REJ09B0079-0200

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

32

SH7710 Group

Hardware Manual

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

SH7710 HD6417710

Hardware Manual

Rev.2.00
Revision Date: Dec. 07, 2005

Renesas Technology
www.renesas.com

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.

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 Peripheral 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. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

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

11. Index

Preface

The SH7710 RISC (Reduced Instruction Set Computer) microcomputer 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.
Refer to the SH-3/SH-3E/SH3-DSP Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

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

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7710	HD6417710

- 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.

- In order to understand the details of the CPU's functions

  Read the SH-3/SH-3E/SH3-DSP Programming Manual.

Rules:	Register name:	The following notation is used for cases when the same or a similar function, e.g. serial communication interface, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB (most significant bit) is on the left and the LSB (least significant bit) 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: $\overline{xxxx}$

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

SH7710 manuals:

Document Title	Document No.
SH7710 Hardware Manual	This manual
SH-3/SH-3E/SH3-DSP Programming Manual	ADE-602-096B

User's manuals for development tools:

Document Title	Document No.
SuperH™ RISC engine C/C++ Compiler,Assembler,Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0152-0101
SuperH™ RISC engine High-performance Embedded Workshop 3 Users Manual	REJ10B0025-0200
SuperH RISC engine High-Performance Embedded Workshop 3 Tutorial	REJ10B0023-0200

Application note:

Document Title	Document No.
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B0463-0100

Abbreviations

ACIA	Asynchronous communication interface adapter
AUD	Advanced user debugger
BSC	Bus state controller
CPG	Clock pulse generator
DMA	Direct memory access
DMAC	Direct memory access controller
etu	Elementary time unit
FIFO	First-in first-out
H-UDI	User debugging interface
INTC	Interrupt controller
IPSEC	Security architecture accelerator for internet protocol
JTAG	Joint test action group
LSB	Least significant bit
MMU	Memory management unit
MSB	Most significant bit
PFC	Pin function controller
RISC	Reduced instruction set computer
RTC	Realtime clock
SCIF	Serial communication interface with FIFO
SIOF	Serial I/O with FIFO
TLB	Translation lookaside buffer
TMU	Timer unit
UART	Universal asynchronous receiver/transmitter
UBC	User break controller
WDT	Watchdog timer

Contents

Section 1 Overview and Pin Function 1
1.1 Features 1
1.2 Block Diagram 7
1.3 Pin Description 8
1.3.1 Pin Assignment 8
1.3.2 Pin Functions 19

Section 2 CPU 27
2.1 Processing States and Processing Modes 27
2.1.1 Processing States 27
2.1.2 Processing Modes 28
2.2 Memory Map 29
2.2.1 Logical Address Space 29
2.2.2 External Memory Space 30
2.3 Register Descriptions 32
2.3.1 General Registers 35
2.3.2 System Registers 36
2.3.3 Program Counter 37
2.3.4 Control Registers 38
2.4 Data Formats 42
2.4.1 Register Data Format 42
2.4.2 Memory Data Formats 42
2.5 Features of CPU Core Instructions 44
2.5.1 Instruction Execution Method 44
2.5.2 CPU Instruction Addressing Modes 45
2.5.3 CPU Instruction Formats 48
2.6 Instruction Set 52
2.6.1 CPU Instruction Set Based on Functions 52
2.6.2 Operation Code Map 66

Section 3 DSP Operating Unit 71
3.1 DSP Extended Functions 71
3.2 DSP Mode Resources 73
3.2.1 Processing Modes 73
3.2.2 DSP Mode Memory Map 73
3.2.3 CPU Register Sets 74

3.2.4 DSP Registers 77
3.3 CPU Extended Instructions 78
3.3.1 Repeat Control Instructions 78
3.3.2 Extended Repeat Control Instructions 88
3.4 DSP Data Transfer Instructions 93
3.4.1 General Registers 97
3.4.2 DSP Data Addressing 99
3.4.3 Modulo Addressing 100
3.4.4 Memory Data Formats 103
3.4.5 Instruction Formats of Double and Single Transfer Instructions 103
3.5 DSP Data Operation Instructions 106
3.5.1 DSP Registers 106
3.5.2 DSP Operation Instruction Set 111
3.5.3 DSP-Type Data Formats 116
3.5.4 ALU Fixed-Point Operations 118
3.5.5 ALU Integer Operations 123
3.5.6 ALU Logical Operations 125
3.5.7 Fixed-Point Multiply Operation 126
3.5.8 Shift Operations 128
3.5.9 Most Significant Bit Detection Operation 132
3.5.10 Rounding Operation 135
3.5.11 Overflow Protection 137
3.5.12 Local Data Move Instruction 138
3.5.13 Operand Conflict 139
3.6 DSP Extended Function Instruction Set 140
3.6.1 CPU Extended Instructions 140
3.6.2 Double-Data Transfer Instructions 142
3.6.3 Single-Data Transfer Instructions 143
3.6.4 DSP Operation Instructions 145
3.6.5 Operation Code Map in DSP Mode 151

Section 4 Exception Handling 155
4.1 Register Descriptions 155
4.1.1 TRAPA Exception Register (TRA) 156
4.1.2 Exception Event Register (EXPEVT) 157
4.1.3 Interrupt Event Register (INTEVT) 157
4.1.4 Interrupt Event Register 2 (INTEVT2) 158
4.1.5 Exception Address Register (TEA) 158
4.2 Exception Handling Function 159
4.2.1 Exception Handling Flow 159

4.2.2 Exception Vector Addresses 160
4.2.3 Exception Codes 160
4.2.4 Exception Request and BL Bit (Multiple Exception Prevention) 160
4.2.5 Exception Source Acceptance Timing and Priority 161
4.3 Individual Exception Operations 165
4.3.1 Resets 165
4.3.2 General Exceptions 166
4.3.3 General Exceptions (MMU Exceptions) 169
4.4 Exception Processing while DSP Extension Function is Valid 172
4.4.1 Illegal Instruction Exception and Slot Illegal Instruction Exception 172
4.4.2 CPU Address Error 172
4.4.3 Exception in Repeat Control Period 172
4.5 Usage Notes 179

Section 5 Memory Management Unit (MMU) 181
5.1 Role of MMU 181
5.1.1 MMU of This LSI 183
5.2 Register Descriptions 189
5.2.1 Page Table Entry Register High (PTEH) 190
5.2.2 Page Table Entry Register Low (PTEL) 191
5.2.3 Translation Table Base Register (TTB) 191
5.2.4 MMU Control Register (MMUCR) 191
5.3 TLB Functions 193
5.3.1 Configuration of the TLB 193
5.3.2 TLB Indexing 195
5.3.3 TLB Address Comparison 196
5.3.4 Page Management Information 198
5.4 MMU Functions 200
5.4.1 MMU Hardware Management 200
5.4.2 MMU Software Management 200
5.4.3 MMU Instruction (LDTLB) 201
5.4.4 Avoiding Synonym Problems 202
5.5 MMU Exceptions 205
5.5.1 TLB Miss Exception 205
5.5.2 TLB Protection Violation Exception 206
5.5.3 TLB Invalid Exception 207
5.5.4 Initial Page Write Exception 208
5.5.5 MMU Exception in Repeat Loop 209
5.6 Memory-Mapped TLB 211
5.6.1 Address Array 211

5.6.2 Data Array 211
5.6.3 Usage Examples 213
5.7 Usage Note 213

Section 6 Cache 215
6.1 Features 215
6.1.1 Cache Structure 215
6.2 Register Descriptions 217
6.2.1 Cache Control Register 1 (CCR1) 217
6.2.2 Cache Control Register 2 (CCR2) 218
6.2.3 Cache Control Register 3 (CCR3) 221
6.3 Operation 222
6.3.1 Searching the Cache 222
6.3.2 Read Access 223
6.3.3 Prefetch Operation 224
6.3.4 Write Access 224
6.3.5 Write-Back Buffer 224
6.3.6 Coherency of Cache and External Memory 225
6.4 Memory-Mapped Cache 226
6.4.1 Address Array 226
6.4.2 Data Array 227
6.4.3 Usage Examples 230

Section 7 X/Y Memory 231
7.1 Features 231
7.2 Operation 232
7.2.1 Access from CPU 232
7.2.2 Access from DSP 232
7.2.3 Access from DMAC, E-DMAC, and IPSEC 233
7.3 Usage Notes 233
7.3.1 Page Conflict 233
7.3.2 Bus Conflict 233
7.3.3 MMU and Cache Settings 233
7.3.4 Sleep Mode 234
7.3.5 Address Error 234

Section 8 Interrupt Controller (INTC) 235
8.1 Features 235
8.1.1 Block Diagram 235
8.2 Input/Output Pins 237

8.3 Interrupt Sources 237
8.3.1 NMI Interrupt 237
8.3.2 IRQ Interrupts 238
8.3.3 IRL Interrupts 238
8.3.4 On-Chip Peripheral Module Interrupts 239
8.3.5 Interrupt Exception Handling and Priority 240
8.4 Register Descriptions 246
8.4.1 Interrupt Priority Registers A to I (IPRA to IPRI) 246
8.4.2 Interrupt Control Register 0 (ICR0) 248
8.4.3 Interrupt Control Register 1 (ICR1) 249
8.4.4 Interrupt Request Register 0 (IRR0) 251
8.4.5 Interrupt Request Register 1 (IRR1) 251
8.4.6 Interrupt Request Register 2 (IRR2) 253
8.4.7 Interrupt Request Register 3 (IRR3) 254
8.4.8 Interrupt Request Register 4 (IRR4) 255
8.4.9 Interrupt Request Register 5 (IRR5) 256
8.4.10 Interrupt Request Register 7 (IRR7) 257
8.4.11 Interrupt Request Register 8 (IRR8) 258
8.5 Operation 260
8.5.1 Interrupt Sequence 260
8.5.2 Multiple Interrupts 262

Section 9 User Break Controller 263
9.1 Features 263
9.2 Register Descriptions 266
9.2.1 Break Address Register A (BARA) 266
9.2.2 Break Address Mask Register A (BAMRA) 267
9.2.3 Break Bus Cycle Register A (BBRA) 267
9.2.4 Break Address Register B (BARB) 269
9.2.5 Break Address Mask Register B (BAMRB) 270
9.2.6 Break Data Register B (BDRB) 270
9.2.7 Break Data Mask Register B (BDMRB) 271
9.2.8 Break Bus Cycle Register B (BBRB) 272
9.2.9 Break Control Register (BRCR) 274
9.2.10 Execution Times Break Register (BETR) 278
9.2.11 Branch Source Register (BRSR) 279
9.2.12 Branch Destination Register (BRDR) 280
9.2.13 Break ASID Register A (BASRA) 280
9.2.14 Break ASID Register B (BASRB) 281
9.3 Operation 281

9.3.1 Flow of the User Break Operation 281
9.3.2 Break on Instruction Fetch Cycle 283
9.3.3 Break on Data Access Cycle 283
9.3.4 Break on X/Y-Memory Bus Cycle 285
9.3.5 Sequential Break 285
9.3.6 Value of Saved Program Counter 286
9.3.7 PC Trace 287
9.3.8 Usage Examples 287
9.4 Usage Notes 292

Section 10 Power-Down Modes 295
10.1 Overview 295
10.1.1 Power-Down Modes 295
10.1.2 Reset 296
10.1.3 Input/Output Pins 298
10.2 Register Descriptions 298
10.2.1 Standby Control Register (STBCR) 298
10.2.2 Standby Control Register 2 (STBCR2) 300
10.2.3 Standby Control Register 3 (STBCR3) 301
10.3 Operation 302
10.3.1 Sleep Mode 302
10.3.2 Software Standby Mode 303
10.3.3 Module Standby Function 305
10.3.4 STATUS Pin Change Timings 306

Section 11 On-Chip Oscillation Circuits 311
11.1 Overview 311
11.1.1 Features 311
11.2 Overview of CPG 313
11.2.1 CPG Block Diagram 313
11.2.2 Input/Output Pins 315
11.3 Clock Operating Modes 315
11.4 Register Description 320
11.4.1 Frequency Control Register (FRQCR) 320
11.5 Changing Frequency 322
11.5.1 Changing Multiplication Rate 322
11.5.2 Changing Division Ratio 322
11.6 Overview of WDT 323
11.6.1 Block Diagram of WDT 323
11.7 Register Descriptions of WDT 324

11.7.1 Watchdog Timer Counter (WTCNT) ... 324
11.7.2 Watchdog Timer Control/Status Register (WTCSR) ... 324
11.7.3 Notes on Register Access ... 326
11.8 Using WDT ... 327
11.8.1 Canceling Standbys ... 327
11.8.2 Changing Frequency ... 328
11.8.3 Using Watchdog Timer Mode ... 328
11.8.4 Using Interval Timer Mode ... 328
11.9 Notes on Board Design ... 329

Section 12 Bus State Controller (BSC) ... 331
12.1 Features ... 331
12.2 Input/Output Pins ... 334
12.3 Area Overview ... 336
12.3.1 Area Division ... 336
12.3.2 Shadow Area ... 336
12.3.3 Address Map ... 338
12.3.4 Area 0 Memory Type and Memory Bus Width ... 340
12.3.5 Data Alignment ... 340
12.4 Register Descriptions ... 341
12.4.1 Common Control Register (CMNCR) ... 342
12.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B) ... 345
12.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B) ... 350
12.4.4 SDRAM Control Register (SDCR) ... 377
12.4.5 Refresh Timer Control/Status Register (RTCSR) ... 380
12.4.6 Refresh Timer Counter (RTCNT) ... 381
12.4.7 Refresh Time Constant Register (RTCOR) ... 382
12.5 Operation ... 383
12.5.1 Endian/Access Size and Data Alignment ... 383
12.5.2 Normal Space Interface ... 389
12.5.3 Access Wait Control ... 395
12.5.4 $\overline{\text{CSn}}$ Assert Period Expansion ... 397
12.5.5 SDRAM Interface ... 398
12.5.6 Burst ROM (Clock Asynchronous) Interface ... 439
12.5.7 Byte-Selection SRAM Interface ... 441
12.5.8 PCMCIA Interface ... 446
12.5.9 Burst ROM (Clock Synchronous) Interface ... 452
12.5.10 Wait between Access Cycles ... 453
12.5.11 Bus Arbitration ... 453
12.5.12 Others ... 457

Section 13 Direct Memory Access Controller (DMAC) 459
13.1 Features 459
13.2 Input/Output Pins 461
13.3 Register Descriptions 462
13.3.1 DMA Source Address Register (SAR) 463
13.3.2 DMA Destination Address Register (DAR) 463
13.3.3 DMA Transfer Count Register (DMATCR) 464
13.3.4 DMA Channel Control Register (CHCR) 464
13.3.5 DMA Operation Register (DMAOR) 469
13.3.6 DMA Extension Resource Selector 0 to 2 (DMARS0 to DMARS2) 471
13.4 Operation 474
13.4.1 DMA Transfer Flow 474
13.4.2 DMA Transfer Requests 477
13.4.3 Channel Priority 479
13.4.4 DMA Transfer Types 482
13.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing 489
13.5 Usage Note 493

Section 14 Timer Unit (TMU) 495
14.1 Features 495
14.1.1 Block Diagram 495
14.2 Register Descriptions 497
14.2.1 Timer Start Register (TSTR) 497
14.2.2 Timer Control Registers (TCR) 498
14.2.3 Timer Constant Registers (TCOR) 499
14.2.4 Timer Counters (TCNT) 499
14.3 TMU Operation 500
14.3.1 Counter Operation 500
14.4 Interrupts 503
14.4.1 Status Flag Set Timing 503
14.4.2 Status Flag Clear Timing 503
14.4.3 Interrupt Sources and Priorities 504
14.5 Usage Notes 504
14.5.1 Writing to Registers 504
14.5.2 Reading Registers 504

Section 15 Realtime Clock (RTC) 505
15.1 Feature 505
15.2 Input/Output Pins 507
15.3 Register Descriptions 507

15.3.1 64-Hz Counter (R64CNT) 508
15.3.2 Second Counter (RSECCNT) 508
15.3.3 Minute Counter (RMINCNT) 509
15.3.4 Hour Counter (RHRCNT) 509
15.3.5 Day of Week Counter (RWKCNT) 510
15.3.6 Date Counter (RDAYCNT) 512
15.3.7 Month Counter (RMONCNT) 512
15.3.8 Year Counter (RYRCNT) 513
15.3.9 Second Alarm Register (RSECAR) 514
15.3.10 Minute Alarm Register (RMINAR) 514
15.3.11 Hour Alarm Register (RHRAR) 515
15.3.12 Day of Week Alarm Register (RWKAR) 516
15.3.13 Date Alarm Register (RDAYAR) 517
15.3.14 Month Alarm Register (RMONAR) 518
15.3.15 Year Alarm Register (RYRAR) 519
15.3.16 RTC Control Register 1 (RCR1) 520
15.3.17 RTC Control Register 2 (RCR2) 522
15.3.18 RTC Control Register 3 (RCR3) 524
15.4 Operation 525
15.4.1 Initial Settings of Registers after Power-On 525
15.4.2 Setting Time 525
15.4.3 Reading Time 525
15.4.4 Alarm Function 527
15.4.5 Crystal Oscillator Circuit 528
15.5 Usage Notes 529
15.5.1 Register Writing during RTC Count 529
15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts 529
15.5.3 Transition to Standby Mode after Setting Register 529
15.5.4 Usage Note about RTC Power Supply 530

Section 16 Serial Communication Interface with FIFO (SCIF) 531
16.1 Features 531
16.2 Input/Output Pins 534
16.3 Register Descriptions 535
16.3.1 Receive Shift Register (SCRSR) 536
16.3.2 Receive FIFO Data Register (SCFRDR) 536
16.3.3 Transmit Shift Register (SCTSR) 537
16.3.4 Transmit FIFO Data Register (SCFTDR) 537
16.3.5 Serial Mode Register (SCSMR) 537
16.3.6 Serial Control Register (SCSCR) 541

16.3.7 Serial Status Register (SCFSR) 545
16.3.8 Bit Rate Register (SCBRR) 553
16.3.9 FIFO Control Register (SCFCR) 554
16.3.10 FIFO Data Count Register (SCFDR) 556
16.3.11 Line Status Register (SCLSR) 558
16.4 Operation 559
16.4.1 Overview 559
16.4.2 Serial Operation in Asynchronous Mode 561
16.4.3 Serial Operation in Clock Synchronous Mode 572
16.5 SCIF Interrupt Sources and DMAC 582
16.6 Usage Notes 583

Section 17 Serial I/O with FIFO (SIOF) 587
17.1 Features 587
17.1.1 Block Diagram 588
17.2 Input/Output Pins 589
17.3 Register Descriptions 590
17.3.1 SIOF Mode Register (SIMDR) 591
17.3.2 Serial Clock Select Register (SISCR) 593
17.3.3 Serial Transmit Data Assign Register (SITDAR) 594
17.3.4 Serial Receive Data Assign Register (SIRDAR) 595
17.3.5 Serial Control Data Assign Register (SICDAR) 596
17.3.6 SIOF Control Register (SICTR) 598
17.3.7 SIOF FIFO Control Register (SIFCTR) 601
17.3.8 SIOF Status Register (SISTR) 603
17.3.9 SIOF Interrupt Enable Register (SIIER) 607
17.3.10 Serial Transmit Data Register (SITDR) 609
17.3.11 Serial Receive Data Register (SIRDR) 610
17.3.12 Serial Transmit Control Data Register (SITCR) 611
17.3.13 Serial Receive Control Data Register (SIRCR) 612
17.4 Operation 613
17.4.1 Serial Clocks 613
17.4.2 Serial Timing 614
17.4.3 Transfer Data Format 616
17.4.4 Register Allocation of Transfer Data 617
17.4.5 Control Data Interface 620
17.4.6 FIFO 621
17.4.7 Transmission and Reception Procedures 623
17.4.8 Interrupts 628
17.4.9 Transmission and Reception Timing 630

17.5 Usage Notes 635

Section 18 Ethernet Controller (EtherC) 637
18.1 Features 637
18.2 Input/Output Pins 639
18.3 Register Descriptions 641
18.3.1 Software Reset Register (ARSTR) 644
18.3.2 EtherC Mode Register (ECMR) 645
18.3.3 EtherC Status Register (ECSR) 648
18.3.4 EtherC Interrupt Permission Register (ECSIPR) 649
18.3.5 PHY Interface Register (PIR) 650
18.3.6 MAC Address High Register (MAHR) 651
18.3.7 MAC Address Low Register (MALR) 651
18.3.8 Receive Frame Length Register (RFLR) 652
18.3.9 PHY Status Register (PSR) 653
18.3.10 Transmit Retry Over Counter Register (TROCR) 653
18.3.11 Delayed Collision Detect Counter Register (CDCR) 654
18.3.12 Lost Carrier Counter Register (LCCR) 654
18.3.13 Carrier Not Detect Counter Register (CNDCR) 654
18.3.14 CRC Error Frame Receive Counter Register (CEFCR) 655
18.3.15 Frame Receive Error Counter Register (FRECR) 655
18.3.16 Too-Short Frame Receive Counter Register (TSFRCR) 655
18.3.17 Too-Long Frame Receive Counter Register (TLFRCR) 656
18.3.18 Residual-Bit Frame Receive Counter Register (RFCR) 656
18.3.19 Multicast Address Frame Receive Counter Register (MAFCR) 657
18.3.20 IPG Register (IPGR) 657
18.3.21 TSU Counter Reset Register (TSU_CTRST) 658
18.3.22 Relay Enable Register (Port 0 to 1) (TSU_FWEN0) 658
18.3.23 Relay Enable Register (Port 1 to 0) (TSU_FWEN1) 659
18.3.24 Relay FIFO Size Select Register (TSU_FCM) 660
18.3.25 Relay FIFO Overflow Alert Set Register (Port 0) (TSU_BSYSL0) 661
18.3.26 Relay FIFO Overflow Alert Set Register (Port 1) (TSU_BSYSL1) 662
18.3.27 Transmit/Relay Priority Control Mode Register (Port 0) (TSU_PRISL0) 663
18.3.28 Transmit/Relay Priority Control Mode Register (Port 1) (TSU_PRISL1) 664
18.3.29 Receive/Relay Function Set Register (Port 0 to 1) (TSU_FWSL0) 666
18.3.30 Receive/Relay Function Set Register (Port 1 to 0) (TSU_FWSL1) 667
18.3.31 Relay Function Set Register (Common) (TSU_FWSLC) 669
18.3.32 Qtag Addition/Deletion Set Register (Port 0 to 1) (TSU_QTAGM0) 671
18.3.33 Qtag Addition/Deletion Set Register (Port 1 to 0) (TSU_QTAGM1) 672
18.3.34 Relay Status Register (TSU_FWSR) 673

18.3.35 Relay Status Interrupt Mask Register (TSU_FWINMK) 675
18.3.36 Added Qtag Value Set Register (Port 0 to 1) (TSU_ADQT0) 679
18.3.37 Added Qtag Value Set Register (Port 1 to 0) (TSU_ADQT1) 680
18.3.38 CAM Entry Table Busy Register (TSU_ADSBSY) 681
18.3.39 CAM Entry Table Enable Register (TSU_TEN) 682
18.3.40 CAM Entry Table POST1 Register (TSU_POST1) 686
18.3.41 CAM Entry Table POST2 Register (TSU_POST2) 689
18.3.42 CAM Entry Table POST3 Register (TSU_POST3) 692
18.3.43 CAM Entry Table POST4 Register (TSU_POST4) 695
18.3.44 CAM Entry Table 0 to 31 H Registers (TSU_ADRH0 to TSU_ADRH31) 698
18.3.45 CAM Entry Table 0 to 31 L Registers (TSU_ADRL0 to TSU_ADRL31) 699
18.3.46 Transmit Frame Counter Register (Port 0) (Normal Transmission Only) (TXNLCR0) 699
18.3.47 Transmit Frame Counter Register (Port 0) (Normal and Error Transmission) (TXALCR0) 700
18.3.48 Receive Frame Counter Register (Port 0) (Normal Reception Only) (RXNLCR0) 700
18.3.49 Receive Frame Counter Register (Port 0) (Normal and Error Reception) (RXALCR0) 701
18.3.50 Relay Frame Counter Register (Port 1 to 0) (Normal Relay Only) (FWNLCR0) 701
18.3.51 Relay Frame Counter Register (Port 1 to 0) (Normal and Error Relay) (FWALCR0) 702
18.3.52 Transmit Frame Counter Register (Port 1) (Normal Transmission Only) (TXNLCR1) 702
18.3.53 Transmit Frame Counter Register (Port 1) (Normal and Error Transmission) (TXALCR1) 703
18.3.54 Receive Frame Counter Register (Port 1) (Normal Reception Only) (RXNLCR1) 703
18.3.55 Receive Frame Counter Register (Port 1) (Normal and Error Reception) (RXALCR1) 704
18.3.56 Relay Frame Counter Register (Port 0 to 1) (Normal Relay Only) (FWNLCR1) 704
18.3.57 Relay Frame Counter Register (Port 0 to 1) (Normal and Error Relay) (FWALCR1) 705
18.4 Operation 706
18.4.1 Transmission 707
18.4.2 Reception 709
18.4.3 Relay 711
18.4.4 CAM Function 711

18.4.5 MII Frame Timing 717
18.4.6 Accessing MII Registers 719
18.4.7 Magic Packet Detection 722
18.4.8 Operation by IPG Setting 723
18.4.9 Direction for IEEE802.1Q Qtag 723
18.5 Connection to LSI 725

Section 19 Ethernet Controller Direct Memory Access Controller (E-DMAC) 727
19.1 Features 727
19.2 Register Descriptions 728
19.2.1 E-DMAC Mode Register (EDMR) 730
19.2.2 E-DMAC Transmit Request Register (EDTRR) 731
19.2.3 E-DMAC Receive Request Register (EDRRR) 732
19.2.4 Transmit Descriptor List Address Register (TDLAR) 733
19.2.5 Receive Descriptor List Address Register (RDLAR) 734
19.2.6 EtherC/E-DMAC Status Register (EESR) 734
19.2.7 EtherC/E-DMAC Status Interrupt Permission Register (EESIPR) 740
19.2.8 Transmit/Receive Status Copy Enable Register (TRSCER) 743
19.2.9 Receive Missed-Frame Counter Register (RMFCR) 744
19.2.10 Transmit FIFO Threshold Register (TFTR) 745
19.2.11 FIFO Depth Register (FDR) 747
19.2.12 Receiving Method Control Register (RMCR) 748
19.2.13 E-DMAC Operation Control Register (EDOCR) 749
19.2.14 Receive Buffer Write Address Register (RBWAR) 750
19.2.15 Receive Descriptor Fetch Address Register (RDFAR) 750
19.2.16 Transmit Buffer Read Address Register (TBRAR) 750
19.2.17 Transmit Descriptor Fetch Address Register (TDFAR) 751
19.2.18 Overflow Alert FIFO Threshold Register (FCFTR) 751
19.2.19 Transmit Interrupt Register (TRIMD) 753
19.3 Operation 753
19.3.1 Descriptors and Descriptor List 754
19.3.2 Transmission 767
19.3.3 Reception 769
19.3.4 Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/ Multi-Descriptor) 771
19.3.5 Receive FIFO Overflow Alert Signal ($\overline{\text{ARBUSY}}$) 773
19.4 Usage Notes 776
19.4.1 Using of EDTRR and EDRRR 776
19.4.2 Endian Support in E-DMAC 777

Section 20 IP Security Accelerator (IPSEC) 779

Section 21 Pin Function Controller (PFC) 781
21.1 Overview 781
21.2 Register Configuration 782
21.3 Register Descriptions 783
21.3.1 Port A Control Register (PACR) 783
21.3.2 Port B Control Register (PBCR) 784
21.3.3 Port C Control Register (PCCR) 785
21.3.4 Ethernet Controller Pin Control Register (PETCR) 786

Section 22 I/O Ports 789
22.1 Overview 789
22.2 Register Descriptions 789
22.2.1 Port A Data Register (PADR) 789
22.2.2 Port B Data Register (PBDR) 790
22.2.3 Port C Data Register (PCDR) 792

Section 23 User Debugging Interface (H-UDI) 793
23.1 Features 793
23.2 Input/Output Pins 794
23.3 Register Descriptions 795
23.3.1 Bypass Register (SDBPR) 795
23.3.2 Instruction Register (SDIR) 795
23.3.3 Boundary Scan Register (SDBSR) 796
23.3.4 ID Register (SDID) 803
23.4 Operation 804
23.4.1 TAP Controller 804
23.4.2 Reset Configuration 805
23.4.3 TDO Output Timing 805
23.4.4 H-UDI Reset 806
23.4.5 H-UDI Interrupt 806
23.5 Boundary Scan 807
23.5.1 Supported Instructions 807
23.5.2 Points for Attention 808
23.6 Usage Notes 808
23.7 Advanced User Debugger (AUD) 808

Section 24 List of Registers 809
24.1 Register Addresses
(by functional module, in order of the corresponding section numbers) 810
24.2 Register Bits 823
24.3 Register States in Each Operating Mode 846

Section 25 Electrical Characteristics 855
25.1 Absolute Maximum Ratings 855
25.2 DC Characteristics 857
25.3 AC Characteristics 859
25.3.1 Clock Timing 860
25.3.2 Control Signal Timing 865
25.3.3 AC Bus Timing 868
25.3.4 Basic Timing 870
25.3.5 Burst ROM Timing 874
25.3.6 Synchronous DRAM Timing 875
25.3.7 DMAC Signal Timing 901
25.3.8 RTC Signal Timing 902
25.3.9 SCIF Module Signal Timing 903
25.3.10 SIOF Module Signal Timing 904
25.3.11 Ethernet Controller Timing 908
25.3.12 Port Input/Output Timing 912
25.3.13 H-UDI Related Pin Timing 913
25.3.14 AC Characteristics Measurement Conditions 915
25.4 Delay Time Variation Due to Load Capacitance 916

Appendix 917
A. Pin States and States of Unused Pins 917
B. Package Dimensions 925

Main Revisions and Additions in this Edition 929

Index 947

Figures

Section 1 Overview
Figure 1.1 Block Diagram 7
Figure 1.2 Pin Assignment (HQFP2828-256 (FP-256G/GV)) 8
Figure 1.3 Pin Assignment (P-LFBGA1717-256 (BP-256H/HV)) 9

