD25	Input	171	$\overline{\text{IRDY}}$	Input
203	IDSEL	Output	170	CBE2	Output
202	IDSEL	Control	169	CBE2	Control
201	IDSEL	Input	168	CBE2	Input
200	AD24	Output	167	$\overline{\text{TRDY}}$	Output
199	AD24	Control	166	$\overline{\text{TRDY}}$	Control
198	AD24	Input	165	$\overline{\text{TRDY}}$	Input
197	AD21	Output	164	$\overline{\text{PCIFRAME}}$	Output
196	AD21	Control	163	$\overline{\text{PCIFRAME}}$	Control
195	AD21	Input	162	$\overline{\text{PCIFRAME}}$	Input
194	AD23	Output	161	$\overline{\text{STOP}}$	Output
193	AD23	Control	160	$\overline{\text{STOP}}$	Control
192	AD23	Input	159	$\overline{\text{STOP}}$	Input
191	AD20	Output	158	PAR	Output
190	AD20	Control	157	PAR	Control
189	AD20	Input	156	PAR	Input
188	AD22	Output	155	$\overline{\text{DEVSEL}}$	Output
187	AD22	Control	154	$\overline{\text{DEVSEL}}$	Control
186	AD22	Input	153	$\overline{\text{DEVSEL}}$	Input
185	AD17	Output	152	$\overline{\text{LOCK}}$	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
151	$\overline{\text{LOCK}}$	Control	118	AD12	Control
150	$\overline{\text{LOCK}}$	Input	117	AD12	Input
149	AD15	Output	116	AD10	Output
148	AD15	Control	115	AD10	Control
147	AD15	Input	114	AD10	Input
146	AD13	Output	113	AD4	Output
145	AD13	Control	112	AD4	Control
144	AD13	Input	111	AD4	Input
143	$\overline{\text{PERR}}$	Output	110	AD2	Output
142	$\overline{\text{PERR}}$	Control	109	AD2	Control
141	$\overline{\text{PERR}}$	Input	108	AD2	Input
140	$\overline{\text{SERR}}$	Output	107	AD8	Output
139	$\overline{\text{SERR}}$	Control	106	AD8	Control
138	$\overline{\text{SERR}}$	Input	105	AD8	Input
137	AD11	Output	104	AD7	Output
136	AD11	Control	103	AD7	Control
135	AD11	Input	102	AD7	Input
134	AD9	Output	101	AD0	Output
133	AD9	Control	100	AD0	Control
132	AD9	Input	99	AD0	Input
131	CBE1	Output	98	AD5	Output
130	CBE1	Control	97	AD5	Control
129	CBE1	Input	96	AD5	Input
128	AD14	Output	95	AD3	Output
127	AD14	Control	94	AD3	Control
126	AD14	Input	93	AD3	Input
125	CBE0	Output	92	AD1	Output
124	CBE0	Control	91	AD1	Control
123	CBE0	Input	90	AD1	Input
122	AD6	Output	89	NMI	Output
121	AD6	Control	88	NMI	Control
120	AD6	Input	87	NMI	Input
119	AD12	Output	86	IRQ/$\overline{\text{IRL0}}$	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
85	IRQ/$\overline{\text{IRL0}}$	Control	58	TCLK/$\overline{\text{IOIS16}}$	
84	IRQ/$\overline{\text{IRL0}}$	Input	57	TCLK/$\overline{\text{IOIS16}}$	
83	IRQ/$\overline{\text{IRL1}}$	Output	56	SCIF0_TXD/HSPI_TX/ $\overline{\text{FWE}}$/MODE8	Output
82	IRQ/$\overline{\text{IRL1}}$	Control	55	SCIF0_TXD/HSPI_TX/ $\overline{\text{FWE}}$/MODE8	Control
81	IRQ/$\overline{\text{IRL1}}$	Input	54	SCIF0_TXD/HSPI_TX/ $\overline{\text{FWE}}$/MODE8	Input
80	IRQ/$\overline{\text{IRL2}}$	Output	53	SCIF0_RXD/HSPI_RX/FRB	Output
79	IRQ/$\overline{\text{IRL2}}$	Control	52	SCIF0_RXD/HSPI_RX/FRB	Control
78	IRQ/$\overline{\text{IRL2}}$	Input	51	SCIF0_RXD/HSPI_RX/FRB	Input
77	IRQ/$\overline{\text{IRL3}}$	Output	50	$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE	Output
76	IRQ/$\overline{\text{IRL3}}$	Control	49	$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE	Control
75	IRQ/$\overline{\text{IRL3}}$	Input	48	$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE	Input
74	IRQ/$\overline{\text{IRL4}}$/FD4/MODE3	Output	47	$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CS}}$/$\overline{\text{FSE}}$	Output
73	IRQ/$\overline{\text{IRL4}}$/FD4/MODE3	Control	46	$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CS}}$/$\overline{\text{FSE}}$	Control
72	IRQ/$\overline{\text{IRL4}}$/FD4/MODE3	Input	45	$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CS}}$/$\overline{\text{FSE}}$	Input
71	IRQ/$\overline{\text{IRL5}}$/FD5/MODE4	Output	44	SCIF1_SCK/MCCMD	Output
70	IRQ/$\overline{\text{IRL5}}$/FD5/MODE4	Control	43	SCIF1_SCK/MCCMD	Control
69	IRQ/$\overline{\text{IRL5}}$/FD5/MODE4	Input	42	SCIF1_SCK/MCCMD	Input
68	IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	Output	41	SCIF1_TXD/MCCLK/MODE5	Output
67	IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	Control	40	SCIF1_TXD/MCCLK/MODE5	Control
66	IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	Input	39	SCIF1_TXD/MCCLK/MODE5	Input
65	IRQ/$\overline{\text{IRL7}}$/FD7	Output	38	SCIF1_RXD/MCDAT	Output
64	IRQ/$\overline{\text{IRL7}}$/FD7	Control	37	SCIF1_RXD/MCDAT	Control
63	IRQ/$\overline{\text{IRL7}}$/FD7	Input	36	SCIF1_RXD/MCDAT	Input
62	SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$	Output	35	SIOF_TXD/HAC_SDOUT/ SSI_SDATA	Output
61	SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$	Control	34	SIOF_TXD/HAC_SDOUT/ SSI_SDATA	Control
60	SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$	Input	33	SIOF_TXD/HAC_SDOUT/ SSI_SDATA	Input
59	TCLK/$\overline{\text{IOIS16}}$		32	SIOF_RXD/HAC_SDIN/ SSI_SCK	Output

Number	Pin Name	I/O*	Number	Pin Name	I/O*
31	SIOF_RXD/HAC_SDIN/ SSI_SCK	Control	14	AUDATA2/FD2	Output
30	SIOF_RXD/HAC_SDIN/ SSI_SCK	Input	13	AUDATA2/FD2	Control
29	SIOF_SYNC/HAC_SYNC/ SSI_WS	Output	12	AUDATA2/FD2	Input
28	SIOF_SYNC/HAC_SYNC/ SSI_WS	Control	11	AUDATA3/FD3	Output
27	SIOF_SYNC/HAC_SYNC/ SSI_WS	Input	10	AUDATA3/FD3	Control
26	SIOF_MCLK/$\overline{\text{HAC\_RES}}$	Output	9	AUDATA3/FD3	Input
25	SIOF_MCLK/$\overline{\text{HAC\_RES}}$	Control	8	AUDCK/FALE	Output
24	SIOF_MCLK/$\overline{\text{HAC\_RES}}$	Input	7	AUDCK/FALE	Control
23	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Output	6	AUDCK/FALE	Input
22	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Control	5	AUDSYNC/$\overline{\text{FCE}}$	Output
21	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Input	4	AUDSYNC/$\overline{\text{FCE}}$	Control
20	AUDATA0/FD0	Output	3	AUDSYNC/$\overline{\text{FCE}}$	Input
19	AUDATA0/FD0	Control	2	$\overline{\text{ASEBRK}}$/BRKACK	Output
18	AUDATA0/FD0	Input	1	$\overline{\text{ASEBRK}}$/BRKACK	Control
17	AUDATA1/FD1	Output	0	$\overline{\text{ASEBRK}}$/BRKACK	Input
16	AUDATA1/FD1	Control		To TDO	
15	AUDATA1/FD1	Input			

Note: * Control is an active-high signal. When Control is driven high, the corresponding pin is driven according to the OUT value.

30.5 Operation

30.5.1 TAP Control

Figure 30.3 shows the internal states of the TAP controller. The state transitions basically conform to the JTAG standard.

- State transitions occur according to the TMS value at the rising edge of the TCK signal.
- The TDI value is sampled at the rising edge of the TCK signal and shifted at the falling edge of the TCK signal.
- The TDO value is changed at the falling edge of the TCK signal. The TDO signal is in a Hi-Z 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 asynchronously with the TCK signal.

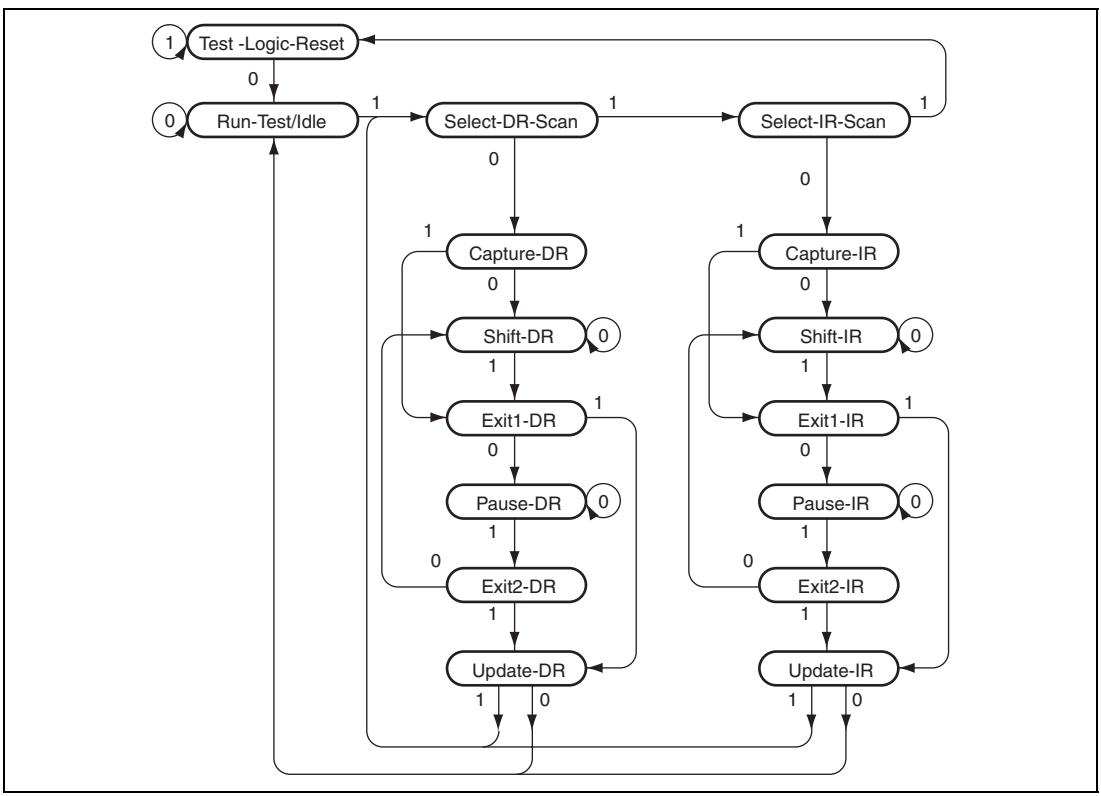


Figure 30.3 TAP Controller State Transitions

30.5.2 H-UDI Reset

A power-on reset is generated by the SDIR command. After the H-UDI reset assert command has been sent from the H-UDI pin, sending the H-UDI reset negate command resets the CPU (see figure 30.4). The required time between the H-UDI reset assert and H-UDI reset negate commands is the same as the time for holding the reset pin low in order to reset this LSI by a power-on reset.

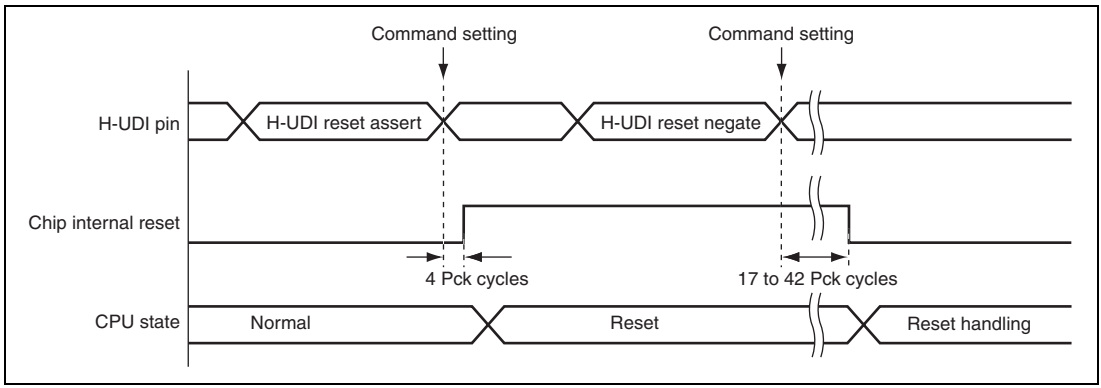


Figure 30.4 H-UDI Reset

30.5.3 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting the appropriate command in SDIR from the H-UDI. An H-UDI interrupt is a general exception/interrupt operation, resulting in branching to the VBR address. The H-UDI returns from the interrupt handling routine with a RTE instruction. When an H-UDI interrupt occurs, the exception code H'600 is stored in the interrupt event register (INTEVT). The priority level for the H-UDI interrupt can be specified by the bits 28 to 24 in INT2PRI3. An H-UDI interrupt request signal is asserted when the INTREQ bit in SDINT is set to 1 by setting the appropriate command. Since the interrupt request signal is not negated until the INTREQ bit is cleared to 0 by software, it is not possible to lose the interrupt request. While an H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins.

30.6 Usage Notes

- Once an SDIR command is set, it will be changed only by an assertion of the $\overline{\text{TRST}}$ signal, making the TAP controller Test-Logic-Reset state, or writing other commands from the H-UDI.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.
- An H-UDI interrupt or an H-UDI reset can be accepted to cancel sleep mode.

Section 31 Electrical Characteristics

31.1 Absolute Maximum Ratings

Table 31.1 Absolute Maximum Ratings[*1, *2]

Item	Symbol	Value	Unit
I/O, RTC, DDRIF power supply voltage	V_{DDQ} V_{DD-RTC}	–0.3 to 4.6	V
	$V_{CCQ-DDR}$	–0.3 to 3.6	
Internal power supply voltage	V_{DD} $V_{DD-PLL1/2/3}$ $V_{DD-DLL1/2}$	–0.3 to 1.8	V
Input voltage	V_{in}	–0.3 to V_{DDQ} + 0.3[*3]	V
	V_{in-DDR}	–0.3 to $V_{CCQ-DDR}$ + 0.3[*3]	
Operating temperature	T_{opr}	–20 to 75	°C
		–40 to 85[*4]	
Storage temperature	T_{stg}	–55 to 125	°C

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 connected to GND.
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 On and Off Power Supply.
6. It is prohibited to input signals to the following seven pins immediately after power-on 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)

31.2 DC Characteristics

Table 31.2 DC Characteristics (T_a = –20 to 75°C / –40 to 85°C)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		V_{DDQ}	3.0	3.3	3.6	V	Normal operation, sleep mode, module standby mode
		V_{DD-RTC}	3.0	3.3	3.6		
			2.0	—	3.6		RTC backup mode
		V_{DD} $V_{DD-PLL1/2/3}$ $V_{DD-DLL1/2}$	1.15	1.25	1.35		Normal operation, sleep mode
		$V_{CCQ-DDR}$	2.3	2.5	2.7		Normal operation, sleep mode, DDR backup mode
Reference voltage		DDR-V_{REF}	1.15	1.25	1.35		
Current dissipation	Normal operation	I_{DD}	—	740	1300	mA	Ick = 400MHz
	Sleep mode		—	—	530		
	Normal operation	I_{DDQ}	—	150	220		Ick = 400MHz
	Sleep mode		—	—	90		Bck = 100MHz
	Normal operation	ΣI_{DD-PLL}	—	—	25		
		ΣI_{DD-DLL}	—	—	400	µA	
	DDR Normal operation	$I_{CCQ-DDR}$	—	—	530	mA	DDRck = 160MHz
	DDR backup mode		—	—	160	mA	Supply only DDR I/O ($V_{CCQ-DDR}$ = 2.5V, DDR-V_{REF} = 1.25V)
	RTC operation	I_{DD-RTC}	—	—	660	µA	V_{DD-RTC} = 3.3V
	RTC backup mode		—	—	8	µA	32.768kHz operation Supply only V_{DD-RTC} = 2.0V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	$\overline{\text{PRESET}}$, 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_{IH}	$V_{DDQ} \times 0.9$	—	$V_{DDQ} + 0.3$	V	V_{DDQ} = 3.0 to 3.6 V
	DDR pins		DDR-V_{REF} + 0.15	—	$V_{CCQ-DDR}$ + 0.3		DDR-V_{REF} = 1.15 to 1.35V $V_{CCQ-DDR}$ = 2.3 to 2.7V
	PCICLK		$V_{DDQ} \times 0.6$	—	$V_{DDQ} + 0.3$		V_{DDQ} = 3.0 to 3.6V
	Other PCI pins		$V_{DDQ} \times 0.5$	—	$V_{DDQ} + 0.3$		
	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	—	$V_{DDQ} \times 0.1$		V_{DDQ} = 3.0 to 3.6 V
	DDR pins		–0.3	—	DDR-V_{REF} – 0.15		DDR-V_{REF} = 1.15 to 1.35V
	PCICLK		–0.3	—	$V_{DDQ} \times 0.2$		V_{DDQ} = 3.0 to 3.6V
	Other PCI pins		–0.3	—	$V_{DDQ} \times 0.3$		
	Other input pins		–0.3	—	$V_{DDQ} \times 0.2$		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leak current	DDR pins	|L|	—	—	2	µA	V_{IN} = 0.5, $V_{CCQ-DDR}$ –0.5V
	All input pins	|Iin|	—	—	1		V_{IN} = 0.5, V_{DDQ} –0.5V
Three-state leak current	I/O, all output pins (off condition)	|Isti|	—	—	1	µA	V_{IN} = 0.5, V_{DDQ} –0.5V
Output voltage	PCI pins	V_{OH}	2.4	—	—	V	V_{DDQ} = 3.0 to 3.6V I_{OH} = –4mA
	DDR pins		1.84	—	—		$V_{CCQ-DDR}$ = 2.3V I_{OH} = –7.6mA
	Other output pins		2.4	—	—		V_{DDQ} = 3.0 to 3.6V I_{OH} = –2mA
	PCI pins	V_{OL}	—	—	0.55		V_{DDQ} = 3.0 to 3.6V I_{OL} = 4mA
	DDR pins		—	—	0.54		$V_{CCQ-DDR}$ = 2.3V I_{OL} = 7.6mA
	Other output pins		—	—	0.55		V_{DDQ} = 3.0 to 3.6V 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 with all output pins unload.

Table 31.3 Permissible Output Currents

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin; DDR pins)	I_{OL}	—	—	16	mA
Permissible output low current (per pin; PCI pins)		—	—	4	
Permissible output low current (per pin; other than DDR and PCI pins)		—	—	2	
Permissible output low current (total)	ΣI_{OL}	—	—	120	
Permissible output high current (per pin; DDR pins)	$-I_{OH}$	—	—	16	
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 that the setup time and hold times for each input signal are observed.

Table 31.4 Clock Timing

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	CPU, FPU, cache, TLB	f	2	—	402	MHz
	DDR-SDRAM bus		112	—	164	
	External bus		2	—	101	
	PCI bus		DC	—	67	
	Peripheral modules		2.5	—	51	
	RTC oscillator		32	—	33	kHz

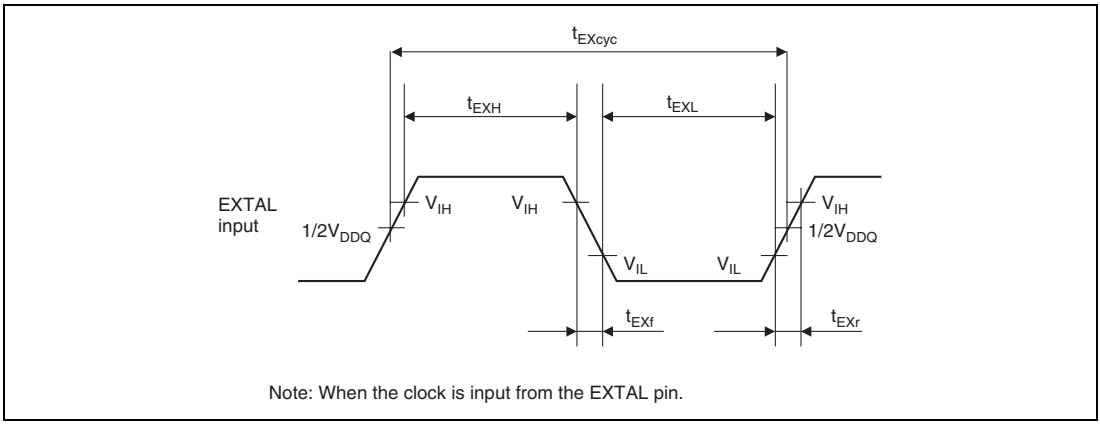
31.3.1 Clock and Control Signal Timing

Table 31.5 Clock and 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	Figure
EXTAL clock input frequency	PLL1 24-times/PLL2 operation	f_{EX}	2	33.4	MHz	
EXTAL clock input cycle time		t_{EXcyc}	30	500	ns	31.1
EXTAL clock input low-level pulse width		t_{EXL}	3.5	—	ns	31.1
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 frequency is 33.4MHz. when a 3rd overtone crystal resonator is used, an external tank circuit is necessary.
2. The load copacitance connected to the CLKOUT pin should be a maximum of 50 pF.
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

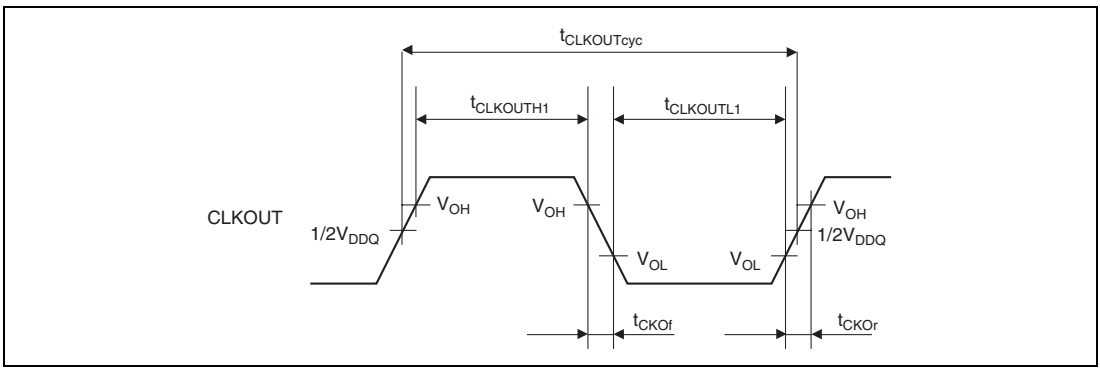


Figure 31.2 CLKOUT Clock Output Timing (1)

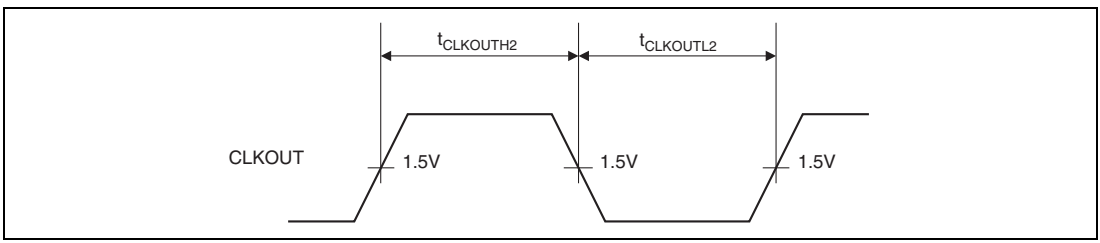
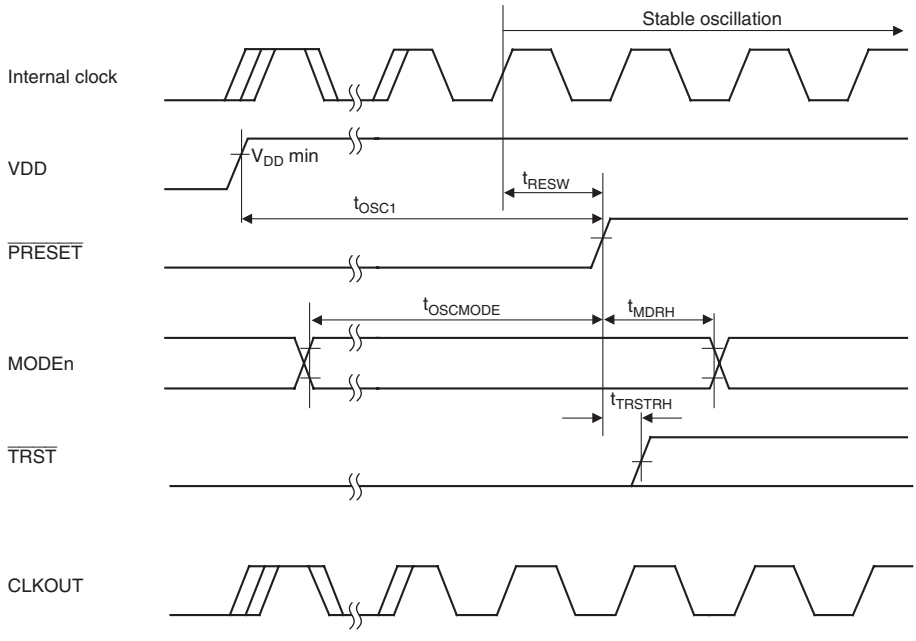


Figure 31.3 CLKOUT Clock Output Timing (2)



Note: Oscillation settling time when on-chip resonator is used.

Figure 31.4 Power-On Oscillation Settling Time

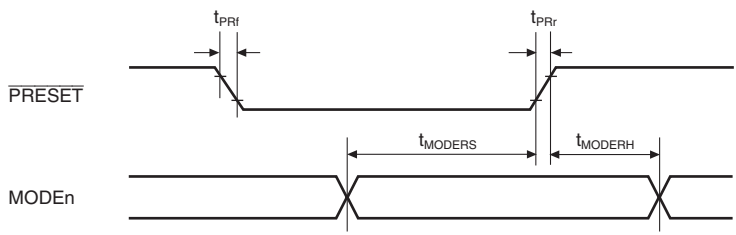


Figure 31.5 MODE pins Setup/Hold Timing

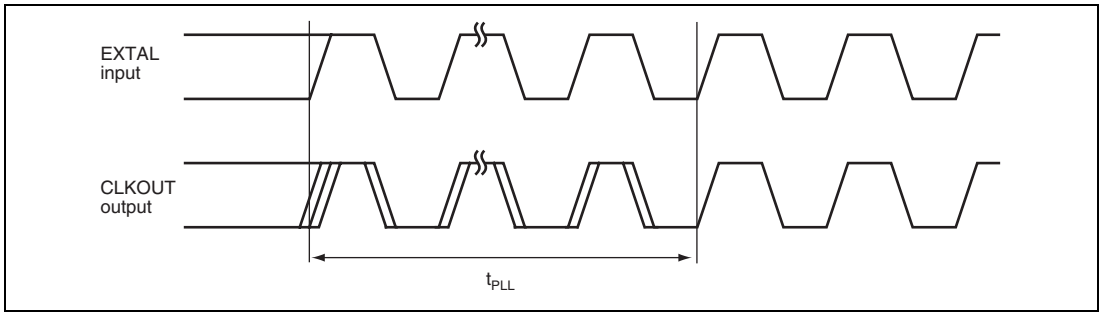


Figure 31.6 PLL Synchronization Settling Time

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	Figure
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	2.5	—	ns	31.7
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1.5	—	ns	31.7
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	6	ns	31.7
Bus three-state delay time	t_{BOFF1}	—	12	ns	31.7
Bus buffer on time	t_{BON1}	—	12	ns	31.7

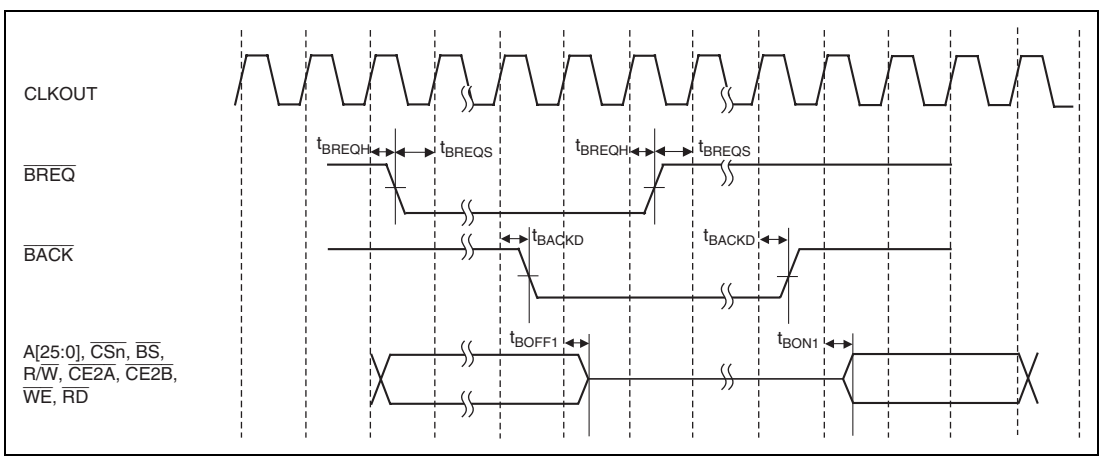


Figure 31.7 Control Signal Timing

31.3.3 Bus Timing

Table 31.7 Bus 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	Notes
Address delay time	t_{AD}	1.5	6	ns	
$\overline{BS}$ delay time	t_{BSD}	1.5	6	ns	
$\overline{CS}$ delay time	t_{CSD}	1.5	6	ns	
R/$\overline{W}$ delay time	t_{RWD}	1.5	6	ns	
$\overline{RD}$ 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 falling edge
$\overline{WE}$ 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	

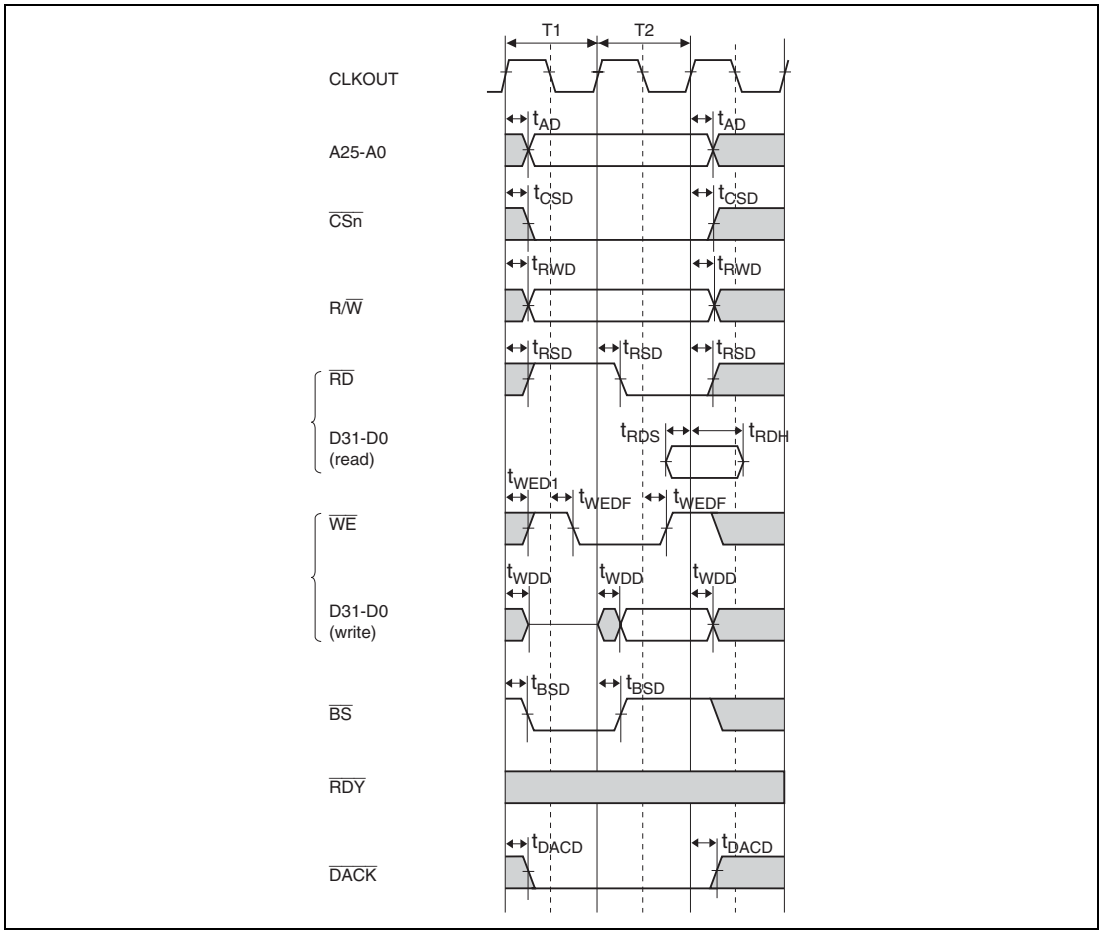


Figure 31.8 SRAM Bus Cycle: Basic Bus Cycle (No Wait)