Section 2 CPU
Figure 2.1 Processing State Transitions 28
Figure 2.2 Logical Address to External Memory Space Mapping 31
Figure 2.3 Register Configuration in Each Processing Mode 34
Figure 2.4 General Registers 36
Figure 2.5 System Registers and Program Counter 37
Figure 2.6 Control Register Configuration 41
Figure 2.7 Data Format on Memory (Big Endian Mode) 43
Figure 2.8 Data Format on Memory (Little Endian Mode) 43

Section 3 DSP Operating Unit
Figure 3.1 DSP Instruction Format 72
Figure 3.2 CPU Registers in DSP Mode 74
Figure 3.3 DSP Register Configuration 77
Figure 3.4 DSP Registers and Bus Connections 94
Figure 3.5 General Registers (DSP Mode) 97
Figure 3.6 Sample Parallel Instruction Program 113
Figure 3.7 Examples of Conditional Operations and Data Transfer Instructions 115
Figure 3.8 Data Formats 117
Figure 3.9 ALU Fixed-Point Arithmetic Operation Flow 118
Figure 3.10 Operation Sequence Example 120
Figure 3.11 DC Bit Generation Examples in Carry or Borrow Mode 121
Figure 3.12 DC Bit Generation Examples in Negative Value Mode 121
Figure 3.13 DC Bit Generation Examples in Overflow Mode 122
Figure 3.14 ALU Integer Arithmetic Operation Flow 123
Figure 3.15 ALU Logical Operation Flow 125
Figure 3.16 Fixed-Point Multiply Operation Flow 127
Figure 3.17 Arithmetic Shift Operation Flow 129
Figure 3.18 Logical Shift Operation Flow 131
Figure 3.19 PDMSB Operation Flow 133
Figure 3.20 Rounding Operation Flow 136
Figure 3.21 Definition of Rounding Operation 136
Figure 3.22 Local Data Move Instruction Flow 138

Section 4 Exception Handling
Figure 4.1 Register Bit Configuration 156

Section 5 Memory Management Unit (MMU)
Figure 5.1 MMU Functions 183
Figure 5.2 Virtual Address Space (MMUCR.AT = 1) 185
Figure 5.3 Virtual Address Space (MMUCR.AT = 0) 186
Figure 5.4 P4 Area 187
Figure 5.5 External Memory Space 188
Figure 5.6 Overall Configuration of the TLB 193
Figure 5.7 Virtual Address and TLB Structure 194
Figure 5.8 TLB Indexing (IX = 1) 195
Figure 5.9 TLB Indexing (IX = 0) 196
Figure 5.10 Objects of Address Comparison 197
Figure 5.11 Operation of LDTLB Instruction 202
Figure 5.12 Synonym Problem (32-kbyte Cache) 204
Figure 5.13 MMU Exception Generation Flowchart 210
Figure 5.14 Specifying Address and Data for Memory-Mapped TLB Access 212

Section 6 Cache
Figure 6.1 Cache Structure 215
Figure 6.2 Cache Search Scheme 223
Figure 6.3 Write-Back Buffer Configuration 225
Figure 6.4 Specifying Address and Data for Memory-Mapped Cache Access (16 kbytes mode) 228
Figure 6.5 Specifying Address and Data for Memory-Mapped Cache Access (32 kbytes mode) 229

Section 8 Interrupt Controller (INTC)
Figure 8.1 Block Diagram of INTC 236
Figure 8.2 Example of IRL Interrupt Connection 239
Figure 8.3 Interrupt Operation Flowchart 261

Section 9 User Break Controller
Figure 9.1 Block Diagram of User Break Controller 265

Section 10 Power-Down Modes
Figure 10.1 Canceling Standby Mode with STBCR.STBY 305
Figure 10.2 STATUS Output at Power-On Reset 306
Figure 10.3 STATUS Output at Manual Reset 307
Figure 10.4 STATUS Output when Software Standby Mode is Canceled by Interrupt 307
Figure 10.5 STATUS Output when Software Standby Mode is Canceled by Power-on Reset 308
Figure 10.6 STATUS Output when Software Standby Mode is Canceled by Manual Reset 308

Figure 10.7 STATUS Output when Sleep Mode is Canceled by Interrupt 309
Figure 10.8 STATUS Output when Sleep Mode is Canceled by Power-on Reset 309
Figure 10.9 STATUS Output when Sleep Mode is Canceled by Manual Reset 310

Section 11 On-Chip Oscillation Circuits

Figure 11.1 Block Diagram of CPG 313
Figure 11.2 Block Diagram of WDT 323
Figure 11.3 Writing to WTCNT and WTCSR 327
Figure 11.4 Points for Attention when Using Crystal Resonator 329
Figure 11.5 Points for Attention when Using PLL Oscillator Circuit 330

Section 12 Bus State Controller (BSC)

Figure 12.1 Block Diagram of BSC 333
Figure 12.2 Address Space 337
Figure 12.3 Normal Space Basic Access Timing (Access Wait 0) 389
Figure 12.4 Continuous Access for Normal Space 1, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0) 391
Figure 12.5 Continuous Access for Normal Space 2, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0) 392
Figure 12.6 Example of 32-Bit Data-Width SRAM Connection 393
Figure 12.7 Example of 16-Bit Data-Width SRAM Connection 394
Figure 12.8 Example of 8-Bit Data-Width SRAM Connection 394
Figure 12.9 Wait Timing for Normal Space Access (Software Wait Only) 395
Figure 12.10 Wait State Timing for Normal Space Access (Wait State Insertion by $\overline{\text{WAIT}}$ Signal) 396
Figure 12.11 $\overline{\text{CSn}}$ Assert Period Expansion 397
Figure 12.12 Example of 32-Bit Data-Width SDRAM Connection 399
Figure 12.13 Example of 16-Bit Data-Width SDRAM Connection 400
Figure 12.14 Burst Read Basic Timing (Auto Precharge) 415
Figure 12.15 Burst Read Wait Specification Timing (Auto Precharge) 416
Figure 12.16 Basic Timing for Single Read (Auto Precharge) 417
Figure 12.17 Basic Timing for Burst Write (Auto Precharge) 419
Figure 12.18 Basic Timing for Single Write (Auto-Precharge) 420
Figure 12.19 Burst Read Timing (No Auto Precharge) 422
Figure 12.20 Burst Read Timing (Bank Active, Same Row Address) 423
Figure 12.21 Burst Read Timing (Bank Active, Different Row Addresses) 424
Figure 12.22 Single Write Timing (No Auto Precharge) 425
Figure 12.23 Single Write Timing (Bank Active, Same Row Address) 426
Figure 12.24 Single Write Timing (Bank Active, Different Row Addresses) 427
Figure 12.25 Auto-Refresh Timing 429
Figure 12.26 Self-Refresh Timing 431

Figure 12.27 Access Timing in Low-Frequency Mode 432
Figure 12.28 Access Timing in Power-Down Mode 433
Figure 12.29 Write Timing for SDRAM Mode Register (Based on JEDEC) 436
Figure 12.30 EMRS Command Issue Timing 438
Figure 12.31 Transition Timing in Deep Power-Down Mode 439
Figure 12.32 Burst ROM (Clock Asynchronous) Access (Bus Width = 32 Bits, 16-byte Transfer (Number of Bursts = 4), Access Wait for First Time = 2, Access Wait for 2nd Time and after = 1) 441
Figure 12.33 Basic Access Timing for Byte-Selection SRAM (BAS = 0) 442
Figure 12.34 Basic Access Timing for Byte-Selection SRAM (BAS = 1) 443
Figure 12.35 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only) 444
Figure 12.36 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM 445
Figure 12.37 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM 445
Figure 12.38 Example of PCMCIA Interface Connection 447
Figure 12.39 Basic Access Timing for PCMCIA Memory Card Interface 448
Figure 12.40 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait = 1) 448
Figure 12.41 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B′10, CS6BWCR.SA[1:0] = B′10) 449
Figure 12.42 Basic Timing for PCMCIA I/O Card Interface 450
Figure 12.43 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait = 1) 451
Figure 12.44 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Waits = 3) 451
Figure 12.45 Burst ROM (Clock Synchronous) Access Timing (Burst Length = 8, Wait Cycles inserted in First Access = 2, Wait Cycles inserted in Second and Subsequent Accesses = 1) 452
Figure 12.46 Bus Arbitration Timing 456

Section 13 Direct Memory Access Controller (DMAC)

Figure 13.1 Block Diagram of DMAC 460
Figure 13.2 DMA Transfer Flowchart 476
Figure 13.3 Round-Robin Mode 480
Figure 13.4 Changes in Channel Priority in Round-Robin Mode 481
Figure 13.5 Data Flow in Dual Address Mode 483
Figure 13.6 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary memory, Destination: Ordinary memory) 484
Figure 13.7 Data Flow in Single Address Mode 485
Figure 13.8 Example of DMA Transfer Timing in Single Address Mode 486
Figure 13.9 DMA Transfer Example in Cycle-Steal Mode (Dual Address, DREQ Low Level Detection) 487

Figure 13.10 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection) 487
Figure 13.11 Bus State when Multiple Channels are Operating 489
Figure 13.12 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection 490
Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Level Detection 490
Figure 13.14 Example of DREQ Input Detection in Burst Mode Edge Detection 491
Figure 13.15 Example of DREQ Input Detection in Burst Mode Level Detection 491
Figure 13.16 Example of DMA Transfer End Timing (Cycle Steal Level Detection) 491
Figure 13.17 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle = 1, Longword Access to 16-bit Device) 492

Section 14 Timer Unit (TMU)

Figure 14.1 TMU Block Diagram 496
Figure 14.2 Setting Count Operation 501
Figure 14.3 Auto-Reload Count Operation 501
Figure 14.4 Count Timing when Internal Clock Is Operating 502
Figure 14.5 UNF Set Timing 503
Figure 14.6 Status Flag Clear Timing 503

Section 15 Realtime Clock (RTC)

Figure 15.1 RTC Block Diagram 506
Figure 15.2 Setting Time 525
Figure 15.3 Reading Time 526
Figure 15.4 Using Alarm Function 527
Figure 15.5 Example of Crystal Oscillator Circuit Connection 528
Figure 15.6 Using Periodic Interrupt Function 529

Section 16 Serial Communication Interface with FIFO (SCIF)

Figure 16.1 Block Diagram of SCIF 533
Figure 16.2 Data Format in Asynchronous Communication (Example of 8-Bit Data with Parity and 2 Stop Bits) 561
Figure 16.3 Sample the SCIF Initialization Flowchart 564
Figure 16.4 Sample Serial Transmission Flowchart 565
Figure 16.5 Example of Transmit Operation (Example of 8-Bit Data with Parity and 1 Stop Bit) 567
Figure 16.6 Sample Serial Reception Flowchart (1) 568
Figure 16.7 Sample Serial Reception Flowchart (2) 569
Figure 16.8 Example of SCIF Receive Operation (Example of 8-Bit Data with Parity and 1 Stop Bit) 571
Figure 16.9 $\overline{\text{CTS}}$ Control Operation 571
Figure 16.10 $\overline{\text{RTS}}$ Control Operation 572
Figure 16.11 Data Format in Clock Synchronous Communication 572

Figure 16.12 Sample the SCIF Initialization Flowchart 574
Figure 16.13 Sample Serial Transmission Flowchart 575
Figure 16.14 Example of the SCIF Transmit Operation 576
Figure 16.15 Sample Serial Reception Flowchart 577
Figure 16.16 Sample Serial Reception Flowchart 578
Figure 16.17 Example of the SCIF Receive Operation 579
Figure 16.18 Sample Serial Data Transmission/Reception Flowchart 581
Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode 584
Figure 16.20 Sample Transfer of Synchronous Clock by DMAC 585

Section 17 Serial I/O with FIFO (SIOF)

Figure 17.1 Block Diagram of SIOF 588
Figure 17.2 Serial Clock Supply 613
Figure 17.3 Serial Data Synchronization Timing 615
Figure 17.4 SIOF Transmit/Receive Timing 616
Figure 17.5 Transmit/Receive Data Bit Alignment 618
Figure 17.6 Control Data Bit Alignment 619
Figure 17.7 Control Data Interface (Slot Position) 620
Figure 17.8 Control Data Interface (Secondary FS) 621
Figure 17.9 Example of Transmission Operation in Master Mode 624
Figure 17.10 Example of Reception Operation in Master Mode 625
Figure 17.11 Example of Transmission Operation in Slave Mode 626
Figure 17.12 Example of Reception Operation in Slave Mode 627
Figure 17.13 Transmission and Reception Timings (8-Bit Monaural Data (1)) 631
Figure 17.14 Transmission and Reception Timings (8-Bit Monaural Data (2)) 631
Figure 17.15 Transmission and Reception Timings (16-Bit Monaural Data (1)) 632
Figure 17.16 Transmission and Reception Timings (16-Bit Stereo Data (1)) 632
Figure 17.17 Transmission and Reception Timings (16-Bit Stereo Data (2)) 633
Figure 17.18 Transmission and Reception Timings (16-Bit Stereo Data (3)) 633
Figure 17.19 Transmission and Reception Timings (16-Bit Monaural Data (2)) 634

Section 18 Ethernet Controller (EtherC)

Figure 18.1 Configuration of EtherC 638
Figure 18.2 EtherC Data Path and Various Settings 707
Figure 18.3 EtherC Transmitter State Transitions 708
Figure 18.4 EtherC Receiver State Transmissions 710
Figure 18.5 Example of External CAM Connection 714
Figure 18.6 External CAM Signal Timing 716
Figure 18.7 (1) MII Frame Transmit Timing (Normal Transmission) 717
Figure 18.7 (2) MII Frame Transmit Timing (Collision) 717
Figure 18.7 (3) MII Frame Transmit Timing (Transmit Error) 718

Figure 18.7 (4) MII Frame Receive Timing (Normal Reception) 718
Figure 18.7 (5) MII Frame Receive Timing (Reception Error (1)) 718
Figure 18.7 (6) MII Fame Receive Timing (Reception Error (2)) 718
Figure 18.8 MII Management Frame Format 719
Figure 18.9 (1) 1-Bit Data Write Flowchart 720
Figure 18.9 (2) Bus Release Flowchart (TA in Read in Figure 18.8) 721
Figure 18.9 (3) 1-Bit Data Read Flowchart 721
Figure 18.9 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 18.8) 722
Figure 18.10 Changing IPG and Transmission Efficiency 723
Figure 18.11 Diagram of Qtag Additional Functions 724
Figure 18.12 Comparison of Normal Ethernet Frame and IEEE802.1Q Frame (with Qtag) 724
Figure 18.13 Example of Connection to DP83847 725

Section 19 Ethernet Controller Direct Memory Access Controller (E-DMAC)

Figure 19.1 Configuration of E-DMAC, and Descriptors and Buffers 728
Figure 19.2 Relationship between Transmit Descriptor and Transmit Buffer 755
Figure 19.3 Relationship between Receive Descriptor and Receive Buffer 761
Figure 19.4 Sample Transmission Flowchart (Single-Frame/Two-Descriptor) 768
Figure 19.5 Sample Reception Flowchart (Single-Frame/Two-Descriptor) 770
Figure 19.6 E-DMAC Operation after Transmit Error 771
Figure 19.7 E-DMAC Operation after Receive Error 772
Figure 19.8 Configuration of $\overline{\text{ARBUSY}}$ 773
Figure 19.9 Summary of Receive FIFO Overflow Alert Signal 774
Figure 19.10 $\overline{\text{ARBUSY}}$ Signal Change and Minimum Pulse Width Depending on Increase and Decrease of FIFO 775

Section 23 User Debugging Interface (H-UDI)

Figure 23.1 Block Diagram of H-UDI 793
Figure 23.2 TAP Controller State Transitions 804
Figure 23.3 H-UDI Data Transfer Timing 806
Figure 23.4 H-UDI Reset 806

Section 25 Electrical Characteristics

Figure 25.1 Power On/Off Sequence 856
Figure 25.2 EXTAL Clock Input Timing 861
Figure 25.3 CKIO Clock Input Timing 861
Figure 25.4 CKIO Clock Output Timing 862
Figure 25.5 Power-On Oscillation Settling Time 862
Figure 25.6 Oscillation Settling Time at Standby Return (Return by Reset) 862
Figure 25.7 Oscillation Settling Time at Standby Return (Return by NMI) 863
Figure 25.8 Oscillation Settling Time at Standby Return (Return by IRQ5 to IRQ0 and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$) 863

Figure 25.9 PLL Synchronization Settling Time by Reset or NMI 863
Figure 25.10 PLL Synchronization Settling Time by IRQ/IRL Interrupts 864
Figure 25.11 PLL Synchronization Settling Time when Frequency Multiplication Ratio Modified 864
Figure 25.12 Reset Input Timing 866
Figure 25.13 Interrupt Signal Input Timing 866
Figure 25.14 Bus Release Timing 866
Figure 25.15 Pin Drive Timing at Standby 867
Figure 25.16 $\overline{\text{IRQOUT}}$ Output Delay Time 867
Figure 25.17 Basic Bus Cycle (No Wait) 870
Figure 25.18 Basic Bus Cycle (One Software Wait) 871
Figure 25.19 Basic Bus Cycle (One External Wait) 872
Figure 25.20 Basic Bus Cycle (One Software Wait, External Wait Enabled (WM bit = 0), No Idle Cycle Setting) 873
Figure 25.21 Burst ROM Read Cycle (One Access Wait, One External Wait, One Burst Wait, Two Bursts) 874
Figure 25.22 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 1 Cycle) 875
Figure 25.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 2 Cycle) 876
Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4), (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 2 Cycle) 877
Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4), (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 1 Cycle) 878
Figure 25.26 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 2 Cycle) 879
Figure 25.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRCD = 3 Cycle, TRWL = 2 Cycle) 880
Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4), (Auto Precharge, TRCD = 1 Cycle, TRWL = 2 Cycle) 881
Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4), (Auto Precharge, TRCD = 2 Cycle, TRWL = 2 Cycle) 882
Figure 25.30 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode, ACTV + READ Commands, CAS Latency = 2, TRCD = 1 Cycle) 883
Figure 25.31 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode, READ Command, Same Row Address, CAS Latency = 2, TRCD = 1 Cycle) 884

Figure 25.32 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode, PRE + ACTV + READ Commands,
Different Row Address, CAS Latency = 2, TRCD = 1 Cycle) 885
Figure 25.33 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4)
(Bank Active Mode, ACTV + WRITE Commands, TRCD = 1 Cycle,
TRWL = 1 Cycle) 886
Figure 25.34 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4)
(Bank Active Mode, WRITE Command, Same Row Address, TRCD = 1 Cycle,
TRWL = 1 Cycle) 887
Figure 25.35 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4)
(Bank Active Mode, PRE + ACTV + WRITE Commands,
Different Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle) 888
Figure 25.36 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle) 889
Figure 25.37 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle) 890
Figure 25.38 Synchronous DRAM Mode Register Write Timing (TRP = 2 Cycle) 891
Figure 25.39 PCMCIA Memory Card Interface Bus Timing 892
Figure 25.40 PCMCIA Memory Card Interface Bus Timing (TED[3:0] = B'0010,
TEH[3:0] = B'0001, One Software Wait, One Hardware Wait) 893
Figure 25.41 PCMCIA I/O Card Interface Bus Timing 894
Figure 25.42 PCMCIA I/O Card Interface Bus Timing (TED[3:0] = B'0010,
TEH[3:0] = B'0001, One Software Wait, One Hardware Wait) 895
Figure 25.43 $\overline{\text{REFOUT}}$ Delay Time 895
Figure 25.44 Access Timing in Low-Frequency Mode (Auto Precharge) 897
Figure 25.45 Synchronous DRAM Auto-Refresh Timing
(TRP = 2 Cycle, Low-Frequency Mode) 898
Figure 25.46 Synchronous DRAM Self-Refresh Timing
(TRP = 2 Cycle, Low-Frequency Mode) 899
Figure 25.47 Synchronous DRAM Mode Register Write Timing
(TRP = 2 Cycle, Low-Frequency Mode) 900
Figure 25.48 DREQn Input Timing 901
Figure 25.49 TENDn, DACKn Output Timing 901
Figure 25.50 Oscillation Settling Time when RTC Crystal Oscillator is Turned On 902
Figure 25.51 SCIFnCK Input Clock Timing 903
Figure 25.52 SCIF Input/Output Timing in Clock Synchronous Mode 904
Figure 25.53 SIOMCLK Input Timing 905
Figure 25.54 SIOF Transmit/Receive Timing (Master Mode 1: Fall Sampling Time) 905
Figure 25.55 SIOF Transmit/Receive Timing (Master Mode 1: Rise Sampling Time) 906
Figure 25.56 SIOF Transmit/Receive Timing (Master Mode 2: Fall Sampling Time) 906
Figure 25.57 SIOF Transmit/Receive Timing (Master Mode 2: Rise Sampling Time) 907
Figure 25.58 SIOF Transmit/Receive Timing (Slave Mode 1 and Slave Mode 2) 907

Figure 25.59 MII Transmit Timing (Normal Operation) 909
Figure 25.60 MII Transmit Timing (Case of Conflict) 909
Figure 25.61 MII Receive Timing (Normal Operation) 910
Figure 25.62 MII Receive Timing (Case of Error) 910
Figure 25.63 MDIO Input Timing 910
Figure 25.64 MDIO Output Timing 910
Figure 25.65 WOL Output Timing 911
Figure 25.66 EXOUT Output Timing 911
Figure 25.67 CAMSEN Input Timing 911
Figure 25.68 $\overline{\text{ARBUBY}}$ Output Timing 911
Figure 25.69 I/O Port Timing 912
Figure 25.70 TCK Input Timing 913
Figure 25.71 $\overline{\text{TRST}}$ Input Timing (Reset Hold) 914
Figure 25.72 H-UDI Data Transfer Timing 914
Figure 25.73 $\overline{\text{ASEMD0}}$ Input Timing 914
Figure 25.74 $\overline{\text{ASEBRKAK}}$ Delay Time 914
Figure 25.75 Output Load Circuit 915
Figure 25.76 Load Capacitance vs. Delay Time 916

Appendix

Figure B.1 Package Dimensions (HQFP2828-256 (FP-256G/GV)) 926
Figure B.2 Package Dimensions (P-LFBGA1717-256 (BP-256H/HV)) 927

Tables

Section 1 Overview
Table 1.1 Pin Assigument 10
Table 1.2 Pin Functions 19

Section 2 CPU
Table 2.1 Logical Address Space 30
Table 2.2 Register Initial Values 33
Table 2.3 Addressing Modes and Effective Addresses for CPU Instructions 45
Table 2.4 CPU Instruction Formats 49
Table 2.5 CPU Instruction Types 52
Table 2.6 Data Transfer Instructions 56
Table 2.7 Arithmetic Operation Instructions 58
Table 2.8 Logic Operation Instructions 60
Table 2.9 Shift Instructions 61
Table 2.10 Branch Instructions 62
Table 2.11 System Control Instructions 63
Table 2.12 Operation Code Map 66

Section 3 DSP Operating Unit
Table 3.1 Logical Address Space 73
Table 3.2 Operation of SR Bits in Each Processing Mode 76
Table 3.3 RS and RE Setting Rule 82
Table 3.4 Repeat Control Instructions 82
Table 3.5 Repeat Control Macros 83
Table 3.6 DSP Mode Extended System Control Instructions 84
Table 3.7 PC Value during Repeat Control (When RC[11:0] ≥ 2) 87
Table 3.8 Extended Repeat Control Instructions 91
Table 3.9 Extended System Control Instructions in DSP Mode 96
Table 3.10 Overview of Data Transfer Instructions 99
Table 3.11 Modulo Addressing Control Instructions 101
Table 3.12 Double Data Transfer Instruction Formats 104
Table 3.13 Single Data Transfer Instruction Formats 105
Table 3.14 Destination Register in DSP Instructions 107
Table 3.15 Source Register in DSP Operations 108
Table 3.16 DSR Register Bits 109
Table 3.17 DSP Operation Instruction Formats 112
Table 3.18 Correspondence between DSP Instruction Operands and Registers 112
Table 3.19 DC Bit Update Definitions 114

Table 3.20 Examples of NOPX and NOPY Instruction Codes 116
Table 3.21 Variation of ALU Fixed-Point Operations 119
Table 3.22 Correspondence between Operands and Registers 119
Table 3.23 Variation of ALU Integer Operations 124
Table 3.24 Variation of ALU Logical Operations 125
Table 3.25 Variation of Fixed-Point Multiply Operation 127
Table 3.26 Correspondence between Operands and Registers 127
Table 3.27 Variation of Shift Operations 128
Table 3.28 Operation Definition of PDMSB 134
Table 3.29 Variation of PDMSB Operation 135
Table 3.30 Variation of Rounding Operation 136
Table 3.31 Definition of Overflow Protection for Fixed-Point Arithmetic Operations 137
Table 3.32 Definition of Overflow Protection for Integer Arithmetic Operations 137
Table 3.33 Variation of Local Data Move Operations 138
Table 3.34 Correspondence between Operands and Registers 139
Table 3.35 DSP Mode Extended System Control Instructions 140
Table 3.36 Double Data Transfer Instruction 142
Table 3.37 Single Data Transfer Instructions 143
Table 3.38 Correspondence between DSP Data Transfer Operands and Registers 144
Table 3.39 DSP Operation Instructions 145
Table 3.40 Operation Code Map 151

Section 4 Exception Handling

Table 4.1 Exception Event Vectors 163
Table 4.2 Instruction Positions and Restriction Types 173
Table 4.3 SPC Value when Re-Execution Type Exception Occurs in Repeat Control (RC[11:0] ≥ 2) 176
Table 4.4 Exception Acceptance in Repeat Loop 177
Table 4.5 Instruction Where a Specific Exception Occurs when Memory Access Exception Occurs in Repeat Control (SR.RC[11:0] ≥ 1) 178

Section 5 Memory Management Unit (MMU)

Table 5.1 Access States Designated by D, C, and PR Bits 199

Section 6 Cache

Table 6.1 LRU and Way Replacement (when Cache Locking Mechanism is Disabled) 216
Table 6.2 Way Replacement when a PREF Instruction Misses the Cache 220
Table 6.3 Way Replacement when Instructions other than the PREF Instruction Miss the Cache 220
Table 6.4 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =0) 220
Table 6.5 LRU and Way Replacement (when W2LOCK = 0 and W3LOCK =1) 221
Table 6.6 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =1) 221

Section 7 X/Y Memory
Table 7.1 X/Y Memory Logical Addresses 231
Table 7.2 MMU and Cache Settings 234

Section 8 Interrupt Controller (INTC)
Table 8.1 Pin Configuration 237
Table 8.2 Interrupt Exception Handling Sources and Priority (IRQ Mode) 240
Table 8.3 Interrupt Exception Handling Sources and Priority (IRL Mode) 243
Table 8.4 Interrupt Level and INTEVT Code 245
Table 8.5 Interrupt Sources and IPRA to IPRI 247

Section 9 User Break Controller
Table 9.1 Specifying Break Address Register 269
Table 9.2 Specifying Break Data Register 271
Table 9.3 Data Access Cycle Addresses and Operand Size Comparison Conditions 284

Section 10 Power-Down Modes
Table 10.1 States of Power-Down Modes 296
Table 10.2 Pin Configuration 298
Table 10.3 Register States in Software Standby Mode 303

Section 11 On-Chip Oscillation Circuits
Table 11.1 Pin Configuration 315
Table 11.2 Clock Operating Modes 315
Table 11.3 Possible Combination of Clock Mode and FRQCR Values 317

Section 12 Bus State Controller (BSC)
Table 12.1 Pin Configuration 334
Table 12.2 Address Space Map 1 (CMNCR.MAP = 0) 338
Table 12.3 Address Space Map 2 (CMNCR.MAP = 1) 339
Table 12.4 Correspondence between External Pins (MD3 and MD4), Memory Type of CS0, and Memory Bus Width 340
Table 12.5 Correspondence between External Pin (MD5) and Endians 340
Table 12.6 32-Bit External Device/Big Endian Access and Data Alignment 383
Table 12.7 16-Bit External Device/Big Endian Access and Data Alignment 384
Table 12.8 8-Bit External Device/Big Endian Access and Data Alignment 385
Table 12.9 32-Bit External Device/Little Endian Access and Data Alignment 386
Table 12.10 16-Bit External Device/Little Endian Access and Data Alignment 387
Table 12.11 8-Bit External Device/Little Endian Access and Data Alignment 388
Table 12.12 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1 401
Table 12.12 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2 403

Table 12.13 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1 404
Table 12.13 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2 405
Table 12.14 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3) 407
Table 12.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1 408
Table 12.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2 409
Table 12.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1 410
Table 12.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2 411
Table 12.17 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1 412
Table 12.17 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2 413
Table 12.18 Relationship between Access Size and Number of Bursts 414
Table 12.19 Access Address in SDRAM Mode Register Write 434
Table 12.20 Output Addresses when EMRS Command is Issued 437
Table 12.21 Relationship between Bus Width, Access Size, and Number of Bursts 440

Section 13 Direct Memory Access Controller (DMAC)

Table 13.1 Pin Configuration 461
Table 13.2 DMARS Setting 474
Table 13.3 Selecting External Request Modes with RS Bits 477
Table 13.4 Selecting External Request Detection with DL, DS Bits 478
Table 13.5 Selecting External Request Detection with DO Bit 478
Table 13.6 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits 479
Table 13.7 Supported DMA Transfers 482
Table 13.8 Relationship of Request Modes and Bus Modes by DMA Transfer Category 488

Section 14 Timer Unit (TMU)

Table 14.1 TMU Interrupt Sources 504

Section 15 Realtime Clock (RTC)

Table 15.1 Pin Configuration 507
Table 15.2 Recommended Oscillator Circuit Constants (Recommended Values) 528

Section 16 Serial Communication Interface with FIFO (SCIF)

Table 16.1 Pin Configuration 534

Table 16.2 Relationship between n and Clock 553
Table 16.3 SCSMR Settings for Serial Transfer Format Selection 560
Table 16.4 SCSMR and SCSCR Settings for the SCIF Clock Source Selection 560
Table 16.5 Serial Transfer Formats 562
Table 16.6 The SCIF Interrupt Sources 583

Section 17 Serial I/O with FIFO (SIOF)
Table 17.1 Pin Configuration 589
Table 17.2 SIOF Serial Clock Frequency 614
Table 17.3 Serial Transfer Modes 616
Table 17.4 Frame Length 617
Table 17.5 Audio Mode Specification for Transmit Data 619
Table 17.6 Audio Mode Specification for Receive Data 619
Table 17.7 Setting for Number of Control Data Channels 620
Table 17.8 Conditions to Issue Transmit Request 622
Table 17.9 Conditions to Issue Receive Request 622
Table 17.10 Transmission and Reception Reset 628
Table 17.11 SIOF Interrupt Sources 629
Table 17.12 Setting Condition of Transmit/Receive Interrupt Flag 630

Section 18 Ethernet Controller (EtherC)
Table 18.1 Pin Configuration 639
Table 18.2 Transfer Frame Processing (Without CAM) 711
Table 18.3 Reception Frame Process 713
Table 18.4 Relay Frame Process (With CAM) 713
Table 18.5 Receive Frame Process (When External CAM Logic is Used) 715
Table 18.6 Relay Frame Process (When External CAM Logic is Used) 716

Section 21 Pin Function Controller (PFC)
Table 21.1 List of Multiplexed Pins (1) 781
Table 21.2 List of Multiplexed Pins (2) 782

Section 22 I/O Ports
Table 22.1 Port A Data Register (PADR) Read/Write Operations 790
Table 22.2 Port B Data Register (PBDR) Read/Write Operations (1) 791
Table 22.3 Port B Data Register (PBDR) Read/Write Operations (2) 791
Table 22.4 Port C Data Register (PCDR) Read/Write Operations 792

Section 23 User Debugging Interface (H-UDI)
Table 23.1 Pin Configuration 794
Table 23.2 H-UDI Commands 796
Table 23.3 This LSI's Pins and Boundary Scan Register Bits 797
Table 23.4 Reset Configuration 805

Section 25 Electrical Characteristics

Table 25.1 Absolute Maximum Ratings 855
Table 25.2 DC Characteristics (1) 857
Table 25.2 DC Characteristics (2) 858
Table 25.3 Permitted Output Current Values 859
Table 25.4 Maximum Operating Frequencies 859
Table 25.5 Clock Timing 860
Table 25.6 Control Signal Timing 865
Table 25.7 Bus Timing (1) 868
Table 25.8 Bus Timing (2) 896
Table 25.9 DMAC Signal Timing 901
Table 25.10 RTC Signal Timing 902
Table 25.11 SCIF Module Signal Timing 903
Table 25.12 SIOF Module Signal Timing 904
Table 25.13 Ethernet Controller Timing 908
Table 25.14 Port Input/Output Timing 912
Table 25.15 H-UDI Related Pin Timing 913

Section 1 Overview and Pin Function

This LSI is a 32-bit reduced instruction set computer (RISC) microprocessor that is built on the SuperH architecture.