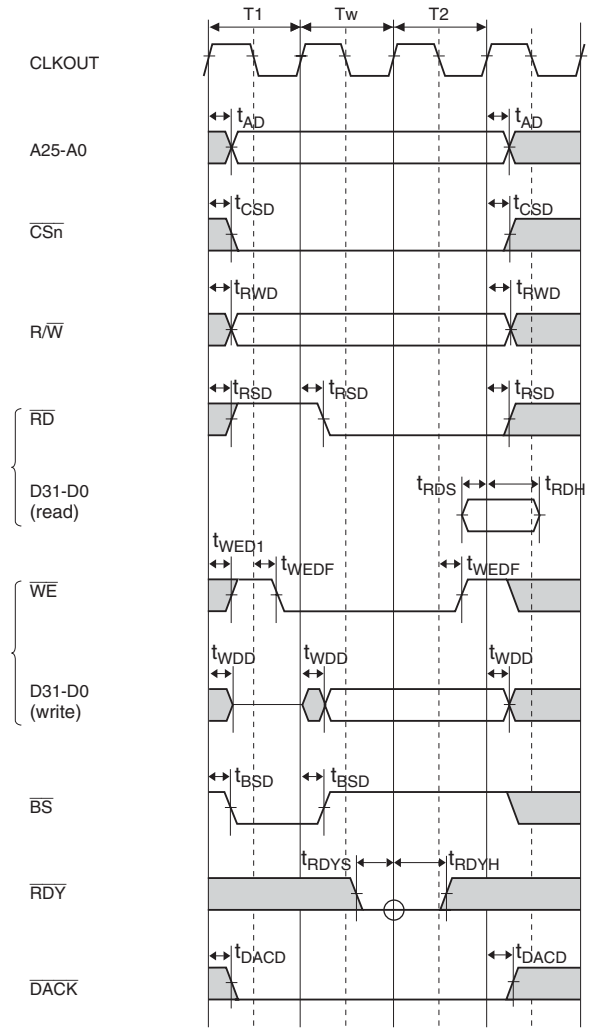


Figure 31.9 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait)

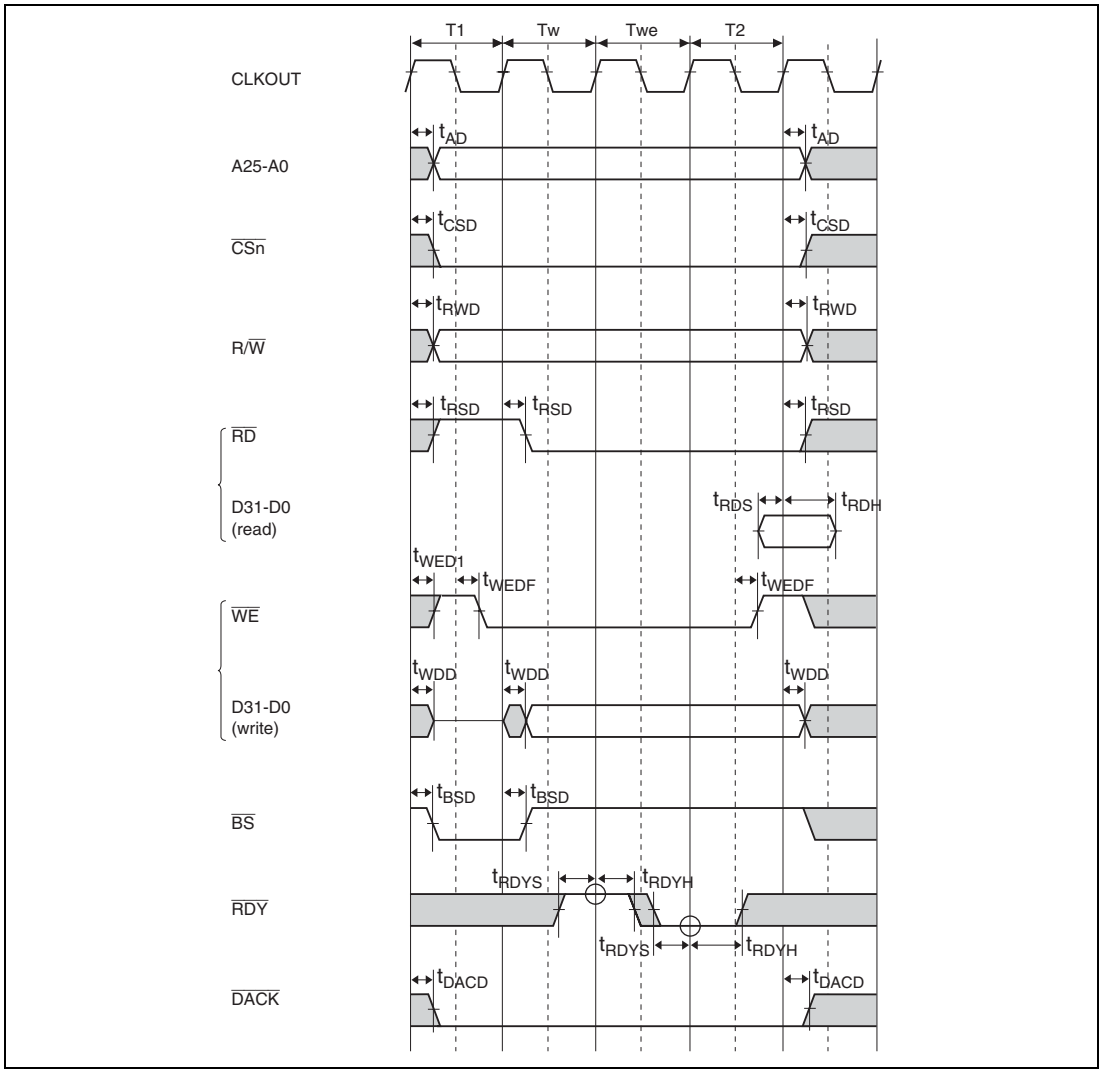


Figure 31.10 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait + One External Wait)

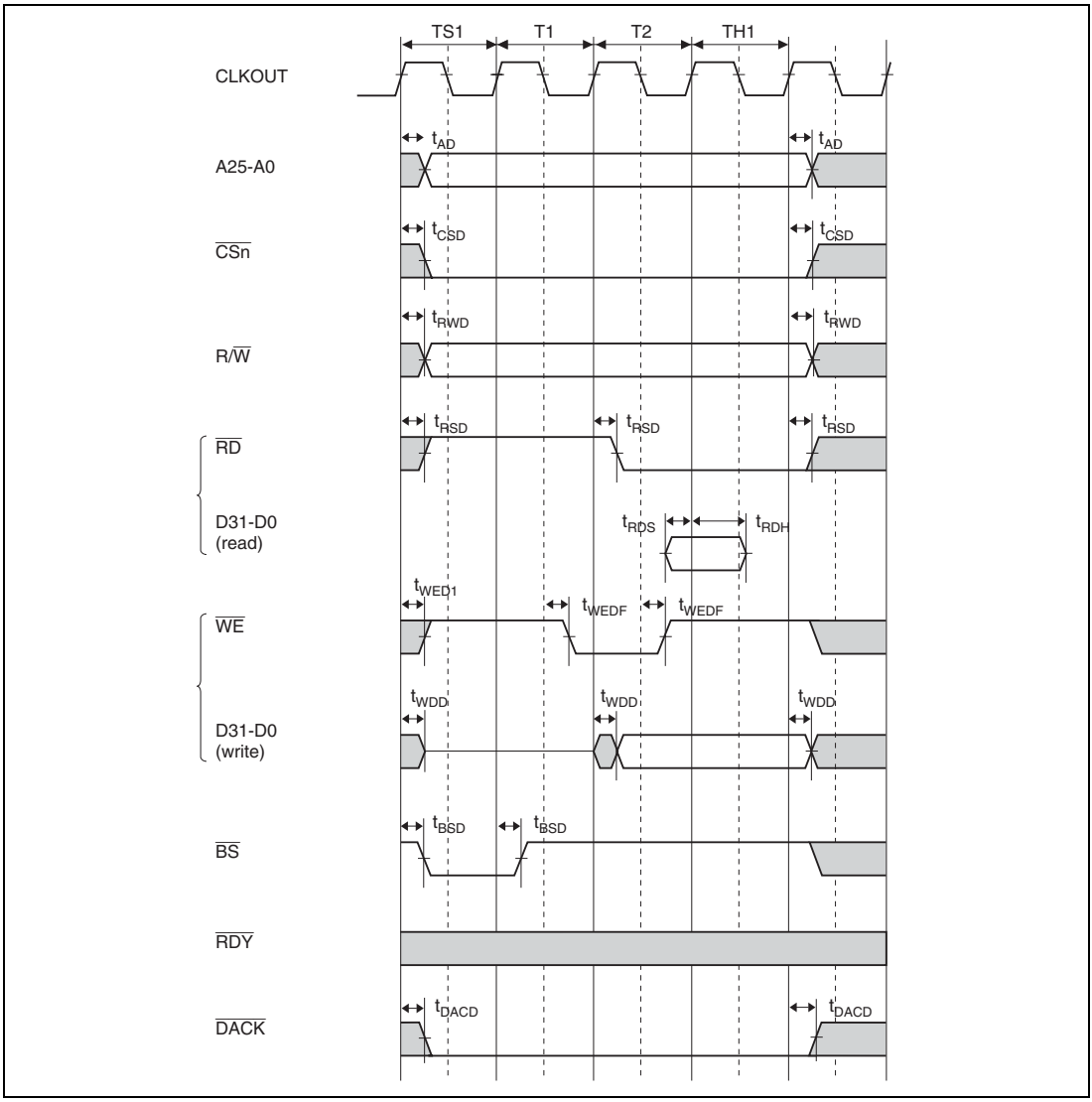


Figure 31.11 SRAM Bus Cycle: Basic Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0, WTS = 1, WTH = 1)

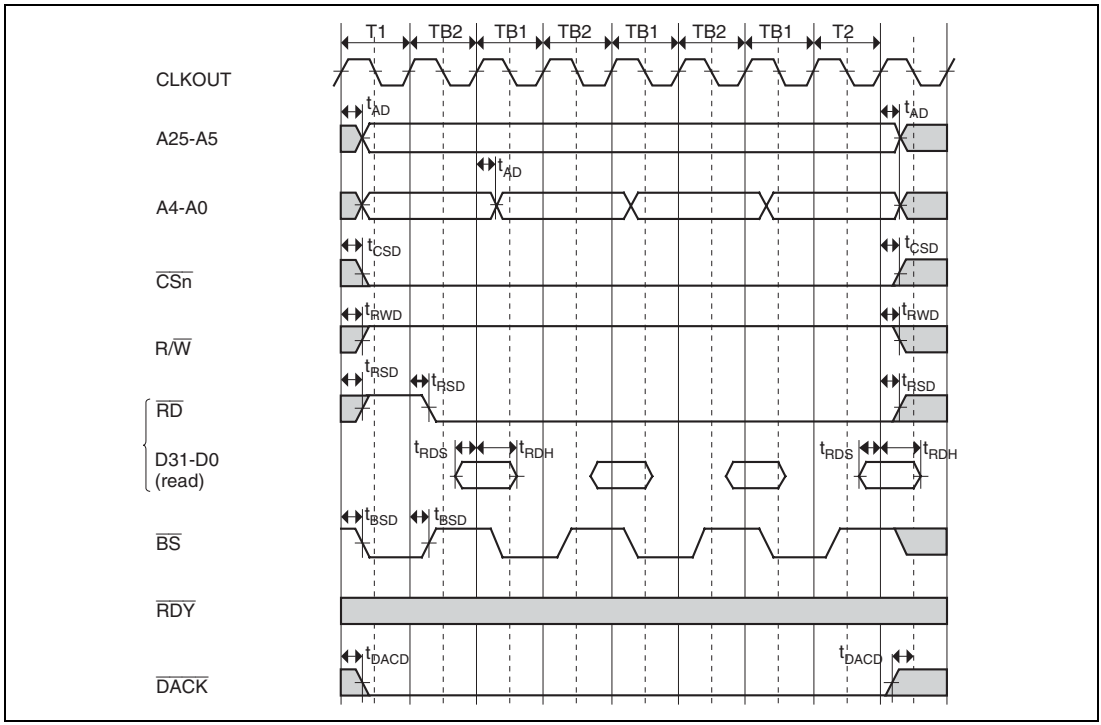


Figure 31.12 Burst ROM Bus Cycle (No Wait)

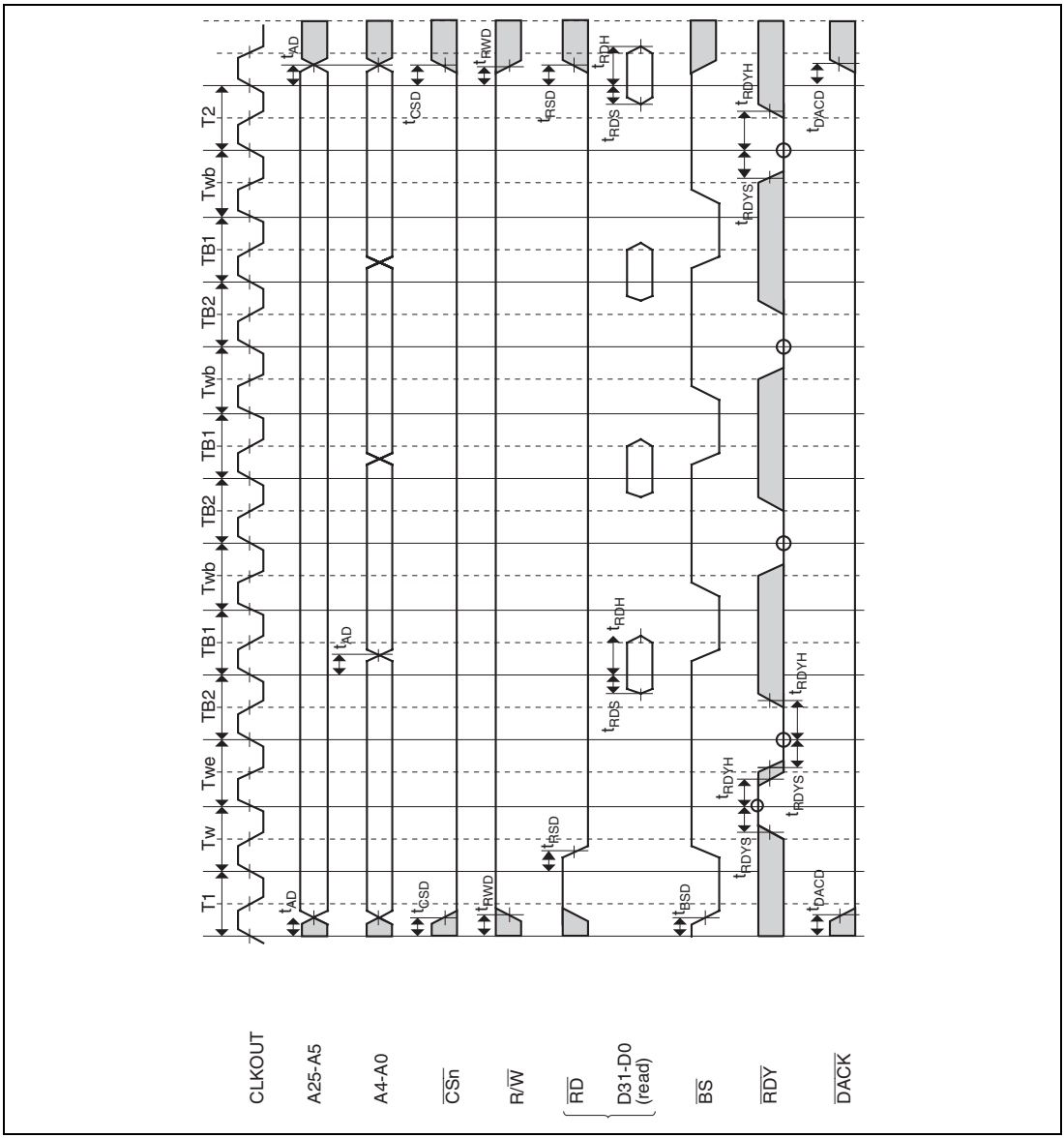


Figure 31.13 Burst ROM Bus Cycle
(1st Data: One Internal Wait + One External Wait ; 2nd/3rd/4th Data: One Internal Wait)

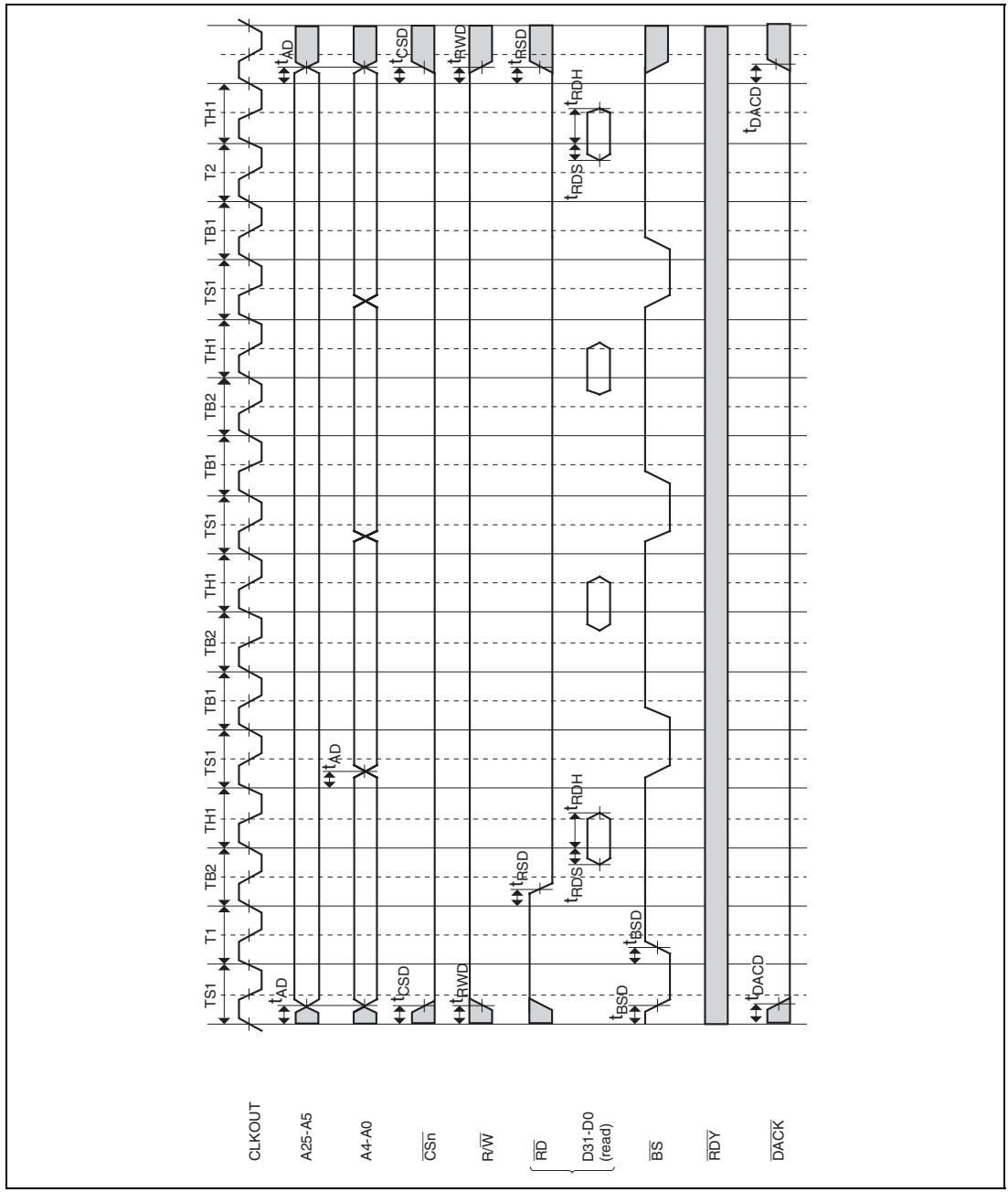


Figure 31.14 Burst ROM Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

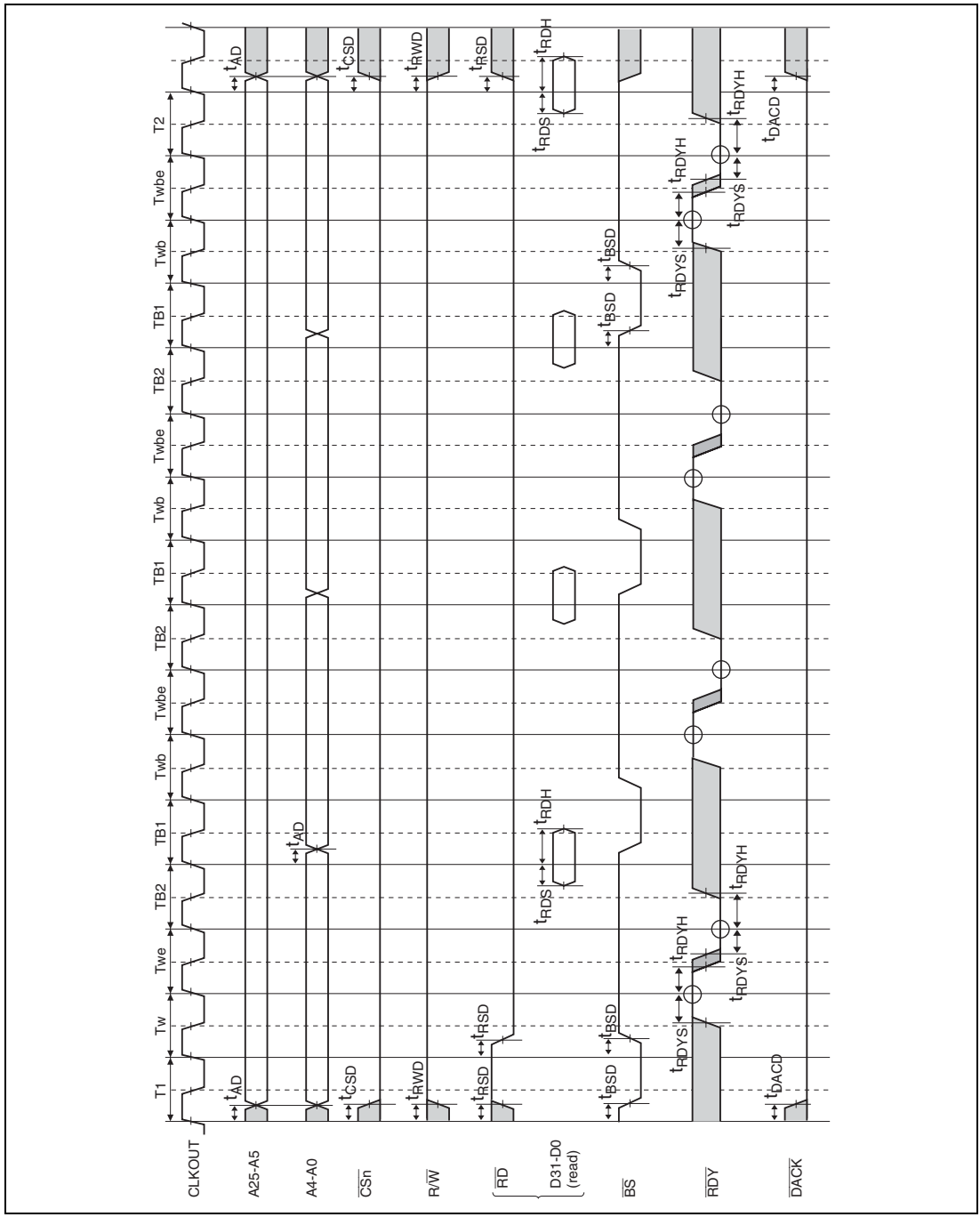


Figure 31.15 Burst ROM Bus Cycle (One Internal Wait + One External Wait)

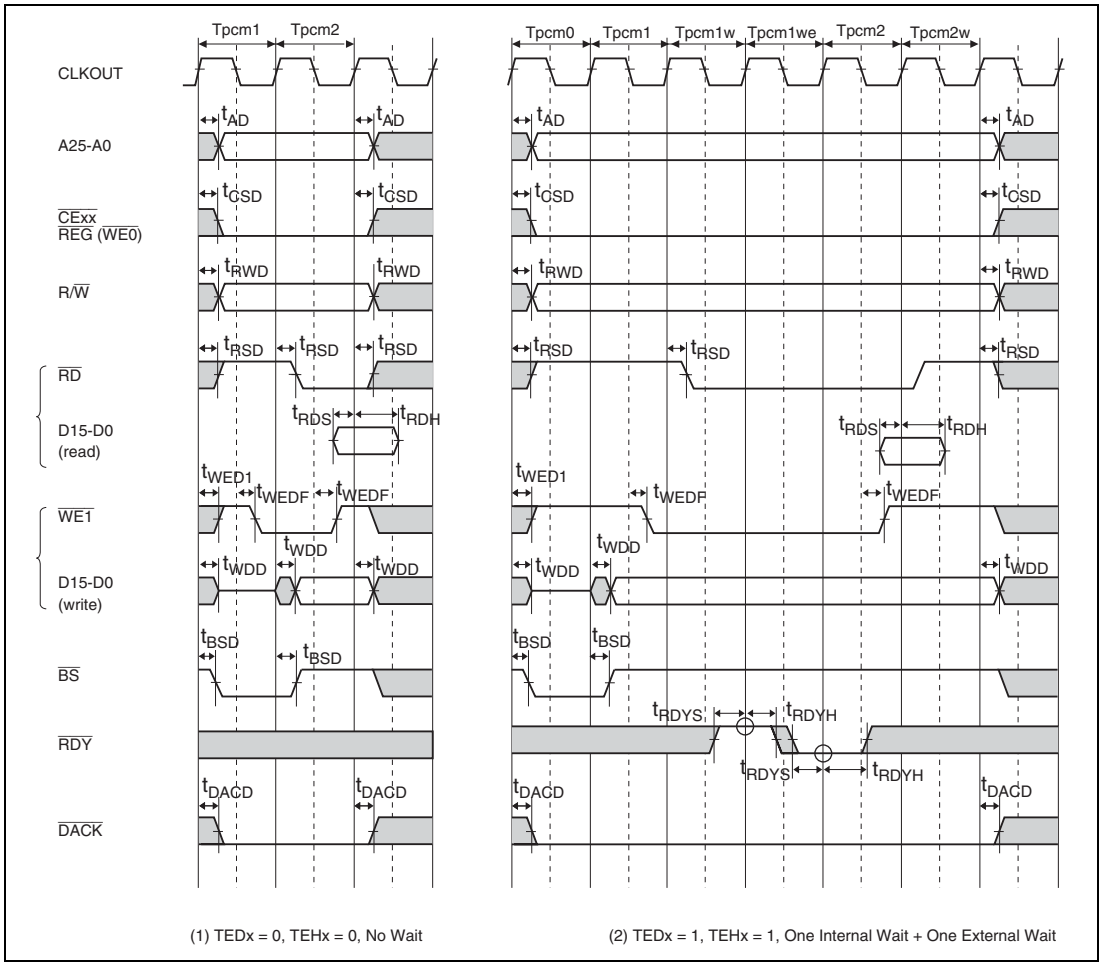


Figure 31.16 PCMCIA Memory Bus Cycle

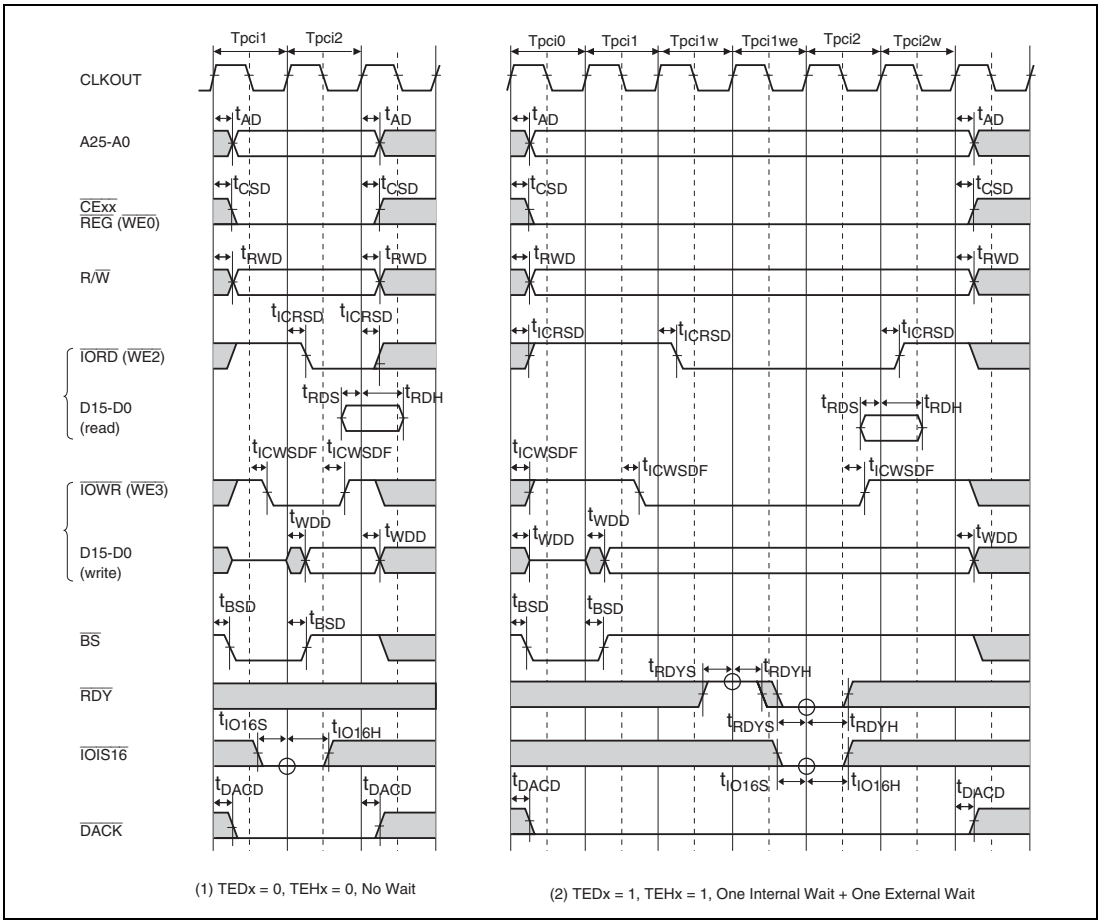


Figure 31.17 PCMCIA I/O Bus Cycle

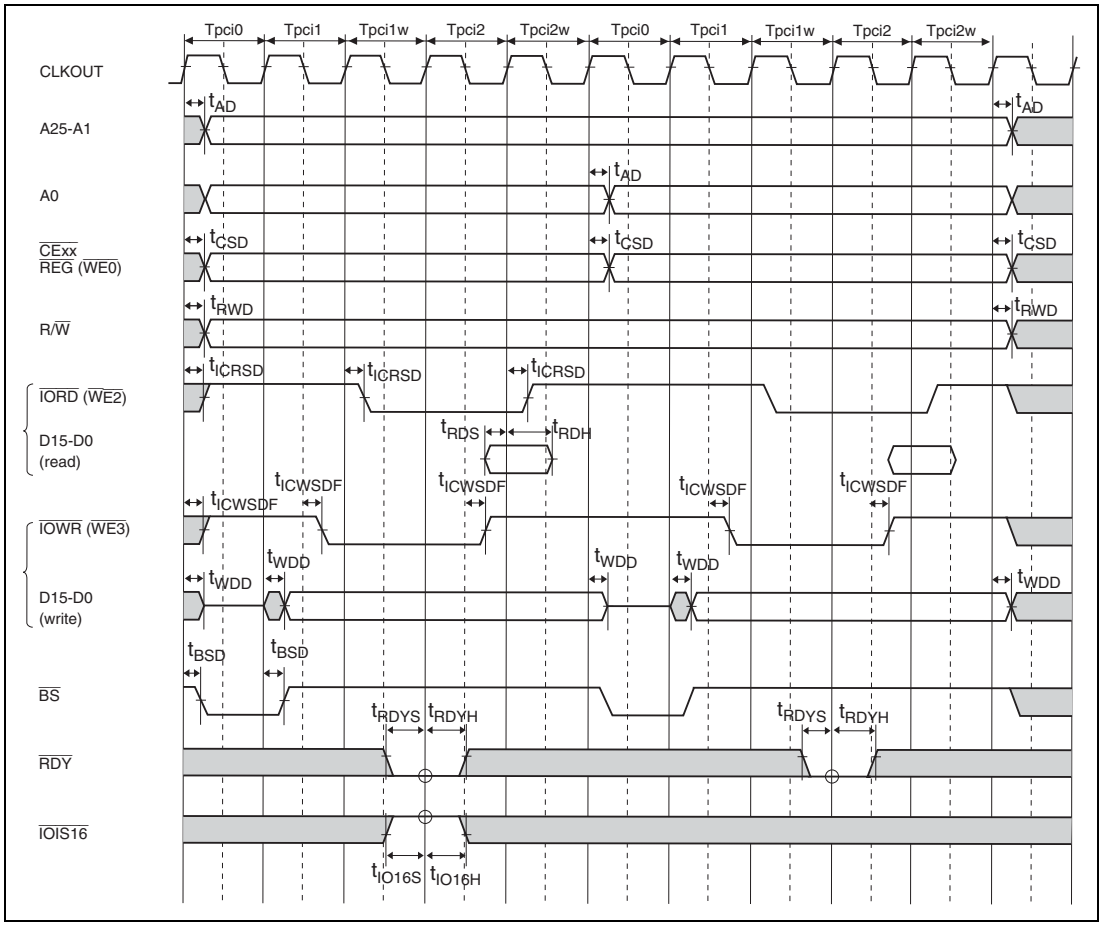


Figure 31.18 PCMCIA I/O Bus Cycle
(TEDx = 1, THEx = 1, IW/PCIW = 1, One Internal Wait, Dynamic Bus Sizing)

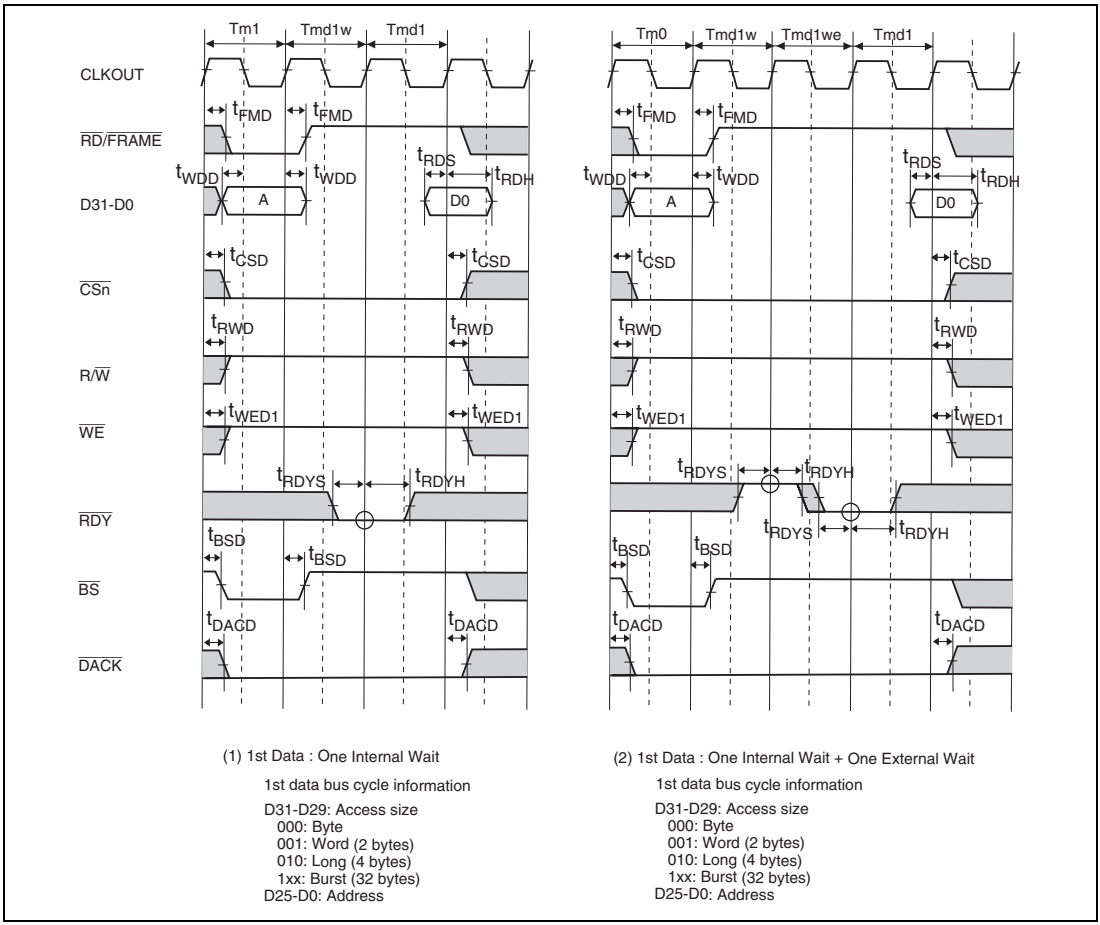


Figure 31.19 MPX Basic Bus Cycle: Read

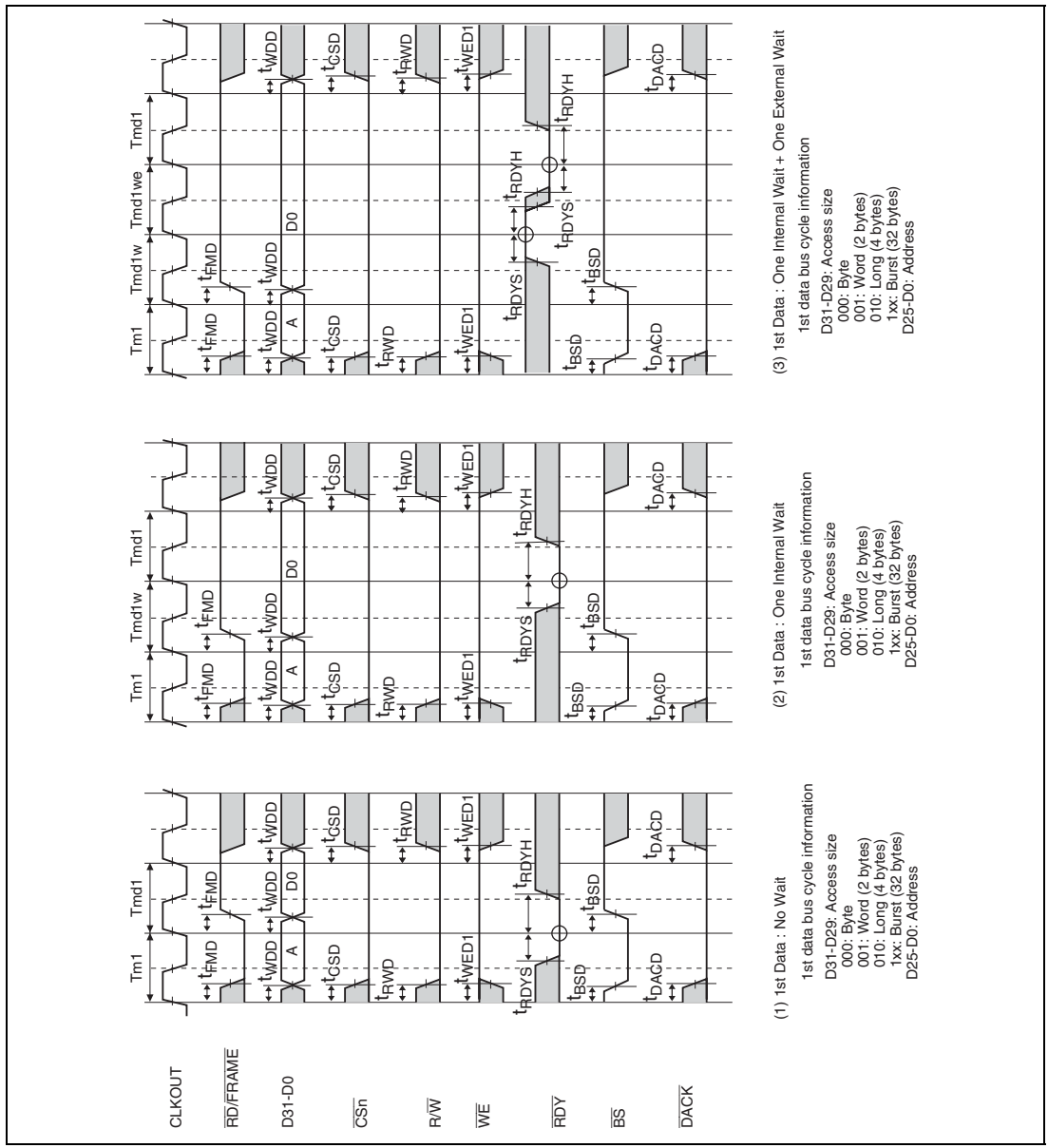


Figure 31.20 MPX Basic Bus Cycle: Write

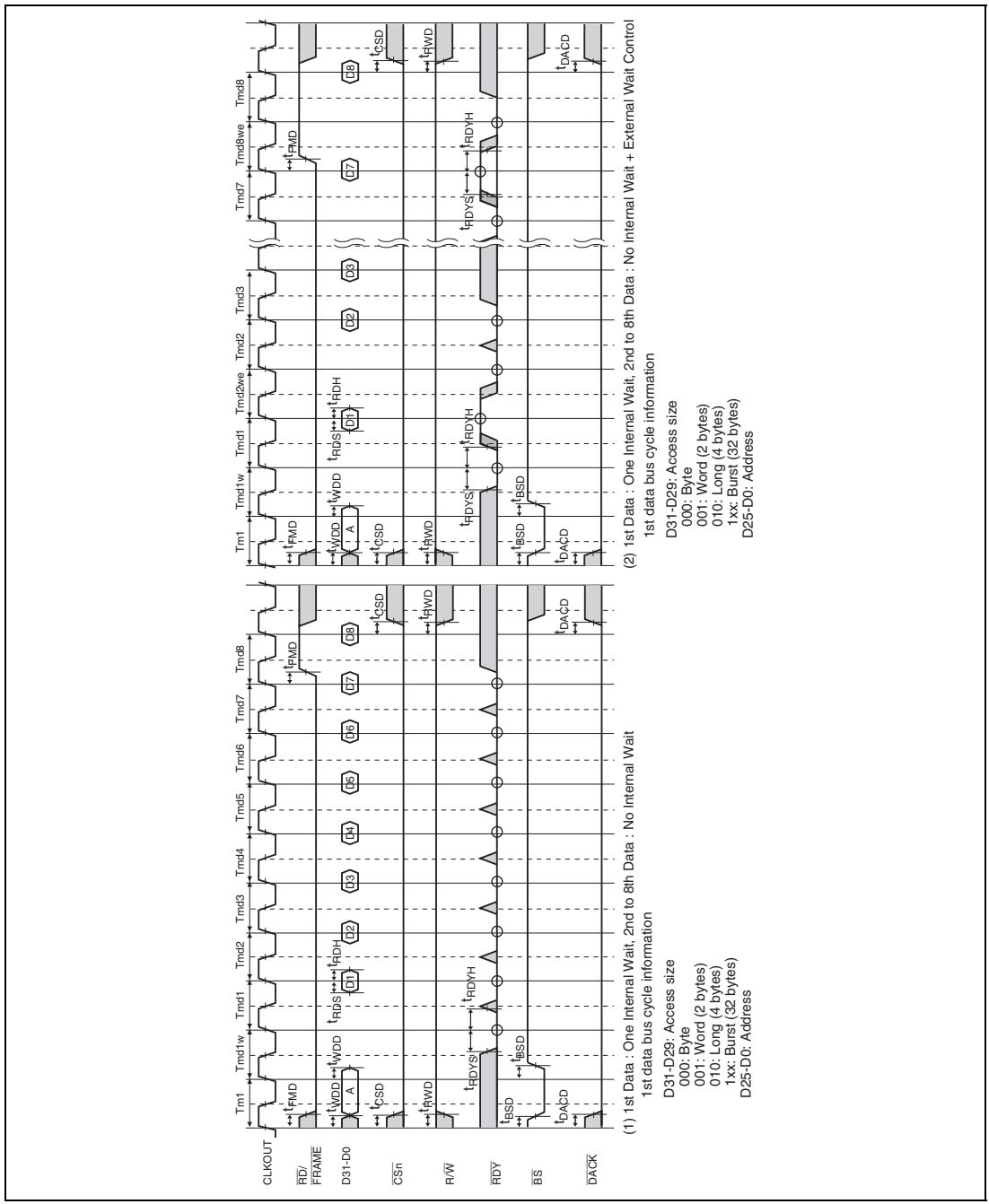


Figure 31.21 MPX Bus Cycle: Burst Read

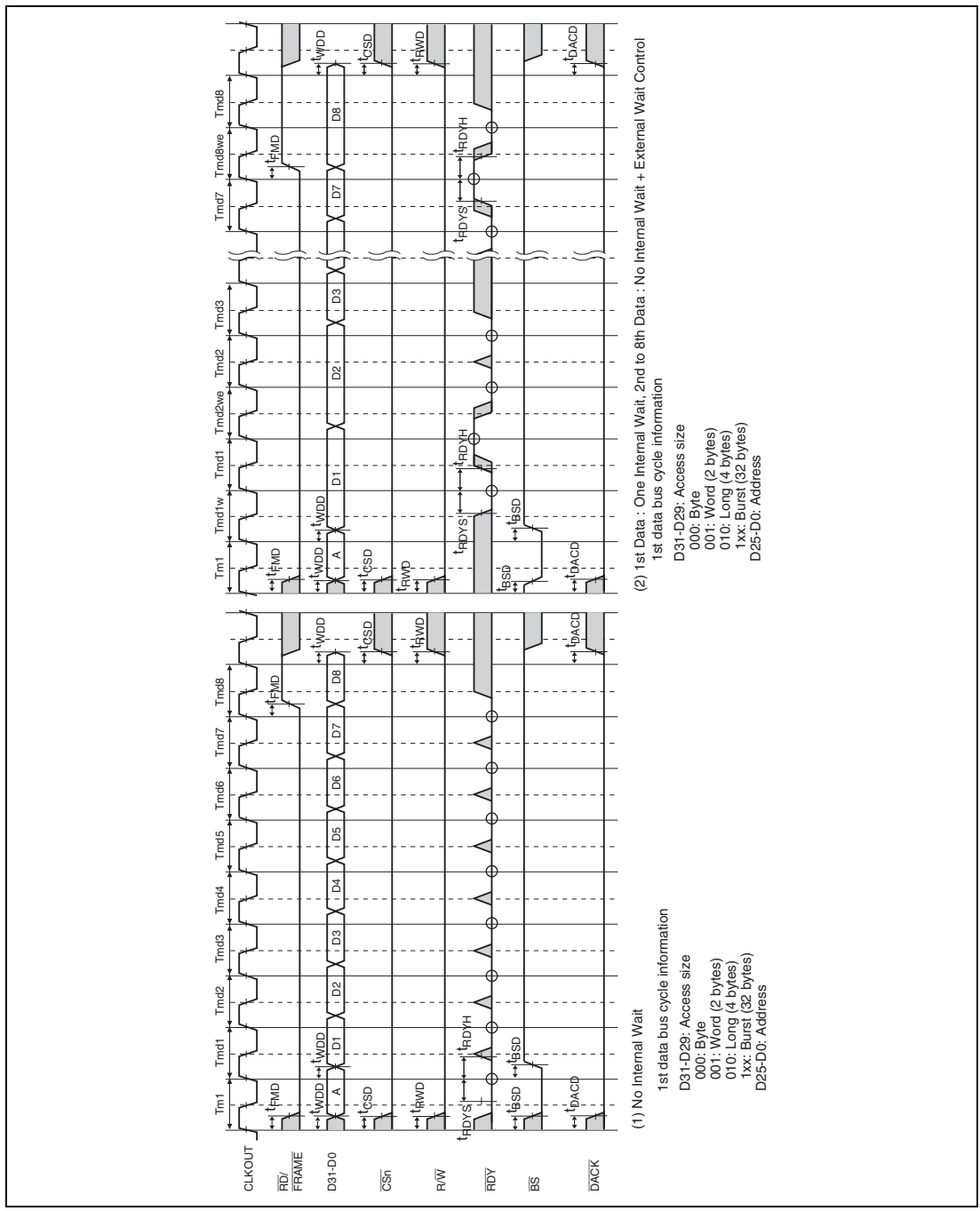


Figure 31.22 MPX Bus Cycle: Burst Write

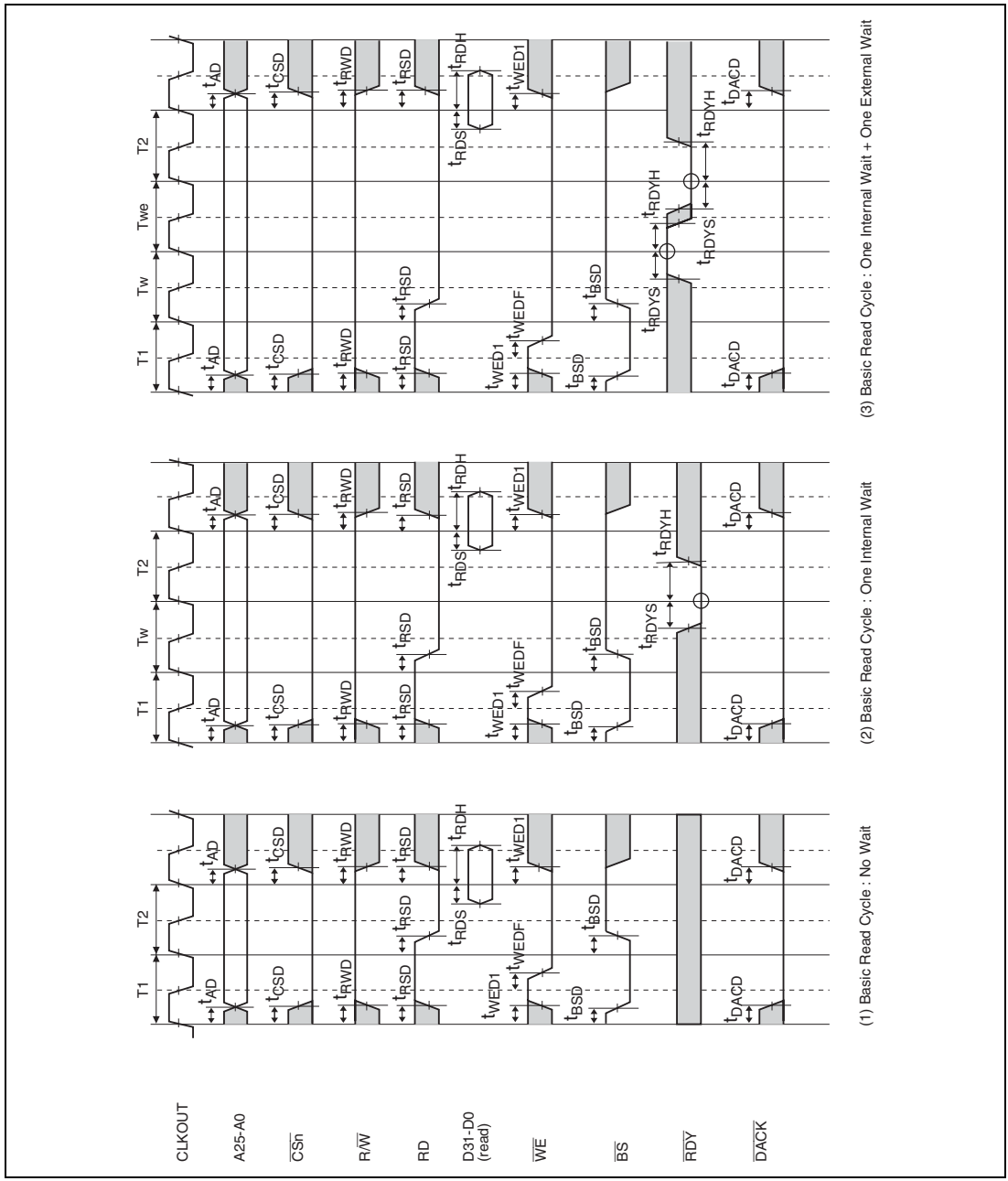


Figure 31.23 Byte Control SRAM Bus Cycle

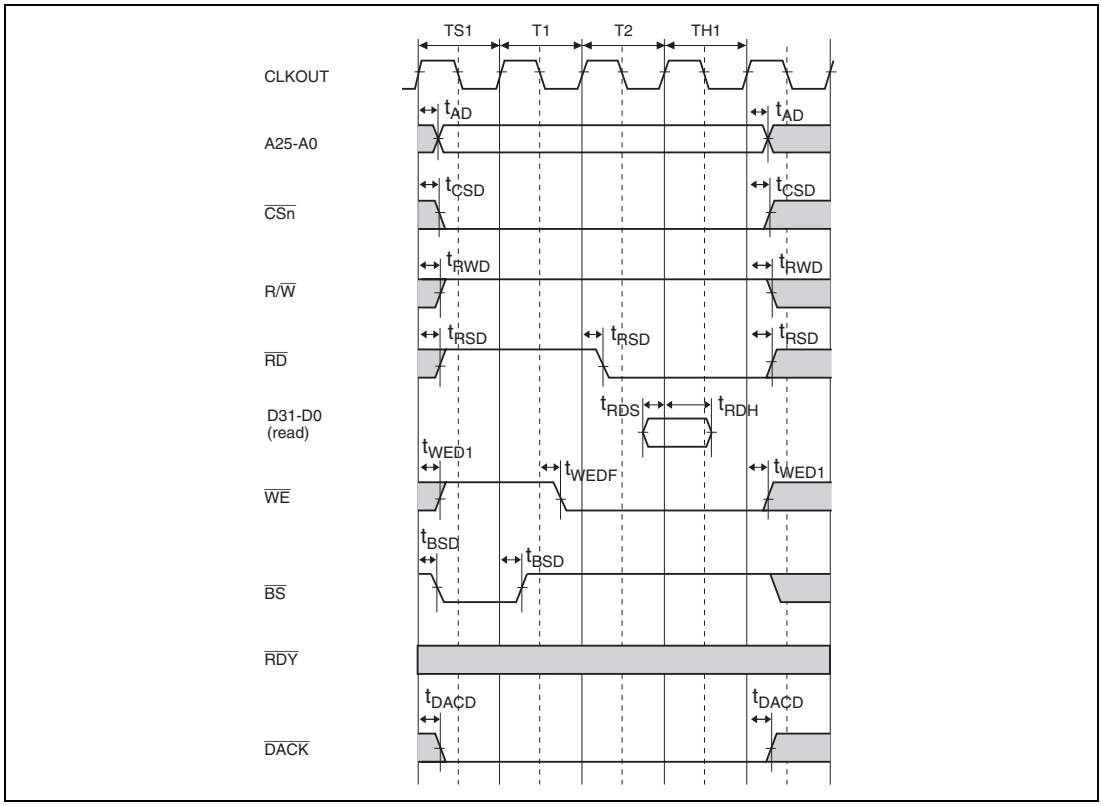


Figure 31.24 Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

31.3.4 DDRIF Signal Timing

Table 31.8 DDRIF Signal Timing

($V_{CCQ-DDR}$ = 2.3 to 2.7V, DDR-V_{REF} = 1.25V, V_{DD} = 1.25V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30pF, R_T = 50 Ω, DLL1/2 on, fast slew rate)

Item	Symbol	Min.	Max.	Unit	Figure	Notes
MCLK output cycle	t_{MCLK}	6.2	12	ns	31.25	DDR320
		7.5	12			DDR266
MCLK output high-level pulse width	t_{MCLKH}	0.45	0.55	t_{MCLK}	31.25	
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	

Item	Symbol	Min.	Max.	Unit	Figure	Notes
MDQS high-level pulse width (write)	t_{WMDQSH}	0.35	—	t_{MCLK}	31.27	
MDQS low-level pulse width (write)	t_{WMDQSL}	0.35	—	t_{MCLK}	31.27	
MDA and MDQM setup time to MDQS rising/ falling edge (write)	t_{WDS}	0.7	—	ns	31.27	DDR320
		0.75	—			DDR266
MDA and MDQM hold time to MDQS rising/ falling edge (write)	t_{WDH}	0.7	—	ns	31.27	DDR320
		0.75	—			DDR266

Note: t_{MCLK} : one MCLK cycle time

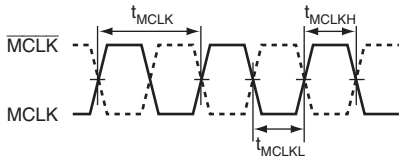


Figure 31.25 MCLK Output Timing

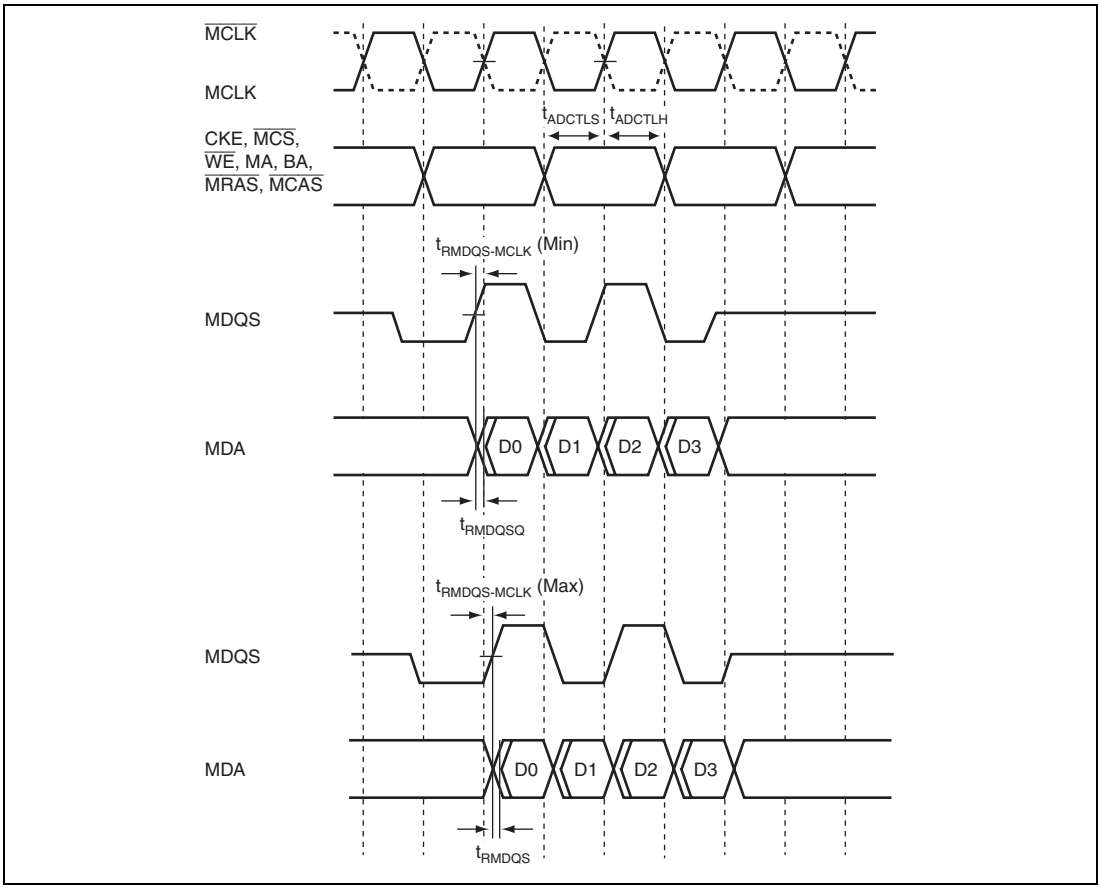


Figure 31.26 Read Timing of DDR-SDRAM (2 Burst Read)

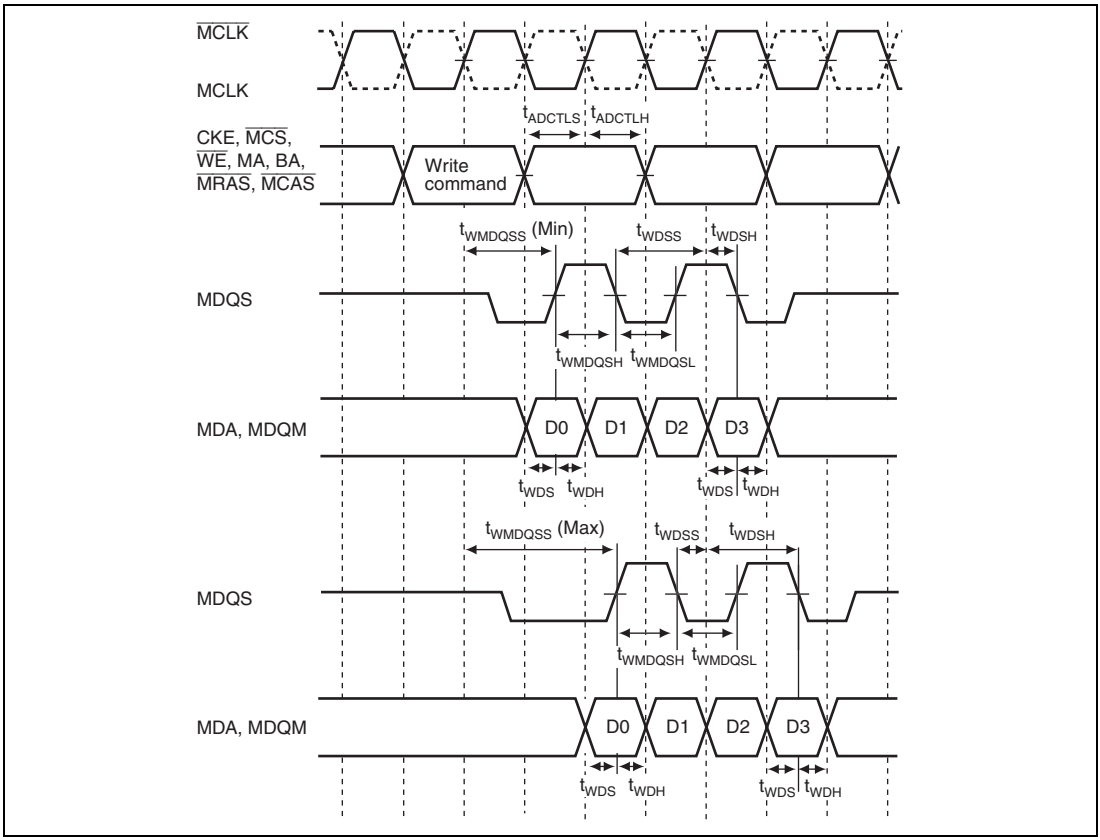


Figure 31.27 Write Timing of DDR-SDRAM (2 Burst Write)