Its core is a RISC-type CPU with a Digital Signal Processor (DSP) as a functional extension. A single chip microprocessor integrates peripheral functions required for building an Ethernet system.

The LSI comprises two channels of Ethernet controllers. They include a media access controller (MAC) and a media independent interface (MII) standard unit that conforms to the IEE802.3u standard and provide 10/100 Mbps LAN connection. The on-chip IP security accelerator enables efficient security management of network data.

The LSI has a large capacity (32-kbyte) cache memory, 16-kbyte on-chip X/Y memory, and an interrupt controller for system configuration to enable flexible system design. It supports high-speed data transfer using an on-chip direct memory access controller (DMAC). Its external memory access support provides direct connection to various types of memory.

The strong on-chip power saving function reduces power consumption even during high-speed operation.

1.1 Features

The features of this LSI are shown below.

CPU:

- Original Renesas-Technology SuperH architecture
- Compatible with SH-1, SH-2, and SH-3 at object code level
- 32-bit internal data bus
- Various groups of built-in registers
  General registers: Sixteen 32-bit registers (including eight 32-bit bank registers)
  Control registers: Five 32-bit registers
  System registers: Four 32-bit registers
- Supports RISC-type instruction set
  Instruction length: 16-bit fixed length for improved code efficiency
  Load/store architecture
  Delayed branch instructions
  Instruction set based on C language
- Instruction execution time: one instruction/cycle for basic instructions

- Logical address space: 4 Gbytes
- Space identifier ASID: 8 bits, 256 logical address spaces
- Supports five-stage pipeline

DSP:

- Mixture of 16-bit and 32-bit instructions
- 32-/40-bit internal data bus
- Multiplier, ALU, and barrel shifter
- 16-bit × 16-bit → 32-bit one cycle multiplier
- Large-capacity DSP data registers
  Six 32-bit data registers
  Two 40-bit data registers
- Supports extended harvard architecture for DSP data bus
  Two data buses
  One instruction bus
- Maximum four parallel operations
  ALU, multiply, and two load/store
- Two addressing units to generate addresses for two memory access
- Supports DSP data addressing modes
  Increment and indexing (with or without modulo addressing)
- Zero-overhead repeat loop control
- Conditional execution instructions
- Supports user DSP mode and privileged DSP mode

Memory management unit (MMU):

- 4 Gbytes of address space, 256 address spaces (8-bit ASID)
- Page unit sharing
- Supports multiple page sizes
  1 kbyte or 4 kbytes
- Supports 128-entry, 4-way set associative TLB
- Supports software selection of replacement way and random-replacement algorithms
- Contents of TLB are directly accessible by address mapping

Cache memory:

- 32-kbyte cache, mixture of instructions and data
- 512-entry, 4-way set associative, 16-byte block length
- Write-back, write-through, LRU replacement algorithm
- 1-stage write-back buffer

X/Y memory:

- Three independent read/write ports
  8-/16-/32-bit access from the CPU
  Maximum two 16-bit accesses from the DSP
  8-/16-/32-bit access from the DMAC or E-DMAC
- A total of 16 kbytes memory (8-kbyte RAM each for X- and Y-memory)

Interrupt controller (INTC):

- Supports seven external interrupt pins (NMI, IRQ5 to IRQ0)
- Supports fifteen level interrupt pins ($\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$)
- Supports one interrupt request output pin ($\overline{\text{IRQOUT}}$)
- On-chip peripheral interrupt: Priority level is independently selected for each module
- Supports software vector mode
- Selection of falling/rising/high/low

User break controller (UBC):

- Address, data value, access type, and data size are available for setting as break conditions
- Supports the sequential break function
- Two break channels

On-Chip Oscillation Circuits:

- Clock source selectable between an external supply (EXTAL or CKIO) and crystal resonator
  The internal clock and peripheral clock can be adjusted by setting the PLL circuit and division ratio.
- Three types of clocks generated:
  CPU clock (I clock): 200 MHz (max)
  Bus clock (B clock): 66 MHz (max)
  Peripheral clock (P clock): 33 MHz (max)

- Supports power-down modes:
  Sleep mode
  Software standby mode
  Module standby mode
- A single channel on-chip watch dog timer
  Watchdog timer mode and interval timer mode is selectable.
  An interrupt can be generated in interval timer mode.

Bus state controller (BSC):

- Physical address space is divided into eight areas: area 0, areas 2 to 4; each a maximum of 64 Mbytes, and areas 5A, 5B, 6A, 6B; each a maximum of 32 Mbytes.
- The following features are settable for each area.
  Bus size (8, 16 or 32 bits): The supported bus size differs for each area.
  Number of access wait cycles: Numbers of wait-state cycles during reading and writing are independently selectable for some areas.
  Setting of idle wait cycles: For the same area or different area.
  Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, burst ROM (synchronization/asynchronous), SDRAM and PCMCIA.
- Outputs chip select signal ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A/B}$, and $\overline{CS6A/B}$) for corresponding area (The CS assert/negate timing can be selected by software.)

Direct memory access controller (DMAC):

- Six channels. Two of these channels (ch0 and ch1) support external requests.
- Supports burst mode and cycle-stealing mode

Timer unit (TMU):

- 3-channel auto reload 32-bit on-chip timer
- 4 types of counter input clocks can be selected

Realtime clock (RTC)*[1]:

- On-chip clock, calendar, and alarm
- On-chip 32 kHz crystal oscillator with 1/256-second resolution (interrupt cycle)

Serial communication interface with FIFO (SCIF):

- 16 bytes each for transmit/receive FIFO
- Two channels (SCIF0 and SCIF1)
- CTS/RTS (flow control) support
- Asynchronous and synchronous modes
- Full-duplex communication support
- DMA transfer

Serial I/O with FIFO (SIOF):

- 64 bytes each for transmit/receive FIFO
- 8-/16-/16-bit stereo-audio input/output supported
- Two channels (SIOF0 and SIOF1)
- DMA transfer
- Frame synchronous signal

Ethernet controller (EtherC):

- MAC (Media Access Control)
- Data frame assembly/disassembly (frame format conforming to IEEE802.3u)
- CSMA/CD link management (collision prevention and collision processing)
- CRC processing
- Full-duplex transmit/receive support
- Detects short frames and long frames
- Conforms to MII (Media Independent Interface) standard *[2]
- Conversion from 8-bit stream data in MAC layer to MII nibble (4-bit) stream
- Station management (STA function)
- 10/100 Mbps transfer rate adjustable
- WOL (Wake-On-LAN) signal output with Magic Packet*[3] detection
- CAM sense signal input

Ethernet Controller Direct Memory Access Controller (E-DMAC):

- EtherC — Transfer between external and internal memories
- 16-byte burst transfer
- Single address transfer
- Chain block transfer

- Transfer data width: 32 bits
- Address space: 4 Gbytes
- On-chip FIFO (2-kbytes each for transmit/receive)

IP Security Accelerator (IPSEC):

- DES/Triple-DES encryption/decryption according to DES (Data Encryption Standard)
- Hash algorithm generation according to MD5 Message Digest Algorithm
- Hash algorithm generation according to SHA-1 Source Hash Standard
- On-chip DMAC dedicated for data transfer
- Interrupt request support

User debugging interface (H-UDI):

- Supports the E10A emulator
- Realtime branch trace
- 1-kbyte of on-chip RAM for executing the high-speed emulation program

Notes: 1. As the power supply is connected, power should always be supplied to all power supplies even if only RTC operates.
2. +5 V I/O is not supported.
3. Magic Packet is the registered trademark of Advanced Micro Devices Inc.

Product Lineup:

	Power supply voltage		Maximum operating		
Abbreviation	**I/O**	**Internal**	**frequency**	**Type name**	**Package**
SH7710	3.3 V ± 0.3 V	1.5 V ± 0.1 V	200 MHz	HD6417710BP/BPV	256-pin CSP (BP-256H/HV)
				HD6417710F/FV	256-pin HQFP (FP-256G/GV)

1.2 Block Diagram

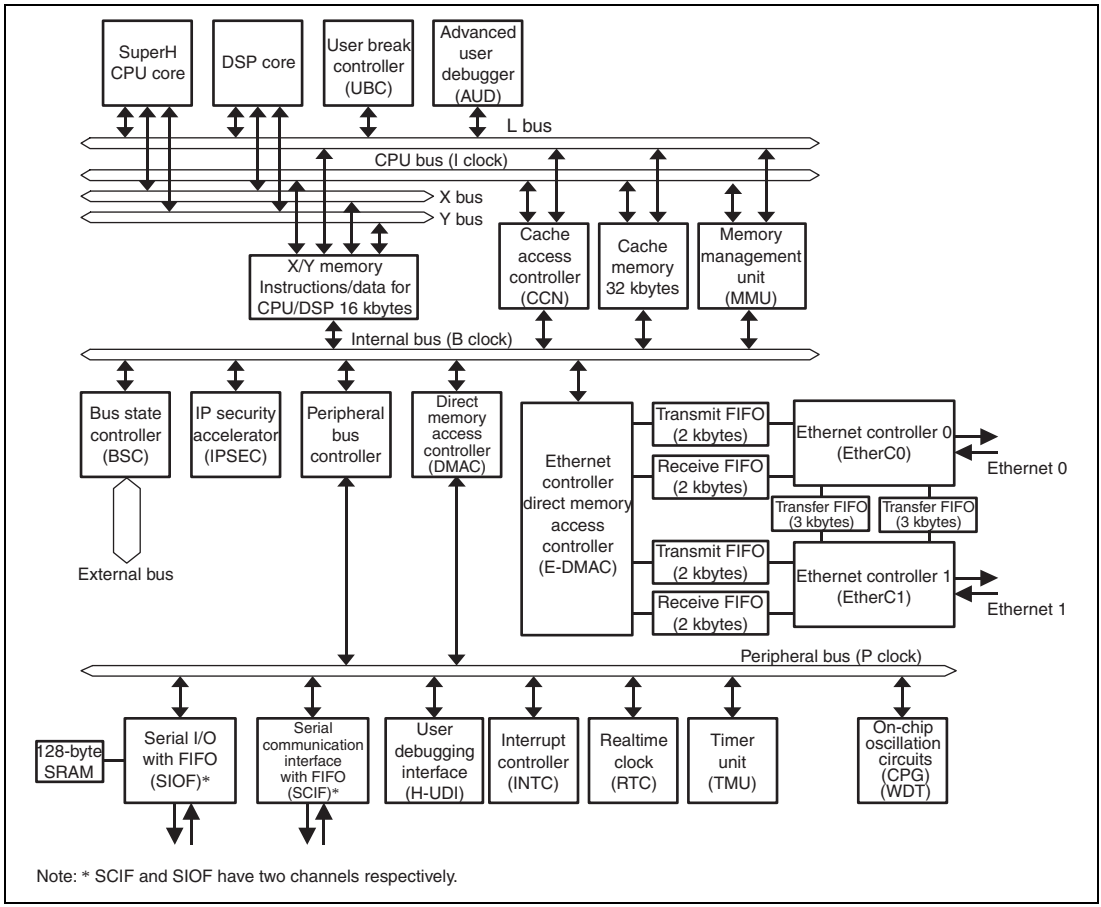


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Assignment

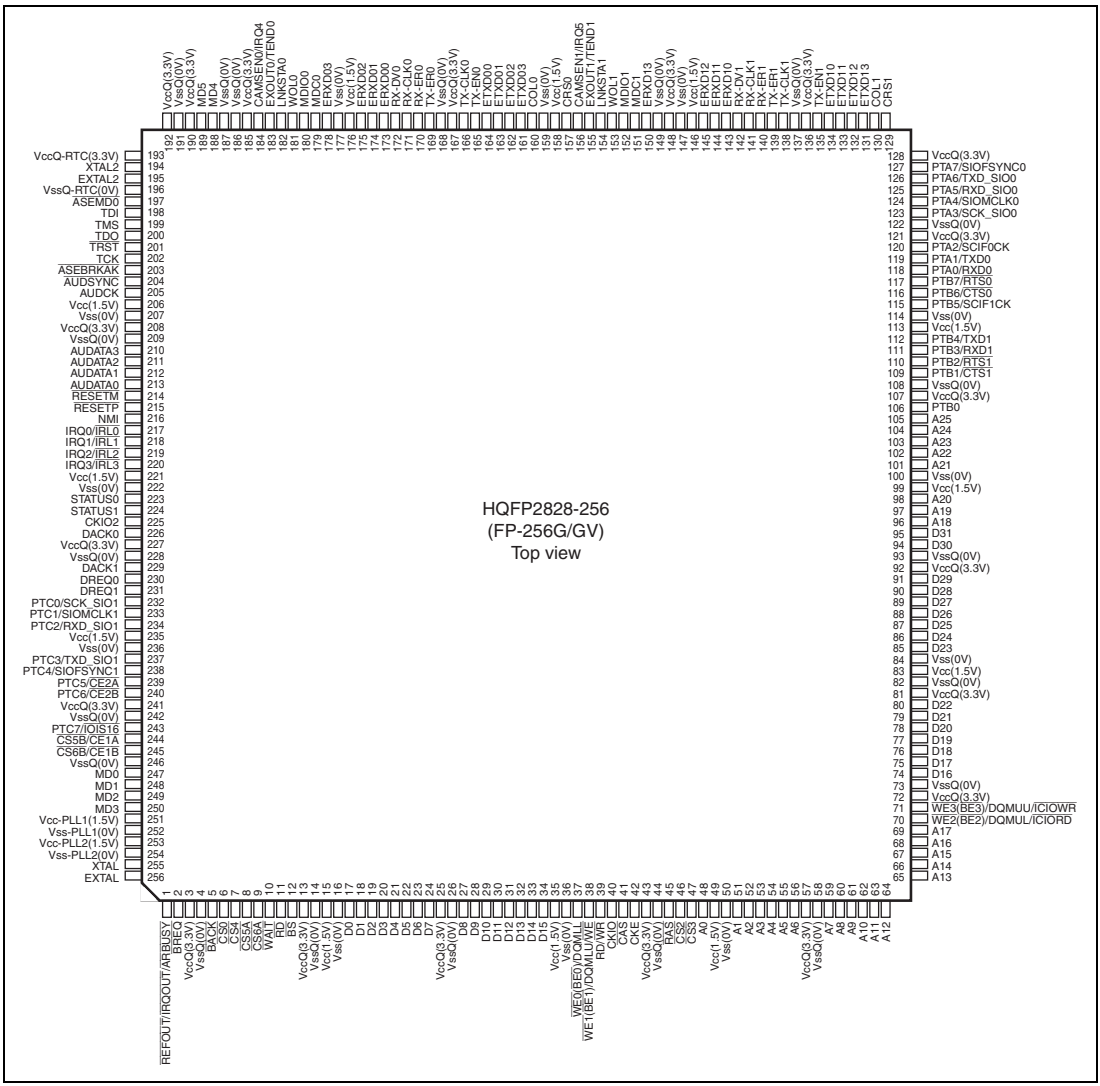


Figure 1.2 Pin Assignment (HQFP2828-256 (FP-256G/GV))

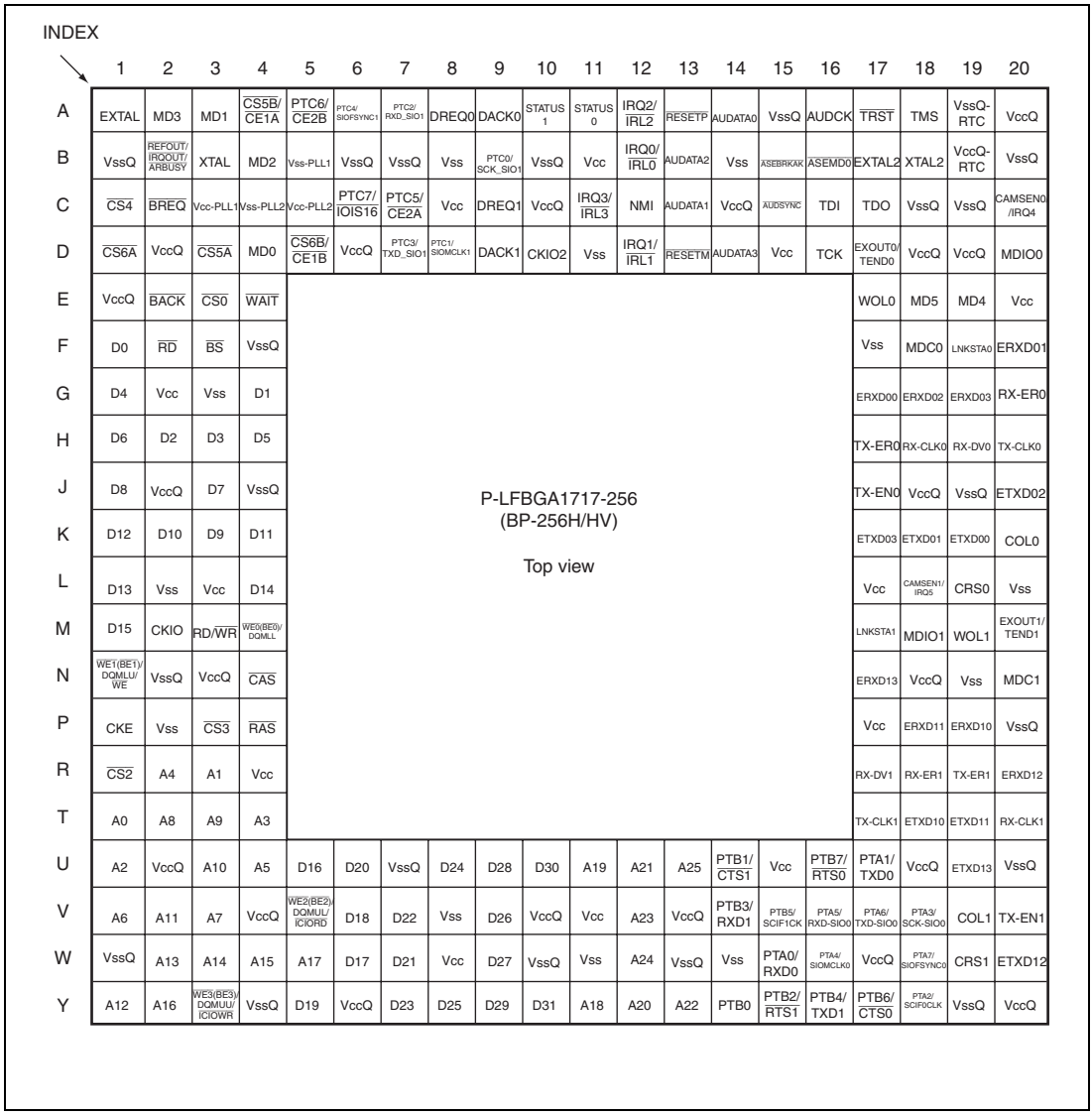


Figure 1.3 Pin Assignment (P-LFBGA1717-256 (BP-256H/HV))

Table 1.1 Pin Assigument

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
1	B2	$\overline{\text{REFOUT}}$/IRQOUT/ $\overline{\text{ARBUSY}}$	O/O/O	Bus release request output
2	C2	$\overline{\text{BREQ}}$	I	Bus request
3	D2	VccQ		I/O power supply (3.3 V)
4	B1	VssQ		I/O power supply (0 V)
5	E2	$\overline{\text{BACK}}$	O	Bus acknowledge
6	E3	$\overline{\text{CS0}}$	O	Chip select 0
7	C1	$\overline{\text{CS4}}$	O	Chip select 4
8	D3	$\overline{\text{CS5A}}$	O	Chip select 5 A
9	D1	$\overline{\text{CS6A}}$	O	Chip select 6 A
10	E4	$\overline{\text{WAIT}}$	I	Hardware wait request
11	F2	$\overline{\text{RD}}$	O	Read strobe
12	F3	$\overline{\text{BS}}$	O	Bus cycle start signal
13	E1	VccQ		I/O power supply (3.3 V)
14	F4	VssQ		I/O power supply (0 V)
15	G2	Vcc		Internal power supply (1.5 V)
16	G3	Vss		Internal power supply (0 V)
17	F1	D0	IO	Data bus
18	G4	D1	IO	Data bus
19	H2	D2	IO	Data bus
20	H3	D3	IO	Data bus
21	G1	D4	IO	Data bus
22	H4	D5	IO	Data bus
23	H1	D6	IO	Data bus
24	J3	D7	IO	Data bus
25	J2	VccQ		I/O power supply (3.3 V)
26	J4	VssQ		I/O power supply (0 V)
27	J1	D8	IO	Data bus
28	K3	D9	IO	Data bus
29	K2	D10	IO	Data bus

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
30	K4	D11	IO	Data bus
31	K1	D12	IO	Data bus
32	L1	D13	IO	Data bus
33	L4	D14	IO	Data bus
34	M1	D15	IO	Data bus
35	L3	Vcc		Internal power supply (1.5 V)
36	L2	Vss		Internal power supply (0 V)
37	M4	$\overline{WE0}$($\overline{BE0}$)/DQMLL	O/O	D7 to D0-select signal/DQM (SDRAM)
38	N1	$\overline{WE1}$($\overline{BE1}$)/DQMLU/ $\overline{WE}$	O/O/O	D15 to D8-select signal/DQM (SDRAM)/PCMCIA write cycle strobe
39	M3	RD/$\overline{WR}$	O	Read/write
40	M2	CKIO	IO	System clock I/O
41	N4	$\overline{CAS}$	O	CAS (SDRAM)
42	P1	CKE	O	CK enable (SDRAM)
43	N3	VccQ		I/O power supply (3.3 V)
44	N2	VssQ		I/O power supply (0 V)
45	P4	$\overline{RAS}$	O	RAS (SDRAM)
46	R1	$\overline{CS2}$	O	Chip select 2
47	P3	$\overline{CS3}$	O	Chip select 3
48	T1	A0	O	Address bus
49	R4	Vcc		Internal power supply (1.5 V)
50	P2	Vss		Internal power supply (0 V)
51	R3	A1	O	Address bus
52	U1	A2	O	Address bus
53	T4	A3	O	Address bus
54	R2	A4	O	Address bus
55	U4	A5	O	Address bus
56	V1	A6	O	Address bus
57	U2	VccQ		I/O power supply (3.3 V)
58	W1	VssQ		I/O power supply (0 V)
59	V3	A7	O	Address bus

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
60	T2	A8	O	Address bus
61	T3	A9	O	Address bus
62	U3	A10	O	Address bus
63	V2	A11	O	Address bus
64	Y1	A12	O	Address bus
65	W2	A13	O	Address bus
66	W3	A14	O	Address bus
67	W4	A15	O	Address bus
68	Y2	A16	O	Address bus
69	W5	A17	O	Address bus
70	V5	$\overline{WE2}$($\overline{BE2}$)/DQMUL/ $\overline{ICIORD}$	O/O/O	D23 to D16-select signal/DQM (SDRAM)/PCMCIA I/O read
71	Y3	$\overline{WE3}$($\overline{BE3}$)/DQMUU/ $\overline{ICIOWR}$	O/O/O	D31 to D24-select signal/DQM (SDRAM)/PCMCIA I/O write
72	V4	VccQ		I/O power supply (3.3 V)
73	Y4	VssQ		I/O power supply (0 V)
74	U5	D16	IO	Data bus
75	W6	D17	IO	Data bus
76	V6	D18	IO	Data bus
77	Y5	D19	IO	Data bus
78	U6	D20	IO	Data bus
79	W7	D21	IO	Data bus
80	V7	D22	IO	Data bus
81	Y6	VccQ		I/O power supply (3.3 V)
82	U7	VssQ		I/O power supply (0 V)
83	W8	Vcc		Internal power supply (1.5 V)
84	V8	Vss		Internal power supply (0 V)
85	Y7	D23	IO	Data bus
86	U8	D24	IO	Data bus
87	Y8	D25	IO	Data bus
88	V9	D26	IO	Data bus

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
89	W9	D27	IO	Data bus
90	U9	D28	IO	Data bus
91	Y9	D29	IO	Data bus
92	V10	VccQ		I/O power supply (3.3 V)
93	W10	VssQ		I/O power supply (0 V)
94	U10	D30	IO	Data bus
95	Y10	D31	IO	Data bus
96	Y11	A18	O	Address bus
97	U11	A19	O	Address bus
98	Y12	A20	O	Address bus
99	V11	Vcc		Internal power supply (1.5 V)
100	W11	Vss		Internal power supply (0 V)
101	U12	A21	O	Address bus
102	Y13	A22	O	Address bus
103	V12	A23	O	Address bus
104	W12	A24	O	Address bus
105	U13	A25	O	Address bus
106	Y14	PTB0	IO	I/O port B
107	V13	VccQ		I/O power supply (3.3 V)
108	W13	VssQ		I/O power supply (0 V)
109	U14	PTB1/$\overline{\text{CTS1}}$	IO/I	I/O port B/SCIF1 transmit clear
110	Y15	PTB2/$\overline{\text{RTS1}}$	IO/O	I/O port B/SCIF1 transmit request
111	V14	PTB3/RXD1	IO/I	I/O port B/SCIF1 receive data
112	Y16	PTB4/TXD1	IO/O	I/O port B/SCIF1 transmit data
113	U15	Vcc		Internal power supply (1.5 V)
114	W14	Vss		Internal power supply (0 V)
115	V15	PTB5/SCIF1CK	IO/IO	I/O port B/SCIF1 serial clock
116	Y17	PTB6/$\overline{\text{CTS0}}$	IO/I	I/O port B/SCIF0 transmit clear
117	U16	PTB7/$\overline{\text{RTS0}}$	IO/O	I/O port B/SCIF0 transmit request
118	W15	PTA0/RXD0	IO/I	I/O port A/SCIF0 receive data
119	U17	PTA1/TXD0	IO/O	I/O port A/SCIF0 transmit data

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
120	Y18	PTA2/SCIF0CK	IO/IO	I/O port A/SCIF0 serial clock
121	W17	VccQ		I/O power supply (3.3 V)
122	Y19	VssQ		I/O power supply (0 V)
123	V18	PTA3/SCK_SIO0	IO/IO	I/O port A/SIOF0 communication clock
124	W16	PTA4/SIOMCLK0	IO/I	I/O port A/SIOF0 clock input
125	V16	PTA5/RXD_SIO0	IO/I	I/O port A/SIOF0 receive data
126	V17	PTA6/TXD_SIO0	IO/O	I/O port A/SIOF0 transmit data
127	W18	PTA7/SIOFSYNC0	IO/IO	I/O port A/SIOF0 frame sync
128	Y20	VccQ		I/O power supply (3.3 V)
129	W19	CRS1	I	MAC1 carrier detection
130	V19	COL1	I	MAC1 collision detection
131	U19	ETXD13	O	MAC1 transmit data 3
132	W20	ETXD12	O	MAC1 transmit data 2
133	T19	ETXD11	O	MAC1 transmit data 1
134	T18	ETXD10	O	MAC1 transmit data 0
135	V20	TX-EN1	O	MAC1 transmit enable
136	U18	VccQ		I/O power supply (3.3 V)
137	U20	VssQ		I/O power supply (0 V)
138	T17	TX-CLK1	I	MAC1 transmit clock
139	R19	TX-ER1	O	MAC1 transmit error
140	R18	RX-ER1	I	MAC1 receive error
141	T20	RX-CLK1	I	MAC1 receive clock
142	R17	RX-DV1	I	MAC1 receive data valid
143	P19	ERXD10	I	MAC1 receive data 0
144	P18	ERXD11	I	MAC1 receive data 1
145	R20	ERXD12	I	MAC1 receive data 2
146	P17	Vcc		Internal power supply (1.5 V)
147	N19	Vss		Internal power supply (0 V)
148	N18	VccQ		I/O power supply (3.3 V)
149	P20	VssQ		I/O power supply (0 V)