31.3.5 INTC Module Signal Timing

Table 31.9 INTC Module 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	Figure	Notes
NMI pulse width (High)	t_{NMIH}	5	—	t_{cyc}	31.28	normal mode sleep mode
NMI pulse width (Low)	t_{NMIL}	5	—	t_{cyc}	31.28	normal mode sleep mode
IRQ/$\overline{IRL7}$ to IRQ/$\overline{IRL0}$ setup time	t_{IRQS}	3.5	—	ns	31.29	IRQ input
IRQ/$\overline{IRL7}$ to IRQ/$\overline{IRL0}$ hold time	t_{IRQH}	1.5	—	ns	31.29	IRQ input
IRQ/$\overline{IRL7}$ to IRQ/$\overline{IRL0}$ setup time	t_{IRLS}	3.5	—	ns	31.29	IRL input
IRQ/$\overline{IRL7}$ to IRQ/$\overline{IRL0}$ hold time	t_{IRLH}	1.5	—	ns	31.29	IRL input
GPIO interrupt setup time (Port E6-E0, H1, H0, J0, K5, K4)	t_{GPIOS}	3.5	—	ns	31.29	GPIO interrupt input
GPIO interrupt hold time (Port E6-E0, H1, H0, J0, K5, K4)	t_{GPIOH}	1.5	—	ns	31.29	GPIO interrupt input
$\overline{IRQOUT}$ output delay time	t_{IRQOD}	1.5	6	ns	31.29	IRQOUT output

Note: t_{cyc} : one CLKOUT cycle time

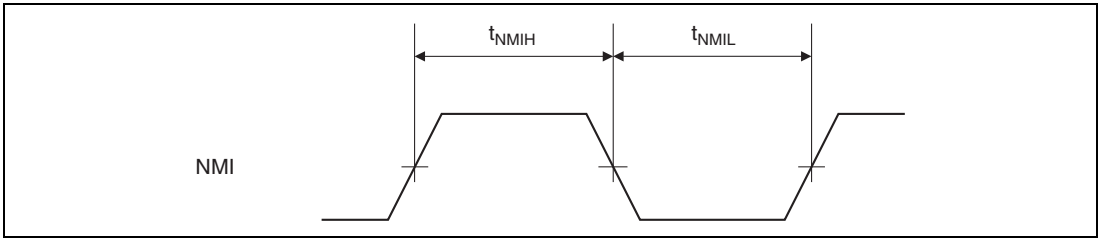


Figure 31.28 NMI Input Timing

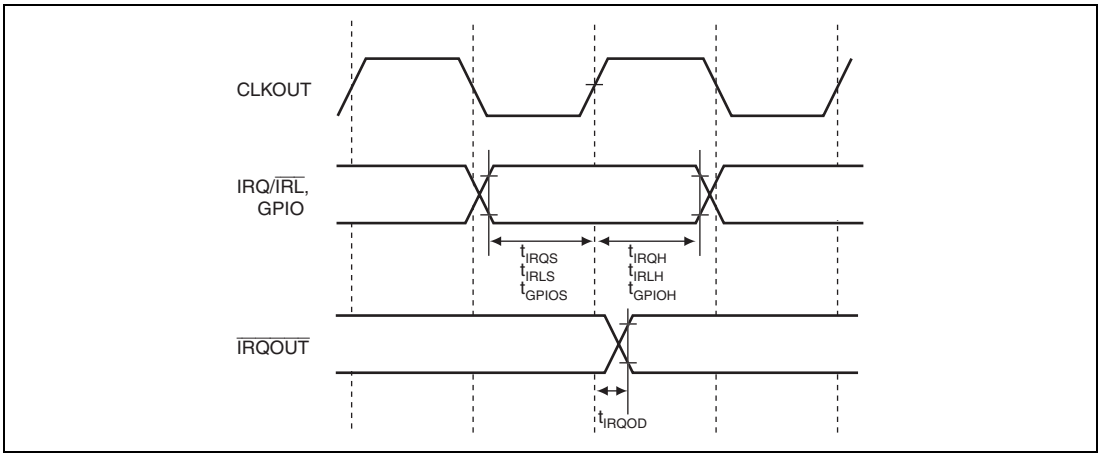


Figure 31.29 IRQ/IRL, GPIO Interrupt Input and IRQOUT Output Timing

31.3.6 PCIC Module Signal Timing

Table 31.10 PCIC Signal Timing (in PCIREQ/PCIGNT Non-Port Mode) (1)

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.25 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF)

			33 MHz		66 MHz			
Pin	**Item**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
PCICLK	Clock cycle	t_{PCICYC}	30	—	15	30	ns	31.30
	Clock pulse width (high)	$t_{PCIHIGH}$	11	—	6	—	ns	31.30
	Clock pulse width (low)	t_{PCILOW}	11	—	6	—	ns	31.30
	Clock rise time	t_{PCIr}	—	4	—	1.5	ns	31.30
	Clock fall time	t_{PCIf}	—	4	—	1.5	ns	31.30
IDSEL	Input setup time	t_{PCISU}	3	—	3	—	ns	31.32
	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns	31.32
AD31–AD0	Output data delay time	t_{PCIVAL}	2	10	2	6	ns	31.31
CBE3–CBE0	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns	31.31
PAR	Tri-state high-impedance delay time	t_{PCIOFF}	2	12	2	6	ns	31.31
$\overline{\text{PCIFRAME}}$								
$\overline{\text{IRDY}}$	Input setup time	t_{PCISU}	3	—	3	—	ns	31.32
$\overline{\text{TRDY}}$	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns	31.32
$\overline{\text{STOP}}$								
$\overline{\text{LOCK}}$								
$\overline{\text{DEVSEL}}$								
$\overline{\text{PERR}}$								
$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$	Output data delay time	t_{PCIVAL}	2	10	2	6	ns	31.31
	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns	31.31
$\overline{\text{REQ3}}$ – $\overline{\text{REQ1}}$	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	—	6	ns	31.31
$\overline{\text{GNT0}}$/ $\overline{\text{GNTIN}}$								
$\overline{\text{GNT3}}$ – $\overline{\text{GNT1}}$	Input setup time	t_{PCISU}	3	—	3	—	ns	31.32
	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns	31.32
$\overline{\text{SERR}}$	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns	31.31
$\overline{\text{INTA}}$ – $\overline{\text{INTD}}$	Tri-state high-impedance delay time	t_{PCIOFF}	2	12	2	6	ns	31.31
	Input setup time	t_{PCISU}	3	—	3	—	ns	31.32
	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns	31.32

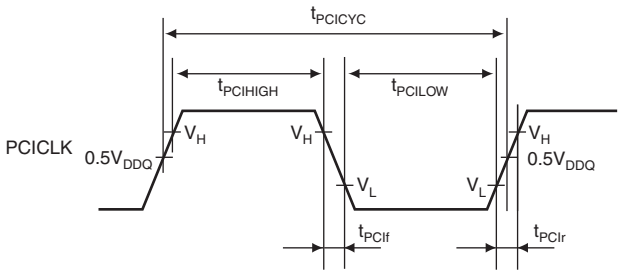


Figure 31.30 PCI Clock Input Timing

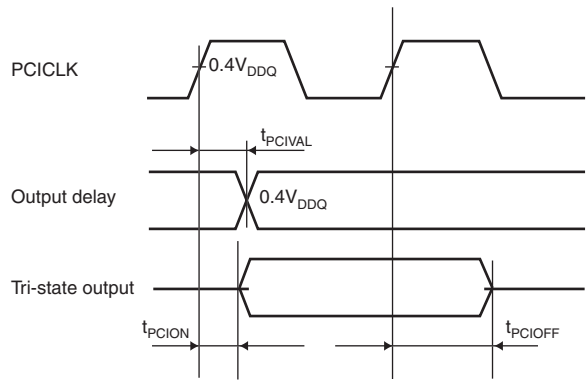


Figure 31.31 Output Signal Timing

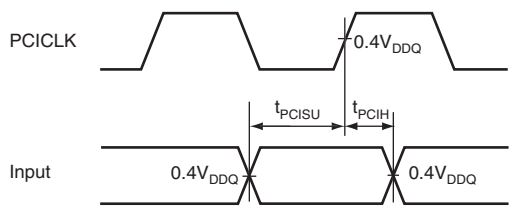


Figure 31.32 Input Signal Timing

31.3.7 DMAC Module Signal Timing

Table 31.11 DMAC Module 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	Figure	Notes
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	2.5	—	ns	31.33	
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	1.5	—	ns	31.33	
$\overline{\text{DRAK}}$ delay time	t_{DRAKD}	1.5	5.3	ns	31.33	
$\overline{\text{DACK}}$ delay time	t_{DACD}	1.5	6	ns	31.8 etc.	

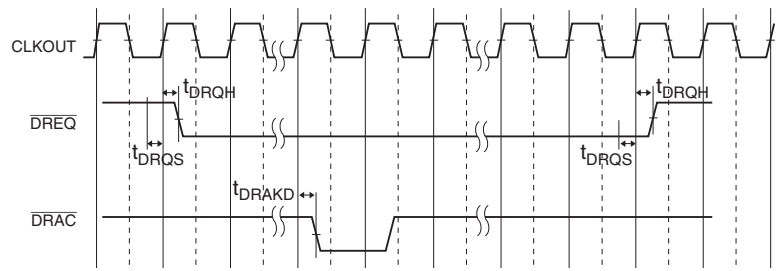


Figure 31.33 $\overline{\text{DREQ}}$ and $\overline{\text{DRAK}}$ Timing

31.3.8 TMU Module Signal Timing

Table 31.12 TMU Module 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	Figure	Notes
Timer clock pulse width (High)	t_{TCLKWH}	4	—	t_{Pcyc}	31.34	
Timer clock pulse width (Low)	t_{TCLKWL}	4	—	t_{Pcyc}	31.34	
Timer clock rise time	t_{TCLKr}	—	0.8	t_{Pcyc}	31.34	
Timer clock fall time	t_{TCLKf}	—	0.8	t_{Pcyc}	31.34	

Note: t_{Pcyc} : one Pck cycle tim

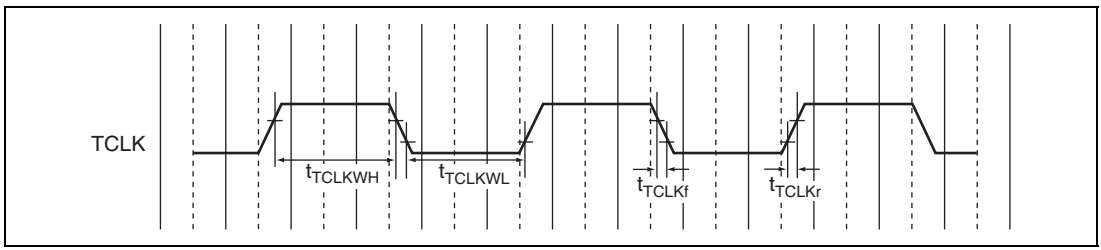


Figure 31.34 TCLK Input Timing

31.3.9 CMT Module Signal Timing

Table 31.13 CMT Module 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	Figure
CMT_CTR output delay time	t_{TMD}	—	8	ns	31.35
CMT_CTR input setup time	t_{TMS}	5	—	ns	31.35
CMT_CTR input hold time	t_{TMH}	5	—	ns	31.35
Timer clock low level width	t_{TMLOW}	1.5	—	t_{cyc}	31.36
Timer clock high level width	t_{TMHIGH}	1.5	—	t_{cyc}	31.36

Note: t_{cyc} : one CLKOUT cycle time

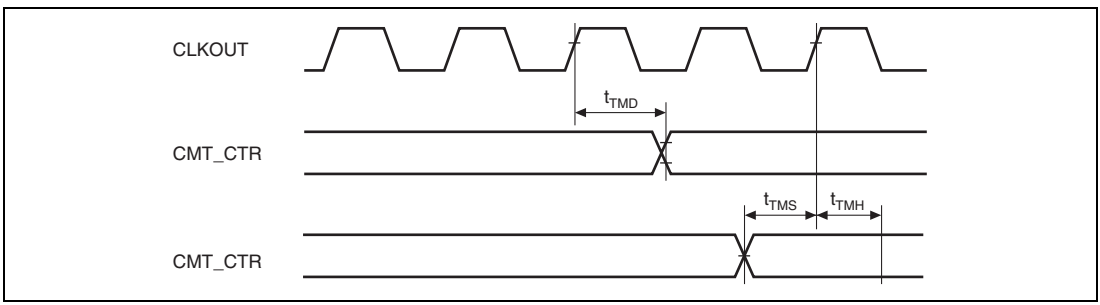


Figure 31.35 CMT Timing (1)

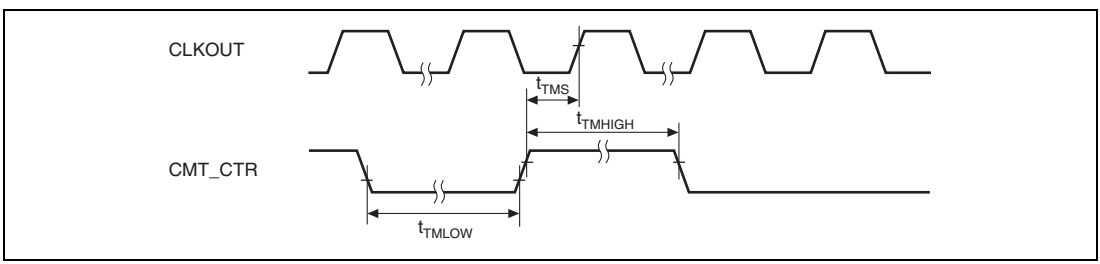


Figure 31.36 CMT Timing (2)

31.3.10 SCIF Module Signal Timing

Table 31.14 SCIF Module 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	Figure	Notes
Input clock cycle (asynchronous)	t_{Scyc}	4	—	t_{Pcyc}	31.37	
Input clock cycle (synchronous)		6	—	t_{Pcyc}	31.37	
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	31.37	
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

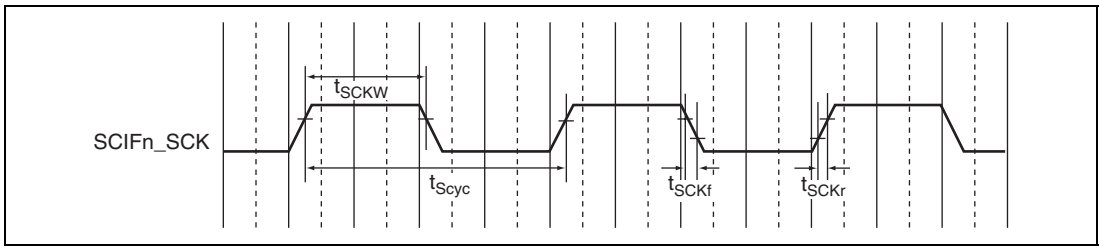


Figure 31.37 SCIFn_SCK Input Clock Timing (n = 0, 1)

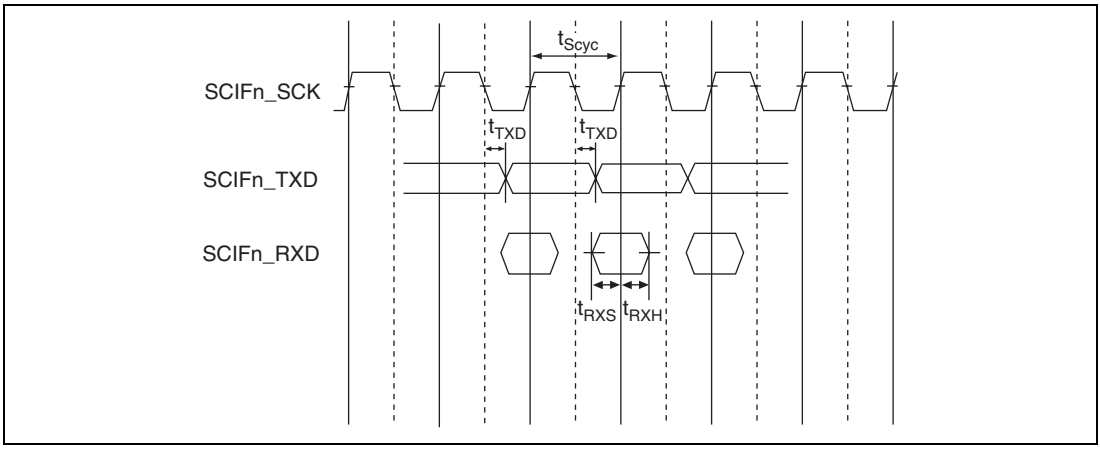


Figure 31.38 SCIF Channel n I/O Synchronous Mode Clock Timing (n = 0, 1)

31.3.11 SIOF Module Signal Timing

Table 31.15 SIOF Module 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	Figure
SIOF_MCLK clock input cycle time	t_{MCYC}	t_{pcyc}*	—	ns	31.39
SIOF_MCLK input high level width	t_{MWH}	0.4 × t_{MCYC}	—	ns	31.39
SIOF_MCLK input low level width	t_{MWL}	0.4 × t_{MCYC}	—	ns	31.39
SIOF_SCK clock cycle time	t_{SICYC}	t_{pcyc}*	—	ns	31.40 to 31.44
SIOF_SCK output high level width	t_{SWHO}	0.4 × t_{SICYC}	—	ns	31.40 to 31.43
SIOF_SCK output low level width	t_{SWLO}	0.4 × t_{SICYC}	—	ns	31.40 to 31.43
SIOF_SYNC output delay time	t_{FSD}	—	10	ns	31.40 to 31.43
SIOF_SCK input high level width	t_{SWHI}	0.4 × t_{SICYC}	—	ns	31.44
SIOF_SCK input low level width	t_{SWLI}	0.4 × 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 to 31.44
SIOF_RXD input setup time	t_{SRDS}	10	—	ns	31.40 to 31.44
SIOF_RXD input hold time	t_{SRDH}	10	—	ns	31.40 to 31.44

Note: * t_{pcyc} is a cycle time of a peripheral clock (Pck).

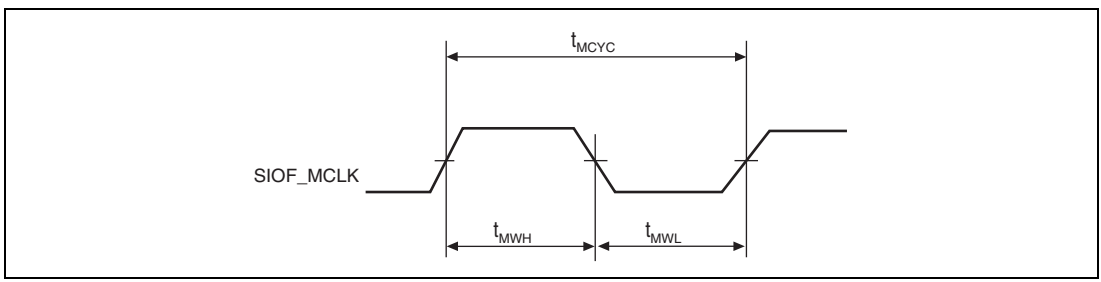


Figure 31.39 SIOF_MCLK Input Timing

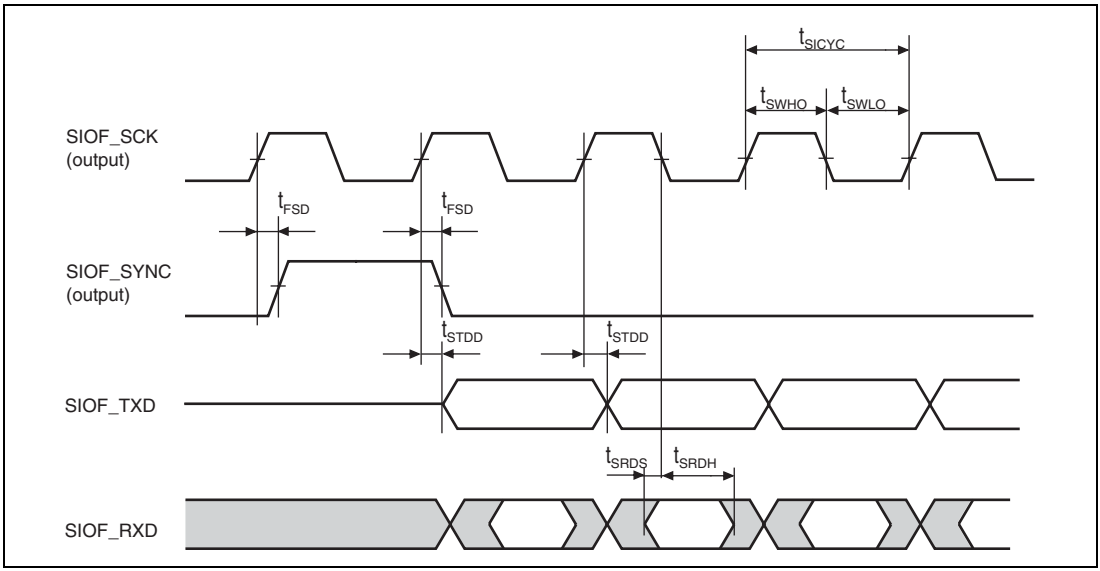


Figure 31.40 SIOF Transmission/Reception Timing (Master Mode 1, Fall Sampling)

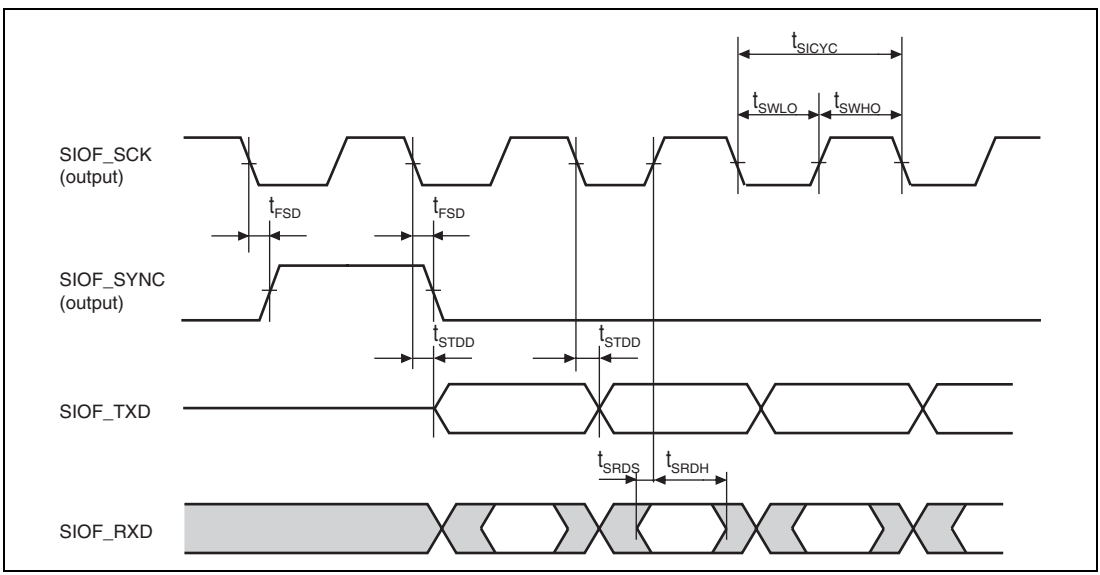


Figure 31.41 SIOF Transmission/Reception Timing (Master Mode 1, Rise Sampling)

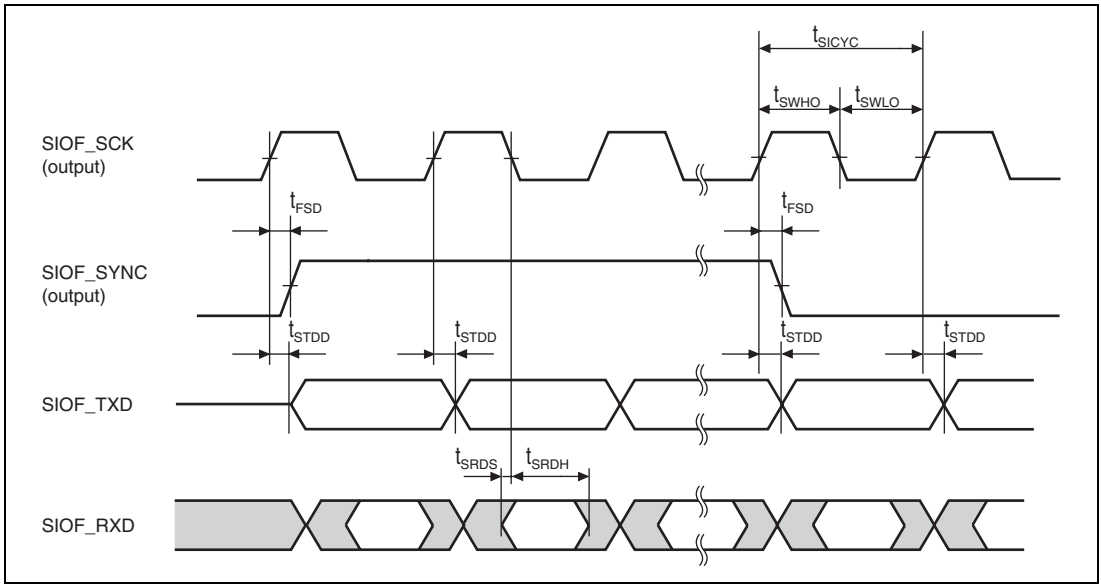


Figure 31.42 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sampling)

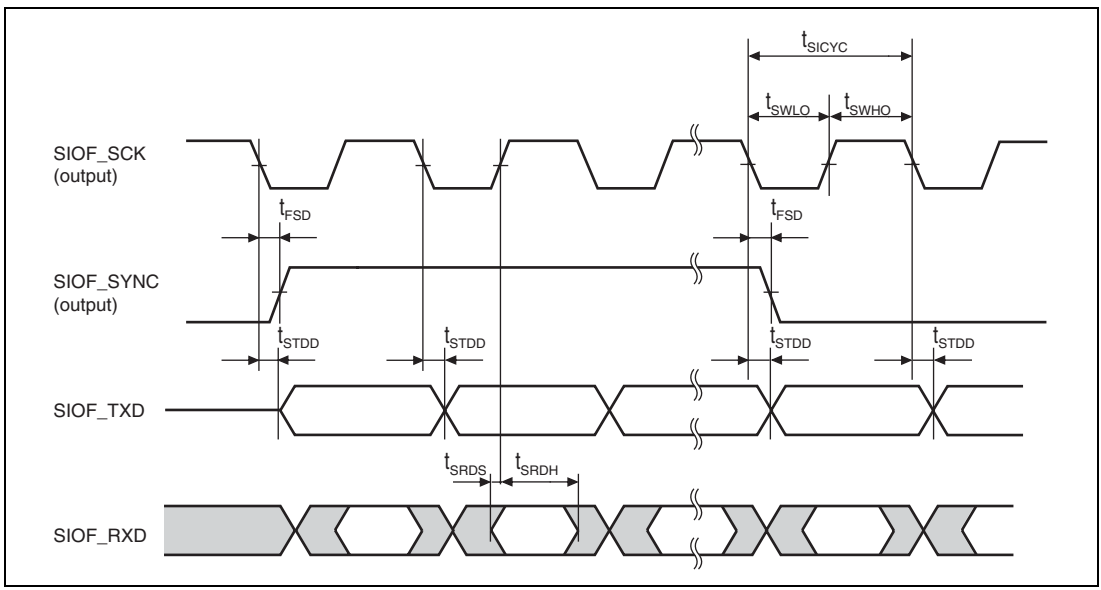


Figure 31.43 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sampling)

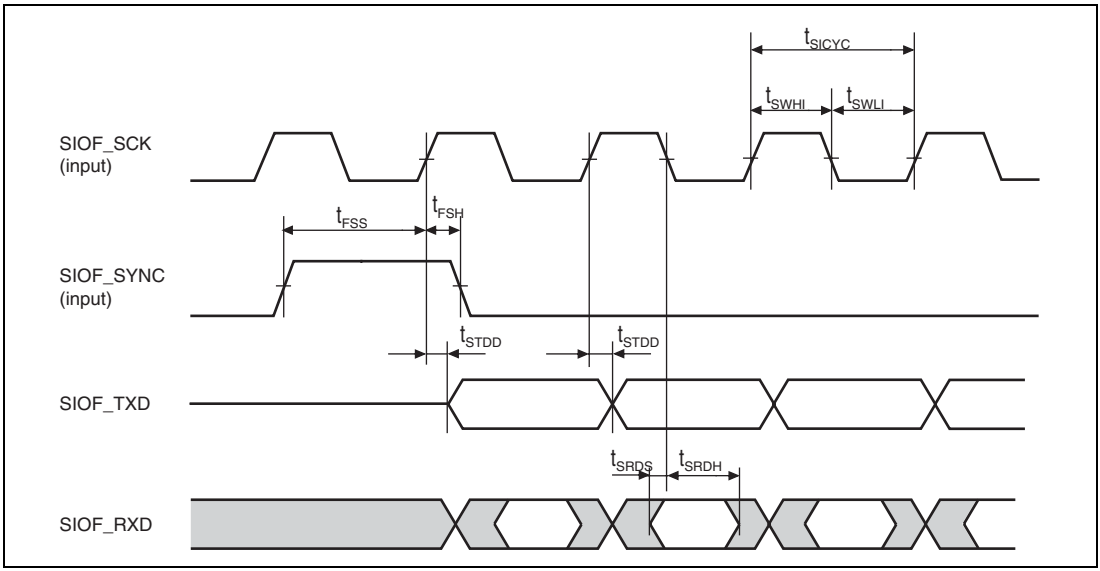


Figure 31.44 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2)

31.3.12 HSPI Module Signal Timing

Table 31.16 HSPI Module 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	Figure
HSPI_CLK frequency	t_{SPIcyc}	—	Pck/8	Hz	31.45
HSPI clock high level width	t_{SPIHW}	60	—	ns	31.45
HSPI clock low level width	t_{SPILW}	60	—	ns	31.45
HSPI_TX setup time (master mode)	$t_{SUSPITX}$	20	—	ns	31.45
HSPI_TX delay time (master mode)	t_{DSPITX}	—	20	ns	31.45
HSPI_TX setup time (slave mode)	$t_{SUSPITX}$	10	—	ns	31.45
HSPI_TX delay time (slave mode)	t_{DSPITX}	—	80	ns	31.45
HSPI_RX setup time	$t_{SUSPIRX}$	20	—	ns	31.45
HSPI_RX hold time	$t_{HLSPIRX}$	20	—	ns	31.45
HSPI_CS lead time	t_{CSLEAD}	100	—	ns	31.45

Note: Pck : Peripheral clock frequency

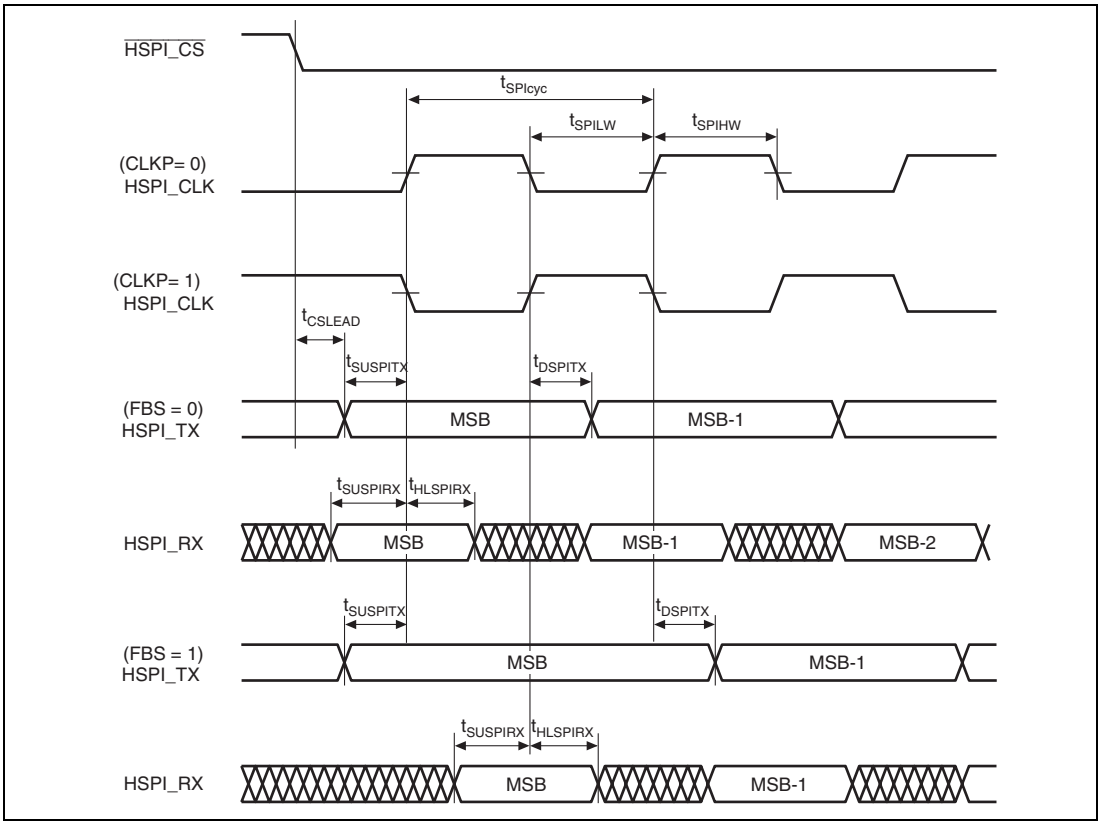


Figure 31.45 HSPI Data Output/Input Timing

31.3.13 MMCIF Module Signal Timing

Table 31.17 MMCIF Module 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	Figure
MCCLK clock cycle time	t_{MMcyc}	50	—	ns	31.46
MCCLK clock high level width	t_{MMWH}	0.4 × t_{MMcyc}	—	ns	31.46
MCCLK clock low level width	t_{MMWL}	0.4 × t_{MMcyc}	—	ns	31.46
MCCMD output data delay time	t_{MMTCD}	—	10	ns	31.46
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