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
150	N17	ERXD13	I	MAC1 receive data 3
151	N20	MDC1	O	MAC1 management data clock
152	M18	MDIO1	IO	MAC1 management data I/O
153	M19	WOL1	O	MAC1 Wake-On-LAN
154	M17	LNKSTA1	I	MAC1 link status
155	M20	EXOUT1/TEND1	O/O	MAC1 general-purpose external output/DMA transfer end notification 1
156	L18	CAMSEN1/IRQ5	I/I	MAC1 CAM input/external interrupt request
157	L19	CRS0	I	MAC0 carrier detection
158	L17	Vcc		Internal power supply (1.5 V)
159	L20	Vss		Internal power supply (0 V)
160	K20	COL0	I	MAC0 collision detection
161	K17	ETXD03	O	MAC0 transmit data 3
162	J20	ETXD02	O	MAC0 transmit data 2
163	K18	ETXD01	O	MAC0 transmit data 1
164	K19	ETXD00	O	MAC0 transmit data 0
165	J17	TX-EN0	O	MAC0 transmit enable
166	H20	TX-CLK0	I	MAC0 transmit clock
167	J18	VccQ		I/O power supply (3.3 V)
168	J19	VssQ		I/O power supply (0 V)
169	H17	TX-ER0	O	MAC0 transmit error
170	G20	RX-ER0	I	MAC0 receive error
171	H18	RX-CLK0	I	MAC0 receive clock
172	H19	RX-DV0	I	MAC0 receive data valid
173	G17	ERXD00	I	MAC0 receive data 0
174	F20	ERXD01	I	MAC0 receive data 1
175	G18	ERXD02	I	MAC0 receive data 2
176	E20	Vcc		Internal power supply (1.5 V)
177	F17	Vss		Internal power supply (0 V)
178	G19	ERXD03	I	MAC0 receive data 3

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
179	F18	MDC0	O	MAC0 management data clock
180	D20	MDIO0	IO	MAC0 management data I/O
181	E17	WOL0	O	MAC0 Wake-On-LAN
182	F19	LNKSTA0	I	MAC0 link status
183	D17	EXOUT0/TEND0	O/O	MAC0 general-purpose external output/DMA transfer end notification 0
184	C20	CAMSEN0/IRQ4	I/I	MAC0 CAM input/external interrupt request
185	D19	VccQ		I/O power supply (3.3 V)
186	B20	VssQ		I/O power supply (0 V)
187	C18	VssQ		I/O power supply (0 V)
188	E19	MD4	I	Specifies area 0 bus width
189	E18	MD5	I	Endian select
190	D18	VccQ		I/O power supply (3.3 V)
191	C19	VssQ		I/O power supply (0 V)
192	A20	VccQ		I/O power supply (3.3 V)
193	B19	VccQ-RTC		RTC oscillator power supply (3.3 V)
194	B18	XTAL2	O	On-chip RTC crystal oscillator pin
195	B17	EXTAL2	I	On-chip RTC crystal oscillator pin
196	A19	VssQ-RTC		RTC oscillator power supply (0 V)
197	B16	$\overline{\text{ASEMD0}}$	I	ASE mode
198	C16	TDI	I	Test data input
199	A18	TMS	I	Test mode select
200	C17	TDO	O	Test data output
201	A17	$\overline{\text{TRST}}$	I	Test reset
202	D16	TCK	I	Test clock
203	B15	$\overline{\text{ASEBRKAK}}$	O	ASE break acknowledge
204	C15	$\overline{\text{AUDSYNC}}$	O	AUD synchronous
205	A16	AUDCK	O	AUD clock
206	D15	Vcc		Internal power supply (1.5 V)
207	B14	Vss		Internal power supply (0 V)

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
208	C14	VccQ		I/O power supply (3.3 V)
209	A15	VssQ		I/O power supply (0 V)
210	D14	AUDATA3	O	AUD data
211	B13	AUDATA2	O	AUD data
212	C13	AUDATA1	O	AUD data
213	A14	AUDATA0	O	AUD data
214	D13	$\overline{\text{RESETM}}$	I	Manual reset request
215	A13	$\overline{\text{RESETP}}$	I	Power-on reset request
216	C12	NMI	I	Non-maskable interrupt request
217	B12	IRQ0/$\overline{\text{IRL0}}$	I	External interrupt request
218	D12	IRQ1/$\overline{\text{IRL1}}$	I	External interrupt request
219	A12	IRQ2/$\overline{\text{IRL2}}$	I	External interrupt request
220	C11	IRQ3/$\overline{\text{IRL3}}$	I	External interrupt request
221	B11	Vcc		Internal power supply (1.5 V)
222	D11	Vss		Internal power supply (0 V)
223	A11	STATUS0	O	Processor status
224	A10	STATUS1	O	Processor status
225	D10	CKIO2	O	System clock output
226	A9	DACK0	O	DMA acknowledge 0
227	C10	VccQ		I/O power supply (3.3 V)
228	B10	VssQ		I/O power supply (0 V)
229	D9	DACK1	O	DMA acknowledge 1
230	A8	DREQ0	I	DMA request 0
231	C9	DREQ1	I	DMA request 1
232	B9	PTC0/SCK_SIO1	IO/IO	I/O port C/SIOF1 communication clock
233	D8	PTC1/SIOMCLK1	IO/I	I/O port C/SIOF1 clock input
234	A7	PTC2/RXD_SIO1	IO/I	I/O port C/SIOF1 receive data
235	C8	Vcc		Internal power supply (1.5 V)
236	B8	Vss		Internal power supply (0 V)
237	D7	PTC3/TXD_SIO1	IO/O	I/O port C/SIOF1 transmit data
238	A6	PTC4/SIOFSYNC1	IO/IO	I/O port C/SIOF1 frame sync

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name	I/O	Description
239	C7	PTC5/$\overline{CE2A}$	IO/O	I/O port C/area 5 PCMCIA card enable
240	A5	PTC6/$\overline{CE2B}$	IO/O	I/O port C/area 6 PCMCIA card enable
241	D6	VccQ		I/O power supply (3.3 V)
242	B7	VssQ		I/O power supply (0 V)
243	C6	PTC7/$\overline{IOIS16}$	IO/I	I/O port C/PCMCIA 16-bit I/O select
244	A4	$\overline{CS5B}$/$\overline{CE1A}$	O/O	Chip select 5B/area 5 PCMCIA card enable
245	D5	$\overline{CS6B}$/$\overline{CE1B}$	O/O	Chip select 6B/area 6 PCMCIA card enable
246	B6	VssQ		I/O power supply (0 V)
247	D4	MD0	I	Clock mode select
248	A3	MD1	I	Clock mode select
249	B4	MD2	I	Clock mode select
250	A2	MD3	I	Area 0 bus width
251	C3	Vcc-PLL1		PLL1 power supply (1.5 V)
252	B5	Vss-PLL1		PLL1 power supply (0 V)
253	C5	Vcc-PLL2		PLL2 power supply (1.5 V)
254	C4	Vss-PLL2		PLL2 power supply (0 V)
255	B3	XTAL	O	Clock oscillator pin
256	A1	EXTAL	I	External clock/crystal oscillator pin

Notes: 1. VccQ-RTC must be supplied even if the realtime clock (RTC) is not used.
2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal oscillator circuit for RTC operates with VccQ-RTC. The control circuit and the RTC counter operate with Vcc (common to the internal circuit). Therefore, all power supplies other than VccQ-RTC should always be turned on even if only RTC operates.
3. Vcc-PLL1/Vcc-PLL2 must be supplied even if the on-chip CPG is not used.
4. VccQ (3.3 V), Vcc (1.5 V), VssQ, and Vss must be connected to the system power supply (for uninterrupted supply).

1.3.2 Pin Functions

Table 1.2 lists the pin functions.

Table 1.2 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	—	Power supply	Power supply for the internal LSI and ports for the system. Connect all Vcc pins to the system power supply. There will be no operation if any pins are open.
	Vss	—	Ground	Ground pin. Connect all Vss pins to the system power supply (0 V). There will be no operation if any pins are open.
	VccQ	—	Power supply	Power supply for I/O pins. Connect all VccQ pins to the system power supply. There will be no operation if any pins are open.
	VssQ	—	Ground	Ground pin. Connect all VssQ pins to the system power supply (0 V). There will be no operation if any pins are open.
Clock	Vcc-PLL1	I	PLL1 power supply	Power supply for the on-chip PLL1 oscillator.
	Vss-PLL1	I	PLL1 ground	Ground pin for the on-chip PLL1 oscillator.
	Vcc-PLL2	I	PLL2 power supply	Power supply for the on-chip PLL2 oscillator.
	Vss-PLL2	I	PLL2 ground	Ground pin for the on-chip PLL2 oscillator.
	EXTAL	I	External clock	For connection to a crystal resonator. This pin can be also used for external clock input. For examples of crystal resonator connection and external clock input, see section 11, On-Chip Oscillation Circuits.

Classification	Symbol	I/O	Name	Function
Clock	XTAL	O	Crystal	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 11, On-Chip Oscillation Circuits.
	CKIO	I/O	System clock	Supplies the system clock to external devices. This pin can be also used for external clock input.
	CKIO2	O	System clock	Supplies the system clock to external devices.
Operating mode control	MD5 to MD0	I	Mode set	These pins set the operating mode. Do not change values on these pins during operation. MD2 to MD0 set the clock mode, MD4 and MD3 set the bus-width mode of area 0, and MD5 sets the endian.
System control	$\overline{\text{RESETP}}$	I	Power-on reset	When low, the system enters the power-on reset state.
	$\overline{\text{RESETM}}$	I	Manual reset	When low, the system enters the manual reset state.
	STATUS1 STATUS0	O	Status output	Indicates that this LSI is in software standby mode, reset, or sleep.
	$\overline{\text{BREQ}}$	I	Bus request	Low when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	Bus request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus mastership.

Classification	Symbol	I/O	Name	Function
Interrupts	NMI	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix to high when not in use.
	IRQ5 to IRQ0	I	Interrupt requests 5 to 0	Maskable interrupt request pins. Selectable as level input or edge input. The rising edge, falling edge, and both edges are selectable as edges.
	$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	I	Interrupt request	15-level interrupt request pins.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Indicates that the interrupt request is occurred.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	32-bit bidirectional bus.
Bus control	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS4}}$, $\overline{\text{CS5A}}$, $\overline{\text{CS6A}}$, $\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$, $\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$	O	Chip select 0, 2 to 4, 5A, 5B, 6A, 6B PCMCIA card select	Chip-select signals for external memory or devices. PCMCIA card select signal when PCMCIA is used.
	$\overline{\text{RD}}$	O	Read	Indicates reading of data from external devices.
	RD/$\overline{\text{WR}}$	O	Read/write	Read/write signal
	$\overline{\text{BS}}$	O	Bus start	Bus-cycle start signal
	$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/ $\overline{\text{ICIOWR}}$	O	Byte write	Indicates that bits 31 to 24 of the data in the external memory or device are being written. I/O write strobe signal when PCMCIA is used.
	$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/ $\overline{\text{ICIORD}}$	O	Byte write	Indicates that bits 23 to 16 of the data in the external memory or device are being written. I/O read strobe signal when PCMCIA is used.
	$\overline{\text{WE1}}$($\overline{\text{BE1}}$)/ $\overline{\text{WE}}$	O	Byte write	Indicates that bits 15 to 8 of the data in the external memory or device are being written. Memory write strobe signal when PCMCIA is used.

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{WE0}$($\overline{BE0}$)	O	Byte write	Indicates that bits 7 to 0 of the data in the external memory or device are being written.
	$\overline{RAS}$	O	RAS	Connects RAS pin during access to the SDRAM.
	$\overline{CAS}$	O	CAS	Connects CAS pin during access to the SDRAM.
	CKE	O	CK enable	Connects CKE pin during access to the SDRAM.
	$\overline{IOIS16}$	I	16-bit I/O selection	Indicates 16-bit I/O for PCMCIA.
	DQMUU	O	DQM	Selects D31 to D24 during access to the SDRAM.
	DQMUL	O	DQM	Selects D23 to D16 during access to the SDRAM.
	DQMLU	O	DQM	Selects D15 to D8 during access to the SDRAM.
	DQMLL	O	DQM	Selects D7 to D0 during access to the SDRAM.
	$\overline{REFOUT}$	O	Refresh request output	Outputs the refresh request in master mode or bus release.
	$\overline{WAIT}$	I	Wait	Inserts a wait cycle into the bus cycles during access to the external space.
Direct memory access controller (DMAC)	DREQ0, DREQ1	I	DMA-transfer request	Input pin for external requests for DMA transfer.
	DACK0, DACK1	O	DMA-transfer request accept	Output pin for request acceptance, in response to external requests for DMA transfer.
	TEND0, TEND1	O	DMA-transfer end output	Output pin for DMA transfer end signal.

Classification	Symbol	I/O	Name	Function
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Inputs the test-mode select signal.
	TDI	I	Test data input	Serial input pin for instructions and data.
	TDO	O	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	I	Test reset	Initializing signal input pin.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	O	AUD data	Data output pin in AUD trace mode.
	AUDCK	O	AUD clock	Synchronous-clock output pin in AUD trace mode.
	$\overline{\text{AUDSYNC}}$	O	AUD synchronous signal	Data start-position acknowledge-signal output pin in AUD trace mode.
E10A interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A emulator has entered its break mode. For the connection with the E10A, see the SH7710 E10A Emulator User's Manual (tentative title).
	$\overline{\text{ASEMD0}}$	I	ASE mode	Sets the ASE mode.
Realtime clock (RTC)	VccQ-RTC	I	RTC oscillator power supply	Power supply pin for the on-chip RTC
	VssQ-RTC	I	RTC oscillator ground	Ground pin for the on-chip RTC
	EXTAL2	I	RTC external clock	Clock input pin for the on-chip RTC clock (32.768 MHz). For a connection example, refer to section 15, Realtime Clock (RTC).
	XTAL2	O	RTC crystal	Clock output pin for the on-chip RTC clock (32.768 MHz). For a connection example, refer to section 15, Realtime Clock (RTC).

Classification	Symbol	I/O	Name	Function
Ethernet controller (EtherC1/0)	CRS1, CRS0	I	MAC1/0 carrier detection	Carrier detection pin. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	COL1, COL0	I	MAC1/0 collision detection	Collision detection pin. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	ETXD13 to ETXD10	O	MAC1 transmit data	4-bit transmit data pins. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	ETXD03 to ETXD00	O	MAC0 transmit data	4-bit transmit data pins. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	TX-EN1, TX-EN0	O	MAC1/0 transmit enable	These pins indicate that transmit data is ready on ETXD13 to ETXD10 and ETXD03 to ETXD00. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	TX-CLK1, TX-CLK0	I	MAC1/0 transmit clock	Timing reference pins (clock) for TX-EN1/0, TX-ER1/0, ETXD13 to ETXD10 and ETXD03 to ETXD00. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	TX-ER1, TX-ER0	O	MAC1/0 transmit error	These pins notify an error during transmission to the PHY-LSI. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	RX-ER1, RX-ER0	I	MAC1/0 receive error	These pins notify an error during data reception. For a connection example, refer to section 18, Ethernet Controller (EtherC).

Classification	Symbol	I/O	Name	Function
Ethernet controller (EtherC1/0)	RX-CLK1, RX-CLK0	I	MAC1/0 receive clock	Timing reference pins (clock) for RX-DV1/0, RX-ER1/0, ERXD13 to ERXD10 and ERXD03 to ERXD00. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	RX-DV1, RX-DV0	I	MAC1/0 receive data valid	These pins indicate that valid receive data is on ERXD13 to ERXD10 and ERXD03 to ERXD00. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	ERXD13 to ERXD10	I	MAC1 receive data	4-bit receive data pins. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	ERXD03 to ERXD00	I	MAC0 receive data	4-bit receive data pins. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	MDC1, MDC0	O	MAC1/0 management data clock	Reference clock pins for information transfer via MDIO. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	MDIO1, MDIO0	I/O	MAC1/0 management data I/O	Bidirectional pins for exchanging management information. For a connection example, refer to section 18, Ethernet Controller (EtherC).
	WOL1, WOL0	O	MAC1/0 Wake-On-LAN	These pins indicate that a Magic Packet has been received.
	LNKSTA1, LNKSTA0	I	MAC1/0 link status	Link state input pins from the PHY-LSI
	EXOUT1, EXOUT0	O	MAC1/0 general-purpose external output	External output pins
	CAMSEN1, CAMSEN0	I	MAC1/0 CAM input	CAM interface pins input

Classification	Symbol	I/O	Name	Function
Ethernet controller (EtherC1/0)	$\overline{\text{ARBUSY}}$	O	Bus release request	This pin outputs a bus release request when the amount of data in the receive FIFO reaches the threshold.
Serial communication interface with FIFO (SCIF1/0)	$\overline{\text{CTS1}}$, $\overline{\text{CTS0}}$	I	SCIF1/0 transmission clear	Modem control pins
	$\overline{\text{RTS1}}$, $\overline{\text{RTS0}}$	O	SCIF1/0 transmit request	Modem control pins
	RXD1, RXD0	I	SCIF1/0 receive data	Receive data pins
	TXD1, TXD0	O	SCIF1/0 transmit data	Transmit data pins
	SCIF1CK, SCIF0CK	I/O	SCIF1/0 serial clock	Clock I/O pins
Serial I/O with FIFO (SIOF1/0)	SCK_SIO1, SCK_SIO0	I/O	SIOF1/0 communication clock	Transmit/receive communication clock I/O pins
	SIOMCLK1, SIOMCLK 0	I	SIOF1/0 clock input	Master-clock input pins
	RXD_SIO1, RXD_SIO0	I	SIOF1/0 receive data	Receive data pins
	TXD_SIO1, TXD_SIO0	O	SIOF1/0 transmit data	Transmit data pins
	SIOFSYNC1, SIOFSYNC0	I/O	SIOF1/0 Frame synchronous signal	Transmit/receive frame-synchronous-signal I/O pins
I/O port	PTA7 to PTA0	I/O	General purpose I/O port A	8-bit general-purpose I/O port pins
	PTB7 to PTB0	I/O	General purpose I/O port B	8-bit general-purpose I/O port pins
	PTC7 to PTC0	I/O	General purpose I/O port C	8-bit general-purpose I/O port pins

Section 2 CPU

2.1 Processing States and Processing Modes

2.1.1 Processing States

This LSI supports four types of processing states: a reset state, an exception handling state, a program execution state, and a low-power consumption state, according to the CPU processing states.

Reset State: In the reset state, the CPU is reset. The LSI supports two types of resets: power-on reset and manual reset. For details on resets, refer to section 4, Exception Handling.

In power-on reset, the registers and internal statuses of all LSI on-chip modules are initialized. In manual reset, the register contents of a part of the LSI on-chip modules, such as the bus state controller (BSC), are retained. For details, refer to section 24, List of Registers. The CPU internal statuses and registers are initialized both in power-on reset and manual reset. After initialization, the program branches to address H'A0000000 to pass control to the reset processing program to be executed.

Exception Handling State: In the exception handling state, the CPU processing flow is changed temporarily by a general exception or interrupt exception processing. The program counter (PC) and status register (SR) are saved in the save program counter (SPC) and save status register (SSR), respectively. The program branches to an address obtained by adding a vector offset to the vector base register (VBR) and passes control to the exception processing program defined by the user to be executed. For details on reset, refer to section 4, Exception Handling.

Program Execution State: The CPU executes programs sequentially.

Low-Power Consumption State: The CPU stops operation to reduce power consumption. The low-power consumption state can be entered by executing the SLEEP instruction. For details on the low-power consumption state, refer to section 10, Power-Down Modes.

Figure 2.1 shows a status transition diagram.

2.1.2 Processing Modes

This LSI supports two processing modes: user mode and privileged mode. These processing modes can be determined by the processing mode bit (MD) of the status register (SR). If the MD bit is cleared to 0, the user mode is selected. If the MD bit is set to 1, the privileged mode is selected. The CPU enters the privileged mode by a transition to reset state or exception handling state. In the privileged mode, any registers and resources in address spaces can be accessed.

Clearing the MD bit of the SR to 0 puts the CPU in the user mode. In the user mode, some of the registers, including SR, and some of the address spaces cannot be accessed by the user program and system control instructions cannot be executed. This function effectively protects the system resources from the user program. To change the processing mode from user to privileged mode, a transition to exception handling state is required*.

Note: * To call a service routine used in privileged mode from user mode, the LSI supports an unconditional trap instruction (TRAPA). When a transition from user mode to privileged mode occurs, the contents of the SR and PC are saved. A program execution in user mode can be resumed by restoring the contents of the SR and PC. To return from an exception processing program, the LSI supports an RTE instruction.

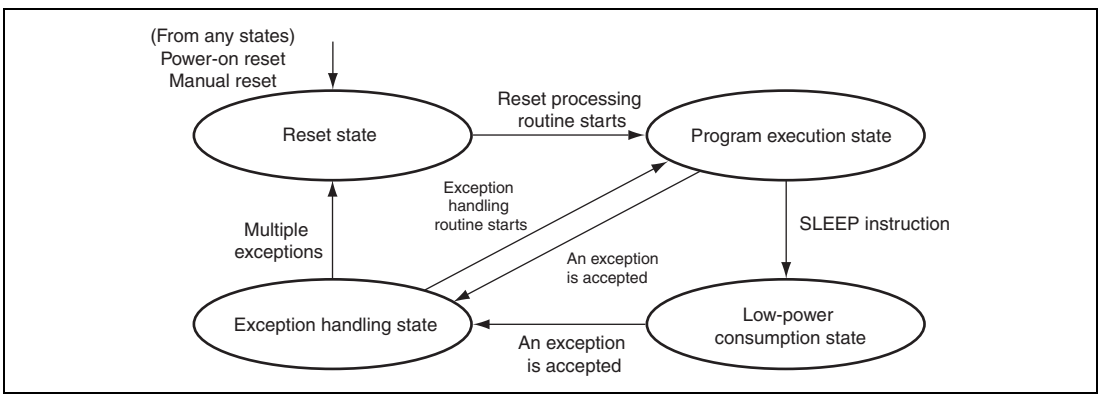


Figure 2.1 Processing State Transitions

2.2 Memory Map

2.2.1 Logical Address Space

The LSI supports 32-bit logical addresses and accesses system resources using the 4-Gbytes of logical address space. User programs and data are accessed from the logical address space. The logical address space is divided into several areas as shown in table 2.1.

P0/U0 Area: This area is called the P0 area when the CPU is in privileged mode and the U0 area when in user mode. For the P0 and U0 areas, access using the cache is enabled. The P0 and U0 areas are handled as address translatable areas.

If the cache is enabled, access to the P0 or U0 area is cached. If a P0 or U0 address is specified while the address translation unit is enabled, the P0 or U0 address is translated into a physical address based on translation information defined by the user.

If the CPU is in user mode, only the U0 area can be accessed. If P1, P2, P3, or P4 is accessed in user mode, a transition to an address error exception occurs.

P1 Area: The P1 area is defined as a cacheable but non-address translatable area. Normally, programs executed at high speed in privileged mode, such as exception processing handlers, which are at the core of the operating system (S), are assigned to the P1 area.

P2 Area: The P2 area is defined as a non-cacheable but non-address translatable area. A reset processing program to be called from the reset state is described at the start address (H'A0000000) of the P2 area. Normally, programs such as system initialization routines and OS initiation programs are assigned to the P2 area. To access a part of an on-chip I/O, its corresponding program should be assigned to the P2 area.

P3 Area: The P3 area is defined as a cacheable and address translatable area. This area is used if an address translation is required for a privileged program.

P4 Area: The P4 area is defined as a control area which is non-cacheable and non-address translatable. This area can be accessed only in privileged mode. A part of the LSI's on-chip I/O is assigned to this area.

Table 2.1 Logical Address Space

Address Range	Name	Mode	Description
H'00000000 to H'7FFFFFFF	P0/U0	Privileged/user mode	2-Gbyte physical space, cacheable, address translatable In user mode, only this address space can be accessed.
H'80000000 to H'9FFFFFFF	P1	Privileged mode	0.5-Gbyte physical space, cacheable
H'A0000000 to H'BFFFFFFF	P2	Privileged mode	0.5-Gbyte physical space, non-cacheable
H'C0000000 to H'DFFFFFFF	P3	Privileged mode	0.5-Gbyte physical space, cacheable, address translatable
H'E0000000 to H'FFFFFFFF	P4	Privileged mode	0.5-Gbyte control space, non-cacheable

2.2.2 External Memory Space

The LSI uses 29 bits of the 32-bit logical address to access external memory. In this case, 0.5-Gbyte of external memory space can be accessed. The external memory space is managed in area units. Different types of memory can be connected to each area, as shown in figure 2.2. For details, please refer to section 12, Bus State Controller (BSC).

In addition, area 1 in the external memory space is used as an on-chip I/O space where most of this LSI's on-chip I/Os are mapped.*[1]

Normally, the upper three bits of the 32-bit logical address are masked and the lower 29 bits are used for external memory addresses.*[2] For example, address H'00000100 in the P0 area, address H'80000100 in the P1 area, address H'A0000100 in the P2 area, and address H'C0000100 in the P3 area of the logical address space are mapped into address H'00000100 of area 0 in the external memory space. The P4 area in the logical address space is not mapped into the external memory address. If an address in the P4 area is accessed, an external memory cannot be accessed.

Notes: 1. To access an on-chip I/O mapped into area 1 in the external memory space, access the address from the P2 area which is not cached in the logical address space.
2. If the address translation unit is enabled, arbitrary mapping in page units can be specified. For details, refer to section 5, Memory Management Unit (MMU).

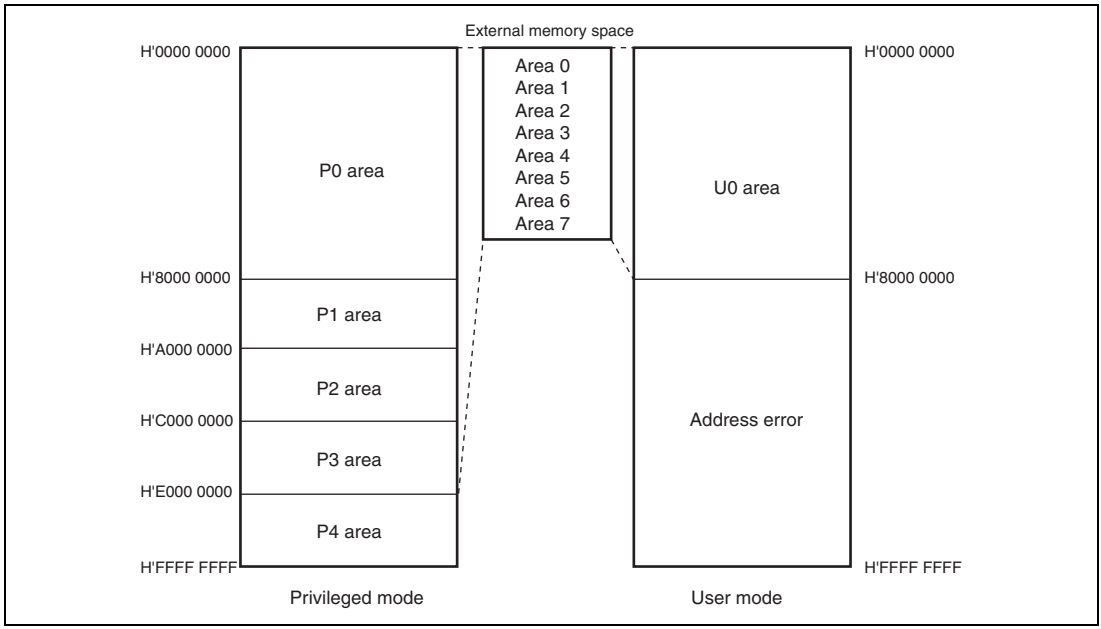


Figure 2.2 Logical Address to External Memory Space Mapping

2.3 Register Descriptions

This LSI provides thirty-three 32-bit registers: 24 general registers, five control registers, three system registers, and one program counter.

General Registers: This LSI incorporates 24 general registers: R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1 and R8 to R15. R0 to R7 are banked. The process mode and the register bank (RB) bit in the status register (SR) define which set of banked registers (R0_BANK0 to R7_BANK0 or R0_BANK1 to R7_BANK1) are accessed as general registers.

System Registers: This LSI incorporates the multiply and accumulate registers (MACH/MACL) and procedure register (PR) as system registers. These registers can be accessed regardless of the processing mode.

Program Counter: The program counter stores the value obtained by adding 4 to the current instruction address.

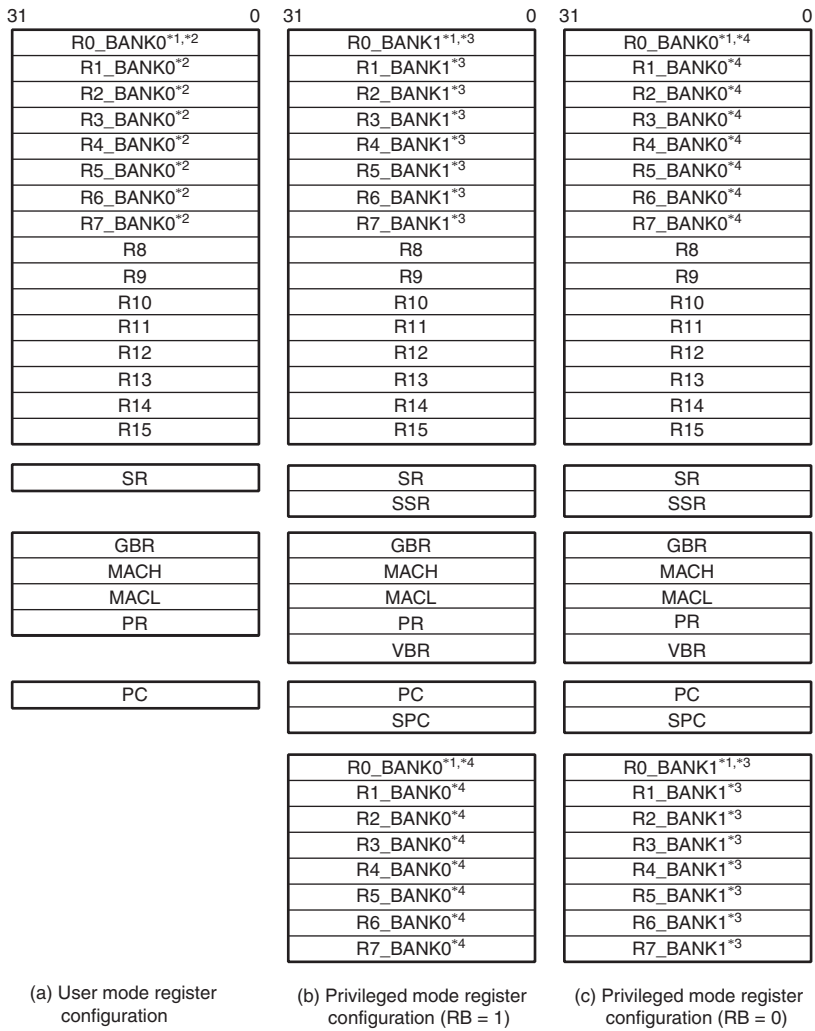
Control Registers: This LSI incorporates the status register (SR), global base register (GBR), save status register (SSR), save program counter (SPC), and vector base register as control register. Only the GBR can be accessed in user mode. Control registers other than the GBR can be accessed only in privileged mode.