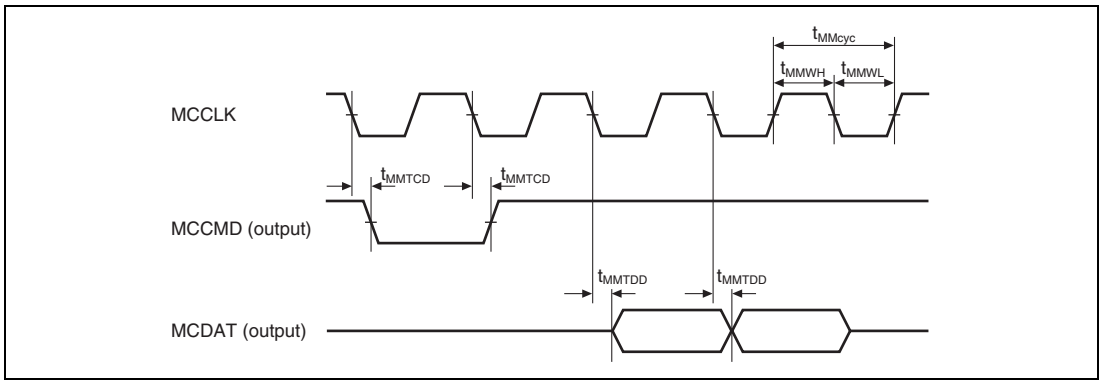


Figure 31.46 MMCIF Transmit Timing

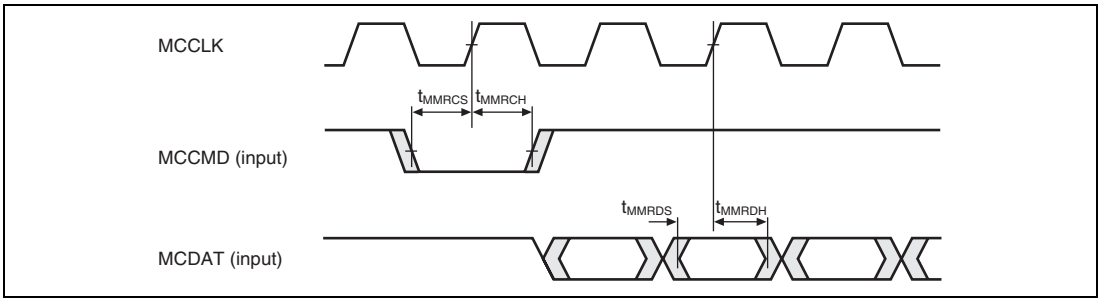


Figure 31.47 MMCIF Receive Timing

31.3.14 HAC Interface Module Signal Timing

Table 31.18 HAC Interface Module 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	Figure
$\overline{\text{HAC\_RES}}$ active low pulse width	t_{RST_LOW}	1000	—	ns	31.48
HAC_SYNC active high pulse width	t_{SYN_HIGH}	1000	—	ns	31.49
HAC_SYNC delay time 1	t_{SYNCD1}	0	15	ns	31.51
HAC_SYNC delay time 2	t_{SYNCD2}	0	15	ns	31.51
HAC_SD_OUT delay time	t_{SDOUTD}	0	15	ns	31.51
HAC_SD_IN setup time	t_{SDINS}	10	—	ns	31.51
HAC_SD_IN hold time	t_{SDINH}	10	—	ns	31.51
HAC_BIT_CLK input high level width	t_{ICL_HIGH}	t_{Pcyc}/2	—	ns	31.50
HAC_BIT_CLK input low level width	t_{ICL_LOW}	t_{Pcyc}/2	—	ns	31.50

Note: t_{Pcyc} : one Pck cycle time

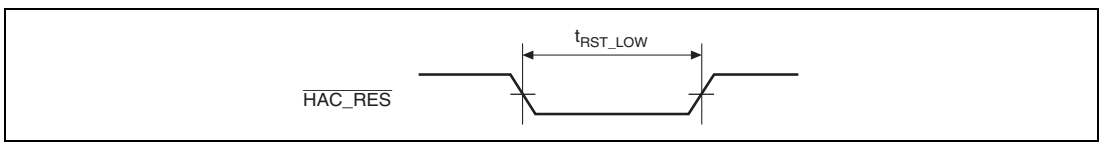


Figure 31.48 HAC Cold Reset Timing

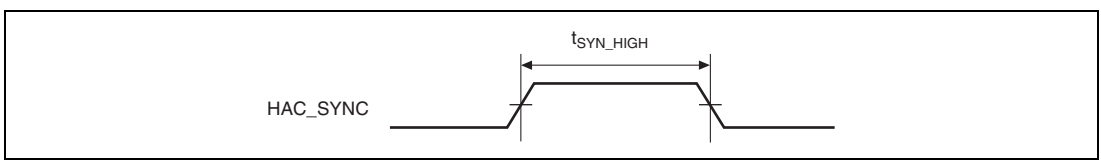


Figure 31.49 HAC SYNC Output Timing

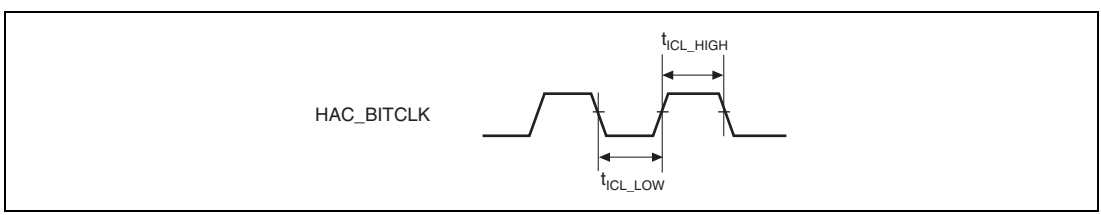


Figure 31.50 HAC Clock Input Timing

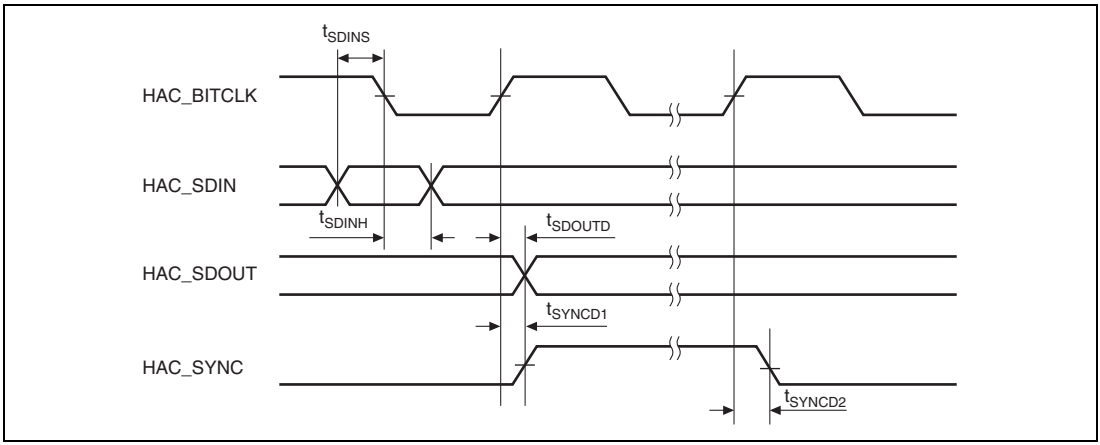


Figure 31.51 HAC Interface Module Signal Timing

31.3.15 SSI Interface Module Signal Timing

Table 31.19 SSI Interface Module 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	Notes	Figure
Output cycle time	t_{OSCK}	40	710	ns	output	31.52
Input cycle time	t_{ISCK}	80	3300	ns	input	31.52
Input high level width/Output high level width	t_{IHC}/t_{OHC}	65	—	ns	input, output	31.52
Input low level width/Output low level width	t_{ILC}/t_{OLC}	65	—	ns	input, output	31.52
SSI_SCK Output rise time	t_{RC}	—	60	ns	output	31.52
SSI_SDATA/WS Output delay time	t_{DTR}	—	10	ns	transmit	31.53, 31.54
SSI_SDATA/WS Input setup time	t_{SR}	10	—	ns	receive	31.55, 31.56
SSI_SDATA/WS Input hold time	t_{HTR}	10	—	ns	receive	31.55, 31.56

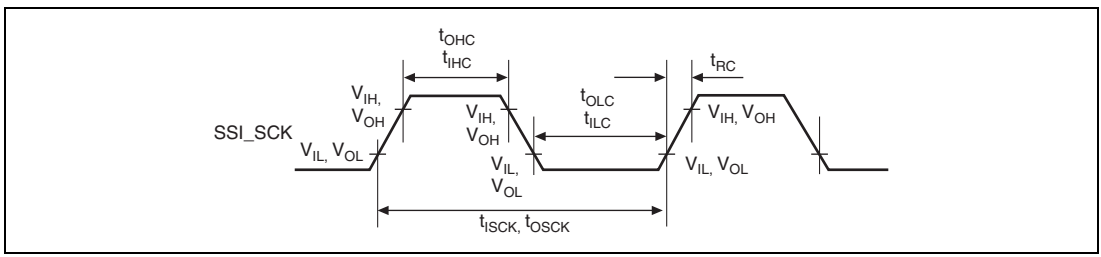


Figure 31.52 SSI Clock Input/Output Timing

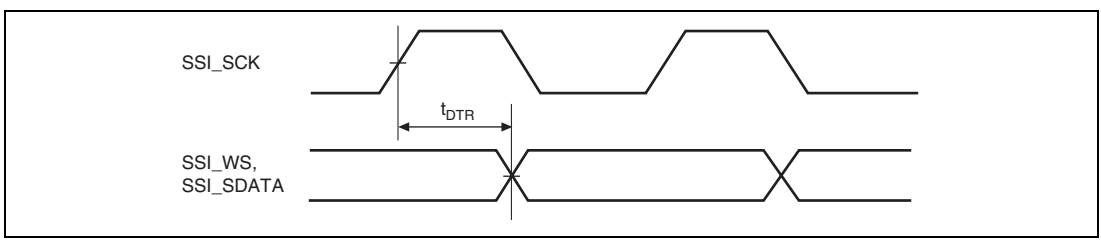


Figure 31.53 SSI Transmit Timing (1)

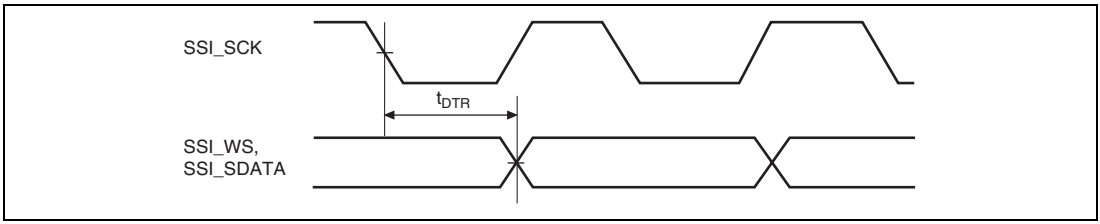


Figure 31.54 SSI Transmit Timing (2)

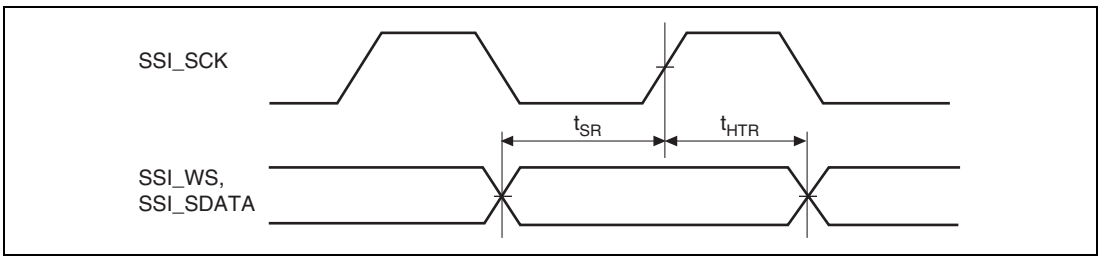


Figure 31.55 SSI Receive Timing (1)

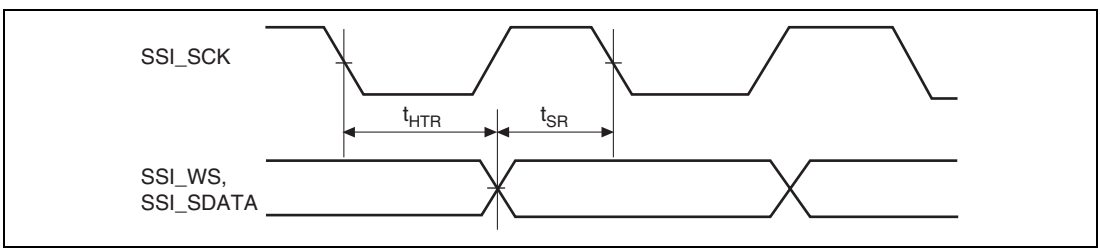


Figure 31.56 SSI Receive Timing (2)

31.3.16 FLCTL Module Signal Timing

Table 31.20 FLCTL Module 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, No wait)

Item	Symbol	Min.	Max.	Unit	Figure
Command output setup time	t_{NCDS}	$2 \times t_{Fcyc} - 10$	—	ns	31.57
Command output hold time	t_{NCHD}	$1.5 \times t_{Fcyc} - 5$	—	ns	
Data output setup time	t_{NDOS}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.57, 31.58
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, 31.58
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, 31.60
$\overline{FWE}$ low pulse width	t_{NWP}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.57, 31.58, 31.60, 31.61
$\overline{FWE}$ high pulse width	t_{NWH}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.58, 31.60
Address to ready/busy transition time	t_{NADRB}	—	$32 \times t_{Pcyc}$	ns	31.58, 31.59
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, 31.61
$\overline{FRE}$ high pulse width	t_{NSPH}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.59
Read data setup time	t_{NRDS}	24	—	ns	31.59, 31.61
Read data hold time	t_{NRDH}	5	—	ns	
Data write setup time	t_{NDWS}	$32 \times t_{Pcyc}$	—	ns	31.59

Item	Symbol	Min.	Max.	Unit	Figure
Command to status read transition time	t_{NCDSR}	$4 \times t_{Fcyc}$	—	ns	31.60
Command output off to status read transition time	t_{NCDFSR}	$3.5 \times t_{Fcyc}$	—	ns	
Status read setup time	t_{NSTS}	$2.5 \times t_{Fcyc}$	—	ns	

Notes: 1. t_{Fcyc}: one FLCTL clock cycle time
2. t_{Pcyc}: one Pck cycle time

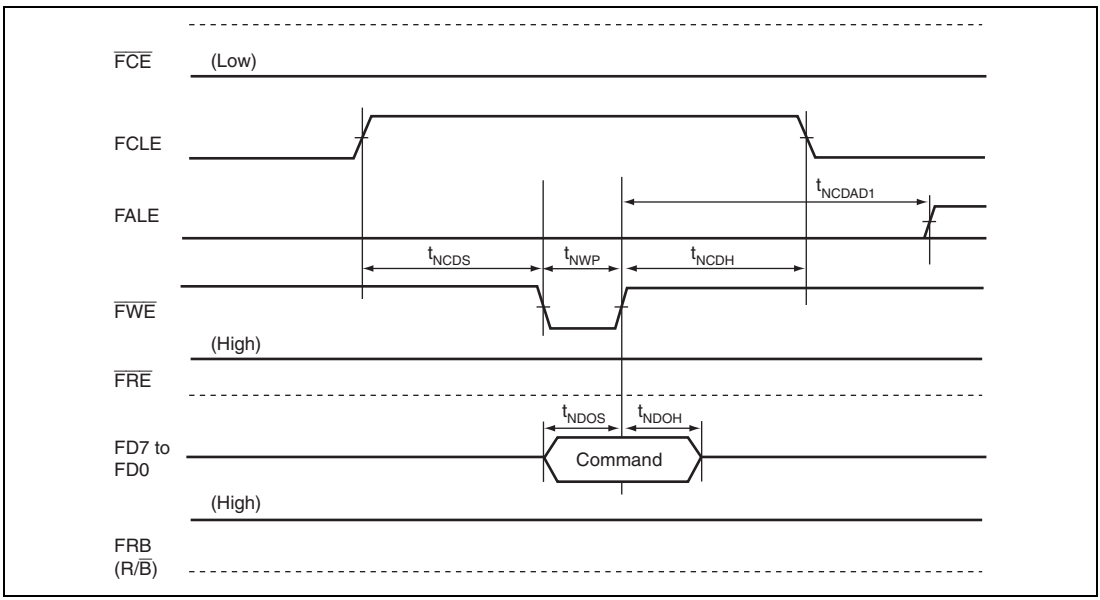


Figure 31.57 Command Issue Timing of NAND-type Flash Memory

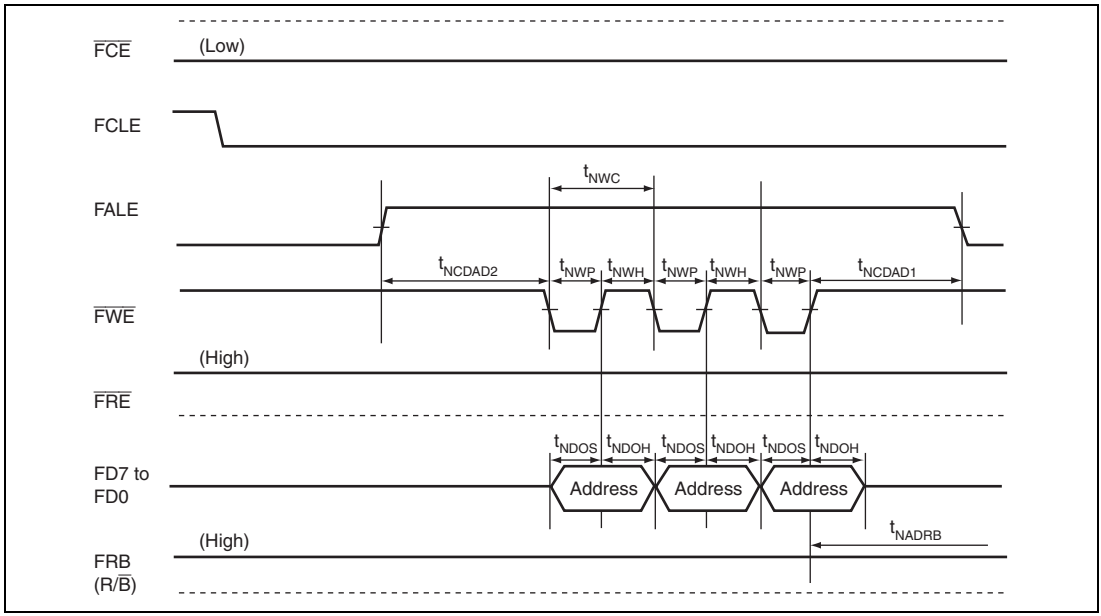


Figure 31.58 Address Issue Timing of NAND-type Flash Memory

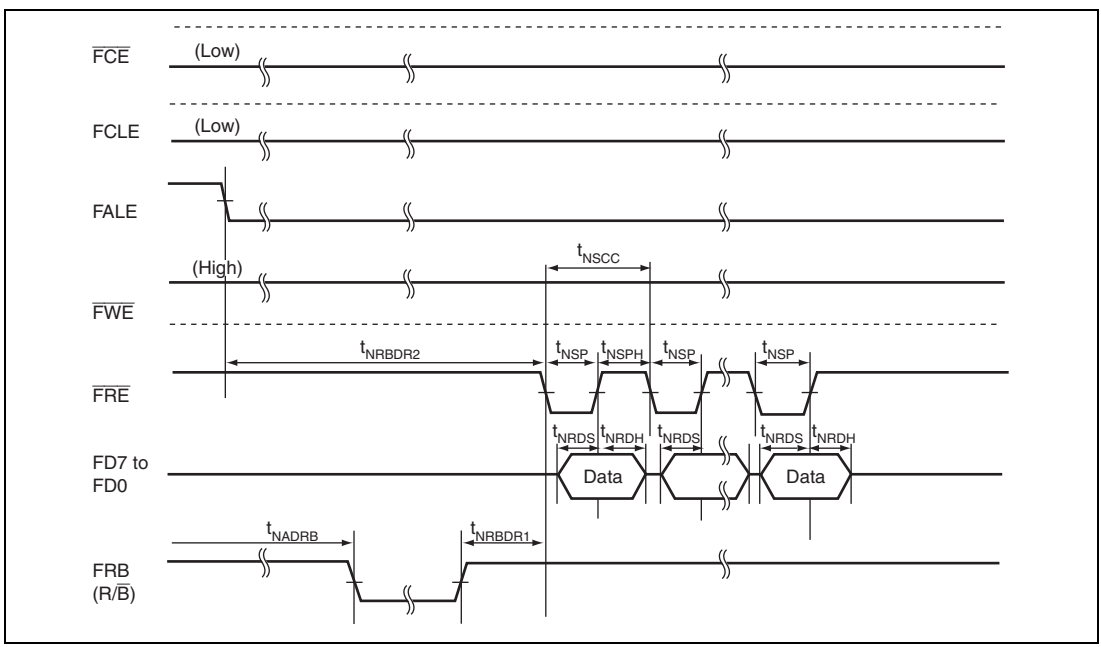


Figure 31.59 Data Read Timing of NAND-type Flash Memory

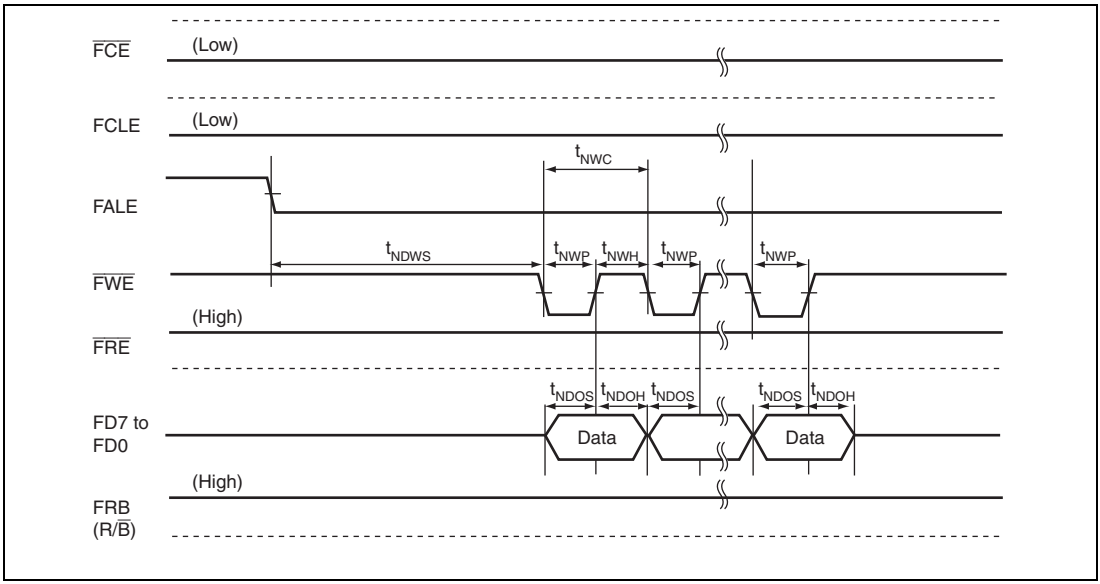


Figure 31.60 Data Write Timing of NAND-type Flash Memory

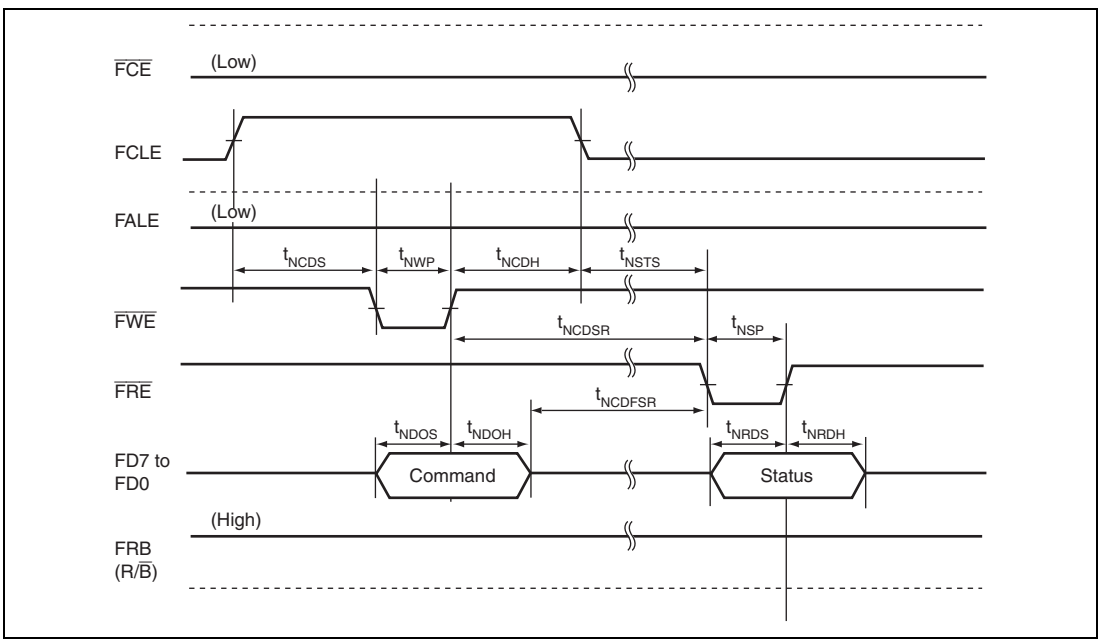


Figure 31.61 Status Read Timing of NAND-type Flash Memory

31.3.17 GPIO Signal Timing

Table 31.21 GPIO 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	Figure
GPIO output delay time	t_{IOPD}	1.5	6	ns	31.62
GPIO input setup time	t_{IOPS}	3.5	—	ns	31.62
GPIO input hold time	t_{IOPH}	1.5	—	ns	31.62

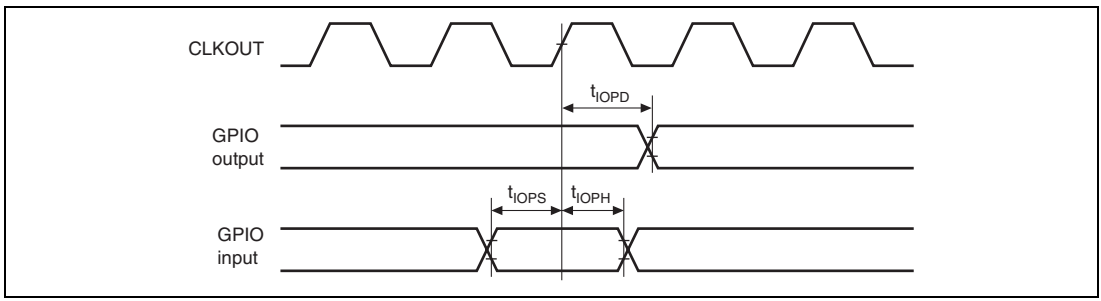


Figure 31.62 GPIO Timing

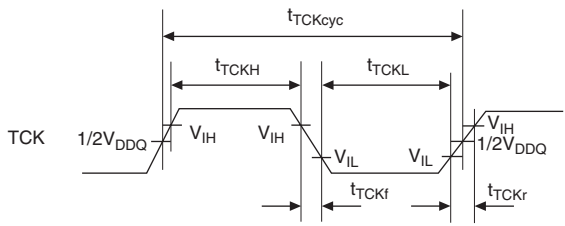
31.3.18 H-UDI Module Signal Timing

Table 31.22 H-UDI Module 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	Figure	Notes
Input clock cycle	t_{TCKcyc}	50	—	ns	31.63, 31.65	
Input clock pulse width (High)	t_{TCKH}	15	—	ns	31.63	
Input clock pulse width (Low)	t_{TCKL}	15	—	ns	31.63	
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

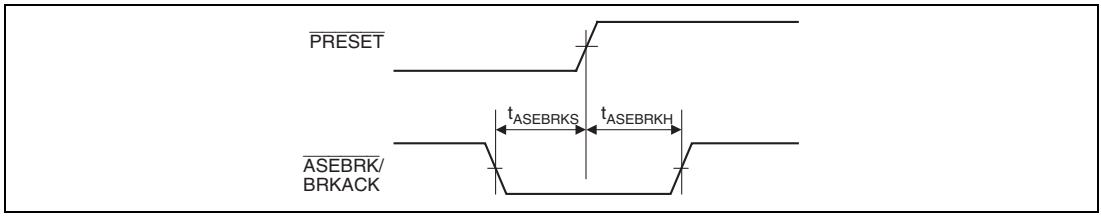


Figure 31.64 PRESET Hold Timing

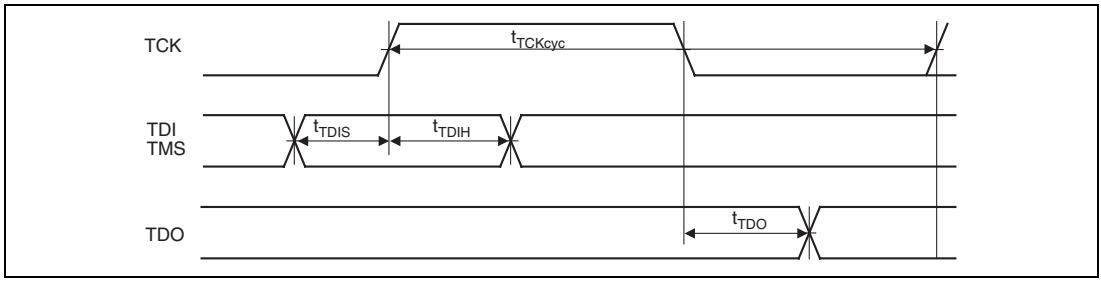


Figure 31.65 H-UDI Data Transfer Timing

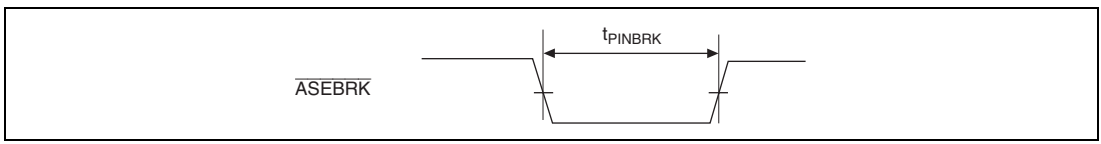


Figure 31.66 ASEBRK Pin Break Timing

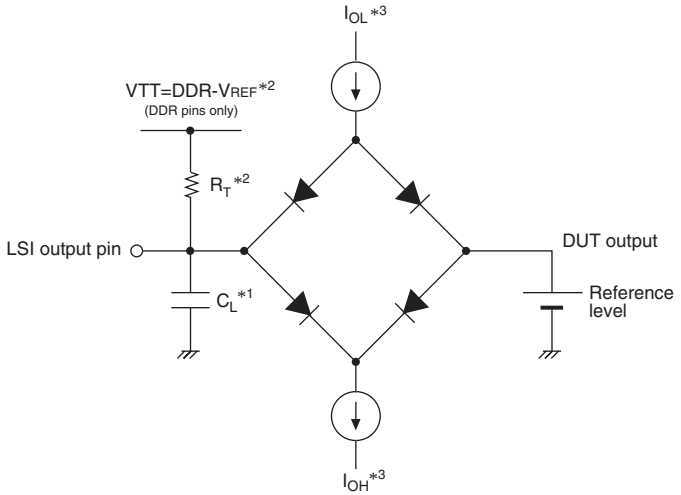
31.4 AC Characteristic Test Conditions

The AC characteristic test conditions are as follows :

- Input/output signal reference level: V*/2
- Input pulse level: V_{SSQ} to V*
- Input rise/fall time: 1 ns

Note: V*: V_{DDQ}, $V_{CCQ-DDR}$ (V_{DDQ} = 3.0 to 3.6V, $V_{CCQ-DDR}$ = 2.3 to 2.7V)

The output load circuit is shown in figure 31.67



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

31.5 Change in Delay Time Based on Load Capacitance

Figure 31.68 is a chart showing the changes in the delay time (reference data) when a load capacitance equal to or larger than the stipulated value (30 pF) is connected to the LSI pins. When connecting an external device with a load capacitance exceeding the regulation, use the chart in figure 31.68 as reference for system design.

Note that if the load capacitance to be connected exceeds the range shown in figure 31.68 the graph will not be a straight line.

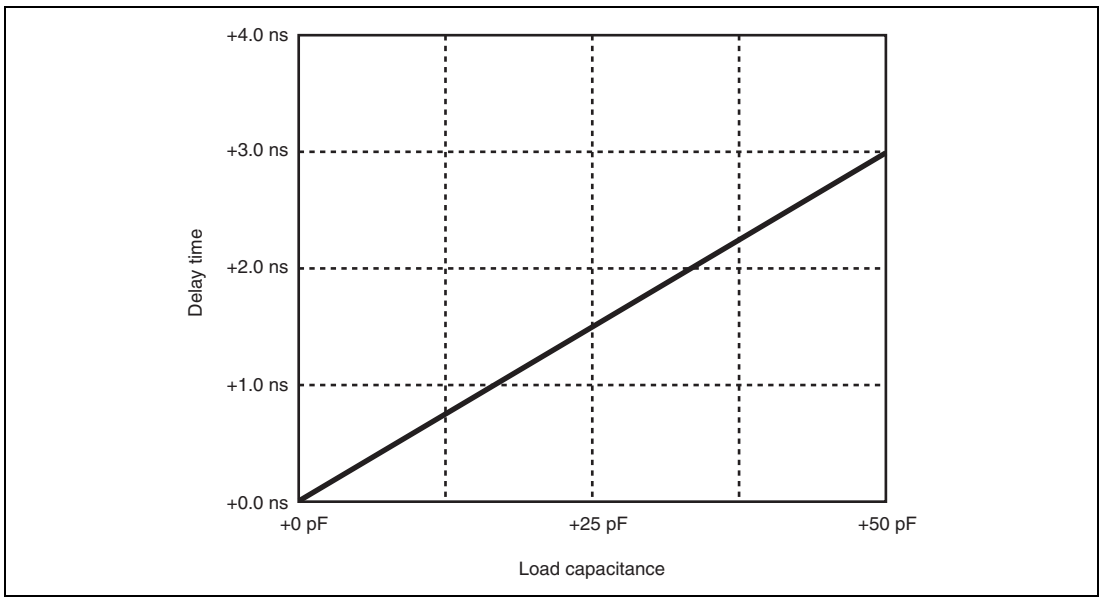


Figure 31.68 Load Capacitance-Delay Time

Appendix

A. CPU Operation Mode Register (CPUOPM)

The CPUOPM is used to control the CPU operation mode. This register can be read from or written to the address H'FF2F 0000 in P4 area or H'1F2F 0000 in area 7 as 32-bit size. The write value to the reserved bits should be the initial value. The operation is not guaranteed if the write value is not the initial value.

The CPUOPM register should be updated by the CPU store instruction not the access from SuperHyway bus master except CPU.

After the CPUOPM is updated, read CPUOPM once, and execute one of the following two 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 updated CPUOPM value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	H'000000F	R	Reserved The write value must be the initial value.
5	RABD	1	R/W	Speculative execution bit for subroutine return 0: Instruction fetch for subroutine return is issued speculatively. When this bit is set to 0, refer to Appendix C, Speculative Execution for Subroutine Return. 1: Instruction fetch for subroutine return is not issued speculatively.
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 is accepted. 1: SR.IMASK is changed to the accepted interrupt level.
2 to 0	—	All 0	R	Reserved The write value must be the initial value.

B. Instruction Prefetching and Its Side Effects

This LSI is provided with an internal buffer for holding pre-read instructions, and always performs pre-reading. Therefore, program code must not be located in the last 64-byte area of any memory space. If program code is located in these areas, a bus access for instruction prefetch may occur exceeding the memory areas boundary. A case in which this is a problem is shown below.

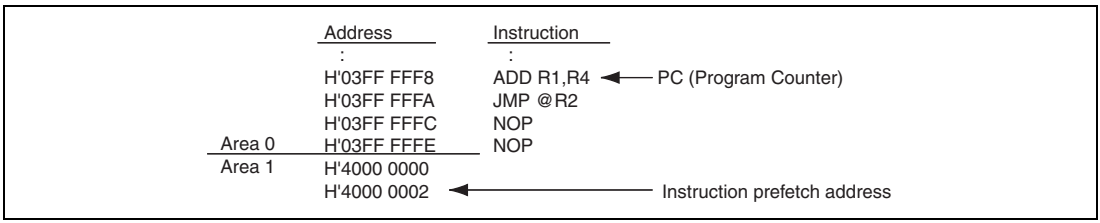


Figure B.1 Instruction Prefetch

Figure B.1 presupposes a case in which the instruction (ADD) indicated by the program counter (PC) and the address H'04000002 instruction prefetch are executed simultaneously. It is also assumed that the program branches to an area other than area 1 after executing the following JMP instruction and delay slot instruction.

In this case, a bus access (instruction prefetch) to area 1 may unintentionally occur from the programming flow.

Instruction Prefetch Side Effects:

1. It is possible that an external bus access caused by an instruction prefetch may result in 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 prefetch, hang-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 area.

C. Speculative Execution for Subroutine Return

The SH-4A has the mechanism to issue an instruction fetch speculatively when returning from subroutine. By issueing an instruction fetch speculatively, the execution cycles to return from subroutine may be shortened.

This function is enabled by setting 0 to the bit 5 (RABD) of CPU Operation Mode register (CPUOPM). But this speculative instruction fetch may issue the access to the address that should not be accessed from the program. Therefore a bus access to an unexpected area or an internal instruction address error may cause a problem. As for the effect of this bus access to unexpected memory area, refer to Appendix B, Instruction Prefetch Side Effects:.

Usage Condition: When the speculative execution for subroutine return is enabled, the RTS instruction should be used to return to the address set in PR by the JSR, BSR or BSRF instructions. It can prevent the access to unexpected address and avoid the problem.

D. Register Address Map

The address map gives information on the on-chip I/O registers. The below listed address is the big endian byte order. When registers consist of 16 or 32 bits, the addresses of the MSBs are given. Access size indicates the number of bits.

Note: Access to reserved addresses and access with under access size are prohibited. Since operation or continued operation is not guaranteed, do not attempt such access.

Legend: The initial value “x” means undefined or depends on the setting of the external pins. For details, refer to the each module section.

H-UDI (H'FC00 0000-H'FC7F FFFF; 8M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC00 0000 to H'FC10 FFFF	Reserved (1,114,112 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC11 0000	Instruction register	SDIR	H'0EFF	R	16	H-UDI
H'FC11 0018	Interrupt source register	SDINT	H'0000	R/W	16	H-UDI
H'FC11 001A to H'FC7F FFFF	Reserved (7,274,470 bytes)	—	—	—	—	—

DMAC (H'FC80 0000-H'FCFF FFFF; 8M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC80 0000 to H'FC80 7FFF	Reserved (32,768 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC80 8000 to H'FC80 801F	Reserved (32 bytes)	—	—	—	—	—
H'FC80 8020	DMA source address register 0	SAR0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8024	DMA destination address register 0	DAR0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8028	DMA transfer count register 0	TCR0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 802C	DMA channel control register 0	CHCR0	H'4000 0000	R/W	32	DMAC
H'FC80 8030	DMA source address register 1	SAR1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8034	DMA destination address register 1	DAR1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8038	DMA transfer count register 1	TCR1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 803C	DMA channel control register 1	CHCR1	H'4000 0000	R/W	32	DMAC
H'FC80 8040	DMA source address register 2	SAR2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8044	DMA destination address register 2	DAR2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8048	DMA transfer count register 2	TCR2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 804C	DMA channel control register 2	CHCR2	H'4000 0000	R/W	32	DMAC
H'FC80 8050	DMA source address register 3	SAR3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8054	DMA destination address register 3	DAR3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8058	DMA transfer count register 3	TCR3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 805C	DMA channel control register 3	CHCR3	H'4000 0000	R/W	32	DMAC
H'FC80 8060	DMA operation register0	DMAOR0	H'0000	R/W	16	DMAC
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	DMAC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC80 8074	DMA destination address register 4	DAR4	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8078	DMA transfer count register 4	TCR4	H'xxxx xxxx	R/W	32	DMAC
H'FC80 807C	DMA channel control register 4	CHCR4	H'4000 0000	R/W	32	DMAC
H'FC80 8080	DMA source address register 5	SAR5	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8084	DMA destination address register 5	DAR5	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8088	DMA transfer count register 5	TCR5	H'xxxx xxxx	R/W	32	DMAC
H'FC80 808C	DMA channel control register 5	CHCR5	H'4000 0000	R/W	32	DMAC
H'FC80 8090 to H'FC80 80FF	Reserved (112 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC80 8100 to H'FC80 811F	Reserved (32 bytes)	—	—	—	—	—
H'FC80 8120	DMA source address register B 0	SARB0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8124	DMA destination address register B 0	DARB0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8128	DMA transfer count register B 0	TCRB0	H'xxxx xxxx	R/W	32	DMAC
H'FC80 812C	Reserved (4 bytes)	—	—	—	—	—
H'FC80 8130	DMA source address register B 1	SARB1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8134	DMA destination address register B 1	DARB1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8138	DMA transfer count register B 1	TCRB1	H'xxxx xxxx	R/W	32	DMAC
H'FC80 813C	Reserved (4 bytes)	—	—	—	—	—
H'FC80 8140	DMA source address register B 2	SARB2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8144	DMA destination address register B 2	DARB2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8148	DMA transfer count register B 2	TCRB2	H'xxxx xxxx	R/W	32	DMAC
H'FC80 814C	Reserved (4 bytes)	—	—	—	—	—
H'FC80 8150	DMA source address register B 3	SARB3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8154	DMA destination address register B 3	DARB3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 8158	DMA transfer count register B 3	TCRB3	H'xxxx xxxx	R/W	32	DMAC
H'FC80 815C to H'FC80 8FFF	Reserved (3,748 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC80 9000	DMA extended resource selector 0	DMARS0	H'0000	R/W	16	DMAC
H'FC80 9004	DMA extended resource selector 1	DMARS1	H'0000	R/W	16	DMAC
H'FC80 9008	DMA extended resource selector 2	DMARS2	H'0000	R/W	16	DMAC
H'FC80 900A to H'FC80 7FFF	Reserved (61,430 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC81 8000 to H'FC81 801F	Reserved (32 bytes)	—	—	—	—	—
H'FC81 8020	DMA source address register 6	SAR6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8024	DMA destination address register 6	DAR6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8028	DMA transfer count register 6	TCR6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 802C	DMA channel control register 6	CHCR6	H'4000 0000	R/W	32	DMAC
H'FC81 8030	DMA source address register 7	SAR7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8034	DMA destination address register 7	DAR7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8038	DMA transfer count register 7	TCR7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 803C	DMA channel control register 7	CHCR7	H'4000 0000	R/W	32	DMAC
H'FC81 8040	DMA source address register 8	SAR8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8044	DMA destination address register 8	DAR8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8048	DMA transfer count register 8	TCR8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 804C	DMA channel control register 8	CHCR8	H'4000 0000	R/W	32	DMAC
H'FC81 8050	DMA source address register 9	SAR9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8054	DMA destination address register 9	DAR9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8058	DMA transfer count register 9	TCR9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 805C	DMA channel control register 9	CHCR9	H'4000 0000	R/W	32	DMAC
H'FC81 8060	DMA operation register1	DMAOR1	H'0000	R/W	16	DMAC
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	DMAC
H'FC81 8074	DMA destination address register 10	DAR10	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8078	DMA transfer count register 10	TCR10	H'xxxx xxxx	R/W	32	DMAC
H'FC81 807C	DMA channel control register 10	CHCR10	H'4000 0000	R/W	32	DMAC
H'FC81 8080	DMA source address register 11	SAR11	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8084	DMA destination address register 11	DAR11	H'xxxx xxxx	R/W	32	DMAC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC81 8088	DMA transfer count register 11	TCR11	H'xxxx xxxx	R/W	32	DMAC
H'FC81 808C	DMA channel control register 11	CHCR11	H'4000 0000	R/W	32	DMAC
H'FC81 8090 to H'FC81 80FF	Reserved (112 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FC81 8100 to H'FC81 811F	Reserved (32 bytes)	—	—	—	—	—
H'FC81 8120	DMA source address register B 6	SARB6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8124	DMA destination address register B 6	DARB6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8128	DMA transfer count register B 6	TCRB6	H'xxxx xxxx	R/W	32	DMAC
H'FC81 812C	Reserved (4 bytes)	—	—	—	—	—
H'FC81 8130	DMA source address register B 7	SARB7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8134	DMA destination address register B 7	DARB7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8138	DMA transfer count register B 7	TCRB7	H'xxxx xxxx	R/W	32	DMAC
H'FC81 813C	Reserved (4 bytes)	—	—	—	—	—
H'FC81 8140	DMA source address register B 8	SARB8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8144	DMA destination address register B 8	DARB8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8148	DMA transfer count register B 8	TCRB8	H'xxxx xxxx	R/W	32	DMAC
H'FC81 814C	Reserved (4 bytes)	—	—	—	—	—
H'FC81 8150	DMA source address register B 9	SARB9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8154	DMA destination address register B 9	DARB9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 8158	DMA transfer count register B 9	TCRB9	H'xxxx xxxx	R/W	32	DMAC
H'FC81 815C to H'FCFF FFFF	Reserved (8,289,956 bytes)	—	—	—	—	—

PCI Memory (H'FD00 0000-H'FDFF FFFF; 16M bytes External Memory Area)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FD00 0000 to H'FDFF FFFF	PCI memory space 0 (16,777,216 bytes)	—	—	—	—	—

PCIC (H'FE00 0000-H'FE3F FFFF; 4M bytes)

'R/W' indicates read/write access from the SuperHyway bus.

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE00 0000 to H'FE00 0007	Reserved (8 bytes)	—	—	—	—	—
H'FE00 0008	PCI enable control register	PCIECR	H'0000 0000	R/W	32	PCIC
H'FE00 000C to H'FE03 FFFF	Reserved (262,132 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE04 0000	PCI device ID register	PCIDID	H'0002	R	16	PCIC
H'FE04 0002	PCI vendor ID register	PCIVID	H'1912	R	16	PCIC
H'FE04 0004	PCI status register	PCISTATUS	H'0290	R/W	16	PCIC
H'FE04 0006	PCI command register	PCICMD	H'0080	R/W	16	PCIC
H'FE04 0008	PCI base class code register	PCIBCC	H'00	R/W	8	PCIC
H'FE04 0009	PCI sub class code register	PCISUB	H'00	R/W	8	PCIC
H'FE04 000A	PCI program interface register	PCIPIF	H'00	R/W	8	PCIC
H'FE04 000B	PCI revision ID register	PCIRID	H'00	R	8	PCIC
H'FE04 000C	PCI BIST register	PCIBIST	H'00	R	8	PCIC
H'FE04 000D	PCI header type register	PCIHDR	H'00	R	8	PCIC
H'FE04 000E	PCI latency timer register	PCILTM	H'00	R/W	8	PCIC
H'FE04 000F	PCI cacheline size register	PCICLS	H'20	R	8	PCIC
H'FE04 0010	PCI I/O base address register	PCIIBAR	H'0000 0001	R/W	32	PCIC
H'FE04 0014	PCI Memory base address register 0	PCIMBAR0	H'0000 0000	R/W	32	PCIC
H'FE04 0018	PCI Memory base address register 1	PCIMBAR1	H'0000 0000	R/W	32	PCIC
H'FE04 001C to H'FE04 002B	Reserved (16 bytes)	—	—	—	—	—
H'FE04 002C	PCI subsystem ID register	PCISID	H'0000	R/W	16	PCIC
H'FE04 002E	PCI subsystem vendor ID register	PCISVID	H'0000	R/W	16	PCIC
H'FE04 0030 to H'FE04 0036	Reserved (7 bytes) register	—	—	—	—	—
H'FE04 0037	PCI capabilities pointer register	PCICP	H'40	R	8	PCIC
H'FE04 0038	Reserved (4 bytes)	—	—	—	—	—
H'FE04 003C	PCI maximum latency register	PCIMAXLAT	H'00	R	8	PCIC
H'FE04 003D	PCI minimum grant register	PCIMINGNT	H'00	R	8	PCIC
H'FE04 003E	PCI interrupt pin register	PCIINTPIN	H'01	R/W	8	PCIC
H'FE04 003F	PCI interrupt line register	PCIINTLINE	H'00	R/W	8	PCIC
H'FE04 0040	PCI power management capability register	PCIPMC	H'000A	R/W	16	PCIC
H'FE04 0042	PCI next item pointer register	PCINIP	H'00	R	8	PCIC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE04 0043	PCI capability ID register	PCICID	H'01	R	8	PCIC
H'FE04 0044	PCI power consumption/dissipation data register	PCIPCDD	H'00	R/W	8	PCIC
H'FE04 0045	PCI PMCSR bridge support extension register	PCIPMCSRBSE	H'00	R	8	PCIC
H'FE04 0046	PCI power management control/status register	PCIPMCSR	H'0000	R/W	16	PCIC
H'FE04 0048 to H'FE04 00FF	Reserved (184 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE04 0100	PCI control register	PCICR	H'0000 00xx	R/W	32	PCIC
H'FE04 0104	PCI local space register 0	PCILSR0	H'0000 0000	R/W	32	PCIC
H'FE04 0108	PCI local space register 1	PCILSR1	H'0000 0000	R/W	32	PCIC
H'FE04 010C	PCI local address register 0	PCILAR0	H'0000 0000	R/W	32	PCIC
H'FE04 0110	PCI local address register 1	PCILAR1	H'0000 0000	R/W	32	PCIC
H'FE04 0114	PCI interrupt register	PCIIR	H'0000 0000	R/W	32	PCIC
H'FE04 0118	PCI interrupt mask register	PCIIMR	H'0000 0000	R/W	32	PCIC
H'FE04 011C	PCI error address information register	PCIAIR	H'xxxx xxxx	R	32	PCIC
H'FE04 0120	PCI error command information register	PCICIR	H'xx00 000x	R	32	PCIC
H'FE04 0124 to H'FE04 012F	Reserved (12 bytes)	—	—	—	—	—
H'FE04 0130	PCI arbiter interrupt register	PCIAINT	H'0000 0000	R/W	32	PCIC
H'FE04 0134	PCI arbiter interrupt mask register	PCIAINTM	H'0000 0000	R/W	32	PCIC
H'FE04 0138	PCI arbiter bus master information register	PCIBMIR	H'0000 00xx	R	32	PCIC
H'FE04 013C to H'FE04 01BF	Reserved (132 bytes)	—	—	—	—	—
H'FE04 01C0	PCI PIO address register	PCIPAR	H'80xx xxxx	R/W	32	PCIC
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	PCIC
H'FE04 01D0	PCI power management interrupt mask register	PCIPINTM	H'0000 0000	R/W	32	PCIC
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	PCIC
H'FE04 01E4	PCI memory bank mask register 0	PCIMBMR0	H'0000 0000	R/W	32	PCIC
H'FE04 01E8	PCI memory bank register 1	PCIMBR1	H'0000 0000	R/W	32	PCIC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE04 01EC	PCI memory bank mask register 1	PCIMBMR1	H'0000 0000	R/W	32	PCIC
H'FE04 01F0	PCI memory bank register 2	PCIMBR2	H'0000 0000	R/W	32	PCIC
H'FE04 01F4	PCI memory bank mask register 2	PCIMBMR2	H'0000 0000	R/W	32	PCIC
H'FE04 01F8	PCI I/O bank register	PCIIOBR	H'0000 0000	R/W	32	PCIC
H'FE04 01FC	PCI I/O bank master register	PCIIOBMR	H'0000 0000	R/W	32	PCIC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
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	PCIC
H'FE04 0214	PCI cache snoop control register 1	PCICSCR1	H'0000 0000	R/W	32	PCIC
H'FE04 0218	PCI cache snoop address register 0	PCICSAR0	H'0000 0000	R/W	32	PCIC
H'FE04 021C	PCI cache snoop address register 1	PCICSAR1	H'0000 0000	R/W	32	PCIC
H'FE04 0220	PCI PIO data register	PCIPDR	H'xxxx xxxx	R/W	32	PCIC
H'FE04 022C to H'FE04 03FF	Reserved (468 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE04 0400 to H'FE3F FFFF	Reserved (3,931,136 bytes)	—	—	—	—	—

SuperHyway RAM (H'FE40 0000-H'FE7F FFFF; 4M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE40 0000 to H'FE40 FFFF	Reserved (65,536 bytes)	—	—	—	—	—
H'FE41 0000 to H'FE41 3FFF	SuperHyway RAM 0 (16,384 bytes)	—	Undefined	R/W	*	SuperHyway RAM
H'FE41 4000 to H'FE41 FFFF	Reserved (49,152 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE42 0000 to H'FE42 3FFF	SuperHyway RAM 1 (16,384 bytes)	—	Undefined	R/W	*	SuperHyway RAM
H'FE42 4000 to H'FE7F FFFF	Reserved (4,046,848 bytes)	—	—	—	—	—

Note: * 8-/16-/32-/64-bit and 16-/32-byte

DDRIF (H'FE80 0000-H'FEFF FFFF; 8M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE80 0000 to H'FE80 0007	Reserved (8 bytes)	—	—	—	—	—
H'FE80 0008	Memory interface mode register	MIM (1)	H'0000 0000	R/W	32	DDRIF
H'FE80 000C	Memory interface mode register	MIM (2)	H'0C34 x100	R/W	32	DDRIF
H'FE80 0010	DDR-SDRAM control register	SCR (1)	H'0000 0000	R/W	32	DDRIF
H'FE80 0014	DDR-SDRAM control register	SCR (2)	H'0000 0000	R/W	32	DDRIF
H'FE80 0018	DDR-SDRAM timing register	STR (1)	H'0000 0000	R/W	32	DDRIF
H'FE80 001C	DDR-SDRAM timing register	STR (2)	H'0000 0000	R/W	32	DDRIF
H'FE80 0030	DDR-SDRAM row attribute register	SDR (1)	H'0000 0000	R/W	32	DDRIF
H'FE80 0034	DDR-SDRAM row attribute register	SDR (2)	H'0000 0000	R/W	32	DDRIF
H'FE80 0038 to H'FE80 03FF	Reserved (968 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FE80 0400	DDR-SDRAM back-up register	DBK (1)	H'0000 0000	R	32	DDRIF
H'FE80 0408	DDR-SDRAM back-up register	DBK (2)	H'0000 000x	R	32	DDRIF
H'FE80 040C to H'FEBF FFFF	Reserved (4,193,268 bytes)	—	—	—	—	—
H'FECx xxxx*	DDR-SDRAM mode register	SDMR	—	W	32	DDRIF

Note: * The DDR-SDRAM mode register is placed in the DDR-SDRAM. The setting value is written to the DDR-SDRAM register by accessing this address. For details, refer to section 12, DDR-SDRAM Interface (DDRIF).

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FEC8 0000 to H'FEFF FFFF	Reserved (3,670,016 bytes)	—	—	—	—	—

CPU and L RAM (H'FF00 0000-H'FF3F FFFF; 4M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF00 0000	Page table entry high register	PTEH	H'xxxx xxxx	R/W	32	MMU
H'FF00 0004	Page table entry low register	PTEL	H'xxxx xxxx	R/W	32	MMU
H'FF00 0008	Translation table base register	TTB	H'xxxx xxxx	R/W	32	MMU
H'FF00 000C	TLB exception address register	TEA	H'xxxx xxxx	R/W	32	MMU
H'FF00 0010	MMU control register	MMUCR	H'0000 0000	R/W	32	MMU
H'FF00 0014 to H'FF00 001B	Reserved (8 bytes)	—	—	—	—	—
H'FF00 001C	Cache control register	CCR	H'0000 0000	R/W	32	Cache
H'FF00 0020	TRAPA exception register	TRA	H'xxxx xxxx	R/W	32	Exception Handling
H'FF00 0024	Exception event register	EXPEVT	H'0000 0000	R/W	32	Exception Handling
H'FF00 0028	Interrupt event register	INTEVT	H'xxxx xxxx	R/W	32	Exception Handling
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	Cache
H'FF00 003C	Queue address control register 1	QACR1	H'0000 00xx	R/W	32	Cache
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	L RAM

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF00 0054	L memory transfer source address register 1	LSA1	H'xxxx xxxx	R/W	32	L RAM
H'FF00 0058	L memory transfer destination address register 0	LDA0	H'xxxx xxxx	R/W	32	L RAM
H'FF00 005C	L memory transfer destination address register 1	LDA1	H'xxxx xxxx	R/W	32	L RAM
H'FF00 0060 to H'FF00 006F	Reserved (16 bytes)	—	—	—	—	—
H'FF00 0070	Physical address space control register	PASCR	H'0000 0000	R/W	32	MMU
H'FF00 0074	On-chip memory control register	RAMCR	H'0000 0000	R/W	32	Cache/ L RAM
H'FF00 0078	Instruction re-fetch inhibit control register	IRMCR	H'0000 0000	R/W	32	MMU
H'FF00 007C to H'FF1F FFFF	Reserved (2,097,028 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF20 0000	Match condition setting register 0	CBR0	H'2000 0000	R/W	32	UBC
H'FF20 0004	Match operation setting register 0	CRR0	H'0000 2000	R/W	32	UBC
H'FF20 0008	Match address setting register 0	CAR0	H'xxxx xxxx	R/W	32	UBC
H'FF20 000C	Match address mask setting register 0	CAMR0	H'xxxx xxxx	R/W	32	UBC
H'FF20 0010 to H'FF20 001F	Reserved (16 bytes)	—	—	—	—	—
H'FF20 0020	Match condition setting register 1	CBR1	H'2000 0000	R/W	32	UBC
H'FF20 0024	Match operation setting register 1	CRR1	H'0000 2000	R/W	32	UBC
H'FF20 0028	Match address setting register 1	CAR1	H'xxxx xxxx	R/W	32	UBC
H'FF20 002C	Match address mask setting register 1	CAMR1	H'xxxx xxxx	R/W	32	UBC
H'FF20 0030	Match data setting register 1	CDR1	H'xxxx xxxx	R/W	32	UBC
H'FF20 0034	Match data mask setting register 1	CDMR1	H'xxxx xxxx	R/W	32	UBC
H'FF20 0038	Execution count break register 1	CETR1	H'xxxx xxxx	R/W	32	UBC
H'FF20 003C to H'FF20 05FF	Reserved (1,476 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF20 0600	Channel match flag register	CCMFR	H'0000 0000	R/W	32	UBC
H'FF20 0604 to H'FF20 061F	Reserved (28 bytes)	—	—	—	—	—
H'FF20 0620	Break control register	CBCR	H'0000 0000	R/W	32	UBC
H'FF20 0624 to H'FF2E FFFF	Reserved (981,468bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF2F 0000	CPU Operation Mode Register	CPUOPM	H'0000 03E0	R/W	32	Exception Handling
H'FF2F 0004 to H'FF3F FFFF	Reserved (1,114,108 bytes)	—	—	—	—	—

SuperHyway Router (H'FF40 0000-H'FF7F FFFF; 4M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
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	LBSC

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
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	Module
H'FF80 0000 to H'FF80 0FFF	Reserved (4,096 bytes)	—	—	—	—	—
H'FF80 1000	Bus Control Register	BCR	H'0000 0000	R/W	32	LBSC
H'FF80 1004 to H'FF80 1FFF	Reserved (4,092 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FF80 2000	CS0 Bus Control Register	CS0BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2004	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2008	CS0 Wait Control Register	CS0WCR	H'7777 770F	R/W	32	LBSC
H'FF80 200C	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2010	CS1 Bus Control Register	CS1BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2014	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2018	CS1 Wait Control Register	CS1WCR	H'7777 770F	R/W	32	LBSC
H'FF80 201C	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2020	CS2 Bus Control Register	CS2BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2024	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2028	CS2 Wait Control Register	CS2WCR	H'7777 770F	R/W	32	LBSC
H'FF80 202C to H'FF80 203F	Reserved (20 bytes)	—	—	—	—	—
H'FF80 2040	CS4 Bus Control Register	CS4BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2044	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2048	CS4 Wait Control Register	CS4WCR	H'7777 770F	R/W	32	LBSC
H'FF80 204C	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2050	CS5 Bus Control Register	CS5BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2054	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2058	CS5 Wait Control Register	CS5WCR	H'7777 770F	R/W	32	LBSC
H'FF80 205C	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2060	CS6 Bus Control Register	CS6BCR	H'7777 7770	R/W	32	LBSC
H'FF80 2064	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2068	CS6 Wait Control Register	CS6WCR	H'7777 770F	R/W	32	LBSC
H'FF80 206C	Reserved (4 bytes)	—	—	—	—	—
H'FF80 2070	CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	R/W	32	LBSC
H'FF80 2074 to H'FF80 207F	Reserved (12 bytes)	—	—	—	—	—
H'FF80 2080	CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	R/W	32	LBSC
H'FF80 2084 to H'FFBF FFFF	Reserved (4,185,980 bytes)	—	—	—	—	—