Table 2.2 shows the register values after reset. Figure 2.3 shows the register configurations in each process mode.

Table 2.2 Register Initial Values

Register Type	Registers	Initial Values*
General registers	R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, R8 to R15	Undefined
System registers	MACH, MACL, PR	Undefined
Program counter	PC	H'A0000000
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3 to I0 bits = H'F (1111), reserved bits = all 0, other bits = undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000

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



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

Figure 2.3 Register Configuration in Each Processing Mode

2.3.1 General Registers

There are twenty-four 32-bit general registers: R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, and R8 to R15. R0 to R7 are banked. The process mode and the register bank (RB) bit in the status register (SR) define which set of banked registers (R0_BANK0 to R7_BANK0 or R0_BANK1 to R7_BANK1) are accessed as general registers. R0 to R7 registers in the selected bank are accessed as R0 to R7. R0 to R7 in the non-selected bank is accessed as R0_BANK to R7_BANK by the control register load instruction (LDC) and control register store instruction (STC).

In user mode, bank 0 is selected regardless of he RB bit value. Sixteen registers: R0_BANK0 to R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK1 to R7_BANK1 registers in bank 1 cannot be accessed.

In privileged mode that is entered by a transition to exception handling state, the RB bit is set to 1 to select bank 1. In privileged mode, sixteen registers: R0_BANK1 to R7_BANK1 and R8 to R15 are accessed as general registers R0 to R15. A bank is switched automatically when an exception handling state is entered, registers R0 to R7 need not be saved by the exception handling routine. The R0_BANK0 to R7_BANK0 registers in bank 0 can be accessed as R0_BANK to R7_BANK by the LDC and STC instructions.

In privileged mode, bank 0 can also be used as general registers by clearing the RB bit to 0. In this case, sixteen registers: R0_BANK0 to R7_BANK0 and R8 to R15 are accessed as general registers R0 to R15. The R0_BANK1 to R7_BANK1 registers in bank 1 can be accessed as R0_BANK to R7_BANK by the LDC and STC instructions.

The general registers R0 to R15 are used as equivalent registers for almost all instructions. In some instructions, the R0 register is automatically used or only the R0 register can be used as source or destination registers.

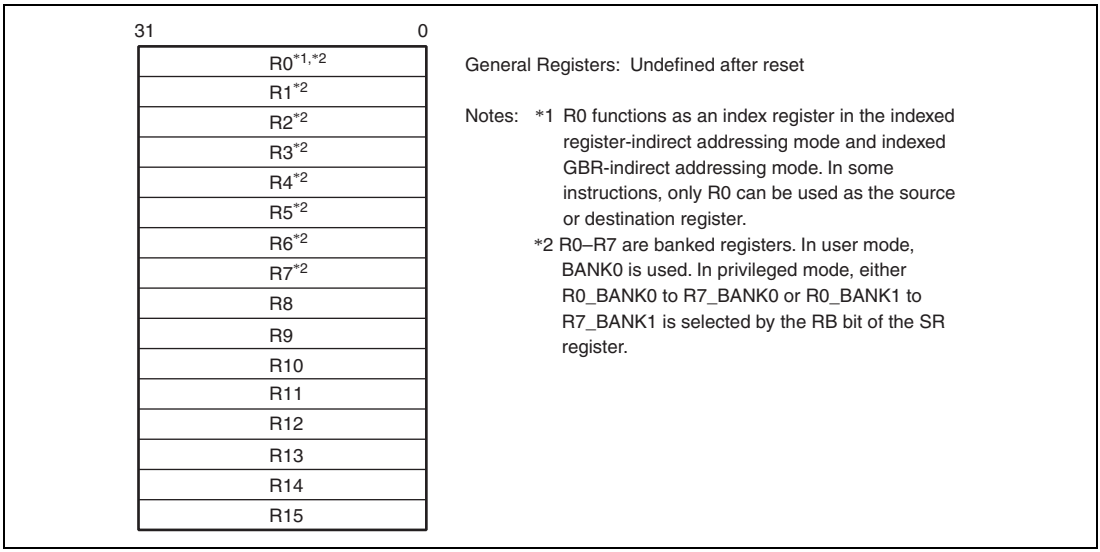


Figure 2.4 General Registers

2.3.2 System Registers

The system registers: multiply and accumulate registers (MACH/MACL) and procedure register (PR) as system registers can be accessed by the LDS and STS instructions.

Multiply and Accumulate Registers (MACH/MACL): The multiply and accumulate registers (MACH/MACL) store the results of multiplication and accumulation instructions or multiplication instructions. The MACH/MACL registers also store addition values for the multiplication and accumulations. After reset, these registers are undefined. The MACH and MACL registers store upper 32 bits and lower 32 bits, respectively.

Procedure Register (PR): The procedure register (PR) stores the return address for a subroutine call using the BSR, BSRF, or JSR instruction. The return address stored in the PR register is restored to the program counter (PC) by the RTS (return from the subroutine) instruction. After reset, this register is undefined.

2.3.3 Program Counter

The program counter (PC) stores the value obtained by adding 4 to the current instruction address. There is no instruction to read the PC directly. Before an exception handling state is entered, the PC is saved in the save program counter (SPC). Before a subroutine call is executed, the PC is saved in the procedure register (PR). In addition, the PC can be used for PC relative addressing mode.

Figure 2.5 shows the system register and program counter configurations.

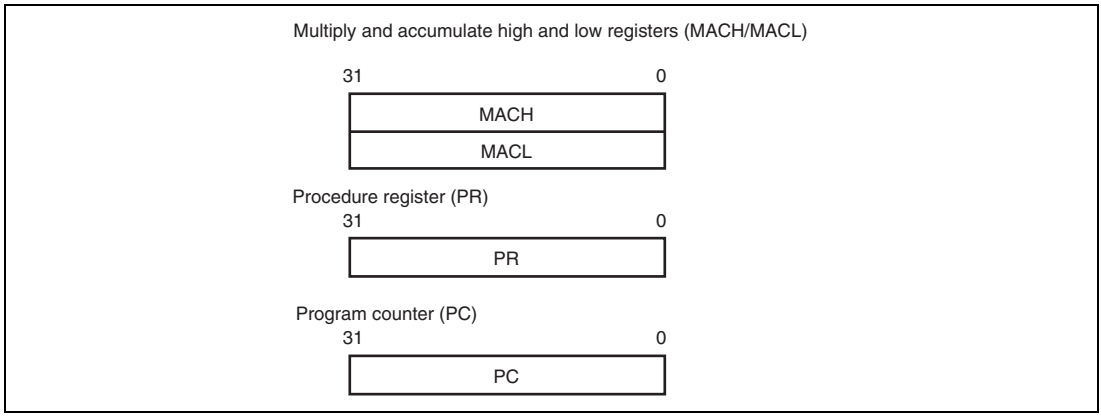


Figure 2.5 System Registers and Program Counter

2.3.4 Control Registers

The control registers (SR, GBR, SSR, SPC, and VBR) can be accessed by the LDC or STC instruction in privileged mode. The GBR register can be accessed in the user mode.

The control registers are described below.

Status Register (SR): The status register (SR) indicates the system status as shown below. The SR register can be accessed only in privileged mode.

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	MD	1	R/W	Processing Mode Indicates the CPU processing mode. 0: User mode 1: Privileged mode The MD bit is set to 1 in reset or exception handling state.
29	RB	1	R/W	Register Bank The general registers R0 to R7 are banked registers. The RB bit selects a bank used in the privileged mode. 0: Selects bank 0 registers. In this case, R0_BANK0 to R7_BANK0 and R8 to R15 are used as general registers. R0_BANK1 to R7_BANK1 can be accessed by the LDC or STR instruction. 1: Selects bank 1 registers. In this case, R0_BANK1 to R7_BANK1 and R8 to R15 are used as general registers. R0_BANK0 to R7_BANK0 can be accessed by the LDC or STR instruction. The RB bit is set to 1 in reset or exception handling state.

Bit	Bit Name	Initial Value	R/W	Description
28	BL	1	R/W	Block Specifies whether an exception, interrupt, or user break is enabled or not. 0: Enables an exception, interrupt, or user break. 1: Disables an exception, interrupt, or user break. The BL bit is set to 1 in reset or exception handling state.
27 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit These bits are used by the DIV0S, DIV0U, and DIV1 instructions. These bits can be changed even in user mode by using the DIV0S, DIV0U, and DIV1 instructions. These bits are undefined at reset. These bits do not change in an exception handling state.
7 to 4	I3 to I0	All 1	R/W	Interrupt Mask Indicates the interrupt mask level. These bits do not change even if an interrupt occurs. At reset, these bits are initialized to B'1111. These bits are not affected in an exception handling state.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	Saturation Mode Specifies the saturation mode for multiply instructions or multiply and accumulate instructions. This bit can be specified by the SETS and CLRS instructions in user mode. At reset, this bit is undefined. This bit is not affected in an exception handling state.

Bit	Bit Name	Initial Value	R/W	Description
0	T	—	R/W	T Bit Indicates true or false for compare instructions or carry or borrow occurrence for an operation instruction with carry or borrow. This bit can be specified by the SETT and CLRT instructions in user mode. At reset, this bit is undefined. This bit is not affected in an exception handling state.

Note: The M, Q, S, and T bits can be set/cleared by the user mode specific instructions. Other bits can be read or written in privileged mode.

Save Status Register (SSR): The save status register (SSR) can be accessed only in privileged mode. Before entering the exception, the contents of the SR register is stored in the SSR register. At reset, the SSR initial value is undefined.

Save Program Counter (SPC): The save program counter (SPC) can be accessed only in privileged mode. Before entering the exception, the contents of the PC is stored in the SPC. At reset, the SPC initial value is undefined.

Global Base Register (GBR): The global base register (GBR) is referenced as a base register in GBR indirect addressing mode. At reset, the GBR initial value is undefined.

Vector Base Register (VBR): The global base register (GBR) can be accessed only in privileged mode. If a transition from reset state to exception handling state occurs, this register is referenced as a base address. For details, refer to section 4, Exception Handling. At reset, the VBR is initialized as H'00000000.

Figure 2.6 shows the control register configuration.

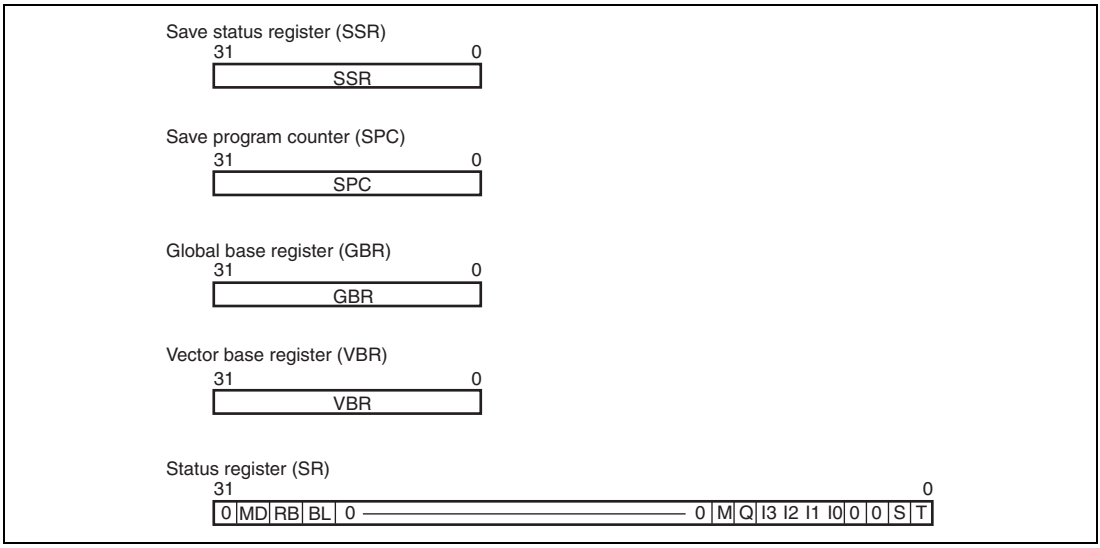
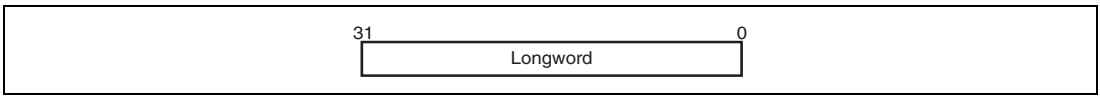


Figure 2.6 Control Register Configuration

2.4 Data Formats

2.4.1 Register Data Format

Register operands are always longwords (32 bits). When the 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.



2.4.2 Memory Data Formats

Memory data formats are classified into byte, word, and longword. Memory can be accessed in byte, word, and longword. When the 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.

An address error will occur if word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed. In such cases, the data accessed cannot be guaranteed.

When a word or longword operand is accessed, the byte positions on the memory corresponding to the word or longword data on the register is determined to the specified endian mode (big endian or little endian).

Figure 2.7 shows a byte correspondence in big endian mode. In big endian mode, the MSB byte in the register corresponds to the lowest address in the memory, and the LSB the in the register corresponds to the highest address. For example, if the contents of the general register R0 is stored at an address indicated by the general register R1 in longword, the MSB byte of the R0 is stored at the address indicated by the R1 and the LSB byte of the R1 register is stored at the address indicated by the (R1 +3).

The on-chip device registers assigned to memory are accessed in big endian mode. Note that the available access size (byte, word, or long word) differs in each register.

Note: The CPU instruction codes of this LSI must be stored in word units. In big endian mode, the instruction code must be stored from upper byte to lower byte in this order from the word boundary of the memory.

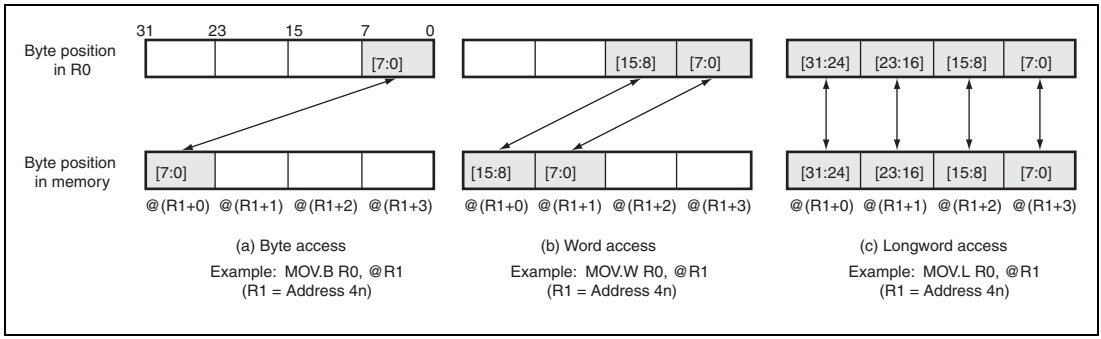


Figure 2.7 Data Format on Memory (Big Endian Mode)

The little endian mode can also be specified as data format. Either big-endian or little-endian mode can be selected according to the external pin (MD5) at a power-on reset. When MD5 is low at reset, the processor operates in big-endian mode. When MD5 is high at reset, the processor operates in little-endian mode. The endian mode cannot be modified dynamically.

In little endian mode, the MSB byte in the register corresponds to the highest address in the memory, and the LSB the in the register corresponds to the lowest address (figure 2.8). For example, if the contents of the general register R0 is stored at an address indicated by the general register R1 in longword, the MSB byte of the R0 is stored at the address indicated by the (R1+3) and the LSB byte of the R1 register is stored at the address indicated by the R1.

If the little endian mode is selected, the on-chip memory are accessed in little endian mode. However, the on-chip device registers assigned to memory are accessed in big endian mode. Note that the available access size (byte, word, or long word) differs in each register.

Note: The CPU instruction codes of this LSI must be stored in word units. In little endian mode, the instruction code must be stored from lower byte to upper byte in this order from the word boundary of the memory.

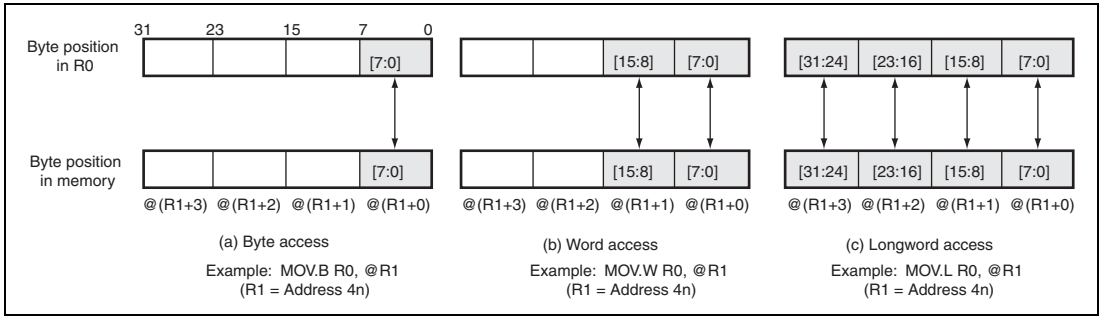


Figure 2.8 Data Format on Memory (Little Endian Mode)

Note: When the external memory is accessed through the E-DMAC or IPSEC module, big endian is supported, but little endian is not supported. Therefore, if the external memory is accessed through the E-DMAC or IPSEC module in little endian mode, data format should be converted from big endian mode to little endian mode through software.

2.5 Features of CPU Core Instructions

2.5.1 Instruction Execution Method

Instruction Length: All instructions have a fixed length of 16 bits and are executed in the sequential pipeline. In the sequential pipeline, almost all instructions can be executed in one cycle. All data items are handles in longword (32 bits). Memory can be accessed in byte, word, or longword. In this case, Memory byte or word data is sign-extended and operated on as longword data. Immediate data is sign-extended to longword size for arithmetic operations (MOV, ADD, and CMP/EQ instructions) or zero-extended to longword size for logical operations (TST, AND, OR, and XOR instructions).

Load/Store Architecture: Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly on memory.

Delayed Branching: Unconditional branch instructions are executed as delayed branches. With a delayed branch instruction, the branch is made after execution of the instruction (called the slot instruction) immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made.

This LSI supports two types of conditional branch instructions: delayed branch instruction or normal branch instruction.

```
Example:  BRA    TARGET
          ADD    R1, R0    ; ADD is executed before branching to the TARGET
```

T Bit: The result of a comparison is indicated by the T bit in the status register (SR), and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

```
Example:  ADD      #1, R0  ; The T bit cannot be modified by the ADD instruction
          CMP/EQ   #0, R0  ; The T bit is set to 1 if R0 is 0.
          BT       Target  ; Branch to TARGET if the T bit is set to 1 (R0=0).
```

Literal Constant: Byte literal constant is placed inside the instruction code as immediate data. Since the instruction length is fixed to 16 bits, word and longword literal constant is not placed inside the instruction code, but in a table in memory. The table in memory is referenced with a MOV instruction using PC-relative addressing mode with displacement.

```
Example:  MOV.W     @(disp, PC)
```

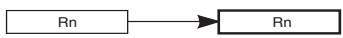
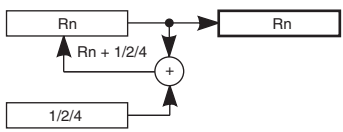
Absolute Addresses: When data is referenced by absolute address, the absolute address value is placed in a table in memory beforehand as well as word or longword literal constant. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using register indirect addressing mode.

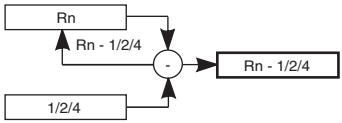
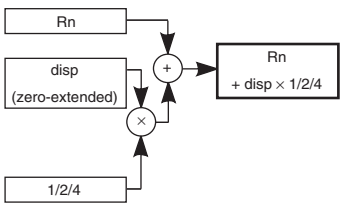
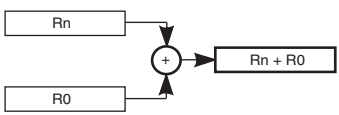
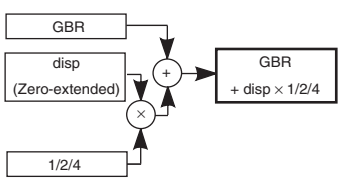
16-Bit/32-Bit Displacement: When data is referenced with a 16- or 32-bit displacement, the displacement value is placed in a table in memory beforehand. Using the method whereby word or longword immediate data is loaded when an instruction is executed, this value is transferred to a register and the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

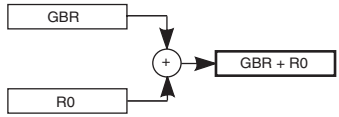
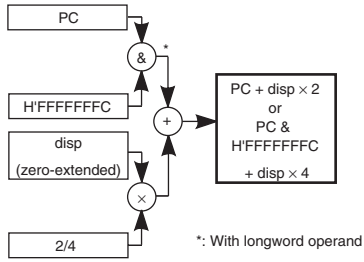
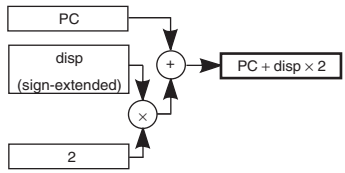
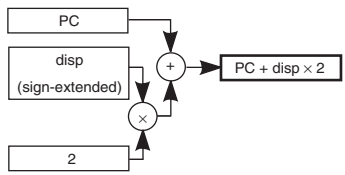
2.5.2 CPU Instruction Addressing Modes

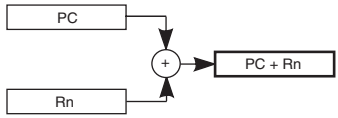
The following table shows addressing modes and effective address calculation methods for instructions executed by the CPU core.

Table 2.3 Addressing Modes and Effective Addresses for CPU Instructions

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
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn

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

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0
PC-relative with displacement	@(disp:8, PC)	Effective address is PC 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 + disp × 2 Longword: PC&H'FFFFFFFC + disp × 4
PC-relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2 	PC + disp × 2

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative	Rn	Effective address is sum of PC and Rn. 	PC + Rn
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 addressing modes with displacement (disp) as shown below, the assembler description in this manual indicates the value before it is scaled (x 1, x2, or x4) according to the operand size to clarify the LSI operation. For details on assembler description, refer to the description rules in each assembler.

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

2.5.3 CPU Instruction Formats

Table 2.4 shows the instruction formats, and the meaning of the source and destination operands, for instructions executed by the CPU core. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

xxxx: Instruction code

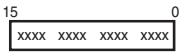
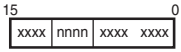
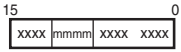
mmmm: Source register

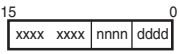
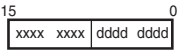
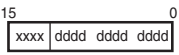
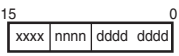
nnnn: Destination register

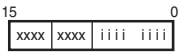
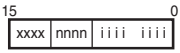
iiii: Immediate data

dddd: Displacement

Table 2.4 CPU Instruction Formats

Instruction Format	Source Operand	Destination Operand	Sample Instruction	
0 type 	—	—	NOP	
n type 	—	nnnn: register direct	MOVT	Rn
	Control register or system register	nnnn: register direct	STS	MACH,Rn
	Control register or system register	nnnn: pre-decrement register indirect	STC.L	SR,@-Rn
m type 	mmmm: register direct	Control register or system register	LDC	Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L	@Rm+,SR
	mmmm: register indirect	—	JMP	@Rm
	PC-relative using Rm	—	BRAF	Rm
nm type 	mmmm: register direct	nnnn: register direct	ADD	Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L	Rm,@Rn
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W	@Rm+,@Rn+

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn
d type 	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
	R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
	dddddddd: PC-relative	—	BF label
d12 type 	ddddddddddddd: PC-relative	—	BRA label (label=disp+PC)
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn

Instruction Format	Source Operand	Destination Operand	Sample Instruction
i type 	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

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

2.6 Instruction Set

2.6.1 CPU Instruction Set Based on Functions

The CPU instruction set consists of 68 basic instruction types divided into six functional groups, as shown in table 2.5. Tables 2.6 to 2.11 show the instruction notation, machine code, execution time, and function.

Table 2.5 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate	33
		MUL	Double-precision multiplication (32 × 32 bits)	
		MULS	Signed multiplication (16 × 16 bits)	
		MULU	Unsigned multiplication (16 × 16 bits)	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with carry	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	Logical AND and T bit setting	
		XOR	Exclusive logical OR	
Shift instructions	12	ROTL	1-bit left shift	16
		ROTR	1-bit right shift	
		ROTCL	1-bit left shift with T bit	
		ROTCR	1-bit right shift with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	
		SHAD	Arithmetic dynamic shift	
		SHLD	Logical dynamic shift	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Branch instructions	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	15	CLRT	T bit clear	75
		CLRMAC	MAC register clear	
		CLRS	S bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		LDTLB	PTEH/PTEL load into TLB	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			188

The instruction code, operation, and number of execution states of the CPU instructions are shown in the following tables, classified by instruction type, using the format shown below.

Instruction	Instruction Code	Operation	Privilege	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Indicates a privileged instruction.	Value when no wait states are inserted*1	Value of T bit after instruction is executed
Explanation of Symbols OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*2	Explanation of Symbols →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift			Explanation of Symbols —: No change

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
   a. When there is a conflict between an instruction fetch and a data access
   b. When the destination register of a load instruction (memory → register) is also used by the following instruction
2. Scaled (x1, x2, or x4) according to the instruction operand size, etc.

Table 2.6 Data Transfer Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
MOV	#imm,Rn	1110nnnniiiiiiii	Imm → Sign extension → Rn	—	1	—
MOV.W	@(disp,PC),Rn	1001nnnndddddddd	(disp x 2+PC)→Sign extension → Rn	—	1	—
MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(disp x 4+PC)→Rn	—	1	—
MOV	Rm,Rn	0110nnnnmmmm0011	Rm→Rn	—	1	—
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm→(Rn)	—	1	—
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm→(Rn)	—	1	—
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm→(Rn)	—	1	—
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm)→Sign extension→Rn	—	1	—
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm)→Sign extension→Rn	—	1	—
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm)→Rn	—	1	—
MOV.B	Rm,@–Rn	0010nnnnmmmm0100	Rn–1→Rn, Rm→(Rn)	—	1	—
MOV.W	Rm,@–Rn	0010nnnnmmmm0101	Rn–2→Rn, Rm→(Rn)	—	1	—
MOV.L	Rm,@–Rn	0010nnnnmmmm0110	Rn–4→Rn, Rm→(Rn)	—	1	—
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm)→Sign extension→Rn, Rm+1→Rm	—	1	—
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm)→Sign extension→Rn, Rm+2→Rm	—	1	—
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm)→Rn, Rm+4→Rm	—	1	—
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0→(disp+Rn)	—	1	—
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0→(disp x 2+Rn)	—	1	—
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm→(disp x 4+Rn)	—	1	—
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp+Rm)→Sign extension→R0	—	1	—
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp x 2+Rm)→Sign extension→R0	—	1	—
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp x 4+Rm)→Rn	—	1	—
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm→(R0+Rn)	—	1	—
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm→(R0+Rn)	—	1	—
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm→(R0+Rn)	—	1	—

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0+Rm)→Sign extension→Rn	—	1	—
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0+Rm)→Sign extension→Rn	—	1	—
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0+Rm)→Rn	—	1	—
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0→(disp+GBR)	—	1	—
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0→(disp x 2+GBR)	—	1	—
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0→(disp x 4+GBR)	—	1	—
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp+GBR)→Sign extension→R0	—	1	—
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp x 2+GBR)→Sign extension→R0	—	1	—
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp x 4+GBR)→R0	—	1	—
MOVA	@(disp,PC),R0	11000111dddddddd	disp x 4+PC→R0	—	1	—
MOVT	Rn	0000nnnn00101001	T→Rn	—	1	—
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm→Swap lowest two bytes→Rn	—	1	—
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm→Swap two consecutive words→Rn	—	1	—
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn →Rn	—	1	—

Table 2.7 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
ADD	Rm,Rn	0011nnnnmmmm1100	Rn+Rm→Rn	—	1	–
ADD	#imm,Rn	0111nnnniiiiiiii	Rn+imm→Rn	—	1	–
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn+Rm+T→Rn, Carry→T	—	1	Carry
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn+Rm→Rn, Overflow→T	—	1	Overflow
CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	—	1	Comparison result
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	—	1	Comparison result
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn ≥ Rm with unsigned data, 1 → T	—	1	Comparison result
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn ≥ Rm with signed data, 1 → T	—	1	Comparison result
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	—	1	Comparison result
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	—	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If Rn ≥ 0, 1 → T	—	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If Rn > 0, 1 → T	—	1	Comparison result
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, 1 → T	—	1	Comparison result
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn/Rm)	—	1	Calculation result
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	—	1	Calculation result
DIV0U		0000000000011001	0 → M/Q/T	—	1	0
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	—	2 (to 5)*	–
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	—	2 (to 5)*	–

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	—	1	Comparison result
EXTS.B	Rm,Rn	0110nnnnmmmm1110	A byte in Rm is sign-extended → Rn	—	1	—
EXTS.W	Rm,Rn	0110nnnnmmmm1111	A word in Rm is sign-extended → Rn	—	1	—
EXTU.B	Rm,Rn	0110nnnnmmmm1100	A byte in Rm is zero-extended → Rn	—	1	—
EXTU.W	Rm,Rn	0110nnnnmmmm1101	A word in Rm is zero-extended → Rn	—	1	—
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC,Rn + 4 → Rn, Rm + 4 → Rm, 32 × 32 + 64 → 64 bits	—	2 (to 5)*	—
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC,Rn + 2 → Rn, Rm + 2 → Rm, 16 × 16 + 64 → 64 bits	—	2 (to 5)*	—
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL, 32 × 32 → 32 bits	—	2 (to 5)*	—
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL, 16 × 16 → 32 bits	—	1(to 3)*	–
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL, 16 × 16 → 32 bits	—	1(to 3)*	—
NEG	Rm,Rn	0110nnnnmmmm1011	0–Rm→Rn	—	1	—
NEGC	Rm,Rn	0110nnnnmmmm1010	0–Rm–T→Rn, Borrow→T	—	1	Borrow
SUB	Rm,Rn	0011nnnnmmmm1000	Rn–Rm→Rn	—	1	—
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn–Rm–T→Rn, Borrow →T	—	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn–Rm→Rn, Underflow→T	—	1	Underflow

Note: * The number of execution cycles indicated within the parentheses ( ) are required when the operation result is read from the MACH/MACL register immediately after the instruction.