Peripheral Modules (H'FFC0 0000-H'FFFF FFFF; 4M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFC0 0000 to H'FFC7 FFFF	Reserved (524,288 bytes)	—	—	—	—	—
H'FFC8 0000	Frequency control register	FRQCR	H'1xxx x3xx	R/W	32	CPG
H'FFC8 0004 to H'FFC8 0023	Reserved (32 bytes)	—	—	—	—	—
H'FFC8 0024	PLL control register	PLLCR	H'0000 E001	R/W	32	CPG
H'FFC8 0028 to H'FFC8 002F	Reserved (8 bytes)	—	—	—	—	—
H'FFC8 0030	Standby control register	MSTPCR	H'0000 0000	R/W	32	CPG
H'FFC8 0034 to H'FFCB FFFF	Reserved (262,092 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFCC 0000	Watchdog timer stop time register	WDTST	H'0000 0000	R/W	32	WDT
H'FFCC 0004	Watchdog timer control/status register	WDTCSR	H'0000 0000	R/W	32	WDT
H'FFCC 0008	Watchdog timer base stop time register	WDTBST	H'0000 0000	R/W	32	WDT
H'FFCC 000C	Reserved (4 bytes)	—	—	—	—	—
H'FFCC 0010	Watchdog timer counter	WDTCNT	H'0000 0000	R	32	WDT
H'FFCC 0014	Reserved (4 bytes)	—	—	—	—	—
H'FFCC 0018	Watchdog timer base counter	WDTBCNT	H'0000 0000	R	32	WDT
H'FFCC 001C to H'FFCF FFFF	Reserved (262,116 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFD0 0000	Interrupt control register 0	ICR0	H'x000 0000	R/W	32	INTC
H'FFD0 0004 to H'FFD0 000F	Reserved (12 bytes)	—	—	—	—	—
H'FFD0 0010	Interrupt priority register	INTPRI	H'0000 0000	R/W	32	INTC
H'FFD0 0014 to H'FFD0 001B	Reserved (8 bytes)	—	—	—	—	—
H'FFD0 001C	Interrupt control register 1	ICR1	H'0000 0000	R/W	32	INTC
H'FFD0 0020	Reserved (4 bytes)	—	—	—	—	—
H'FFD0 0024	Interrupt source register	INTREQ	H'0000 0000	R/W	32	INTC
H'FFD0 0028 to H'FFD0 0043	Reserved (28 bytes)	—	—	—	—	—
H'FFD0 0044	Interrupt mask register 0	INTMSK0	H'0000 0000	R/W	32	INTC
H'FFD0 0048	Interrupt mask register 1	INTMSK1	H'FF00 0000	R/W	32	INTC
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	INTC
H'FFD0 0068	Interrupt mask clear register 1	INTMSKCLR1	H'0000 0000	R/W	32	INTC
H'FFD0 006C to H'FFD0 00BF	Reserved (84 bytes)	—	—	—	—	—
H'FFD0 00C0	NMI flag control register	NMIFCR	H'x000 0000	R/W	32	INTC
H'FFD0 00C4 to H'FFD2 FFFF	Reserved (196,412 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFD3 0000	User interrupt mask level register	USERIMASK	H'0000 0000	R/W	32	INTC
H'FFD3 0004 to H'FFD3 FFFF	Reserved (65,532 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFD4 0000	Interrupt priority registers 0	INT2PRI0	H'0000 0000	R/W	32	INTC
H'FFD4 0004	Interrupt priority registers 1	INT2PRI1	H'0000 0000	R/W	32	INTC
H'FFD4 0008	Interrupt priority registers 2	INT2PRI2	H'0000 0000	R/W	32	INTC
H'FFD4 000C	Interrupt priority registers 3	INT2PRI3	H'0000 0000	R/W	32	INTC
H'FFD4 0010	Interrupt priority registers 4	INT2PRI4	H'0000 0000	R/W	32	INTC
H'FFD4 0014	Interrupt priority registers 5	INT2PRI5	H'0000 0000	R/W	32	INTC
H'FFD4 0018	Interrupt priority registers 6	INT2PRI6	H'0000 0000	R/W	32	INTC
H'FFD4 001C	Interrupt priority registers 7	INT2PRI7	H'0000 0000	R/W	32	INTC
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	INTC
H'FFD4 0034	Interrupt source register (mask state is affected)	INT2A1	H'0000 0000	R	32	INTC
H'FFD4 0038	Interrupt mask register	INT2MSKRG	H'FFFF FFFF	R/W	32	INTC
H'FFD4 003C	Interrupt mask clear register	INT2MSKCR	H'0000 0000	R/W	32	INTC
H'FFD4 0040	Individual module interrupt source registers 0	INT2B0	H'xxxx xxxx	R	32	INTC
H'FFD4 0044	Individual module interrupt source registers 1	INT2B1	H'xxxx xxxx	R	32	INTC
H'FFD4 0048	Individual module interrupt source registers 2	INT2B2	H'xxxx xxxx	R	32	INTC
H'FFD4 004C	Individual module interrupt source registers 3	INT2B3	H'xxxx xxxx	R	32	INTC
H'FFD4 0050	Individual module interrupt source registers 4	INT2B4	H'xxxx xxxx	R	32	INTC
H'FFD4 0054	Individual module interrupt source registers 5	INT2B5	H'xxxx xxxx	R	32	INTC
H'FFD4 0058	Individual module interrupt source registers 6	INT2B6	H'xxxx xxxx	R	32	INTC
H'FFD4 005C	Individual module interrupt source registers 7	INT2B7	H'xxxx xxxx	R	32	INTC
H'FFD4 0060 to H'FFD4 007F	Reserved (32 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFD4 0080	Interrupt mask register 2	INTMSK2	H'FF00 0000	R/W	32	INTC
H'FFD4 0084	Interrupt mask clear register 2	INTMSKCLR2	H'0000 0000	R/W	32	INTC
H'FFD4 0088 to H'FFD4 008F	Reserved (8 bytes)	—	—	—	—	—
H'FFD4 0090	GPIO interrupt set register	INT2GPIC	H'0000 0000	R/W	32	INTC
H'FFD4 0094 to H'FFD7 FFFF	Reserved (261,996 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFD8 0000	Timer output control register	TOCR	H'00	R/W	8	TMU
H'FFD8 0004	Timer start register 0	TSTR0	H'00	R/W	8	TMU
H'FFD8 0008	Timer constant register 0	TCOR0	H'FFFF FFFF	R/W	32	TMU
H'FFD8 000C	Timer counter 0	TCNT0	H'FFFF FFFF	R/W	32	TMU
H'FFD8 0010	Timer control register 0	TCR0	H'0000	R/W	16	TMU
H'FFD8 0014	Timer constant register 1	TCOR1	H'FFFF FFFF	R/W	32	TMU
H'FFD8 0018	Timer counter 1	TCNT1	H'FFFF FFFF	R/W	32	TMU
H'FFD8 001C	Timer control register 1	TCR1	H'0000	R/W	16	TMU
H'FFD8 0020	Timer constant register 2	TCOR2	H'FFFF FFFF	R/W	32	TMU
H'FFD8 0024	Timer counter 2	TCNT2	H'FFFF FFFF	R/W	32	TMU
H'FFD8 0028	Timer control register 2	TCR2	H'0000	R/W	16	TMU
H'FFD8 002C	Input capture register 2	TCPR2	H'xxxx xxxx	R	32	TMU
H'FFD8 0030 to H'FFDB FFFF	Reserved (262,096 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFDC 0000	Reserved (4 bytes)	—	—	—	—	—
H'FFDC 0004	Timer start register 1	TSTR1	H'00	R/W	8	TMU
H'FFDC 0008	Timer constant register 3	TCOR3	H'FFFF FFFF	R/W	32	TMU
H'FFDC 000C	Timer counter 3	TCNT3	H'FFFF FFFF	R/W	32	TMU
H'FFDC 0010	Timer control register 3	TCR3	H'0000	R/W	16	TMU
H'FFDC 0014	Timer constant register 4	TCOR4	H'FFFF FFFF	R/W	32	TMU
H'FFDC 0018	Timer counter 4	TCNT4	H'FFFF FFFF	R/W	32	TMU
H'FFDC 001C	Timer control register 4	TCR4	H'0000	R/W	16	TMU
H'FFDC 0020	Timer constant register 5	TCOR5	H'FFFF FFFF	R/W	32	TMU
H'FFDC 0024	Timer counter 5	TCNT5	H'FFFF FFFF	R/W	32	TMU
H'FFDC 0028	Timer control register 5	TCR5	H'0000	R/W	16	TMU
H'FFDC 002A to H'FFDF FFFF	Reserved (262,102 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE0 0000	Serial mode register 0	SCSMR0	H'0000	R/W	16	SCIF
H'FFE0 0004	Bit rate register 0	SCBRR0	H'FF	R/W	8	SCIF
H'FFE0 0008	Serial control register 0	SCSCR0	H'0000	R/W	16	SCIF
H'FFE0 000C	Transmit FIFO data register 0	SCFTDR0	H'xx	W	8	SCIF
H'FFE0 0010	Serial status register 0	SCFSR0	H'0060	R/W	16	SCIF
H'FFE0 0014	Receive FIFO data register 0	SCFRDR0	H'xx	R	8	SCIF
H'FFE0 0018	FIFO control register 0	SCFCR0	H'0000	R/W	16	SCIF
H'FFE0 001C	Transmit FIFO data count register 0	SCTFDR0	H'0000	R	16	SCIF
H'FFE0 0020	Receive FIFO data count register 0	SCRFDR0	H'0000	R	16	SCIF
H'FFE0 0024	Serial port register 0	SCSPTR0	H'000x	R/W	16	SCIF
H'FFE0 0028	Line status register 0	SCLSR0	H'0000	R/W	16	SCIF
H'FFE0 002C	Serial error register 0	SCRER0	H'0000	R	16	SCIF
H'FFE0 002E to H'FFE0 FFFF	Reserved (65,490 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE1 0000	Serial mode register 1	SCSMR1	H'0000	R/W	16	SCIF
H'FFE1 0004	Bit rate register 1	SCBRR1	H'FF	R/W	8	SCIF
H'FFE1 0008	Serial control register 1	SCSCR1	H'0000	R/W	16	SCIF
H'FFE1 000C	Transmit FIFO data register 1	SCFTDR1	H'xx	W	8	SCIF
H'FFE1 0010	Serial status register 1	SCFSR1	H'0060	R/W*	16	SCIF
H'FFE1 0014	Receive FIFO data register 1	SCFRDR1	H'xx	R	8	SCIF
H'FFE1 0018	FIFO control register 1	SCFCR1	H'0000	R/W	16	SCIF
H'FFE1 001C	Transmit FIFO data count register 1	SCTFDR1	H'0000	R	16	SCIF
H'FFE1 0020	Receive FIFO data count register 1	SCRFDR1	H'0000	R	16	SCIF
H'FFE1 0024	Serial port register 1	SCSPTR1	H'00xx	R/W	16	SCIF
H'FFE1 0028	Line status register 1	SCLSR1	H'0000	R/W	16	SCIF
H'FFE1 002C	Serial error register 1	SCRER1	H'0000	R	16	SCIF
H'FFE1 002E to H'FFE1 FFFF	Reserved (65,490 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE2 0000	Mode register	SIMDR	H'8000	R/W	16	SIOF
H'FFE2 0002	Clock select register	SISCR	H'C000	R/W	16	SIOF
H'FFE2 0004	Transmit data assign register	SITDAR	H'0000	R/W	16	SIOF
H'FFE2 0006	Receive data assign register	SIRDAR	H'0000	R/W	16	SIOF
H'FFE2 0008	Control data assign register	SICDAR	H'0000	R/W	16	SIOF
H'FFE2 000C	Control register	SICTR	H'0000	R/W	16	SIOF
H'FFE2 0010	FIFO control register	SIFCTR	H'1000	R/W	16	SIOF
H'FFE2 0014	Status register	SISTR	H'0000	R/W	16	SIOF
H'FFE2 0016	Interrupt enable register	SIIER	H'0000	R/W	16	SIOF
H'FFE2 0018 to H'FFE2 001F	Reserved (8 bytes)	—	—	—	—	—
H'FFE2 0020	Transmit data register	SITDR	H'xxxx xxxx	W	32	SIOF
H'FFE2 0024	Receive data register	SIRDR	H'xxxx xxxx	R	32	SIOF
H'FFE2 0028	Transmit control data register	SITCR	H'0000 0000	R/W	32	SIOF
H'FFE2 002C	Receive control data register	SIRCR	H'xxxx xxxx	R/W	32	SIOF
H'FFE2 002C to H'FFE2 FFFF	Reserved (65,492 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE3 0000	Configuration register	CMTCFG	H'0000 0000	R/W	32	CMT
H'FFE3 0004	Free-running timer	CMTFRT	H'0000 0000	R	32	CMT
H'FFE3 0008	Control register	CMTCTL	H'0000 0000	R/W	32	CMT
H'FFE3 000C	IRQ status register	CMTIRQS	H'0000 0000	R/W	32	CMT
H'FFE3 0010	Channel 0 time register	CMTCH0T	H'0000 0000	R/W	32	CMT
H'FFE3 0014	Channel 1 time register	CMTCH1T	H'0000 0000	R/W	32	CMT
H'FFE3 0018	Channel 2 time register	CMTCH2T	H'0000 0000	R/W	32	CMT
H'FFE3 001C	Channel 3 time register	CMTCH3T	H'0000 0000	R/W	32	CMT
H'FFE3 0020	Channel 0 stop time register	CMTCH0ST	H'0000 0000	R/W	32	CMT
H'FFE3 0024	Channel 1 stop time register	CMTCH1ST	H'0000 0000	R/W	32	CMT
H'FFE3 0028 to H'FFE3 002F	Reserved (8 bytes)	—	—	—	—	—
H'FFE3 0030	Channel 0 timer/counter	CMTCH0C	H'0000 0000	R/W	32	CMT
H'FFE3 0034	Channel 1 timer/counter	CMTCH1C	H'0000 0000	R/W	32	CMT
H'FFE3 0038	Channel 2 timer/counter	CMTCH2C	H'0000 0000	R/W	32	CMT
H'FFE3 003C	Channel 3 timer/counter	CMTCH3C	H'0000 0000	R/W	32	CMT
H'FFE3 0040 to H'FFE4 5FFF	Reserved (90,048 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE4 6000 to H'FFE4 6007	Reserved (8 bytes)	—	—	—	—	—
H'FFE4 6008	Control and status register	HACCR	H'0000 0200	R/W	32	HAC
H'FFE4 600C to H'FFE4 601F	Reserved (20 bytes)	—	—	—	—	—
H'FFE4 6020	Command/status address register	HACCSAR	H'0000 0000	R/W	32	HAC
H'FFE4 6024	Command/status data register	HACCSDR	H'0000 0000	R/W	32	HAC
H'FFE4 6028	PCM left channel register	HACPCML	H'0000 0000	R/W	32	HAC
H'FFE4 602C	PCM right channel register	HACPCMR	H'0000 0000	R/W	32	HAC
H'FFE4 6030 to H'FFE4 604F	Reserved (32 bytes)	—	—	—	—	—
H'FFE4 6050	TX interrupt enable register	HACTIER	H'0000 0000	R/W	32	HAC
H'FFE4 6054	TX status register	HACTSR	H'F000 0000	R/W	32	HAC
H'FFE4 6058	RX interrupt enable register	HACRIER	H'0000 0000	R/W	32	HAC
H'FFE4 605C	RX status register	HACRSR	H'0000 0000	R/W	32	HAC
H'FFE4 6060	HAC control register	HACACR	H'8400 0000	R/W	32	HAC
H'FFE4 6064 to H'FFE4 FFFF	Reserved (40,860 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE5 0000	Control register	SPCR	H'0000 0000	R/W	32	HSPI
H'FFE5 0004	Status register	SPSR	H'xxxx xx20	R	32	HSPI
H'FFE5 0008	System control register	SPSCR	H'0000 0040	R/W	32	HSPI
H'FFE5 000C	Transmit buffer register	SPTBR	H'0000 0000	R/W	32	HSPI
H'FFE5 0010	Receive buffer register	SPRBR	H'0000 0000	R	32	HSPI
H'FFE5 0014 to H'FFE5 FFFF	Reserved (65,516 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE6 0000	Command register 0	CMDR0	H'00	R/W	8	MMCIF
H'FFE6 0001	Command register 1	CMDR1	H'00	R/W	8	MMCIF
H'FFE6 0002	Command register 2	CMDR2	H'00	R/W	8	MMCIF
H'FFE6 0003	Command register 3	CMDR3	H'00	R/W	8	MMCIF
H'FFE6 0004	Command register 4	CMDR4	H'00	R/W	8	MMCIF
H'FFE6 0005	Command register 5	CMDR5	H'00	R	8	MMCIF
H'FFE6 0006	Command start register	CMDSTRT	H'00	R/W	8	MMCIF
H'FFE6 000A	Operation control register	OPCR	H'00	R/W	8	MMCIF
H'FFE6 000B	Card status register	CSTR	H'0x	R	8	MMCIF
H'FFE6 000C	Interrupt control register 0	INTCR0	H'00	R/W	8	MMCIF
H'FFE6 000D	Interrupt control register 1	INTCR1	H'00	R/W	8	MMCIF
H'FFE6 000E	Interrupt status register 0	INTSTR0	H'00	R/W	8	MMCIF
H'FFE6 000F	Interrupt status register 1	INTSTR1	H'00	R/W	8	MMCIF
H'FFE6 0010	Transfer clock control register	CLKON	H'00	R/W	8	MMCIF
H'FFE6 0011	Command timeout control register	CTOCR	H'00	R/W	8	MMCIF
H'FFE6 0014	Transfer byte number count register	TBCR	H'00	R/W	8	MMCIF
H'FFE6 0016	Mode register	MODER	H'00	R/W	8	MMCIF
H'FFE6 0018	Command type register	CMDTYR	H'00	R/W	8	MMCIF
H'FFE6 0019	Response type register	RSPTYR	H'00	R/W	8	MMCIF
H'FFE6 001A	Transfer block number counter	TBNCR	H'0000	R/W	16	MMCIF
H'FFE6 001C	Reserved (4 bytes)	—	—	—	—	—
H'FFE6 0020	Response register 0	RSPR0	H'00	R/W	8	MMCIF
H'FFE6 0021	Response register 1	RSPR1	H'00	R/W	8	MMCIF
H'FFE6 0022	Response register 2	RSPR2	H'00	R/W	8	MMCIF
H'FFE6 0023	Response register 3	RSPR3	H'00	R/W	8	MMCIF
H'FFE6 0024	Response register 4	RSPR4	H'00	R/W	8	MMCIF
H'FFE6 0025	Response register 5	RSPR5	H'00	R/W	8	MMCIF
H'FFE6 0026	Response register 6	RSPR6	H'00	R/W	8	MMCIF

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE6 0027	Response register 7	RSPR7	H'00	R/W	8	MMCIF
H'FFE6 0028	Response register 8	RSPR8	H'00	R/W	8	MMCIF
H'FFE6 0029	Response register 9	RSPR9	H'00	R/W	8	MMCIF
H'FFE6 002A	Response register 10	RSPR10	H'00	R/W	8	MMCIF
H'FFE6 002B	Response register 11	RSPR11	H'00	R/W	8	MMCIF
H'FFE6 002C	Response register 12	RSPR12	H'00	R/W	8	MMCIF
H'FFE6 002D	Response register 13	RSPR13	H'00	R/W	8	MMCIF
H'FFE6 002E	Response register 14	RSPR14	H'00	R/W	8	MMCIF
H'FFE6 002F	Response register 15	RSPR15	H'00	R/W	8	MMCIF
H'FFE6 0030	Response register 16	RSPR16	H'00	R/W	8	MMCIF
H'FFE6 0031	CRC status register	RSPRD	H'00	R/W	8	MMCIF
H'FFE6 0032	Data timeout register	DTOUTR	H'FFFF	R/W	16	MMCIF
H'FFE6 0034 to H'FFE6 003F	Reserved (12 bytes)	—	—	—	—	—
H'FFE6 0040	Data register	DR	H'xxxx	R/W	16	MMCIF
H'FFE6 0042	FIFO pointer clear register	FIFOCLR	H'00	W	8	MMCIF
H'FFE6 0044	DMA control register	DMACR	H'00	R/W	8	MMCIF
H'FFE6 0046	Interrupt control register 2	INTCR2	H'00	R/W	8	MMCIF
H'FFE6 0048	Interrupt status register 2	INTSTR2	H'0x	R/W	8	MMCIF
H'FFE6 0049 to H'FFE6 FFFF	Reserved (65,463 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE7 0000	Control register	SSICR	H'0000 0000	R/W	32	SSI
H'FFE7 0004	Status register	SSISR	H'0200 0003	R/W	32	SSI
H'FFE7 0008	Transmit data register	SSITDR	H'0000 0000	R/W	32	SSI
H'FFE7 000C	Receive data register	SSIRDR	H'0000 0000	R	32	SSI
H'FFE7 0010 to H'FFE7 FFFF	Reserved (65,520 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE8 0000	64 Hz counter	R64CNT	H'xx	R	8	RTC
H'FFE8 0004	Second counter	RSECCNT	H'xx	R/W	8	RTC
H'FFE8 0008	Minute counter	RMINCNT	H'xx	R/W	8	RTC
H'FFE8 000C	Hour counter	RHRCNT	H'xx	R/W	8	RTC
H'FFE8 0010	Day-of-week counter	RWKCNT	H'xx	R/W	8	RTC
H'FFE8 0014	Day counter	RDAYCNT	H'xx	R/W	8	RTC
H'FFE8 0018	Month counter	RMONCNT	H'xx	R/W	8	RTC
H'FFE8 001C	Year counter	RYRCNT	H'xxxx	R/W	16	RTC
H'FFE8 0020	Second alarm register	RSECAR	H'xx	R/W	8	RTC
H'FFE8 0024	Minute alarm register	RMINAR	H'xx	R/W	8	RTC
H'FFE8 0028	Hour alarm register	RHRAR	H'xx	R/W	8	RTC
H'FFE8 002C	Day-of-week alarm register	RWKAR	H'xx	R/W	8	RTC
H'FFE8 0030	Day alarm register	RDAYAR	H'xx	R/W	8	RTC
H'FFE8 0034	Month alarm register	RMONAR	H'xx	R/W	8	RTC
H'FFE8 0038	RTC control register 1	RCR1	H'xx	R/W	8	RTC
H'FFE8 003C	RTC control register 2	RCR2	H'x9	R/W	8	RTC
H'FFE8 003D to H'FFE8 004F	Reserved (19 bytes)	—	—	—	—	—
H'FFE8 0050	RTC control register 3	RCR3	H'x0	R/W	8	RTC
H'FFE8 0054	Year alarm register	RYRAR	H'xxxx	R/W	16	RTC
H'FFE8 0056 to H'FFE8 FFFF	Reserved (65,450 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFE9 0000	Common control register	FLCMNCR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0004	Command control register	FLCMDCR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0008	Command code register	FLCMCDR	H'0000 0000	R/W	32	FLCTL
H'FFE9 000C	Address register	FLADR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0010	Data register	FLDATAR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0014	Data counter register	FLDTCNTR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0018	Interrupt DMA control register	FLINTDMACR	H'0000 0000	R/W	32	FLCTL
H'FFE9 001C	Ready busy timeout setting register	FLBSYTMR	H'0000 0000	R/W	32	FLCTL
H'FFE9 0020	Ready busy timeout counter	FLBSYCNT	H'0000 0000	R	32	FLCTL
H'FFE9 0024	Data FIFO register	FLDTFIFO	H'xxxx xxxx	R/W	32	FLCTL
H'FFE9 0028	Control code FIFO register	FLECFIFO	H'xxxx xxxx	R/W	32	FLCTL
H'FFE9 002C	Transfer control register	FLTRCR	H'00	R/W	8	FLCTL
H'FFE9 002D to H'FFE9 FFFF	Reserved (65,491 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFEA 0000	Port A control register	PACR	H'0000	R/W	16	GPIO
H'FFEA 0002	Port B control register	PBCR	H'0000	R/W	16	GPIO
H'FFEA 0004	Port C control register	PCCR	H'0000	R/W	16	GPIO
H'FFEA 0006	Port D control register	PDCR	H'0000	R/W	16	GPIO
H'FFEA 0008	Port E control register	PECR	H'3000	R/W	16	GPIO
H'FFEA 000A	Port F control register	PFCR	H'0000	R/W	16	GPIO
H'FFEA 000C	Port G control register	PGCR	H'0000	R/W	16	GPIO
H'FFEA 000E	Port H control register	PHCR	H'FFFF	R/W	16	GPIO
H'FFEA 0010	Port J control register	PJCR	H'FFFF	R/W	16	GPIO
H'FFEA 0012	Port K control register	PKCR	H'FFFF	R/W	16	GPIO
H'FFEA 0014	Port L control register	PLCR	H'FFFF	R/W	16	GPIO
H'FFEA 0016	Port M control register	PMCR	H'FFFF	R/W	16	GPIO
H'FFEA 0018 to H'FFEA 001F	Reserved (8 bytes)	—	—	—	—	—
H'FFEA 0020	Port A data register	PADR	H'00	R/W	8	GPIO
H'FFEA 0022	Port B data register	PBDR	H'00	R/W	8	GPIO
H'FFEA 0024	Port C data register	PCDR	H'00	R/W	8	GPIO
H'FFEA 0026	Port D data register	PDDR	H'00	R/W	8	GPIO
H'FFEA 0028	Port E data register	PEDR	H'x0	R/W	8	GPIO
H'FFEA 002A	Port F data register	PFDR	H'00	R/W	8	GPIO
H'FFEA 002C	Port G data register	PGDR	H'00	R/W	8	GPIO
H'FFEA 002E	Port H data register	PHDR	H'xx	R/W	8	GPIO
H'FFEA 0030	Port J data register	PJDR	H'xx	R/W	8	GPIO
H'FFEA 0032	Port K data register	PKDR	H'xx	R/W	8	GPIO
H'FFEA 0034	Port L data register	PLDR	H'00	R/W	8	GPIO
H'FFEA 0036	Port M data register	PMDR	H'0x	R/W	8	GPIO
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	GPIO
H'FFEA 0049 to H'FFEA 004D	Reserved (5 bytes)	—	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	Module
H'FFEA 004E	Port H pull-up control register	PHPUPR	H'FF	R/W	8	GPIO
H'FFEA 0050	Port J pull-up control register	PJPUPR	H'FF	R/W	8	GPIO
H'FFEA 0052	Port K pull-up control register	PKPUPR	H'FF	R/W	8	GPIO
H'FFEA 0056	Port M pull-up control register	PMPUPR	H'FF	R/W	8	GPIO
H'FFEA 0057 to H'FFEA 005F	Reserved (9 bytes)	—	—	—	—	—
H'FFEA 0060	Input pin pull-up control register 1	PPUPR1	H'FFFF	R/W	16	GPIO
H'FFEA 0062	Input pin pull-up control register 2	PPUPR2	H'FFFF	R/W	16	GPIO
H'FFEA 0064 to H'FFEA 007F	Reserved (28 bytes)	—	—	—	—	—
H'FFEA 0080	On-chip module select register	OMSELR	H'0000	R/W	16	GPIO
H'FFEA 0082 to H'FFFF FFFF	Reserved (1,441,662 bytes)	—	—	—	—	—

E. Package Dimensions

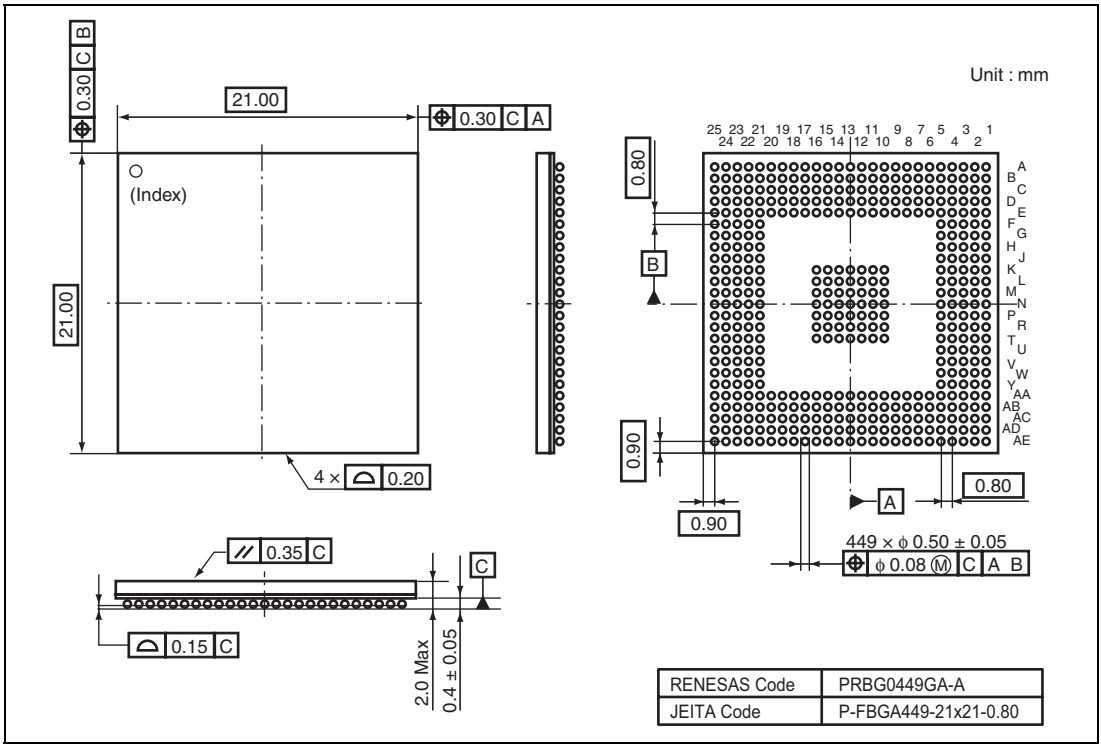


Figure E.1 Package Dimensions (449-Pin BGA)

Note: The Tj (junction temperature) of this LSI becomes over 125°C if operating with the maximum power consumption. So a careful thermal design is necessary. Use a heat sink or forced air cooling to lower the Tj.

F. Mode Pin Settings

The MODE8–MODE0 pin values are input in the event of a power-on reset via the $\overline{\text{PRESET}}$ pin.

[Legend]
H: High level input
L: Low level input

Table F.1 Clock Operating Modes with External Pin Combination

	Pin Value				Frequency (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)	FRQCR Initial Value
0	LL	L	L	On	× 12	× 6	× 3/2	× 24/5	× 3	H'1023 3335
1			H	On	× 12	× 6	× 1	× 24/5	× 2	H'1024 4336
2		H	L	On	× 12	× 6	× 3/2	× 24/5	× 3/2	H'1025 5335
3			H	On	× 12	× 6	× 1	× 24/5	× 1	H'1026 6336
12	HH	L	L	On	× 12	× 4	× 1	× 4	× 2	H'1044 4346

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

Table F.4 PCI Mode

Pin Value	
MODE6	**PCIC Operation Mode**
L	PCIC normal (non-host)
H	PCI host bus bridge

Table F.5 Clock Input

Pin Value	
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.