Table 2.8 Logic Operation Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	—	1	—
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	—	1	—
AND.B	#imm,@(R0, GBR)	11001101iiiiiiii	(R0+GBR) & imm → (R0+GBR)	—	3	—
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	—	1	—
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	—	1	—
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	—	1	—
OR.B	#imm,@(R0, GBR)	11001111iiiiiiii	(R0+GBR) | imm → (R0+GBR)	—	3	—
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	—	4	Test result
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	—	1	Test result
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	—	1	Test result
TST.B	#imm,@(R0, GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	—	3	Test result
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	—	1	—
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	—	1	—
XOR.B	#imm,@(R0, GBR)	11001110iiiiiiii	(R0+GBR) ^ imm → (R0+GBR)	—	3	—

Table 2.9 Shift Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
ROTL	Rn	0100nnnn00000100	T←Rn←MSB	—	1	MSB
ROTR	Rn	0100nnnn00000101	LSB→Rn→T	—	1	LSB
ROTCL	Rn	0100nnnn00100100	T←Rn←T	—	1	MSB
ROTCR	Rn	0100nnnn00100101	T→Rn→T	—	1	LSB
SHAD	Rm, Rn	0100nnnnmmmm1100	Rm ≥ 0: Rn << Rm → Rn Rm < 0: Rn >> Rm → [MSB → Rn]	—	1	—
SHAL	Rn	0100nnnn00100000	T←Rn←0	—	1	MSB
SHAR	Rn	0100nnnn00100001	MSB→Rn→T	—	1	LSB
SHLD	Rm, Rn	0100nnnnmmmm1101	Rm ≥ 0: Rn << Rm → Rn Rm < 0: Rn >> Rm → [0 → Rn]	—	1	—
SHLL	Rn	0100nnnn00000000	T←Rn←0	—	1	MSB
SHLR	Rn	0100nnnn00000001	0→Rn→T	—	1	LSB
SHLL2	Rn	0100nnnn00001000	Rn<<2 → Rn	—	1	—
SHLR2	Rn	0100nnnn00001001	Rn>>2 → Rn	—	1	—
SHLL8	Rn	0100nnnn00011000	Rn<<8 → Rn	—	1	—
SHLR8	Rn	0100nnnn00011001	Rn>>8 → Rn	—	1	—
SHLL16	Rn	0100nnnn00101000	Rn<<16 → Rn	—	1	—
SHLR16	Rn	0100nnnn00101001	Rn>>16 → Rn	—	1	—

Table 2.10 Branch Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
BF	label	10001011dddddddd	If T = 0, disp × 2 + PC → PC; if T = 1, nop	—	3/1*	—
BF/S	label	10001111dddddddd	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	—	2/1*	—
BT	label	10001001dddddddd	If T = 1, disp × 2 + PC → PC; if T = 0, nop	—	3/1*	—
BT/S	label	10001101dddddddd	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	—	2/1*	—
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	—	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch,Rm + PC → PC	—	2	—
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	—	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	—	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	—	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	—	2	—
RTS		0000000000001011	Delayed branch, PR → PC	—	2	—

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

Table 2.11 System Control Instructions

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
CLRMAC		0000000000101000	0→MACH,MACL	—	1	—
CLRS		0000000001001000	0→S	—	1	—
CLRT		0000000000001000	0→T	—	1	0
LDC	Rm,SR	0100mmmm00001110	Rm→SR	√	6	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm→GBR	—	4	—
LDC	Rm,VBR	0100mmmm00101110	Rm→VBR	√	4	—
LDC	Rm,SSR	0100mmmm00111110	Rm→SSR	√	4	—
LDC	Rm,SPC	0100mmmm01001110	Rm→SPC	√	4	—
LDC	Rm,R0_BANK	0100mmmm10001110	Rm→R0_BANK	√	4	—
LDC	Rm,R1_BANK	0100mmmm10011110	Rm→R1_BANK	√	4	—
LDC	Rm,R2_BANK	0100mmmm10101110	Rm→R2_BANK	√	4	—
LDC	Rm,R3_BANK	0100mmmm10111110	Rm→R3_BANK	√	4	—
LDC	Rm,R4_BANK	0100mmmm11001110	Rm→R4_BANK	√	4	—
LDC	Rm,R5_BANK	0100mmmm11011110	Rm→R5_BANK	√	4	—
LDC	Rm,R6_BANK	0100mmmm11101110	Rm→R6_BANK	√	4	—
LDC	Rm,R7_BANK	0100mmmm11111110	Rm→R7_BANK	√	4	—
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	√	8	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	—	4	—
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	√	4	—
LDC.L	@Rm+,SSR	0100mmmm00110111	(Rm)→SSR,Rm+4→Rm	√	4	—
LDC.L	@Rm+,SPC	0100mmmm01000111	(Rm)→SPC,Rm+4→Rm	√	4	—
LDC.L	@Rm+, R0_BANK	0100mmmm10000111	(Rm)→R0_BANK,Rm+4→Rm	√	4	—
LDC.L	@Rm+, R1_BANK	0100mmmm10010111	(Rm)→R1_BANK,Rm+4→Rm	√	4	—
LDC.L	@Rm+, R2_BANK	0100mmmm10100111	(Rm)→R2_BANK,Rm+4→Rm	√	4	—
LDC.L	@Rm+, R3_BANK	0100mmmm10110111	(Rm)→R3_BANK, Rm+4→Rm	√	4	—

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
LDC.L	@Rm+, R4_BANK	0100mmmm11000111	(Rm)→R4_BANK, Rm+4→Rm	√	4	—
LDC.L	@Rm+, R5_BANK	0100mmmm11010111	(Rm)→R5_BANK, Rm+4→Rm	√	4	—
LDC.L	@Rm+, R6_BANK	0100mmmm11100111	(Rm)→R6_BANK, Rm+4→Rm	√	4	—
LDC.L	@Rm+, R7_BANK	0100mmmm11110111	(Rm)→R7_BANK, Rm+4→Rm	√	4	—
LDS	Rm,MACH	0100mmmm00001010	Rm→MACH	—	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm→MACL	—	1	—
LDS	Rm,PR	0100mmmm00101010	Rm→PR	—	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm)→MACH, Rm+4→Rm	—	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm)→MACL, Rm+4→Rm	—	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm)→PR, Rm+4→Rm	—	1	—
LDTLB		0000000000111000	PTEH/PTEL→TLB	√	1	—
NOP		0000000000001001	No operation	–	1	—
PREF	@Rm	0000mmmm10000011	(Rm) → cache	–	1	—
RTE		0000000000101011	Delayed branch, SSR → SR, SPC → PC	√	5	—
SETS		0000000001011000	1→S	—	1	—
SETT		0000000000011000	1→T	—	1	1
SLEEP		0000000000011011	Sleep	√	4*[1]	—
STC	SR,Rn	0000nnnn00000010	SR→Rn	√	1	—
STC	GBR,Rn	0000nnnn00010010	GBR→Rn	—	1	—
STC	VBR,Rn	0000nnnn00100010	VBR→Rn	√	1	—
STC	SSR, Rn	0000nnnn00110010	SSR→Rn	√	1	—
STC	SPC,Rn	0000nnnn01000010	SPC→Rn	√	1	—
STC	R0_BANK,Rn	0000nnnn10000010	R0_BANK→Rn	√	1	—
STC	R1_BANK,Rn	0000nnnn10010010	R1_BANK→Rn	√	1	—
STC	R2_BANK,Rn	0000nnnn10100010	R2_BANK→Rn	√	1	—
STC	R3_BANK,Rn	0000nnnn10110010	R3_BANK→Rn	√	1	—
STC	R4_BANK,Rn	0000nnnn11000010	R4_BANK→Rn	√	1	—

Instruction		Instruction Code	Operation	Privileged Mode	Cycles	T Bit
STC	R5_BANK,Rn	0000nnnn11010010	R5_BANK→Rn	√	1	—
STC	R6_BANK,Rn	0000nnnn11100010	R6_BANK→Rn	√	1	—
STC	R7_BANK,Rn	0000nnnn11110010	R7_BANK→Rn	√	1	—
STC.L	SR,@–Rn	0100nnnn00000011	Rn–4→Rn, SR→(Rn)	√	1	—
STC.L	GBR,@–Rn	0100nnnn00010011	Rn–4→Rn, GBR→(Rn)	–	1	—
STC.L	VBR,@–Rn	0100nnnn00100011	Rn–4→Rn, VBR→(Rn)	√	1	—
STC.L	SSR,@–Rn	0100nnnn00110011	Rn–4→Rn, SSR→(Rn)	√	1	—
STC.L	SPC,@–Rn	0100nnnn01000011	Rn–4→Rn, SPC→(Rn)	√	1	—
STC.L	R0_BANK,@–Rn	0100nnnn10000011	Rn–4→Rn, R0_BANK→(Rn)	√	1	—
STC.L	R1_BANK,@–Rn	0100nnnn10010011	Rn–4→Rn, R1_BANK→(Rn)	√	1	—
STC.L	R2_BANK,@–Rn	0100nnnn10100011	Rn–4→Rn, R2_BANK→(Rn)	√	1	—
STC.L	R3_BANK,@–Rn	0100nnnn10110011	Rn–4→Rn, R3_BANK→(Rn)	√	1	—
STC.L	R4_BANK,@–Rn	0100nnnn11000011	Rn–4→Rn, R4_BANK→(Rn)	√	1	—
STC.L	R5_BANK,@–Rn	0100nnnn11010011	Rn–4→Rn, R5_BANK→(Rn)	√	1	—
STC.L	R6_BANK,@–Rn	0100nnnn11100011	Rn–4→Rn, R6_BANK→(Rn)	√	1	—
STC.L	R7_BANK,@–Rn	0100nnnn11110011	Rn–4→Rn, R7_BANK→(Rn)	√	1	—
STS	MACH,Rn	0000nnnn00001010	MACH→Rn	—	1	—
STS	MACL,Rn	0000nnnn00011010	MACL→Rn	—	1	—
STS	PR,Rn	0000nnnn00101010	PR→Rn	—	1	—
STS.L	MACH,@–Rn	0100nnnn00000010	Rn–4→Rn, MACH→(Rn)	—	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn–4→Rn, MACL→(Rn)	—	1	—
STS.L	PR,@–Rn	0100nnnn00100010	Rn–4→Rn, PR→(Rn)	—	1	—
TRAPA	#imm	11000011iiiiiiii	Unconditional trap exception occurs*2	—	8	—

Notes: 1. Number of states before the chip enters the sleep state.
2. For details, refer to section 4, Exception Handling.

2.6.2 Operation Code Map

Table 2.12 shows the operation code map.

Table 2.12 Operation Code Map

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB		LSB		MD: 00	MD: 01	MD: 10	MD: 11
0000	Rn	Fx	0000				
0000	Rn	Fx	0001				
0000	Rn	00MD	0010	STC SR, Rn	STC GBR, Rn	STC VBR, Rn	STC SSR, Rn
0000	Rn	01MD	0010	STC SPC, Rn			
0000	Rn	10MD	0010	STC R0_BANK, Rn	STC R1_BANK, Rn	STC R2_BANK, Rn	STC R3_BANK, Rn
0000	Rn	11MD	0010	STC R4_BANK, Rn	STC R5_BANK, Rn	STC R6_BANK, Rn	STC R7_BANK, Rn
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rm	10MD	0011	PREF @Rm			
0000	Rn	Rm	01MD	MOV.B Rm, @(R0, Rn)	MOV.W Rm, @(R0, Rn)	MOV.L Rm, @(R0, Rn)	MUL.L Rm, Rn
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	Fx	1010	STS MACH, Rn	STS MACL, Rn	STS PR, Rn	
0000	Rn	Fx	1011				
0000	Rn	Rm	11MD	MOV. B @(R0, Rm), Rn	MOV.W @(R0, Rm), Rn	MOV.L @(R0, Rm), Rn	MAC.L @Rm+,@Rn+
0001	Rn	Rm	disp	MOV.L Rm, @(disp:4, Rn)			
0010	Rn	Rm	00MD	MOV.B Rm, @Rn	MOV.W Rm, @Rn	MOV.L Rm, @Rn	

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0010				Rn	Rm	01MD	MOV.B Rm, @–Rn
0010				Rn	Rm	10MD	TST Rm, Rn
0010				Rn	Rm	11MD	CMP/STR Rm, Rn
0011	Rn	Rm	00MD	CMP/EQ Rm, Rn		CMP/HS Rm, Rn	CMP/GE Rm, Rn
0011	Rn	Rm	01MD	DIV1 Rm, Rn	DMULU.L Rm,Rn	CMP/HI Rm, Rn	CMP/GT Rm, Rn
0011	Rn	Rm	10MD	SUB Rm, Rn		SUBC Rm, Rn	SUBV Rm, Rn
0011	Rn	Rm	11MD	ADD Rm, Rn	DMULS.L Rm,Rn	ADDC Rm, Rn	ADDV Rm, Rn
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn	
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn	
0100	Rn	Fx	0010	STS.L MACH, @–Rn	STS.L MACL, @–Rn	STS.L PR, @–Rn	
0100	Rn	00MD	0011	STC.L SR, @–Rn	STC.L GBR, @–Rn	STC.L VBR, @–Rn	STC.L SSR, @–Rn
0100	Rn	01MD	0011	STC.L SPC, @–Rn			
0100	Rn	10MD	0011	STC.L R0_BANK, @–Rn	STC.L R1_BANK, @–Rn	STC.L R2_BANK, @–Rn	STC.L R3_BANK, @–Rn
0100	Rn	11MD	0011	STC.L R4_BANK, @–Rn	STC.L R5_BANK, @–Rn	STC.L R6_BANK, @–Rn	STC.L R7_BANK, @–Rn
0100	Rn	Fx	0100	ROTL Rn		ROTCL Rn	
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn	
0100	Rm	Fx	0110	LDS.L @Rm+, MACH	LDS.L @Rm+, MACL	LDS.L @Rm+, PR	
0100	Rm	00MD	0111	LDC.L @Rm+, SR	LDC.L @Rm+, GBR	LDC.L @Rm+, VBR	LDC.L @Rm+, SSR
0100	Rm	01MD	0111	LDC.L @Rm+, SPC			
0100	Rm	10MD	0111	LDC.L @Rm+, R0_BANK	LDC.L @Rm+, R1_BANK	LDC.L @Rm+, R2_BANK	LDC.L @Rm+, R3_BANK
0100	Rm	11MD	0111	LDC.L @Rm+, R4_BANK	LDC.L @Rm+, R5_BANK	LDC.L @Rm+, R6_BANK	LDC.L @Rm+, R7_BANK
0100	Rn	Fx	1000	SHLL2 Rn	SHLL8 Rn	SHLL16 Rn	

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB		LSB		MD: 00	MD: 01	MD: 10	MD: 11
0100	Rn	Fx	1001	SHLR2 Rn			
0100	Rm	Fx	1010	LDS Rm, MACH			
0100	Rm/Rn	Fx	1011	JSR @Rm			
0100	Rn	Rm	1100	SHAD Rm, Rn			
0100	Rn	Rm	1101	SHLD Rm, Rn			
0100	Rm	00MD	1110	LDC Rm, SR	LDC Rm, GBR	LDC Rm, VBR	LDC Rm, SSR
0100	Rm	01MD	1110	LDC Rm, SPC			
0100	Rm	10MD	1110	LDC Rm, R0_BANK	LDC Rm, R1_BANK	LDC Rm, R2_BANK	LDC Rm, R3_BANK
0100	Rm	11MD	1110	LDC Rm, R4_BANK	LDC Rm, R5_BANK	LDC Rm, R6_BANK	LDC Rm, R7_BANK
0100	Rn	Rm	1111	MAC.W @Rm+, @Rn+			
0101	Rn	Rm	disp	MOV.L @ (disp:4, Rm), Rn			
0110	Rn	Rm	00MD	MOV.B @Rm, Rn	MOV.W @Rm, Rn	MOV.L @Rm, Rn	MOV Rm, Rn
0110	Rn	Rm	01MD	MOV.B @Rm+, Rn	MOV.W @Rm+, Rn	MOV.L @Rm+, Rn	NOT Rm, Rn
0110	Rn	Rm	10MD	SWAP.B Rm, Rn	SWAP.WRm, Rn	NEGC Rm, Rn	NEG Rm, Rn
0110	Rn	Rm	11MD	EXTU.B Rm, Rn	EXTU.W Rm, Rn	EXTS.B Rm, Rn	EXTS.WRm, Rn
0111	Rn	imm		ADD # imm : 8, Rn			
1000	00MD	Rn	disp	MOV. B R0, @(disp: 4, Rn)	MOV. W R0, @(disp: 4, Rn)		
1000	01MD	Rm	disp	MOV.B @(disp:4, Rm), R0	MOV.W @(disp: 4, Rm), R0		
1000	10MD	imm/disp		CMP/EQ #imm:8, R0	BT disp: 8		BF disp: 8
1000	11MD	imm/disp			BT/S disp: 8		BF/S disp: 8
1001	Rn	disp		MOV.W @ (disp : 8, PC), Rn			
1010	disp			BRA disp: 12			
1011	disp			BSR disp: 12			

Instruction Code			Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB		LSB	MD: 00	MD: 01	MD: 10	MD: 11
1100	00MD	imm/disp	MOV.B R0, @(disp: 8, GBR)	MOV.W R0, @(disp: 8, GBR)	MOV.L R0, @(disp: 8, GBR)	TRAPA #imm: 8
1100	01MD	disp	MOV.B @(disp: 8, GBR), R0	MOV.W @(disp: 8, GBR), R0	MOV.L @(disp: 8, GBR), R0	MOVA @(disp: 8, PC), R0
1100	10MD	imm	TST #imm: 8, R0	AND #imm: 8, R0	XOR #imm: 8, R0	OR #imm: 8, R0
1100	11MD	imm	TST.B #imm: 8, @(R0, GBR)	AND.B #imm: 8, @(R0, GBR)	XOR.B #imm: 8, @(R0, GBR)	OR.B #imm: 8, @(R0, GBR)
1101	Rn	disp	MOV.L @(disp: 8, PC), Rn			
1110	Rn	imm	MOV #imm:8, Rn			
1111	**********					

Note: For details, refer to the SH-3/SH-3E/SH3-DSP Programming Manual.

Section 3 DSP Operating Unit

3.1 DSP Extended Functions

This LSI incorporates a DSP unit and X/Y memory directly connected to the DSP unit. This LSI supports the DSP extended function instruction sets needed to control the DSP unit and X/Y memory. The DSP extended function instructions are divided into four groups.

Extended System Control Instructions for the CPU: If the DSP extended function is enabled, the following extended system control instructions can be used for the CPU.

- Repeat loop control instructions and repeat loop control register access instructions are added. Looped programs can be executed efficiently by using the zero-overhead repeat control unit. For details, refer to section 3.3, CPU Extended Instructions.
- Modulo addressing control instructions and control register access instructions are added. Function allows access to data with a circular structure. For details, refer to section 3.4, DSP Data Transfer Instructions.
- DSP unit register access instructions are added. Some of the DSP unit registers can be used in the same way as the CPU system registers. For details, refer to section 3.4, DSP Data Transfer Instructions.

Data Transfer Instructions for Data Transfers between DSP Unit Registers and On-Chip X/Y memory: Data transfer instructions for data transfers between the DSP unit registers and on-chip X/Y memory are called double-data transfer instructions. Instruction codes for these double-transfer instructions are 16 bit codes as well as CPU instruction codes. These data transfer instructions perform data transfers between the DSP unit and on-chip X/Y memory that is directly connected to the DSP unit. These data transfer instructions can be described in combination with other DSP unit operation instructions. For details, refer to section 3.4, DSP Data Transfer Instructions.

Data Transfer Instructions for Data Transfers between DSP Unit Registers and All Logical Address Spaces: Data transfer instructions for data transfers between DSP unit registers and all logical address spaces are called single-data transfer instructions. Instruction codes for the double-transfer instructions are 16 bit codes as well as CPU instruction codes. These data transfer instructions performs data transfers between the DSP unit registers and all logical address spaces. For details, refer to section 3.4, DSP Data Transfer Instructions.

DSP Unit Operation Instructions: DSP unit operation instructions are called DSP data operation instructions. These instructions are provided to execute digital signal processing operations at high speed using the DSP. Instruction codes for these instructions are 32 bits. The DSP data operation

instruction fields consist of two fields: field A and field B. In field A, a function for double data transfer instructions can be descried. In field B, ALU operation instructions and multiply instructions can be described. The instructions described in fields A and B can be executed in parallel. A maximum of four instructions (ALU operation, multiply, and two data transfers) can be executed in parallel. For details, refer to section 3.5, DSP Data Operation Instructions.

Notes: 1. 32-bit instruction codes are handled as two consecutive 16-bit instruction codes. Accordingly, 32-bit instruction codes can be assigned to a word boundary. 32-bit instruction codes must be stored in memory, upper word and lower word, in this order, in word units.

2. In little endian, the upper and lower words must be stored in memory as data to be accessed in word units.

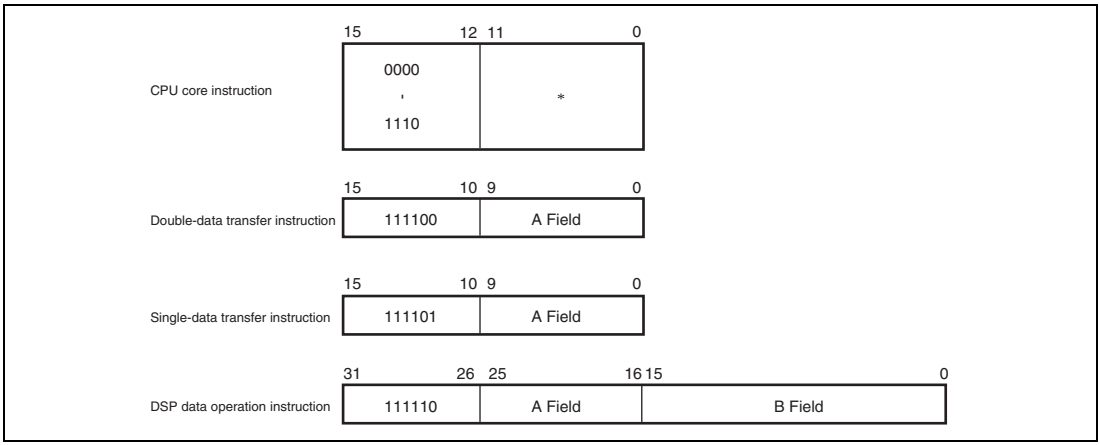


Figure 3.1 DSP Instruction Format

3.2 DSP Mode Resources

3.2.1 Processing Modes

The CPU processing modes can be extended using the mode bit (MD) and DSP bit (DSP) of the status register (SR), as shown below.

			Description	
MD	DSP	Processing Mode	Access of Resources Protected in Privileged Mode or Privileged Instruction Execution	DSP Extended Functions
0	0	User mode	Prohibited	Invalid
0	1	User DSP mode	Prohibited	Valid
1	0	Privileged mode	Allowed	Invalid
1	1	Privileged DSP mode	Allowed	Valid

As shown above, the extension of the DSP function by the DSP bit can be specified independently of the control by the MD bit. Note, however, that the DSP bit can be modified only in privileged mode. Before the DSP bit is modified, a transition to privileged mode or privileged DSP mode is necessary.

3.2.2 DSP Mode Memory Map

In DSP mode, a part of the P2 area in the logical address space can be accessed in user DSP mode. When this area is accessed in user DSP mode, this area is referred to as a Uxy area. X/Y memory is then assigned to this Uxy area. Accordingly, X/Y memory can also be accessed in user DSP mode.

Table 3.1 Logical Address Space

Address Range	Name	Protection	Description
H'A5000000 to H'A5FFFFFF	P2/Uxy	Privileged or DSP	16-Mbyte physical address space, non-cacheable, non-address translatable Can be accessed in privileged mode, privileged DSP mode, and user DSP mode

3.2.3 CPU Register Sets

In DSP mode, the status register (SR) in the CPU unit is extended to add control bits and three control registers: a repeat start register (SR), repeat end register (RE), and module register are added as control registers.

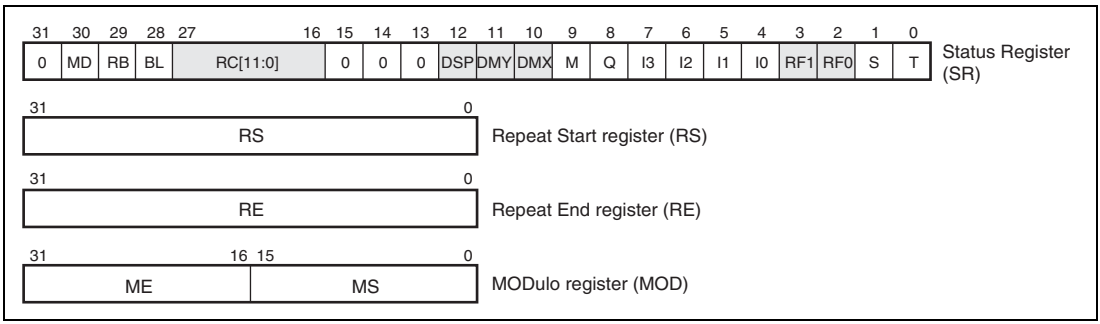


Figure 3.2 CPU Registers in DSP Mode

Extension of Status Register (SR): In DSP mode, the following control bits are added to the status register (SR). These added bits are called DSP extension bits. These DSP extension bits are valid only in DSP mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	—	—	For details, refer to section 2, CPU.
27 to 16	RC11 to RC0	All 0	R/W	Repeat Counter Holds the number of repeat times in order to perform loop control, and can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, this bit is initialized to 0. This bit is not affected in the exception handling state.
15 to 13	—	—	—	For details, refer to section 2, CPU.
12	DSP	0	R/W	DSP Bit Enables or disables the DSP extended functions. If this bit is set to 1, the DSP extended functions are enabled. This bit can be modified in privileged mode or privileged DSP mode. This bit cannot be modified in user DSP mode. At reset, this bit is initialized to 0. This bit is not affected in the exception handling state.

Bit	Bit Name	Initial Value	R/W	Description
11	DMY	0	R/W	Modulo Control Bits
10	DMX	0	R/W	Enable or disable modulo addressing for X/Y memory access. These bits can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, these bits are initialized to 0. These bits are not affected in the exception handling state.
9 to 4	—	—	—	For details, refer to section 2, CPU.
3	FR1	0	R/W	Repeat Flag Bits
2	FR0	0	R/W	Used by repeat control instructions. These bits can be modified in privileged mode, privileged DSP mode, or user DSP mode. At reset, these bits are initialized to 0. These bits are not affected in the exception handling state.
1 to 0	—	—	—	For details, refer to section 2, CPU.

Note: When data is written to the SR register, 0 should be written to bits that are specified as 0.

Repeat Start Register (RS): The repeat start register (RS) holds the start address of a loop repeat module that is controlled by the repeat function. This register can be accessed in DSP mode. At reset, the initial value of this register is undefined. This register is not affected in the exception handling state.

Repeat End Register (RE): The repeat end register (RE) holds the end address of a loop repeat module that is controlled by the repeat function. This register can be accessed in DSP mode. At reset, this register is initialized to 0. This register is not affected in the exception handling state.

Modulo Register (MOD): The modulo register stores the modulo end address and modulo start address for modulo addressing in upper and lower 16 bits. The upper and lower 16 bits of the modulo register are referred to as the ME register and MS register, respectively. This register can be accessed in DSP mode. At reset, the initial value of this register is undefined. This register is not affected in the exception handling state.

The above registers can be accessed by the control register load instruction (LDC) and store instruction (STC). Note that the LDC and STC instructions for the RS, RE, and MOD registers can be used only in privileged DSP mode and user DSP mode. The LDC and STC instruction for the SR register can be executed only when the MD bit is set to 1 or in user DSP mode. Note, however, that the LDC and STC instructions can modify only the RC11 to RC0, RF1 to RF0, DMX, and DMY bits in the SR, as described below.

- In user mode, if the LCD and STC instructions are used for the RS, an illegal instruction exception occurs.

- In privileged and privileged DSP modes, all SR bits can be modified.
- In user DSP mode, the SR can be read by the STC instruction.
- In user DSP mode, the LDC instruction can be issued to the SR but only the DSP extension bits can be modified.

Table 3.2 Operation of SR Bits in Each Processing Mode

Field	Privileged Mode MD = 1 & DSP = 0	User Mode MD = 0 & DSP = 0	Privileged DSP Mode MD = 1 & DSP = 1	User DSP Mode MD = 0 & DSP = 1	Access to DSP-Related Bit with Dedicated Instruction	Initial Value after Reset
MD	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RB	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
BL	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1
RC [11:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	000000000000
DSP	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		0
DMY	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK		0
DMX	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK		0
Q	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
M	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
I[3:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		1111
RF[1:0]	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	x
S	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x
T	S: OK, L: OK	S, L: Invalid instruction	S: OK, L: OK	S: OK, L: NG		x

[Legend]

S: STC instruction

L: LDC instruction

OK: STC/LDC operation is enabled.

Invalid instruction: Exception occurs when an invalid instruction is executed.

NG: Previous value is retained. No change.

x: Undefined

Before entering the exception handling state, all bits including the DSP extension bits of the SR registers are saved in the SSR. Before returning from the exception handling, all bits including the DSP extension bits of the SR must be restored. If the repeat control must be recovered before entering the exception handling state, the RS and RE registers must be recovered to the value that existed before exception handling. In addition, if it is necessary to recover modulo control before entering the exception handling state, the MOD register must be recovered to the value that existed before exception handling.

3.2.4 DSP Registers

The DSP unit incorporates eight data registers (A0, A1, X0, X1, Y0, Y1, M0, and M1) and a status register (DSR). Figure 3.3 shows the DSP register configuration. These are 32-bit width registers with the exception of registers A0 and A1. Registers A0 and A1 include 8 guard bits (fields A0G and A1G), giving them a total width of 40 bits. The DSR register stores the DSP data operation result (zero, negative, others). The DSP register has a DC bit whose function is similar to the T bit of the CPU register. For details on DSR bits, refer to section 3.5, DSP Data Operation Instructions.

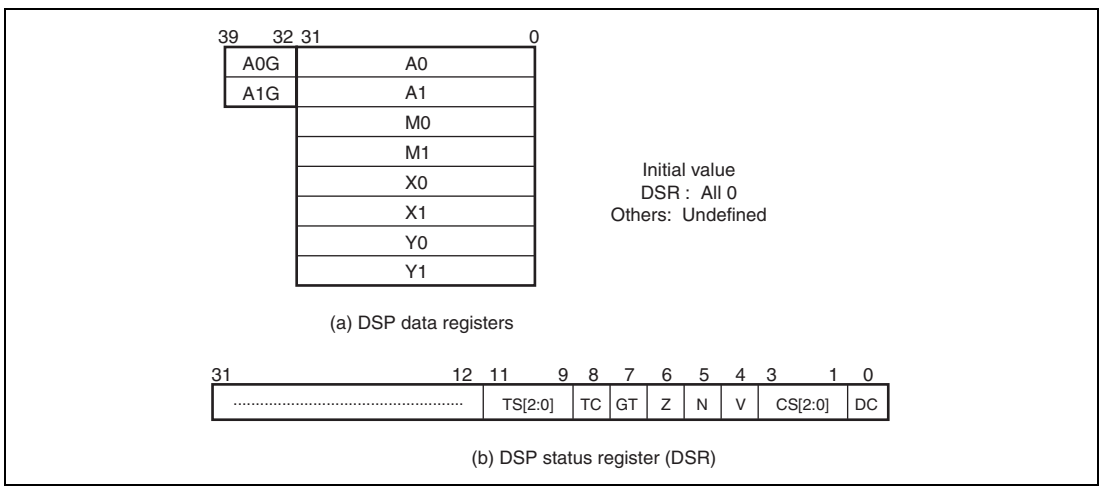


Figure 3.3 DSP Register Configuration

3.3 CPU Extended Instructions

3.3.1 Repeat Control Instructions

In DSP mode, a specific function is provided to execute repeat loops efficiently. By using this function, loop programs can be executed without overhead caused by the compare and branch instructions.

Examples of Repeat Loop Programs: Examples of repeat loop programs are shown below.

- Example 1: Repeat loop consisting of 4 or more instructions

```
            LDRS RptStart       ; Sets repeat start instruction address
                                  to the RS register
            LDRE RptStart +4    ; Sets (repeat detection instruction
                                  address + 4) to the RE register
            SETRC #4            ; Sets the number of repetitions (4) to
                                  the RC[11:0] bits of the SR register
            Instr0              ; At least one instruction is required
                                  from SETRC instruction to [Repeat start
                                  instruction]
RptStart:   instr1              ; [Repeat start instruction]
            ... ...             ;
            ... ...             ;
RptDtct:    instr(N-3)          ; Three instruction prior to the repeat
                                  end instruction is regarded as repeat
                                  detection instruction
RptEnd2:    instr(N-2)          ;
RptEnd1:    instr(N-1)          ;
RptEnd:     instrN              ; [Repeat end instruction]
```

In the above program example, instructions from the RptStart address (instr1 instruction) to the RptEnd address (instrN instruction) are repeated four times. These repeated instructions in the program are called repeat loop. The start and end instructions of the repeat loop are called the repeat start instruction and repeat end instruction, respectively. The CPU sequentially executes instructions and starts repeat loop control if the CPU detects the completion of a specific instruction. This specific instruction is called the repeat detection instruction. In a repeat loop consisting of 4 or more instructions, an instruction three instructions prior to the repeat end instruction is regarded as the repeat detection instruction. In a repeat loop consisting of 4 or more instructions, the same instruction is regarded as the RptStart instruction and RptDtct instruction.

To control the repeat loop, the DSP extended control registers, such as the RE register and RS register and the RC[11:0] and RF[1:0] bits of the SR register, are used. These registers can be specified by the LDRE, LDRS, and SETRC instructions.

- Repeat end register (RE)

  The RE register is specified by the LDRE instruction. The RE register specifies (repeat detection instruction address +4). In a repeat loop consisting of 4 or more instructions, an instruction three instructions prior to the repeat end instruction is regarded as the repeat detection instruction. A repeat loop consisting of three or less instructions is described later.
- Repeat start register (RS)

  The RS register is specified by the LDRS instruction. In a repeat loop consisting of 4 or more instructions, the RS register specifies the repeat start instruction address. In a repeat loop consisting of three or less instructions, a specific address is specified in the RS. This is described later.
- Repeat counter (RC[11:0] bits of the SR)

  The repeat counter is specifies the number of repetitions by the SETRC instruction. During repeat loop execution, the RC holds the remaining number of repetitions.
- Repeat flags (RF[1:0] bits of the SR)

  The repeat flags are automatically specified according to the RS and RE register values during SETRC instruction execution. The repeat flags store information on the number of instructions included in the repeat loop. Normally, the user cannot modify the repeat flag values.

The CPU always executes instructions by comparing the RE register to program counter values. Because the PC stores (the current instruction address +4), if the RE matches the PC during repeat instruction detection execution, a repeat detection instruction can be detected. If a repeat detection instruction is executed without branching and if RC[11:0] > 0, then repeat control is performed. If RC[11:0] ≥ 2 when the repeat end instruction is completed, the RC[11:0] is decremented by 1 and then control is passed to the address specified by the RS register.

Examples 2 to 4 show program examples of the repeat loop consisting of three instructions, two instructions, and one instruction, respectively. In these examples, an instruction immediately prior to the repeat start instruction is regarded as a repeat detection instruction. The RS register specifies the specific value that indicates the number of repeat instructions.

- Example 2: Repeat loop consisting of three instructions

```
            LDRS RptStart +4     ; Sets (repeat detection instruction
                                   address + 4) to the RS register
            LDRE RptStart +4     ; Sets (repeat detection instruction
                                   address + 4) to the RE register
            SETRC #4             ; Sets the number of repetitions (4)
                                   to the RC[11:0] bits of the SR
                                   register
                                 ; If RE-RS==0 during SETRC instruction
                                   execution, the repeat loop is
                                   regarded as three-instruction repeat.
RptDtct:    instr0               ; An instruction prior to the Repeat
                                   start instruction is regarded as a
                                   repeat detection instruction.
RptStart:   instr1               ; [Repeat start instruction]
            Instr2               ;
RptEnd:     instr3               ; [Repeat end instruction]
```

- Example 3: Repeat loop consisting of two instructions

```
            LDRS RptStart +6     ; Sets (repeat detection instruction
                                   address + 6) to the RS register
            LDRE RptStart +4     ; Sets (repeat detection instruction
                                   address + 4) to the RE register
            SETRC #4             ; Sets the number of repetitions (4)
                                   to the RC[11:0] bits of the SR
                                   register
                                 ; If RE-RS==-2 during SETRC instruction
                                   execution, the repeat loop is
                                   regarded as two-instruction repeat.
RptDtct:    instr0               ; An instruction prior to the Repeat
                                   start instruction is regarded as a
                                   repeat detection instruction.
RptStart:   instr1               ; [Repeat start instruction]
RptEnd:     instr2               ; [Repeat end instruction]
```

- Example 4: Repeat loop consisting of one instruction

```
            LDRS RptStart +8     ; Sets (repeat detection instruction
                                   address + 8) to the RS register
            LDRE RptStart +4     ; Sets (repeat detection instruction
                                   address + 4) to the RE register
            SETRC #4             ; Sets the number of repetitions (4)
                                   to the RC[11:0] bits of the SR
                                   register
                                 ; If RE-RS==-4 during SETRC
                                   instruction execution, the repeat
                                   loop is regarded as one-instruction
                                   repeat.
  RptDtct:  instr0               ; An instruction prior to the Repeat
                                   start instruction is regarded as a
                                   repeat detection instruction.
  RptStart:
  RptEnd:   instr1               ; [Repeat start instruction] ==
                                   [Repeat end instruction]
```

In repeat loops consisting of three instructions, two instructions and one instruction, specific addresses are specified in the RS register. RE – RS is calculated during SETRC instruction execution, and the number of instructions included in the repeat loop is determined according to the result. A value of 0, –2,and –4 in the result correspond to 3 instructions, two instructions, and one instruction, respectively.

If repeat instruction execution is completed without branching and if RC[11:0]>0, an instruction following the repeat detection instruction is regarded as a repeat start instruction and instruction execution is repeated for the number of times corresponding to the recognized number of instructions. If RC[11:0] ≥ 2 when the repeat end instruction is completed, the RC[11:0] is decremented by 1 and then control is passed to the address specified by the RS register. If RC[11:0] ==1(or 0) when the repeat end instruction is completed, the RC[11:0] is cleared to 0 and then the control is passed to the next instruction following the repeat end instruction.

Note: If RE – RS is a positive value, the CPU regards the repeat loop as a four-instruction repeat loop. (In a repeat loop consisting of four or more instructions, RE – RS is always a positive value. For details, refer to example 1 above.) If RE – RS is positive, or a value other than 0, –2,and –4, correct operation cannot be guaranteed.

The rule is shown in table 3.3.

Table 3.3 RS and RE Setting Rule

	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**≥ 4**
RS	RptStart0 + 8	RptStart0 + 6	RptStart0 + 4	RptStart
RE	RptStart0 + 4	RptStart0 + 4	RptStart0 + 4	RptEnd3 + 4

Note: The terms used above in table 3.3, are defined as follows.
RptStart: Address of the repeat start instruction
RptStart0: Address of the instruction one instruction prior to the repeat start instruction
RptEnd3: Address of the instruction three instructions prior to the repeat end instruction

Repeat Control Instructions and Repeat Control Macros: To describe a repeat loop, the RS and RE registers must be specified appropriately by the LDRS and LDRS instructions and then the number of repetitions must be specified by the SERTC instruction. An 8-bit immediate data or a general register can be used as an operand of the SETRC instruction. To specify the RC as a value greater than 256, use SETRC Rm type instructions.

Table 3.4 Repeat Control Instructions

Instruction	Operation	Number of Execution States
LDRS @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RS register	1
LDRE @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RE register	1
SETRC #imm	Sets 8-bit immediate data imm to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 255.	1
SETRC Rm	Sets the[11:0] bits of the Rm register to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 4095.	1

The RS and RE registers must be specified appropriately according to the rules shown in table 3.3. The SH assembler supports control macros (REPEAT) as shown in table 3.5 to solve problems.

Table 3.5 Repeat Control Macros

Instruction	Operation	Number of Execution States
REPEAT RptStart, RptEnd, #imm	Specifies RptStart as repeat start instruction, RptEnd as repeat end instruction, and 8-bit immediate data #imm as number of repetitions. This macro is extended to three instructions: LDRS, LDRE, and SETRC which are converted correctly.	3
REPEAT RptStart, RptEnd, Rm	Specifies RptStart as repeat start instruction, RptEnd as repeat end instruction, and the [11:0] bits of Rm as number of repetitions. This macro is extended to three instructions: LDRS, LDRE, and SETRC which are converted correctly.	3

Using the repeat macros shown in table 3.5, examples 1 to 4 shown above can be simplified to examples 5 to 8 as shown below.

- Example 5: Repeat loop consisting of 4 or more instructions (extended to the instruction stream shown in example 1, above)

```
          REPEAT RptStart, RptEnd, #4
          Instr0              ;
RptStart: instr1              ; [Repeat start instruction]
          ... ...             ;
          ... ...             ;
          instr(N-3)          ;
          instr(N-2)          ;
          instr(N-1)          ;
RptEnd:   instrN              ;[Repeat end instruction]
```

- Example 6: Repeat loop consisting of three instructions (extended to the instruction stream shown in example 2, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0              ;
RptStart: instr1              ; [Repeat start instruction]
          Instr2              ;
RptEnd:   instr3              ; [Repeat end instruction]
```

- Example 7: Repeat loop consisting of two instructions (extended to the instruction stream shown in example 3, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0              ;
RptStart: instr1              ; [Repeat start instruction]
RptEnd:   instr2              ; [Repeat end instruction]
```

- Example 8: Repeat loop consisting of one instruction instructions (extended to the instruction stream shown in example 4, above)

```
          REPEAT RptStart, RptEnd, #4
          instr0              ;
RptStart:
RptEnd:   instr1              ; [Repeat start instruction] ==
                                [Repeat end instruction]
```

In the DSP mode, the system control instructions (LDC and STC) that handle the RS and RE registers are extended. The RC[11:0] bits and RF[1:0] bits of the SR can be controlled by the LDC and STC instructions for the SR register. These instructions should be used if an exception is enabled during repeat loop execution. The repeat loop can be resumed correctly by storing the RS and RE register values and RC[11:0] bits and RF[1:0] bits of the SR register before exception handling and by restoring the stored values after exception handling. However, note that there are some restrictions on exception acceptance during repeat loop execution. For details refer to Restrictions on Repeat Loop Control in section 3.3.1, Repeat Control Instructions and section 4, Exceptation Handling.

Table 3.6 DSP Mode Extended System Control Instructions

Instruction	Operation	Number of Execution States
STC RS, Rn	RS→Rn	1
STC RE, Rn	RE→Rn	1
STC.L RS, @-Rn	Rn-4→Rn, RS→(Rn)	1
STC.L RE, @-Rn	Rn-4→Rn, RE→(Rn)	1
LDC.L @Rn+, RS	(Rn)→RS, Rn+4→Rn	4
LDC.L @Rn+, RE	(Rn)→RE, Rn+4→Rn	4
LDC Rn,RS	Rn →RS	4
LDC Rn, RE	Rn→RE	4

Restrictions on Repeat Loop Control

1. Repeat control instruction assignment

   The SETRC instruction must be executed after executing the LDRS and LDRE instructions. In addition, note that at least one instruction is required between the SETRC instruction and a repeat start instruction.

2. Illegal instruction one or more instructions following the repeat detection instruction

   If one of the following instructions is executed between an instruction following a repeat detection instruction to a repeat end instruction, an illegal instruction exception occurs.

   — Branch instructions

   BRA, BSR, BT, BF, BT/F, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA

   — Repeat control instructions

   SETRC, LDRS, LDRE

   — Load instructions for SR, RS, and RE registers

   LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+,RS

Note: This restriction applies to all instructions for a repeat loop consisting of one to three instructions and to three instructions including a repeat end instruction for a repeat loop consisting of four or more instructions.

3. Instructions prohibited during repeat loop (In a repeat loop consisting of four or more instructions)

   The following instructions must not be placed between the repeat start instruction and repeat detection instruction in a repeat loop consisting of four or more instructions. Otherwise, the correct operation cannot be guaranteed.

   — Repeat control instructions

   SETRC, LDRS, LDRE

   — Load instructions for SR, RS, and RE registers

   LDC Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+,RS

Note: Multiple repeat loops cannot be guaranteed. Describe the inner loop by repeat control instructions, and the external loop by other instructions such as DT or BF/S.

4. Restriction on branching to an instruction following the repeat detection instruction and an exception acceptance

Execution of a repeat detection instruction must be completed without any branch so that the CPU can recognize the repeat loop. Therefore, when the execution branches to an instruction following the repeat detection instruction, the control will not be passed to a repeat start instruction after executing a repeat end instruction because the repeat loop is not recognized by the CPU. In this case, the RC[11:0] bits of the SR register will not be changed.

— If a conditional branch instruction is used in the repeat loop, an instruction before a repeat detection instruction must be specified as a branch destination.

— If a subroutine call is used in the repeat loop, a delayed slot instruction of the subroutine call instruction must be placed before a repeat detection instruction.

Here, a branch includes a return from an exception processing routine. If an exception whose return address is placed in an instruction following the repeat detection instruction occurs, the repeat control cannot be returned correctly. Accordingly, an exception acceptance is restricted from the repeat detection instruction to the repeat end instruction. Exceptions such as interrupts that can be retained by the CPU are retained. For exceptions that cannot be retained by the CPU, a transition to an exception occurs but a program cannot be returned to the previous execution state correctly. For details, refer to section 4, Exception Handling.

Notes: 1. If a TRAPA instruction is used as a repeat detection instruction, an instruction following the repeat detection instruction is regarded as a return address. In this case, a control cannot be returned to the repeat control correctly. In a TRAPA instruction, an address of an instruction following the repeat detection address is regarded as return address. Accordingly, to return to the repeat control correctly, place a return address prior to the repeat detection instruction.

2. If a SLEEP instruction is placed following a repeat detection instruction, a transition to the low-power consumption state or an exception acceptance such as interrupts can be performed correctly. In this case, however, the repeat control cannot be returned correctly. To return to the repeat control correctly, the SLEEP instruction must be placed prior to the repeat detection instruction.

5. Branch from a repeat detection instruction

   If a repeat detection instruction is a delayed slot instruction of a delayed branch instruction or a branch instruction, a repeat loop can be acknowledged when a branch does not occur in a branch instruction. If a branch occurs in a branch instruction, a repeat control is not performed and a branch destination instruction is executed.

6. Program counter during repeat control

If RC[11:0] ≥ 2, the program counter (PC) value is not correct for instructions two instructions following a repeat detection instruction. In a repeat loop consisting of one to three instructions, the PC indicates the correct value (instruction address + 4) for an instruction (repeat start instruction) following a repeat detection instruction but the PC continues to indicate the same address (repeat start instruction address) from the subsequent instruction to a repeat end instruction. In a repeat loop consisting of four or more instructions, the PC indicates the correct value (instruction address + 4) for an instruction following a repeat detec-tion instruction, but PC indicates the RS and (RS +2) for instructions two and three instructions following the repeat detection instruction. Here, RS indicates the value stored in the repeat start register (RS). The correct operation cannot be guaranteed for the incorrect PC values.

Accordingly, PC relative addressing instructions placed two or more instructions following the repeat detection instruction cannot be executed correctly and the correct results cannot be obtained.

— PC relative addressing instructions

MOVA @(disp, PC), Rn

MOV.W @(disp, PC), Rn

MOV.L @(disp, PC), Rn

(Including the case when the MOV #imm,Rn is extended to MOV.W @(disp, PC), Rn or MOV.L @(disp, PC), Rn)

Table 3.7 PC Value during Repeat Control (When RC[11:0] ≥ 2)

	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**≥ 4**
RptDtct	RptDtct + 4	RptDtct + 4	RptDtct + 4	RptDtct +4
RptDtct1	RptDtct1 + 4	RptDtct1 + 4	RptDtct1 + 4	RptDtct1 + 4
RptDtct2	—	RptDtct1 + 4	RptDtct1 + 4	RS
RptDtct3	—	—	RptDtct1 + 4	RS + 2

Note: In table 3.7, the following labels are used.

RptDtct: An address of the repeat detection instruction

RptDtct1: An address of the instruction one instruction following the repeat start instruction (In a repeat loop consisting of one to three instructions, RptStart is a repeat start instruction)

RptDtct2: An address of the instruction two instruction following the repeat start instruction

RptDtct3: An address of the instruction three instruction following the repeat start instruction

7. Repeat counter and repeat control

The CPU always executes a program with comparing the repeat end register (RE) and the program counter (PC). If the PC matches the RE while the RC[11:0] bits of the SR register are other than 0, the repeat control function is initiated.

— If RC ≥ 2, a control is passed to a repeat start instruction after a repeat end instruction has been executed. The RC is decremented by 1 at the completion of the repeat end instruction. In this case, restrictions 1 to 6 are also applied.

— If RC == 1, the RC is decremented to 0 at the completion of the repeat end instruction and a control is passed to the subsequent instruction. In this case, restrictions 1 to 6 are also applied.

— If RC == 0, the repeat control function is not initiated even if a repeat detection instruction is executed. The repeat loop is executed once as normal instructions and a control is not be passed to a repeat start instruction even if a repeat end instruction is executed.

3.3.2 Extended Repeat Control Instructions

In the repeat control function described in section 3.3.1, Repeat Control Instructions, there are some restrictions. To reduce these restrictions, this LSI supports the extended repeat instructions to extend the repeat control function. These extended repeat control instructions were not supported in the conventional SH-DSP. To keep compatibility with the conventional SH-DSP, use the conventional repeat control instructions called compatible repeat control instructions.

Program Examples Using the Extended Repeat Control Instructions: Examples of repeat loop programs using the extended repeat control instructions are shown below.

- Example 1: Repeat loop consisting of 4 or more instructions

```
            LDRS RptStart       ; Sets repeat start instruction address
                                  to the RS register
            LDRE RptEnd         ; Sets repeat end instruction address
                                  to the RE register
            LDRC #4             ; Sets the number of repetitions (4) to
                                  the RC[11:0] bits of the SR register
            instr0              ; At least one instruction is required
                                  from LDRC instruction to [Repeat start
                                  instruction]
RptStart:   instr1              ; [Repeat start instruction]
            ... ...             ;
            ... ...             ;
            instr(N-3)          ;
            instr(N-2)          ;
            instr(N-1)          ;
RptEnd:     instrN              ; [Repeat end instruction]
```

- Example 2: Repeat loop consisting of three instructions

```
            LDRS RptStart       ; Sets repeat start instruction address
                                  to the RS register
            LDRE RptEnd         ; Sets repeat end instruction address
                                  to the RE register
            LDRC #4             ; Sets the number of repetitions (4) to
                                  the RC[11:0] bits of the SR register
            instr0              ; At least one instruction is required
                                  from LDRC instruction to [Repeat start
                                  instruction]
RptStart:   instr1              ; [Repeat start instruction]
            instr2              ;
RptEnd:     instr3              ; [Repeat end instruction]
```

- Example 3: Repeat loop consisting of two instructions

```
            LDRS RptStart       ; Sets repeat start instruction address
                                  to the RS register
            LDRE RptEnd         ; Sets repeat end instruction address
                                  to the RE register
            LDRC #4             ; Sets the number of repetitions (4) to
                                  the RC[11:0] bits of the SR register
            instr0              ; At least one instruction is required
                                  from LDRC instruction to [Repeat start
                                  instruction]
RptStart:   instr1              ; [Repeat start instruction]
RptEnd:     instr2              ; [Repeat end instruction]
```

- Example 4: Repeat loop consisting of one instructions

```
            LDRS RptStart       ; Sets repeat start instruction address
                                  to the RS register
            LDRE RptEnd         ; Sets repeat end instruction address
                                  to the RE register
            LDRC #4             ; Sets the number of repetitions (4) to
                                  the RC[11:0] bits of the SR register
            instr0              ; At least one instruction is required
                                  from LDRC instruction to [Repeat start
Rptstart:                         instruction] RptStart:
RptEnd:     instr1              ; [Repeat start instruction]=
                                  [Repeat end instruction]
```

In extended repeat control instructions, a repeat start instruction address and a repeat end instruction address are stored in the RS register and RE register, respectively, regardless of the number of repeat instructions. In addition, the extended repeat control can be performed by using the LDRC instruction instead of the SETRC instruction. During the extended repeat control, a repeat loop can be recognized by executing a repeat end instruction. Therefore, there is no restriction on branches or exceptions.

Extended Repeat Control Instructions: To describe the extended repeat loop, the repeat start and end addresses must be specified to the RS and RE registers by the LDRS and LDRE instructions, respectively. For the LDRS and LDRE instructions of the extended repeat control instructions, the LDRS and LDRE instructions of the compatible repeat control instructions are used. The number of repetitions are specified by the LDRC instruction. An 8-bit immediate data or the general register values can be used as an operand of the LDRC instruction. If 256 or greater value is specified to the RC, use the LDRC Rm type instructions.

Table 3.8 Extended Repeat Control Instructions

Instruction	Operation	Number of Execution States
LDRS @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RS register	1
LDRE @(disp,PC)	Calculates (disp x 2 + PC) and stores the result to the RE register	1
LDRC #imm	Sets 8-bit immediate data imm to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 255. During extended repeat control, bit 0 of the RE register is set to 1.	1
LDRC Rm	Sets the[11:0] bits of the Rm register to the RC[11:0] bits of the SR register and sets the information related to the number of repetitions to the RF[1:0] bits of the SR. RC[11:0] can be specified as 0 to 4095. During extended repeat control, bit 0 of the RE register is set to 1.	1

By executing the LDRC instruction, the CPU performs the extended repeat control function. To indicate that the CPU is being in extended repeat control, bit 0 of the RE register is set to 1 by executing the LDRC instruction. To change the RE register value by a process such as an exception handling, bit 0 of the RE register must be saved and restored correctly. By saving and restoring the RC[11:0] bits, DSP bit, and RF[1:0] bits of the SR register, RE register, and RS register correctly, a control is returned to the extended repeat function correctly after processing such as exception handling.

Restrictions on Extended Repeat Loop Control

1. Extended repeat control instruction assignment

   The LDRC instruction must be executed after executing the LDRS and LDRE instructions. In addition, note that at least one instruction is required between the LDRC instruction and a repeat start instruction.
2. Illegal instruction one or more instructions following the repeat detection instruction

   If one of the following instructions is executed as a repeat end instruction, an illegal instruction exception occurs.

   — Branch instructions

BRA, BSR, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP

— Repeat control instructions
SETRC, LDRS, LDRE, LDRC

— Load instructions for SR, RS, and RE registers
LCD Rn,SR, LDC @Rn+,SR, LDC Rn,RE, LDC @Rn+,RE, LDC Rn,RS, LDC @Rn+,RS

Note: A branch instruction without delay (BT, BF, TRAPA) can be placed as a repeat end instruction. A delay stop of a delayed branch instruction can also be placed as a repeat end instruction. In this case, the RC[11:0] value is decremented by 1 regardless of branch occurrence. If no branch occurs, a control returns to a repeat start instruction. If a branch occurs, a control is passed to a branch destination.

3. Repeat counter and repeat control

The CPU always execute a program with comparing the repeat end register (RE) and the (PC – 4) (current instruction address). If the (PC – 4) [31:1] matches the RE [31:1] while bit 0 of RE register is set to 1 and RC [11:0] of SR register is not 0, the extended repeat control function is initiated.

— If RC ≥ 2, a control is passed to a repeat start instruction after a repeat end instruction has been executed. The RC is decremented by 1 at the completion of the repeat end instruction.

— If RC == 1, the RC is decremented to 0 at the completion of the repeat end instruction and a control is passed to the subsequent instruction.

— If RC == 0, the repeat control function is not initiated even if a repeat detection instruction is executed. The repeat loop is executed once as normal instructions and a control is not be passed to a repeat start instruction even if a repeat end instruction is executed.

3.4 DSP Data Transfer Instructions

In DSP mode, data transfer instructions are added for the DSP unit registers. The newly added instructions are classified into the following three groups.

1. Double data transfer instructions

   The DSP unit is connected to the X memory and Y memory via the specific buses called X bus and Y bus. By using the data transfer instructions using the X and Y buses, two data items can be transferred between the DSP unit and X/Y memories simultaneously. These instructions are called double data transfer instructions. These double data transfer instructions can be described in combination with the DSP operation instructions to execute data transfer and data operation in parallel,
2. Single data transfer instructions

   The DSP unit is also connected to the L bus that is used by the CPU. The DSP registers other than the DSR can access any logical addresses generated by the CPU. In this case, the single data transfer instructions are used. The single data transfer instructions cannot be used in combination with the DSP operation instructions and can access only one data item at a time.
3. System control instructions

   Some of the DSP unit registers are handled as the CPU system registers. To control these system registers, the system control registers are supported. The DSP registers are connected to the CPU general registers via the data transfer bus (C bus).

In any DSP data transfer instructions, an address to be accessed is generated and output by the CPU. For DSP data transfer instructions, some of the CPU general registers are used for address generation and specific addressing modes are used.

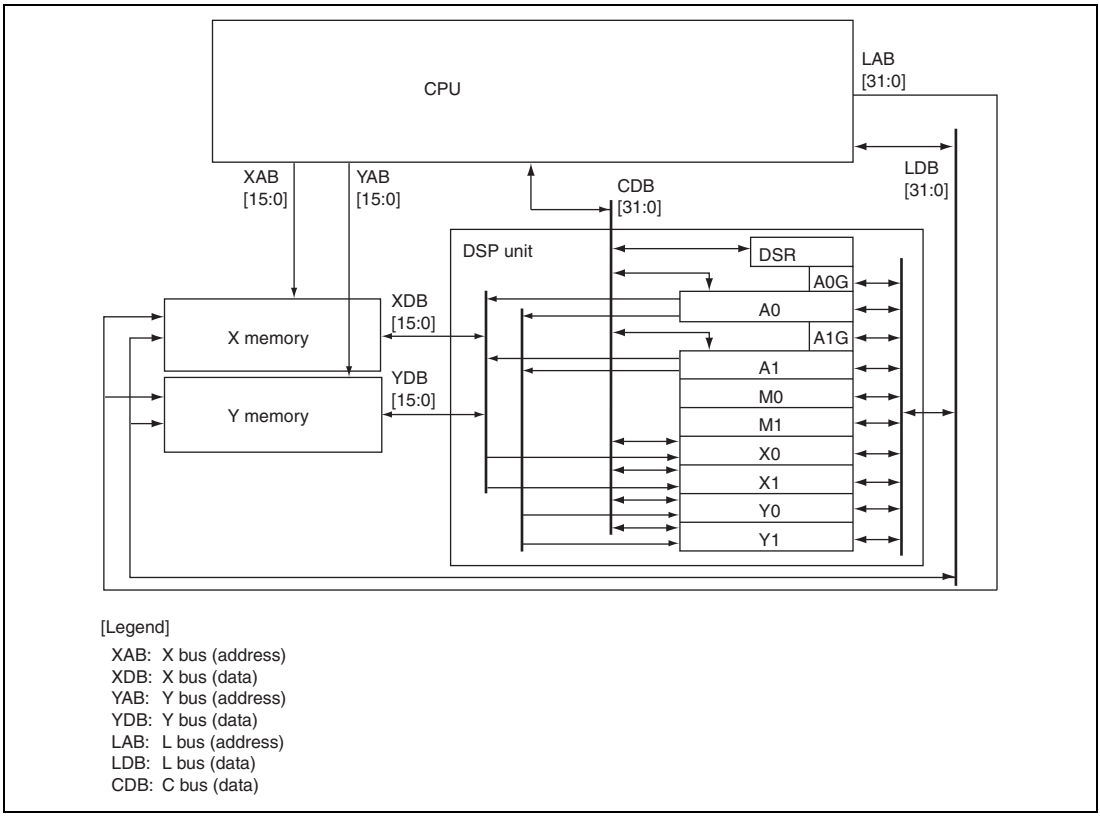


Figure 3.4 DSP Registers and Bus Connections

Double Data Transfer Instructions (MOVX.W, MOVY.W, MOVX.L, MOVY.L): With double data transfer group instructions, X memory and Y memory can be accessed in parallel.