G. Pin Functions

G.1 Pin States

Table G.1 Pin states in Reset, Power-Down State, and Bus-Released State

				Reset				
Pin Name (LSI level)	**Pin Name (Module level)**	**Related Module**	**I/O**	**Power -on**	**Manual**	**Sleep**	**Module Standby**	**Bus Release**
A[25:0]	A[25:0]	LBSC	O	PZ*[1]	PZ/Z	O	—	PZ/Z
D[31:24]	D[31:24] (default)	LBSC	I/O	Z	PZ/Z	PZ/Z	—	PZ/Z
	PortF[7:0]	GPIO	I/O	—	PI/I/O	PI/I/O	—	PI/I/O
D[23:16]	D[23:16] (default)	LBSC	I/O	Z	PZ/Z	PZ/Z	—	PZ/Z
	Port G[7:0]	GPIO	I/O	—	PI/I/O	PI/I/O	—	PI/I/O
D[15:0]	D[15:0]	LBSC	I/O	Z	PZ/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	—	PZ/Z
$\overline{BACK}$	Port M0 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{BACK}$	LBSC	O	—	H	O	—	O
$\overline{BREQ}$	Port M1 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{BREQ}$	LBSC	I	—	I	I	—	I
$\overline{BS}$	$\overline{BS}$	LBSC	O	H	H	O	—	PZ/Z
R/$\overline{W}$	R/$\overline{W}$	LBSC	O	H	H	O	—	PZ/Z
$\overline{RD}$/$\overline{FRAME}$	$\overline{RD}$/$\overline{FRAME}$	LBSC	O	H	O	O	—	PZ/Z/O
$\overline{RDY}$	$\overline{RDY}$	LBSC	I	Z	PI/I	PI/I	—	PI/I
$\overline{WE0}$/$\overline{REG}$	$\overline{WE0}$/$\overline{REG}$	LBSC	O	H	O	O	—	PZ/Z/O
$\overline{WE1}$	$\overline{WE1}$	LBSC	O	H	O	O	—	PZ/Z/O
$\overline{WE2}$/$\overline{IORD}$	$\overline{WE2}$/$\overline{IORD}$	LBSC	O	H	O	O	—	PZ/Z/O
$\overline{WE3}$/$\overline{IOWR}$	$\overline{WE3}$/$\overline{IOWR}$	LBSC	O	H	O	O	—	PZ/Z/O
$\overline{DACK0}$/MODE0	MODE0 (POR)	CPG	I	I	—	—	—	—
	Port L3*[3] (default)	GPIO	O	—	O	O	—	O
	$\overline{DACK0}$	DMAC	O	—	O	O	K	O

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
$\overline{\text{DACK1}}$/MODE1	MODE1 (POR)	CPG	I	I	—	—	—	—
	Port L2*[3] (default)	GPIO	O	—	O	O	—	O
	$\overline{\text{DACK1}}$	DMAC	O	—	O	O	K	O
$\overline{\text{DACK2}}$/ $\overline{\text{MRESETOUT}}$/ AUDATA2	Port K3 (default)	GPIO	I/O	PI*[2]	I/O	I/O	—	I/O
	$\overline{\text{DACK2}}$	DMAC	O	—	O	O	K	O
	$\overline{\text{MRESETOUT}}$	RESET	O	—	L	O	—	O
	AUDATA2	H-UDI	O	—	O	O	—	O
$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/ AUDATA3	Port K2 (default)	GPIO	I/O	PI*[2]	I/O	I/O	—	I/O
	$\overline{\text{DACK3}}$	DMAC	O	—	O	O	K	O
	$\overline{\text{IRQOUT}}$	INTC	O	—	O	O	—	O
	AUDATA3	H-UDI	O	—	O	O	—	O
$\overline{\text{DRAK0}}$/MODE2	MODE2 (POR)	CPG	I	I	—	—	—	—
	Port L1*[3] (default)	GPIO	O	—	O	O	—	O
	$\overline{\text{DRAK0}}$	DMAC	O	—	O	O	K	O
$\overline{\text{DRAK1}}$/MODE7	MODE7 (POR)	CPG	I	I	—	—	—	—
	Port L0*[3] (default)	GPIO	O	—	O	O	—	O
	$\overline{\text{DRAK1}}$	DMAC	O	—	O	O	K	O
$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/ AUDCK	Port K1*[3] (default)	GPIO	O	O	O	O	—	O
	$\overline{\text{DRAK2}}$	DMAC	O	—	O	O	K	O
	$\overline{\text{CE2A}}$	LBSC	O	—	O	O	—	PZ/Z
	AUDCK	H-UDI	O	—	O	O	—	O
$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/ AUDSYNC	Port K0*[3] (default)	GPIO	O	O	O	O	—	O
	$\overline{\text{DRAK3}}$	DMAC	O	—	O	O	K	O
	$\overline{\text{CE2B}}$	LBSC	O	—	O	O	—	PZ/Z
	AUDSYNC	H-UDI	O	—	O	O	—	O

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
$\overline{\text{DREQ0}}$	Port K7 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{\text{DREQ0}}$	DMAC	I	—	PZ/Z	PI/I	PZ/Z	PI/I
$\overline{\text{DREQ1}}$	Port K6 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{\text{DREQ1}}$	DMAC	I	—	PZ/Z	PI/I	PZ/Z	PI/I
$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/ AUDATA0	Port K5 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{\text{DREQ2}}$	DMAC	I	—	PZ/Z	PI/I	PZ/Z	PI/I
	$\overline{\text{INTB}}$	PCIC	I	—	PI/I	PI/I	—	PI/I
	AUDATA0	H-UDI	O	—	O	O	—	O
$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/ AUDATA1	Port K4 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	$\overline{\text{DREQ3}}$	DMAC	I	—	PZ/Z	PI/I	PZ/Z	PI/I
	$\overline{\text{INTC}}$	PCIC	I	—	PI/I	PI/I	—	PI/I
	AUDATA1	H-UDI	O	—	O	O	—	O
MCLK	MCLK	DDRIF	O	O	O	O	—	O
$\overline{\text{MCLK}}$	$\overline{\text{MCLK}}$	DDRIF	O	O	O	O	—	O
MDQS[3:0]	MDQS[3:0]	DDRIF	I/O	Z	Z	I/O	—	I/O
MDQM[3:0]	MDQM[3:0]	DDRIF	O	H	H	O	—	O
MDA[31:0]	MDA[31:0]	DDRIF	I/O	Z	Z	I/O	—	I/O
CKE	CKE	DDRIF	O	O	O	O	—	O
$\overline{\text{MCAS}}$	$\overline{\text{MCAS}}$	DDRIF	O	H	H	O	—	O
$\overline{\text{MRAS}}$	$\overline{\text{MRAS}}$	DDRIF	O	H	H	O	—	O
$\overline{\text{MCS}}$	$\overline{\text{MCS}}$	DDRIF	O	H	H	O	—	O
$\overline{\text{MWE}}$	$\overline{\text{MWE}}$	DDRIF	O	H	H	O	—	O
MA[13:0]	MA[13:0]	DDRIF	O	L	L	O	—	O
BA[1:0]	BA[1:0]	DDRIF	O	L	L	O	—	O
$\overline{\text{BKPRST}}$	$\overline{\text{BKPRST}}$	DDRIF	I	PI	PI	PI	—	PI

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset			Module Standby	Bus Release
				Power -on	Manual	Sleep		
AD [31:24]	AD[31:24] (default)	PCIC	I/O	Z	I/O	I/O	—	I/O
	Port A[7:0]	GPIO	I/O	—	I/O	I/O	—	I/O
AD [23:16]	AD[23:16] (default)	PCIC	I/O	Z	I/O	I/O	—	I/O
	Port B[7:0]	GPIO	I/O	—	I/O	I/O	—	I/O
AD [15:8]	AD[15:8] (default)	PCIC	I/O	Z	I/O	I/O	—	I/O
	Port C[7:0]	GPIO	I/O	—	I/O	I/O	—	I/O
AD [7:0]	AD[7:0] (default)	PCIC	I/O	Z	I/O	I/O	—	I/O
	Port D[7:0]	GPIO	I/O	—	I/O	I/O	—	I/O
CBE [3:0]	CBE[3:0]	PCIC	I/O	Z	I/O	I/O	—	I/O
$\overline{\text{GNT0/GNTIN}}$	$\overline{\text{GNT0/GNTIN}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{GNT[3:1]}}$	$\overline{\text{GNT[3:1]}}$ (default)	PCIC	O	PZ	O	O	—	O
	Port E0-E2	GPIO	I/O	—	PI/O	PI/O	—	PI/O
$\overline{\text{REQ0/REQOUT}}$	$\overline{\text{REQ0/REQOUT}}$	PCIC	I/O	PZ	I/O	I/O	—	I/O
$\overline{\text{REQ[3:1]}}$	$\overline{\text{REQ[3:1]}}$ (default)	PCIC	I	PZ	PI	PI	—	PI
	Port E3-E5	GPIO	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{PCIFRAME}}$	$\overline{\text{PCIFRAME}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
IDSEL	IDSEL	PCIC	I	PZ	PI	PI	—	PI
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{LOCK}}$	$\overline{\text{LOCK}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
PAR	PAR	PCIC	I/O	Z	I/O	I/O	—	I/O
PCICLK	PCICLK	PCIC	I	I	I	I	—	I
$\overline{\text{PCIRESET}}$	$\overline{\text{PCIRESET}}$	PCIC	O	L	K	K	—	O
$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	PCIC	I/O	PZ	PI/O	PI/O	—	PI/O
CLKOUT	CLKOUT	CPG	O	O	Z/O	Z/O	—	Z/O
$\overline{\text{PRESET}}$	$\overline{\text{PRESET}}$	RESET	I	I	I	I	—	I
EXTAL	EXTAL	CPG	I	I	I	I	—	I
XTAL	XTAL	CPG	O	O	O	O	—	O
STATUS0/ CMT_CTR0	STATUS0 (default)	RESET	O	H	H	L	—	O
	CMT_CTR0	CMT	I/O	—	PZ/O	I/O	K	PI/I/O
STATUS1/ CMT_CTR1	STATUS1 (default)	RESET	O	H	H	H	—	O
	CMT_CTR1	CMT	I/O	—	PZ/O	I/O	K	PI/I/O
IRQ/$\overline{\text{IRL}}$[3:0]	IRQ/$\overline{\text{IRL}}$[3:0]	INTC	I	PI*[2]	PI/I	PI/I	—	PI/I
IRQ/$\overline{\text{IRL4}}$/FD4/ MODE3	MODE3 (POR)	LBSC	I	I	—	—	—	—
	IRQ/$\overline{\text{IRL4}}$ (default)	INTC	I	—	I	I	—	I
	FD4	FLCTL	I/O	—	I/O	I/O	K	I/O
IRQ/$\overline{\text{IRL5}}$/FD5/ MODE4	MODE4 (POR)	LBSC	I	I	—	—	—	—
	IRQ/$\overline{\text{IRL5}}$ (default)	INTC	I	—	I	I	—	I
	FD5	FLCTL	I/O	—	I/O	I/O	K	I/O
IRQ/$\overline{\text{IRL6}}$/FD6/ MODE6	MODE6 (POR)	PCIC	I	I	—	—	—	—
	IRQ/$\overline{\text{IRL6}}$ (default)	INTC	I	—	I	I	—	I
	FD6	FLCTL	I/O	—	Z	I/O	K	I/O
IRQ/$\overline{\text{IRL7}}$/FD7	Port E6 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	IRQ/$\overline{\text{IRL7}}$	INTC	I	—	PI/I	PI/I	—	PI/I
	FD7	FLCTL	I/O	—	I/O	PI/I/O	K	PI/I/O
NMI	NMI	INTC	I	PI*[2]	PI/I	PI/I	—	PI/I

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
SCIF0_CTS/ INTD/FCLE	Port H1 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF0_CTS	SCIF	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	INTD	PCIC	I	—	PI/I	PI/I	—	PI/I
SCIF0_RTS/ HSPI_CS/FSE	Port H0 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF0_RTS	SCIF	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	HSPI_CS	HSPI	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	FSE	FLCTL	O	—	O	O	K	O
SCIF0_RXD/ HSPI_RX/FRB	Port H2 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF_RXD	SCIF	I	—	PI/I	PI/I	PZ/Z	PI/I
	HSPI_RX	HSPI	I	—	PI/I	PI/I	PZ/Z	PI/I
	FRB	FLCTL	I	—	PI/I	PI/I	PZ/Z	PI/I
SCIF0_SCK/ HSPI_CLK/FRE	Port H4 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF0_SCK	SCIF	I/O	—	PI/I	PI/I/O	K	PI/I/O
	HSPI_CLK	HSPI	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	FRE	FLCTL	O	—	O	O	K	O
SCIF0_TXD/ HSPI_TX/FWE/ MODE8	MODE8 (POR)	CPG	I	I	—	—	—	—
	Port H3*3 (default)	GPIO	O	—	O	O	—	O
	SCIF0_TXD	SCIF	O	—	PZ/Z	O	K	O
	HSPI_TX	HSPI	O	—	PZ/Z	O	K	O
	FWE	FLCTL	O	—	PZ/Z	O	K	O
SCIF1_RXD/ MCDAT	Port H5 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF1_RXD	SCIF	I	—	PI/I	PI/I	PZ/Z	PI/I
	MCDAT	MMCIF	I/O	—	PI/I	PI/I/O	K	PI/I/O

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
SCIF1_SCK/ MCCMD	Port H7 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SCIF1_SCK	SCIF	I/O	—	PI/I	PI/I/O	K	PI/I/O
	MCCMD	MMCIF	I/O	—	PI/I	PI/I/O	K	PI/I/O
SCIF1_TXD/ MCCLK/MODE5	MODE5 (POR)	LBSC	I	I	—	—	—	—
	Port H6*3 (default)	GPIO	O	—	O	O	—	O
	SCIF1_TXD	SCIF	O	—	O	O	K	O
	MCCLK	MMCIF	O	—	O	O	K	O
SIOF_MCLK/ HAC_RES	Port J2 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SIOF_MCLK	SIOF	I	—	PI/I	PI/I	PZ/Z	PI/I
	HAC_RES	HAC	O	—	O	O	K	O
SIOF_RXD/ HAC_SDIN/ SSI_SCK	Port J4 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SIOF_RXD	SIOF	I	—	PI/I	PI/I	PZ/Z	PI/I
	HAC_SDIN	HAC	I	—	PI/I	PI/I	PZ/Z	PI/I
	SSI_SCK	SSI	I/O	—	PI/I	PI/I/O	K	PI/I/O
SIOF_SCK/ HAC_BITCLK/ SSI_CLK	Port J1 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SIOF_SCK	SIOF	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	HAC_BITCLK	HAC	I	—	PI/I	PI/I	PZ/Z	PI/I
	SSI_CLK	SSI	I/O	—	PI/I	PI/I/O	K	PI/I/O
SIOF_SYNC/ HAC_SYNC/ SSI_WS	Port J3 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SIOF_SYNC	SIOF	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	HAC_SYNC	HAC	O	—	O	O	K	O
	SSI_WS	SSI	I/O	—	PI/I	PI/I/O	K	PI/I/O
SIOF_TXD/ HAC_SDOUT/ SSI_SDATA	Port J5 (default)	GPIO	I/O	PI*2	PI/I/O	PI/I/O	—	PI/I/O
	SIOF_TXD	SIOF	O	—	O	O/Z	K	O/Z
	HAC_SDOUT	HAC	O	—	O	O	K	O
	SSI_SDATA	SSI	I/O	—	PI/I	PI/I/O	K	PI/I/O

Pin Name (LSI level)	Pin Name (Module level)	Related Module	I/O	Reset: Power-on	Reset: Manual	Sleep	Module Standby	Bus Release
TCLK/$\overline{\text{IOIS16}}$	Port J0 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—	PI/I/O
	TCLK	TMU	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
	$\overline{\text{IOIS16}}$	LBSC	I	—	PI/I	PI/I	—	PI/I
$\overline{\text{ASEBRK}}$/ BRKACK	$\overline{\text{ASEBRK}}$/ BRKACK	H-UDI	I/O	PI	PI/O	PI/O	—	PI/O
TCK	TCK	TMU	I	PI	PI	PI	—	PI
$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	H-UDI	I	PI	PI	PI	—	PI
TDI	TDI	H-UDI	I	PI	PI	PI	—	PI
TMS	TMS	H-UDI	I	PI	PI	PI	—	PI
TDO	TDO	H-UDI	O	O	O	O	—	O
AUDCK/FALE	AUDCK (default)	H-UDI	O	O	O	O	—	O
	FALE	FLCTL	O	—	O	O	K	O
AUDSYNC/$\overline{\text{FCE}}$	AUDSYNC (default)	H-UDI	O	O	O	O	—	O
	$\overline{\text{FCE}}$	FLCTL	O	—	O	O	K	O
AUDATA[3:0]/ FD [3:0]	AUDATA[3:0] (default)	H-UDI	O	O	O	O	—	O
	FD[3:0]	FLCTL	I/O	—	PI/I/O	PI/I/O	K	PI/I/O
MPMD	MPMD	H-UDI	I	PI	PI	PI	—	PI
$\overline{\text{XRTCSTBI}}$	$\overline{\text{XRTCSTBI}}$	RTC	I	I	I	I	—	I
XTAL2	XTAL2	RTC	O	O	O	O	—	O
EXTAL2	EXTAL2	RTC	I	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 Bus State Controller (LBSC), section 28, General Purpose I/O (GPIO), and related module section.
K: Input is high impedance and output is held its state.
POR: Power on reset

Notes: 1. High impedance state until the internal clock is stable.
2. Input state until the internal clock is stable.
3. Do not input signals to these pins immediately after power-on reset because these initial states are port outputs.

G.2 Handling of Unused Pins

Table G.2 Treatment of Unused Pins

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
A[25:0]	A[25:0]	LBSC	O	Open
D[31:24]	D[31:24] (default)	LBSC	I/O	Open
	Port F[7:0]	GPIO	I/O	
D[23:16]	D[23:16] (default)	LBSC	I/O	Open
	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 VSSQ*[1]
$\overline{WE0}$/$\overline{REG}$	$\overline{WE0}$/$\overline{REG}$	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 power-on reset
	Port L3 (default)	GPIO	O	Open
	$\overline{DACK0}$	DMAC	O	
$\overline{DACK1}$/MODE1	MODE1 (POR)	DMAC	I	Must be used during power-on reset
	Port L2 (default)	GPIO	O	Open
	$\overline{DACK1}$	DMAC	O	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/ AUDATA2	Port K3 (default)	GPIO	I/O	Open
	$\overline{\text{DACK2}}$	DMAC	O	
	$\overline{\text{MRESETOUT}}$	RESET	O	
	AUDATA2	H-UDI	O	
$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/ AUDATA3	Port K2 (default)	GPIO	I/O	Open
	$\overline{\text{DACK3}}$	DMAC	O	
	$\overline{\text{IRQOUT}}$	INTC	O	
	AUDATA3	H-UDI	O	
$\overline{\text{DRAK0}}$/MODE2	MODE2 (POR)	CPG	I	Must be used during power-on reset
	Port L1 (default)	GPIO	O	Open
	$\overline{\text{DRAK0}}$	DMAC	O	
$\overline{\text{DRAK1}}$/MODE7	MODE7 (POR)	CPG	I	Must be used during power-on reset
	Port L0 (default)	GPIO	O	Open
	$\overline{\text{DRAK1}}$	DMAC	O	
$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK	Port K1 (default)	GPIO	O	Open
	$\overline{\text{DRAK2}}$	DMAC	O	
	$\overline{\text{CE2A}}$	LBSC	O	
	AUDCK	H-UDI	O	
$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/ AUDSYNC	Port K0 (default)	GPIO	O	Open
	$\overline{\text{DRAK3}}$	DMAC	O	
	$\overline{\text{CE2B}}$	LBSC	O	
	AUDSYNC	H-UDI	O	
$\overline{\text{DREQ0}}$	Port K7 (default)	GPIO	I/O	Open
	$\overline{\text{DREQ0}}$	DMAC	I	
$\overline{\text{DREQ1}}$	Port K6 (default)	GPIO	I/O	Open
	$\overline{\text{DREQ1}}$	DMAC	I	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/ AUDATA0	Port K5 (default)	GPIO	I/O	Open
	$\overline{\text{DREQ2}}$	DMAC	I	
	$\overline{\text{INTB}}$	PCIC	I	
	AUDATA0	H-UDI	O	
$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/ AUDATA1	Port K4 (default)	GPIO	I/O	Open
	$\overline{\text{DREQ3}}$	DMAC	I	
	$\overline{\text{INTC}}$	PCIC	I	
	AUDATA1	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-DDR
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	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
CBE[3:0]	CBE[3:0]	PCIC	I/O	Open
$\overline{\text{GNT0/GNTIN}}$	$\overline{\text{GNT0/GNTIN}}$	PCIC	I/O	Open
$\overline{\text{GNT[3:1]}}$	$\overline{\text{GNT[3:1]}}$ (default)	PCIC	O	Open
	Port E0-E2	GPIO	I/O	
$\overline{\text{REQ0/REQOUT}}$	$\overline{\text{REQ0/REQOUT}}$	PCIC	I/O	Open
$\overline{\text{REQ[3:1]}}$	$\overline{\text{REQ[3:1]}}$ (default)	PCIC	I	Open
	Port E3-E5	GPIO	I/O	
$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	PCIC	I/O	Open
$\overline{\text{PCIFRAME}}$	$\overline{\text{PCIFRAME}}$	PCIC	I/O	Open
IDSEL	IDSEL	PCIC	I	Pulled-down to VSSQ
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	PCIC	I/O	Pulled-up to VDDQ
$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	PCIC	I/O	Open
$\overline{\text{LOCK}}$	$\overline{\text{LOCK}}$	PCIC	I/O	Open
PAR	PAR	PCIC	I/O	Open
PCICLK	PCICLK	PCIC	I	Fixed to VSSQ
$\overline{\text{PCIRESET}}$	$\overline{\text{PCIRESET}}$	PCIC	O	Open
$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	PCIC	I/O	Open
$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	PCIC	I/O	Pulled-up to VDDQ
$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	PCIC	I/O	Open
$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	PCIC	I/O	Open
CLKOUT	CLKOUT	CPG	O	Open
$\overline{\text{PRESET}}$	$\overline{\text{PRESET}}$	RESET	I	Must be used during power-on 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/$\overline{\text{IRL}}$[3:0]	IRQ/$\overline{\text{IRL}}$[3:0]	INTC	I	Pulled-up to VDDQ

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
IRQ/$\overline{\text{IRL4}}$/FD4/MODE3	MODE3 (POR)	LBSC	I	Must be used during power-on reset
	IRQ/$\overline{\text{IRL4}}$ (default)	INTC	I	Pulled-up to VDDQ
	FD4	FLCTL	I/O	
IRQ/$\overline{\text{IRL5}}$/FD5/MODE4	MODE4 (POR)	LBSC	I	Must be used during power-on reset
	IRQ/$\overline{\text{IRL5}}$ (default)	INTC	I	Pulled-up to VDDQ
	FD5	FLCTL	I/O	
IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	MODE6 (POR)	PCIC	I	Must be used during power-on 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	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$	Port H4 (default)	GPIO	I/O	Open
	SCIF0_SCK	SCIF	I/O	
	HSPI_CLK	HSPI	I/O	
	$\overline{\text{FRE}}$	FLCTL	O	
SCIF0_TXD/HSPI_TX/$\overline{\text{FWE}}$/MODE8	MODE8 (POR)	CPG	I	Must be used during power-on reset
	Port H3 (default)	GPIO	O	Open
	SCIF0_TXD	SCIF	O	
	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 power-on 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	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
SIOF_SCK/ HAC_BITCLK/SSI_CLK	Port J1 (default)	GPIO	I/O	Open
	SIOF_SCK	SIOF	I/O	
	HAC_BITCLK	HAC	I	
	SSI_CLK	SSI	I/O	
SIOF_SYNC/ HAC_SYNC/SSI_WS	Port J3 (default)	GPIO	I/O	Open
	SIOF_SYNC	SIOF	I/O	
	HAC_SYNC	HAC	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 connected 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	

Pin Name (LSI level)	Pin Name (Module level)	Module	I/O	When Not in Use
MPMD	MPMD	H-UDI	I	Pulled-up to VDDQ*[2]
$\overline{\text{XRTCSTBI}}$	$\overline{\text{XRTCSTBI}}$	RTC	I	Pulled-up to VDD-RTC
XTAL2	XTAL2	RTC	O	Open
EXTAL2	EXTAL2	RTC	I	Pulled-up to VDD-RTC

Notes: Power must be supplied to each power supply pin, even when the function pin is not used. When the pin is not used, do not set the register for the pin.

1. This pin is pulled-up within this LSI after power-on reset. Set the RDYPUP bit in PPUPR1 (GPIO) to 1 to pull-up off the $\overline{\text{RDY}}$ pin's pulled-up.
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 another pin which operates in the same manner as $\overline{\text{PRESET}}$. However, when fixed to a ground pin, the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground pin. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

H. Turning On and Off Power Supply

- Turning On Power Supply
  - — There is no restriction for the order of the power supply between each power supplies. Within 300 ms after turning on a single power supply, turn on all the other power supplies. (except DDR-SDRAM power supply backup and RTC power supply backup)
- Turning Off Power Supply
  - — There is no restriction for the order of the power supply between each power supplies. Within 300 ms after turning off a single power supply, turn off all the other power supplies. (except DDR-SDRAM power supply backup and RTC power supply backup)

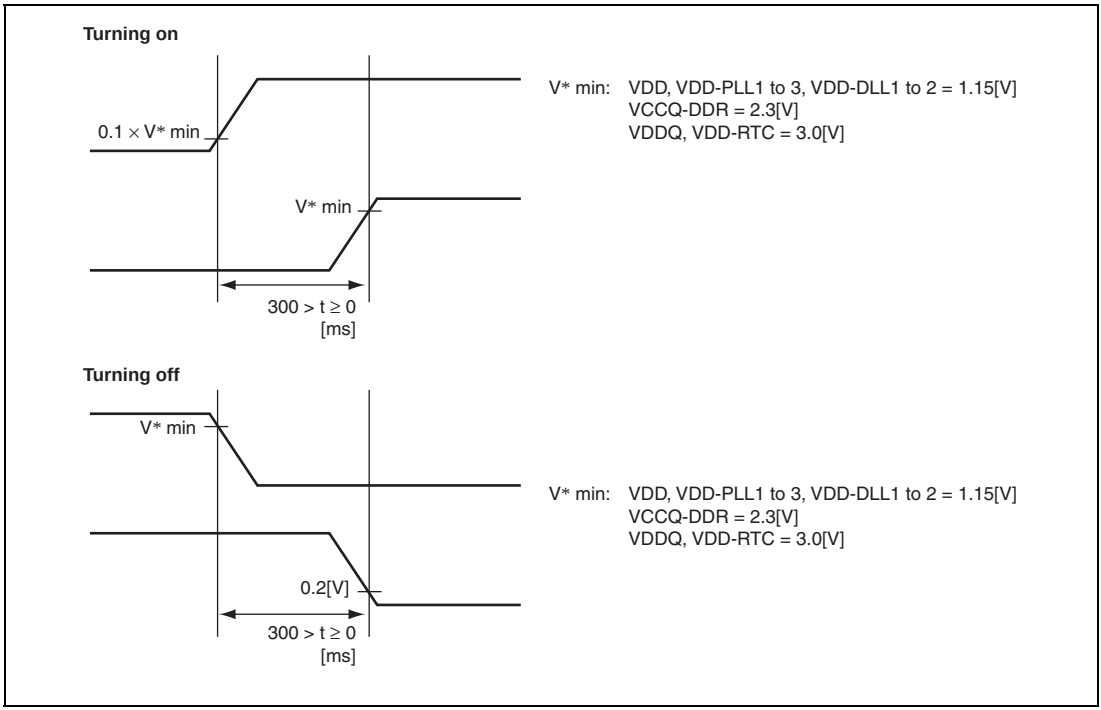


Figure H.1 Sequence of Turning On and Off Power Supply

I. Version Registers (PVR, PRR)

The SH7780 has the read-only registers which show the version of a processor core, and the version of a product. By using the value of these registers, it becomes possible to be able to distinguish the version and product of a processor from software, and to realize the scalability of the high system. Since the values of the version registers differ for every product, please refer to the hardware manual or contact Renesas Technology Corp.

Note: The bit 7 to bit 0 of PVR register and the bit 3 to bit 0 of PRR register should be masked by the software.

Table I.1 Register Configuration

Register Name	Abbrev.	R/W	Initial Value	P4 Address	Area 7 Address	Access Size
Processor Version Register	PVR	R	H'1020 0Axx	H'FF00 0030	H'1F00 0030	32
Product Register	PRR	R	H'0000 092x	H'FF00 0044	H'1F00 0044	32

[Legend] x: Undefined

- Processor Version Register (PVR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	version information															
Initial value:	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	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	1	0
	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	17	16
	version information															
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	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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	—	—	—	—

J. Part Number List

Table J.1 SH7780 Product Lineup

Product Name	Voltage	Operating Frequency	Part Number	Operating Temperature	Package
SH7780	1.25 V	400 MHz	R8A77800ADBG	−40 to 85 °C	449-pin BGA
			R8A77800ADBGV		449-pin BGA (Lead Free)
			R8A77800ANBG	−20 to 75 °C	449-pin BGA
			R8A77800ANBGV		449-pin BGA (Lead Free)

Index

Numerics

16-Bit Timer
Input Capture 697
Output Compare 699
32-Bit address extended mode 188
32-Bit Timer
Input Capture 692
Output Compare 693

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
Burst ROM Interface 370
Bus Arbitration 395
Byte Control SRAM Interface 389

C

Cacheability bit 168
Caches 197
Clock Pulse Generator (CPG) 613
Clocked synchronous serial communication mode 733
Command Access Mode 1044
Compare Match Timer (CMT) 677
Control registers 34
Crystal Oscillator Circuit 730
CUI 730
Cycle-steal mode 597

D

Data address error 115
Data Alignment 406
Data TLB miss exception 110, 180
Data TLB multiple hit exception 180
Data TLB multiple-hit exception 109
Data TLB protection violation exception 113, 181
DDR-SDRAM Interface (DDRIF) 401
DDR-SDRAM Power Supply Backup 650
Delay slot 51
Delayed branches 51
Direct memory access controller (DMAC) 557
Dirty bit 169
Division by zero 142
Double-precision floating-point extended registers 38

Double-precision floating-point registers or single-precision floating-point registers 38
Dual address mode 595

E

Effective address 53
Endian 352
ERI 793
Exception flow 104
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
General interrupt request 125
General Purpose I/O (GPIO) 1055
General registers 34
Geometric operation instructions 144

H

Hardware ITLB miss handling 175
H-UDI reset 108

I

I/O card interface 372
IC memory card interface 372
Inexact exception 142
Initial page write exception 112, 182
Instruction address error 116
Instruction execution state 49
Instruction fetch cycle break 1122
Instruction set 51
Instruction TLB miss exception 111, 178
Instruction TLB multiple hit exception 177
Instruction TLB multiple-hit exception 109
Instruction TLB protection violation exception 114, 179
Intermittent mode 16 598
Interrupt Controller (INTC) 243
Interrupt Response Time 311
Interrupts
  TICPI 674
  TUNI 674
Invalid operation 142
IRL Interrupts 297
IRQ Interrupts 296
Issue rates 87
ITLB 170
ITLB address array 184
ITLB data array 185

L

L memory 227
Little endian 48
Load-store architecture 51
Local Bus State Controller (LBSC) 315
Logic operation instructions 63

M

Manual reset 108
Master Mode 397
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
PCMCIA Support 325
Pipelining 73
Power-Down Mode 428, 643
Power-down state 49
Power-on reset 108
PPN 168
Pre-execution user break/ post-execution user break 122
Prefetch instruction 238
PRI 730
Privileged mode 34
Processing modes 34
Programming model 33
Protection key data 168

R

Realtime Clock (RTC) 707
Registers
- BCR 333
- CAMR0 1114
- CAMR1 1114
- CAR0 1113
- CAR1 1113
- CBCR 1119
- CBR0 1105
- CBR1 1105
- CCMFR 1118
- CCR 201
- CDMR1 1116
- CDR1 1115
- CETR1 1117
- CHCR 572
- CLKON 885
- CMDR 871
- CMDSTRT 872
- CMDTYR 889
- CMTCFG 681
- CMTCHnC 690
- CMTCHnST 689

CMTCHnT 689
CMTCTL 684
CMTFRT 684
CMTIRQS 688
CPUOPM 1217
CRR0 1111
CRR1 1111
CSnBCR 336
CSnPCR 347
CSnWCR 342
CSTR 875
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
HACCR 958
HACCSAR 960
HACCSDR 962
HACPCML 963
HACPCMR 965
HACRIER 969
HACRSR 970
HACTIER 966
HACTSR 967
ICR 255
INT2A 277
INT2B 287
INT2GPIC 294
INT2MSKCR 285
INT2MSKR 282
INT2PRI 276
INTCR 877
INTEVT 100
INTMSK 261
INTMSKCLR 266
INTPRI 259
INTREQ 260
INTSTR 880
IRMCR 165
LDA0 234
LDA1 236
LSA0 230
LSA1 232
MACH 42
MACL 42
MIM 412
MMC Mode 901
MMSELR 331
MMUCR 160
MODER 888
MSTPCR 646
NMIFCR 271
OPCR 873
PACR 1063
PADR 1081

PASCR 164
PBCR 1064
PBDR 1081
PC 42
PCCR 1066
PCDR 1082
PDCR 1067
PDDR 1082
PECR 1069
PEDR 1083
PEPUPR 1087
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
RCR1 721
RCR2 723
RCR3 726
RDAYAR 719
RDAYCNT 715
RHRAR 718
RHRCNT 713
RMINAR 717
RMINCNT 713
RMONAR 720
RMONCNT 716
RSECAR 717
RSECCNT 712
RSPR 895
RSPTYR 890
RWKAR 718
RWKCNT 714
RYRAR 720
RYRCNT 716
SAR 567
SARB 568
SCBRR 758
SCFCR 759
SCFRDR 742
SCFSR 751
SCFTDR 743
SCLSR 767
SCR 416
SCRER 768
SCRFDR 763
SCRSR 742
SCSCR 747
SCSMR 744
SCSPTR 764
SCTFDR 762
SCTSR 743
SDBSR 1143
SDINT 1142
SDIR 1141
SDMR 422

SDR 421
SGR 42
SICDAR 825
SICTR 806
SIFCTR 821
SIIER 819
SIMDR 802
SIRCR 812
SIRDAR 824
SIRDR 810
SISCR 804
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
VBR 42
Relative priorities 102
Reset state 49
Rounding 141
Round-robin mode 592
RTC Power Supply Backup 653
RXI 793

S

Sector Access Mode 1046
Sequential break 1124
Serial Communication Interface with FIFO (SCIF) 733
Serial I/O with FIFO (SIOF) 797
Serial Protocol Interface (HSPI) 849
Serial Sound Interface (SSI) Module 983
Share status bit 168
Shift instructions 64
Sign-extended 47
Single virtual memory mode 155
Single-precision floating-point extended register matrix 39
Single-precision floating-point extended registers 38
Single-precision floating-point registers 38
Single-precision floating-point vector registers 38
Sleep Mode 648
Slot FPU disable exception 121
Slot illegal instruction exception 119
SRAM interface 361
System control instructions 66
System registers 34
System registers related to FPU 35

T

T bit 52
TAP control 1152

TCNT Count Timing 671
Time Setting 727
Timer Unit
Input Capture Function 673
Timer Unit (TMU) 657
TXI 793
Types of exceptions 102

U

Unconditional trap 117
Underflow 142
User break controller 1101
User break operation 1121
User debugging interface 1135
User mode 34
UTLB 167
UTLB address array 186
UTLB data array 187

V

Validity bit 168
Vector addresses 102
Virtual address space 149
VPN 167

W

Wait Cycles between Accesses 393
Watchdog Timer and Reset 625
Write-back instruction 238
Write-through bit 169

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 Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 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, Malaysia
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