In this case, the specific buses called X bus and Y bus are used to access X memory and Y memory, respectively. To fetch the CPU instructions, the L bus is used. Accordingly, no conflict occurs among X, Y, and L buses.

Load instructions for X memory specify the X0 or X1 register as the destination operand. Load instructions for Y memory specify the Y0 or Y1 register as the destination operand. Store registers for X or Y memory specify the A0 or A1 register as the source operand. These instructions use only word data (16 bits). When a word data transfer instruction is executed, the upper word of register operand is used. To load word data, data is loaded to the upper word of the destination register and the lower word of the destination register is automatically cleared to 0.

Double data transfer instructions can be described in parallel to the DSP operation instructions. Even if a conditional operation instruction is specified in parallel to a double data transfer instruction, the specified condition does not affect the data transfer operations. For details, refer to section 3.5, DSP Data Operation Instructions.

Double data transfer instructions can access only the X memory or Y memory and cannot access other memory space. The X bus and Y bus are 16 bits and support 64-byte address spaces corresponding to address areas H'A5000000 to H'A500FFFF and H'A5010000 to H'A501FFFF, respectively. Because these areas are included in the P2/Uxy area, they are not affected by the cache and address translation unit.

Single Data Transfer Instructions: The single data transfer instructions access any memory location. All DSP registers other than the DSR* can be specified as source and destination operands. Guard bit registers A0G and A1G can also be specified as two independent registers. Because these instructions use the L bus (LAB and LDB), these instructions can access any logical space handled by the CPU. If these instructions access the cacheable area while the cache is enabled, the area accessed by these instructions are cached. The X memory and Y memory are mapped to the logical address space and can also be accessed by the single data transfer instructions. In this case, bus conflict may occur between data transfer and instruction fetch because the CPU also uses the L bus for instruction fetches.

The single data transfer instructions can handle both word and longword data. In word data transfer, only the upper word of the operand register is valid. In word data load, word data is loaded into the upper word of the destination registers and the lower word of the destination is automatically cleared to 0. If the guard bits are supported, the sign bit is extended before storage. In longword data load, longword data is loaded into the upper and lower word of the destination register. If the guard bits are supported, the sign bit is extended before storage. When the guard register is stored, the sign bit is extended to the upper 24 bits of the LDB and are loaded onto the LDB bus.

Notes: * Because the DSR register is defined as the system register, it can be accessed by the LDS or STS instruction.

1. Any data transfer instruction is executed at the MA stage of the pipeline.
2. Any data transfer instruction does not modify the condition code bits of the DSR register.

System Control Instructions: The DSR, A0, X0, X1, Y0, and Y1 registers in the DSP unit can also be used as the CPU system registers. Accordingly, data transfer operations between these DSP system registers and general registers or memory can be executed by the STS and LDS instructions. These DSP system registers can be treated as the CPU system register such as PR, MACL and MACH and can use the same addressing modes.

Table 3.9 Extended System Control Instructions in DSP Mode

Instruction		Operation	Execution States
STS	DSR,Rn	DSR → Rn	1
STS	A0,Rn	A0 → Rn	1
STS	X0,Rn	X0 → Rn	1
STS	X1,Rn	X1 → Rn	1
STS	Y0,Rn	Y0 → Rn	1
STS	Y1,Rn	Y1 → Rn	1
STS.L	DSR,@-Rn	Rn – 4 → Rn, DSR → (Rn)	1
STS.L	A0,@-Rn	Rn – 4 → Rn, A0 → (Rn)	1
STS.L	X0,@-Rn	Rn – 4 → Rn, X0 → (Rn)	1
STS.L	X1,@-Rn	Rn – 4 → Rn, X1 → (Rn)	1
STS.L	Y0,@-Rn	Rn – 4 → Rn, Y0 → (Rn)	1
STS.L	Y1,@-Rn	Rn – 4 → Rn, Y1 → (Rn)	1
LDS.L	@Rn+,DSR	(Rn) → DSR, Rn + 4 → Rn	1
LDS.L	@Rn+,A0	(Rn) → A0, Rn + 4 → Rn	1
LDS.L	@Rn+,X0	(Rn) → X0, Rn + 4 → Rn	1
LDS.L	@Rn+,X1	(Rn) → X1, Rn + 4 → Rn	1
LDS.L	@Rn+,Y0	(Rn) → Y0, Rn + 4 → Rn	1
LDS.L	@Rn+,Y1	(Rn) → Y1, Rn + 4 → Rn	1
LDS	Rn,DSR	Rn → DSR	1
LDS	Rn,A0	Rn → A0	1
LDS	Rn,X0	Rn → X0	1
LDS	Rn,X1	Rn → X1	1
LDS	Rn,Y0	Rn → Y0	1
LDS	Rn,Y1	Rn → Y1	1

3.4.1 General Registers

The DSP instructions 10 general registers in the 16 general registers as address pointers or index registers for double data transfers and single data transfers. In the following descriptions, another register function in the DSP instructions is also indicated within parentheses [ ].

- Double data transfer instructions (X memory and Y memory are accessed simultaneously)
  In double data transfers, X memory Y memory can be accessed simultaneously. To specify X and Y memory addresses, two address pointers are supported.

	Address Pointer	Index Register
X memory (MOVX.W)	R4, R5[Ax]	R8 [Ix]
Y memory (MOVY.W)	R6, R7[Ay]	R9 [Iy]

- Single data transfer instructions
  In single data transfer, any logical address space can be accessed via the L bus. The following address pointers and index registers are used.

	Address Pointer	Index Register
Any logical space (MOVS.W/L)	R4, R5, R2, R3[As]	R8 [Is]

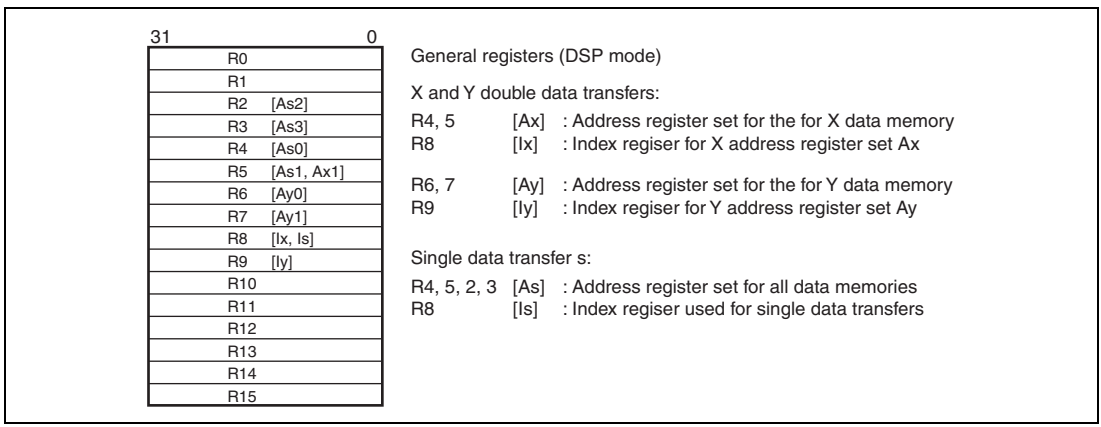


Figure 3.5 General Registers (DSP Mode)

In assembler, R0 to R9 are used as symbols. In the DSP data transfer instructions, the following register names (alias) can also be used. In assembler, described as shown below.

Ix: .REG (R8)

Ix indicates the alias of register 8. Other aliases are shown below.

Ax0: .REG (R4)

Ax1: .REG (R5)

Ix: .REG (R8)

Ay0: .REG (R6)

Ay1: .REG (R7)

Iy: .REG (R9)

As0: .REG (R4); This definition is used for if the alias is required in the single data transfer

As1: .REG (R5); This definition is used for if the alias is required in the single data transfer

As2: .REG (R2)

As3: .REG (R3)

Is: .REG (R8); This definition is used for if the alias is required in the single data transfer

3.4.2 DSP Data Addressing

Table 3.10 shows the relationship between the double data transfer instructions and single data transfer instructions.

Table 3.10 Overview of Data Transfer Instructions

	Double Data Transfer Instructions	Single Data Transfer Instructions
	MOVX.W **MOVY.W**	**MOVS.W** **MOVS.L**
Address register	Ax: R4, R5, Ay: R6, R7	As: R2, R3, R4, R5
Index register	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc (+2)/index addition: post-increment	Nop/Inc (+2, +4)/index addition: post-increment
Addressing	—	Dec (–2, –4): pre-decrement
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	LDB
Data length	16 bits (word)	16/32 bits (word/longword)
Bus conflict	No	Yes
Memory	X/Y data memory	Entire memory space
Source register	Dx, Dy: A0, A1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G
Destination register	Dx: X0/X1 Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G

Addressing Mode for Double Data Transfer Instructions: The double data transfer instructions supports the following three addressing modes.

- Non-update address register addressing

  The Ax and Ay registers are address pointers. They are not updated.
- Increment address register addressing

  The Ax and Ay registers are address pointers. After a data transfer, they are each incremented by 2 (post- increment).
- Addition index register addressing

  The Ax and Ay registers are address pointers. After a data transfer, the value of the Ix or Iy register is added to each (post-increment). The double data transfer instructions do not supports

decrement addressing mode. To perform decrementing, –2 is set in the index register and addition index register addressing is specified.

When using X/Y data addressing, bit 0 of the address pointer is invalid. Accordingly, bit 0 of the address pointer and index register must be cleared to 0 in X/Y data addressing.

When accessing X and Y memory using the X and Y buses, the upper word of Ax and Ay is ignored. The result of Ay+ or Ay+Iy is stored in the lower word of Ay, while the upper word retains its original value. The Ax and Ax +Ix operations are executed in longword (32 bits) and the upper word may be changed according to the result.

Single Data Addressing: The following four kinds of addressing can be used with single data transfer instructions.

- Non-update address register addressing
  The As register is an address pointer. An access to @As is performed but As is not updated.
- Increment address register addressing:
  The As register is an address pointer. After an access to @As, the As register is incremented by 2 or 4 (post-increment).
- Addition index register addressing:
  The As register is an address pointer. After an access to @As, the value of the Is register is added to the As register (post-increment).
- Decrement address register addressing:
  The As register is an address pointer. Before a data transfer, –2 or –4 is added to the As register (i.e. 2 or 4 is subtracted) (pre-decrement).

In single data transfer instructions, all bits in 32-bit address are valid.

3.4.3 Modulo Addressing

In double data transfer instructions, a modulo addressing can be used. If the address pointer value reaches the preset modulo end address while a modulo addressing mode is specified, the address pointer value becomes the modulo start address.

To control modulo addressing, the modulo register (MOD) extended in the DSP mode and the DMX and DMY bits of the SR register are used.

The MOD register is provided to set the start and end addresses of the modulo address area. The upper and lower words of the MOD register store modulo start address (MS) and modulo end

address (ME), respectively. The LDC and STC instructions are extended for MOD register handling.

If the DMX bit of the SR register is set, the modulo addressing is specified for the X address register. If the DMY bit of the SR register is set, the modulo addressing is specified for the Y address register. Modulo addressing is valid for either the X or the Y address register, only; it cannot be set for both at the same time. Therefore, DMX and DMY cannot both be set simultaneously (if they are, the DMY setting will be valid). (In the future, this specification may be changed.) The DMX and DMY bits of the SR can be specified by the STC or LDC instruction for the SR register.

If an exception is accepted during modulo addressing, the DMX and DMY bits of the SR and MOD register must be saved. By restoring these register values, a control is returned to the modulo addressing after an exception handling.

Table 3.11 Modulo Addressing Control Instructions

Instruction	Operation	Execution States
STC MOD, Rn	MOD → Rn	1
STC.L MOD, @–Rn	Rn – 4 → Rn, MOD → (Rn)	1
LDC @Rn+, MOD	(Rn) → MOD, Rn + 4 → Rn	4
LDC Rn, MOD	Rn → MOD	4

An example of the use of modulo addressing is shown below.

```
MOV.L #H'70047000,R10
                    ;Specify MS=H'7000 ME = H'7004
LDC R10,MOD         ;Specify ME:MS to MOD register
STC SR,R10          ;
MOV.L #H'FFFFF3FF,R11    ;
MOV.L #H'00000400,R12    ;
AND R11,R10         ;
OR R12,R10          ;
LDC R10,SR          ; Specify SR.DMX=1,
                      SR.DMY=0, and X modulo addressing mode
MOV.L #H'A5007000,R14
MOVX.W @R4+,X0      ; R4: H'A5007000→ H'A5007002
MOVX.W @R4+,X0      ; R4: H'A5007002→ H'A5007004
MOVX.W @R4+,X0      ; R4: H'A5007004→ H'A5007000
                      (Matches to ME and MS is set)
MOVX.W @R4+,X0      ; R4: H'A5007000→ H'A5007002
MOVX.W @R4+,X0      ; R4: H'A5007000→ H'A5007002
```

The start and end addresses are specified in MS and ME, then the DMX or DMY bit is set to 1. When the X or Y data transfer instruction specified by the DMX or DMY is executed, the address register contents before updating are compared with ME*, and if they match, start address MS is stored in the address register as the value after updating.

When the addressing type of the X/Y data transfer instruction is no-update, the X/Y data transfer instruction is not returned to MS even if they match ME.

When the addressing type of the X/Y data transfer instruction is addition index register addressing, the address pointed way not match the address pointer ME and exceed it. In this case, the address pointer value does not become the modulo start address.

The maximum modulo size is 64 kbytes. This is sufficient to access the X and Y data memory.

Note: * Not only with modulo addressing, but when X and Y data addressing is used, bit 0 is ignored. 0 must always be written to bit 0 of the address pointer, index register, MS, and ME.

3.4.4 Memory Data Formats

Memory data formats that can be used in the DSP instructions are classified into word, and longword. An address error will occur if word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed by MOVS.L, LDS.L, or STS.L instruction. In such cases, the data accessed cannot be guaranteed

An address error will not occur if word data starting from an address other than 2n is accessed by the MOVX.W or MOVY.W instruction. When using the MOVX.W or MOVY.W instruction, an address must be specified on the boundary 2n. If an address is specified other than 2n, the data accessed cannot be guaranteed.

3.4.5 Instruction Formats of Double and Single Transfer Instructions

The format of double data transfer instructions is shown in tables 3.12, and that of single data transfer instructions in table 3.13.

Table 3.12 Double Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory data transfer	NOPX	1	1	1	1	0	0	0		0		0		0	0		
	MOVX.W @Ax,Dx							Ax		Dx		0		0	1		
	MOVX.W @Ax+,Dx													1	0		
	MOVX.W @Ax+Ix,Dx													1	1		
	MOVX.W Da,@Ax									Da		1		0	1		
	MOVX.W Da,@Ax+													1	0		
	MOVX.W Da,@Ax+Ix													1	1		
Y memory data transfer	NOPY	1	1	1	1	0	0		0		0		0			0	0
	MOVY.W @Ay,Dy								Ay		Dy		0			0	1
	MOVY.W @Ay+,Dy															1	0
	MOVY.W @Ay+Iy,Dy															1	1
	MOVY.W Da,@Ay										Da		1			0	1
	MOVY.W Da,@Ay+															1	0
	MOVY.W Da,@Ay+Iy															1	1

Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 3.13 Single Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Single data transfer	MOVS.W @-As,Ds	1	1	1	1	0	1	As			Ds	0:(*)		0	0	0	0
	MOVS.W @As,Ds							0:R4				1:(*)		0	1		
	MOVS.W @As+,Ds							1:R5				2:(*)		1	0		
	MOVS.W @As+Is,Ds							2:R2				3:(*)		1	1		
	MOVS.W Ds,@-As							3:R3				4:(*)		0	0	0	1
	MOVS.W Ds,@As											5:A1		0	1		
	MOVS.W Ds,@As+											6:(*)		1	0		
	MOVS.W Ds,@As+Is											7:A0		1	1		
	MOVS.L @-As,Ds											8:X0		0	0	1	0
	MOVS.L @As,Ds											9:X1		0	1		
	MOVS.L @As+,Ds											A:Y0		1	0		
	MOVS.L @As+Is,Ds											B:Y1		1	1		
	MOVS.L Ds,@-As											C:M0		0	0	1	1
	MOVS.L Ds,@As											D:A1G		0	1		
	MOVS.L Ds,@As+											E:M1		1	0		
	MOVS.L Ds,@As+Is											F:A0G		1	1		

Note: * Codes reserved for system use.

3.5 DSP Data Operation Instructions

3.5.1 DSP Registers

This LSI has eight data registers (A0, A1, X0, X1, Y0, Y1, M0 and M1) and one control register (DSR) as DSP registers (figure 3.3).

Four kinds of operation access the DSP data registers. The first is DSP data processing. When a DSP fixed-point data operation uses A0 or A1 as the source register, it uses the guard bits (bits 39 to 32). When it uses A0 or A1 as the destination register, guard bits 39 to 32 are valid. When a DSP fixed-point data operation uses a DSP register other than A0 or A1 as the source register, it sign-extends the source value to bits 39 to 32. When it uses one of these registers as the destination register, bits 39 to 32 of the result are discarded.

The second kind of operation is an X or Y data transfer operation, MOVX.W, MOVY.W. This operation accesses the X and Y memories through the 16-bit X and Y data buses (figure 3.4). The register to be loaded or stored by this operation always comprises the upper 16 bits (bits 31 to 16). X0 or X1 can be the destination of an X memory load and Y0 or Y1 can be the destination of a Y memory load, but no other register can be the destination register in this operation. When data is read into the upper 16 bits of a register (bits 31 to 16), the lower 16 bits of the register (bits 15 to 0) are automatically cleared. A0 and A1 can be stored in the X or Y memory by this operation, but no other registers can be stored.

The third kind of operation is a single-data transfer instruction, MOVS.W or MOVS.L. These instructions access any memory location through the LDB (figure 3.4). All DSP registers connect to the LDB and can be the source or destination register of the data transfer. These instructions have word and longword access modes. In word mode, registers to be loaded or stored by this instruction comprise the upper 16 bits (bits 31 to 16) for DSP registers except A0G and A1G. When data is loaded into a register other than A0G and A1G in word mode, the lower half of the register is cleared. When A0 or A1 is used, the data is sign-extended to bits 39 to 32 and the lower half is cleared. When A0G or A1G is the destination register in word mode, data is loaded into an 8-bit register, but A0 or A1 is not cleared. In longword mode, when the destination register is A0 or A1, it is sign-extended to bits 39 to 32.

The fourth kind of operation is system control instructions such as LDS, STS, LDS.L, or STS.L. The DSR, A0, X0, X1, Y0, and Y1 registers of the DSP register can be treated as system registers. For these registers, data transfer instructions between the CPU general registers and system registers or memory access instructions are supported.

Tables 3.14 and 3.15 show the data type of registers used in DSP instructions. Some instructions cannot use some registers shown in the tables because of instruction code limitations. For example, PMULS can use A1 as the source register, but cannot use A0. These tables ignore details of register selectability.

Table 3.14 Destination Register in DSP Instructions

			Guard Bits	Register Bits	
Registers	**Instructions**		**39 32**	**31 16**	**15 0**
A0, A1	DSP operation	Fixed-point, PSHA, PMULS	Sign-extended	40-bit result	
		Integer, PDMSB	Sign-extended	24-bit result	Cleared
		Logical, PSHL	Cleared	16-bit result	Cleared
	Data transfer	MOVS.W	Sign-extended	16-bit data	Cleared
		MOVS.L	Sign-extended	32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	No update	
		MOVS.L	Data	No update	
X0, X1 Y0, Y1 M0, M1	DSP operation	Fixed-point, PSHA, PMULS		32-bit result	
		Integer, logical, PDMSB, PSHL		16-bit result	Cleared
	Data transfer	MOVX/Y.W, MOVS.W		16-bit result	Cleared
		MOVS.L		32-bit data	

Table 3.15 Source Register in DSP Operations

			Guard Bits		Register Bits			
Registers	**Instructions**		**39**	**32**	**31**	**16**	**15**	**0**
A0, A1	DSP operation	Fixed-point, PDMSB, PSHA			40-bit data			
		Integer			24-bit data			
		Logical, PSHL, PMULS			16-bit data			
	Data transfer	MOVX/Y.W, MOVS.W			16-bit data			
		MOVS.L			32-bit data			
A0G, A1G	Data transfer	MOVS.W	Data					
		MOVS.L	Data					
X0, X1 Y0, Y1 M0, M1	DSP operation	Fixed-point, PDMSB, PSHA	Sign*		32-bit data			
		Integer	Sign*		16-bit data			
		Logical, PSHL, PMULS			16-bit data			
	Data transfer	MOVS.W			16-bit data			
		MOVS.L			32-bit data			

Note: * The data is sign-extended and input to the ALU.

The DSP unit incorporates one control register and DSP status register (DSR). The DSR register stores the DSP data operation result (zero, negative, others). The DSP register also has the DC bit whose function is similar to the T bit of the CPU register. The DC bit functions as status flag. Conditional DSP data operations are controlled based on the DC bit. These operation control affects only the DSP unit instructions. In other words, these operations control affects only the DSP registers and does not affect address register update and CPU instructions such as load and store instructions. A condition to be reflected on the DC bit should be specified to the DC status selection bits (CS[2:0]).

The unconditional DSP type data instructions other than PMULS, MOVX, MOVY, and MOVS change the condition flag and DC bit. However, the CPU instructions including the MAC instruction do not modify the DC bit. In addition, conditional DSP instructions do not modify the DSR.

Table 3.16 DSR Register Bits

Bits	Bit Name	Initial Value	R/W	Function
31 to 12	—	All 0	R	Reserved Bits These bits are always read as 0. The write value should always be 0.
11 to 9	TS2 to TS0	All 0	R/W	T Bit Status Selection Specifies the operation result status to be set in the T bit in the SR register if the TC bit is 1. If the S bit of the SR register is set to 1, an overflow is detected. 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)
8	TC	0	R/W	TC Bit 0: The T bit of the SR register is not affected by the DSP instruction. 1: The T bit of the SR register changes according to the TS bit of the DSR register while the DSP instruction is executed. Note, however, the T bit does not change during conditional DSP instruction execution.
7	GT	0	R/W	Signed Greater Bit Indicates that the operation result is positive (except 0), or that operand 1 is greater than operand 2 1: Operation result is positive, or operand 1 is greater than operand 2
6	Z	0	R/W	Zero Bit Indicates that the operation result is zero (0), or that operand 1 is equal to operand 2 1: Operation result is zero (0), or operands are equal

Bits	Bit Name	Initial Value	R/W	Function
5	N	0	R/W	Negative Bit Indicates that the operation result is negative, or that operand 1 is smaller than operand 2 1: Operation result is negative, or operand 1 is smaller than operand 2
4	V	0	R/W	Overflow Bit Indicates that the operation result has overflowed 1: Operation result has overflowed
3 to 1	CS2 to CS0	All 0	R/W	DC Bit Status Selection Designate the mode for selecting the operation result status to be set in the DC bit 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed greater than or equal to mode 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)
0	DC	0	R/W	DSP Status Bit Sets the status of the operation result in the mode designated by the CS bits 0: Designated mode status has not occurred (false) 1: Designated mode status has occurred Indicates the operation result by carry or borrow regardless of the CS bit status after the PADDC or PSUBC instruction has been executed.

The DSR is assigned to the system registers. For the DSR, the following load and store instructions are supported.

```
STS DSR,Rn;
STS.L DSR,@-Rn;
LDS Rn,DSR;
LDS.L @Rn+,DSR;
```

If the DSR is read by the STS instruction, upper bits (bits 31 to 16) are all 0

3.5.2 DSP Operation Instruction Set

DSP operation instructions are instructions for digital signal processing performed by the DSP unit. These instructions have a 32-bit instruction code, and multiple instructions can be executed in parallel. The instruction code is divided into an A field and B field; a parallel data transfer instruction is specified in the A field, and a single or double data operation instruction in the B field. Instructions can be specified independently, and are also executed independently.

B-field data operation instructions are of three kinds: double data operation instructions, conditional single data operation instructions, and unconditional single data operation instructions. The formats of the DSP operation instructions are shown in table 3.17. The respective operands are selected independently from the DSP registers. The correspondence between DSP operation instruction operands and registers is shown in table 3.18.

Table 3.17 DSP Operation Instruction Formats

Type	Instruction Formats
Double data operation instructions	`ALUop. Sx, Sy, Du`
	`MLTop. Se, Df, Dg`
Conditional single data operation instructions	`DCT ALUop. Sx, Sy, Dz`
	`DCF ALUop. Sx, Sy, Dz`
	`DCT ALUop. Sx, Dz`
	`DCF ALUop. Sx, Dz`
	`DCT ALUop. Sy, Dz`
	`DCF ALUop. Sy, Dz`
Unconditional single data operation instructions	`ALUop. Sx, Sy, Dz`
	`ALUop. Sx, Dz`
	`ALUop. Sy, Dz`
	`MLTop. Se, Sf, Dg`

Table 3.18 Correspondence between DSP Instruction Operands and Registers

	ALU/Shift Operations				Multiply Operations		
Register	**Sx**	**Sy**	**Dz**	**Du**	**Se**	**Sf**	**Dg**
A0	Yes		Yes	Yes			Yes
A1	Yes		Yes	Yes	Yes	Yes	Yes
M0		Yes	Yes				Yes
M1		Yes	Yes				Yes
X0	Yes		Yes	Yes	Yes	Yes	
X1	Yes		Yes		Yes		
Y0		Yes	Yes	Yes	Yes	Yes	
Y1		Yes	Yes			Yes	

When writing parallel instructions, the B-field instruction is written first, followed by the A-field instruction. A sample parallel processing program is shown in figure 3.6.

```
        PADD   A0, M0, A0    PMULS  X0, Y0, M0    MOVX.W @R4+,    X0    MOVY.W @R6+, Y0
DCF     PINC   M1, A1                             MOVX.W @R5+R8,  X0    MOVY.W @R7+, Y1
        PCMP   M1, M0                             MOVX.W @R4,     X1    [NOPY]
```

Figure 3.6 Sample Parallel Instruction Program

[ ] mean that the contents can be omitted.

The no operation instructions NOPX and NOPY can be omitted. For details on the B field in DSP data operation instructions, refer to section 3.6.4, DSP Operation Instructions.

The DSR register condition code bit (DC) is always updated on the basis of the result of an unconditional ALU or shift operation instruction. Conditional instructions do not update the DC bit. Multiply instructions, also, do not update the DC bit. DC bit updating is performed by means of the CS[2:0] bits in the DSR register. The DC bit update rules are shown in table 3.19.

Table 3.19 DC Bit Update Definitions

CS [2:0]			Condition Mode	Description
0	0	0	Carry or borrow mode	The DC bit is set if an ALU arithmetic operation generates a carry or borrow, and is cleared otherwise. When a PSHA or PSHL shift instruction is executed, the last bit data shifted out is copied into the DC bit. When an ALU logical operation is executed, the DC bit is always cleared.
0	0	1	Negative value mode	When an ALU or shift (PSHA) arithmetic operation is executed, the MSB of the result, including the guard bits, is copied into the DC bit. When an ALU or shift (PSHL) logical operation is executed, the MSB of the result, excluding the guard bits, is copied into the DC bit.
0	1	0	Zero value mode	The DC bit is set if the result of an ALU or shift operation is all-zeros, and is cleared otherwise.
0	1	1	Overflow mode	The DC bit is set if the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, excluding the guard bits, and is cleared otherwise. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared.
1	0	0	Signed greater-than mode	This mode is similar to signed greater-or-equal mode, but DC is cleared if the result is all-zeros. DC = ~{(negative value ^ over-range) | zero value}; In case of arithmetic operation DC = 0; In case of logical operation
1	0	1	Signed greater-or-equal mode	If the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, including the guard bits (over-range), the definition is the same as in negative value mode. If the result is not over-range, the definition is the opposite of that in negative value mode. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared. DC = ~(negative value ^ over-range); In case of arithmetic operation DC = 0 ; In case of logical operation
1	1	0	Reserved (setting prohibited)	
1	1	1	Reserved (setting prohibited)	

- Conditional Operations and Data Transfer

  Some instructions belonging to this class can be executed conditionally, as described earlier. The specified condition is valid only for the B field of the instruction, and is not valid for data transfer instructions for which a parallel specification is made. Examples are shown in figure 3.7.

```
DCT PADD X0,Y0,A0  MOVX.W @R4+,X0  MOVY.W A0,@R6+R9 ;

When condition is True

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00005000, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'0088888888,
                  R4=H'00008002, R6=H'00005004, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456

When condition is False

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00005000, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008002, R6=H'00005004, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456
```

Figure 3.7 Examples of Conditional Operations and Data Transfer Instructions

- Assignment of NOPX and NOPY Instruction Codes

  When there is no data transfer instruction to be parallel-processed simultaneously with a DSP operation instruction, an NOPX or NOPY instruction can be written as the data transfer instruction, or the instruction can be omitted. The instruction code is the same whether an NOPX or NOPY instruction is written or the instruction is omitted. Examples of NOPX and NOPY instruction codes are shown in table 3.20.

Table 3.20 Examples of NOPX and NOPY Instruction Codes

Instruction			Code
PADD X0,Y0,A0	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111100000001011 1011000100000111
PADD X0,Y0,A0	NOPX	MOVY.W @R6+R9,Y0	1111100000000011 1011000100000111
PADD X0,Y0,A0	NOPX	NOPY	1111100000000000 1011000100000111
PADD X0,Y0,A0	NOPX		1111100000000000 1011000100000111
PADD X0,Y0,A0			1111100000000000 1011000100000111
	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111000000001011
	MOVX.W @R4+,X0	NOPY	1111000000001000
	MOVS.W @R4+,X0		1111010010001000
	NOPX	MOVY.W @R6+R9,Y0	1111000000000011
		MOVY.W @R6+R9,Y0	1111000000000011
	NOPX	NOPY	1111000000000000
NOP			0000000000001001

3.5.3 DSP-Type Data Formats

This LSI has several different data formats that depend on the instruction. This section explains the data formats for DSP type instructions.

Figure 3.8 shows three DSP-type data formats with different binary point positions. A CPU-type data format with the binary point to the right of bit 0 is also shown for reference.

The DSP-type fixed point data format has the binary point between bit 31 and bit 30. The DSP-type integer format has the binary point between bit 16 and bit 15. The DSP-type logical format does not have a binary point. The valid data lengths of the data formats depend on the instruction and the DSP register.

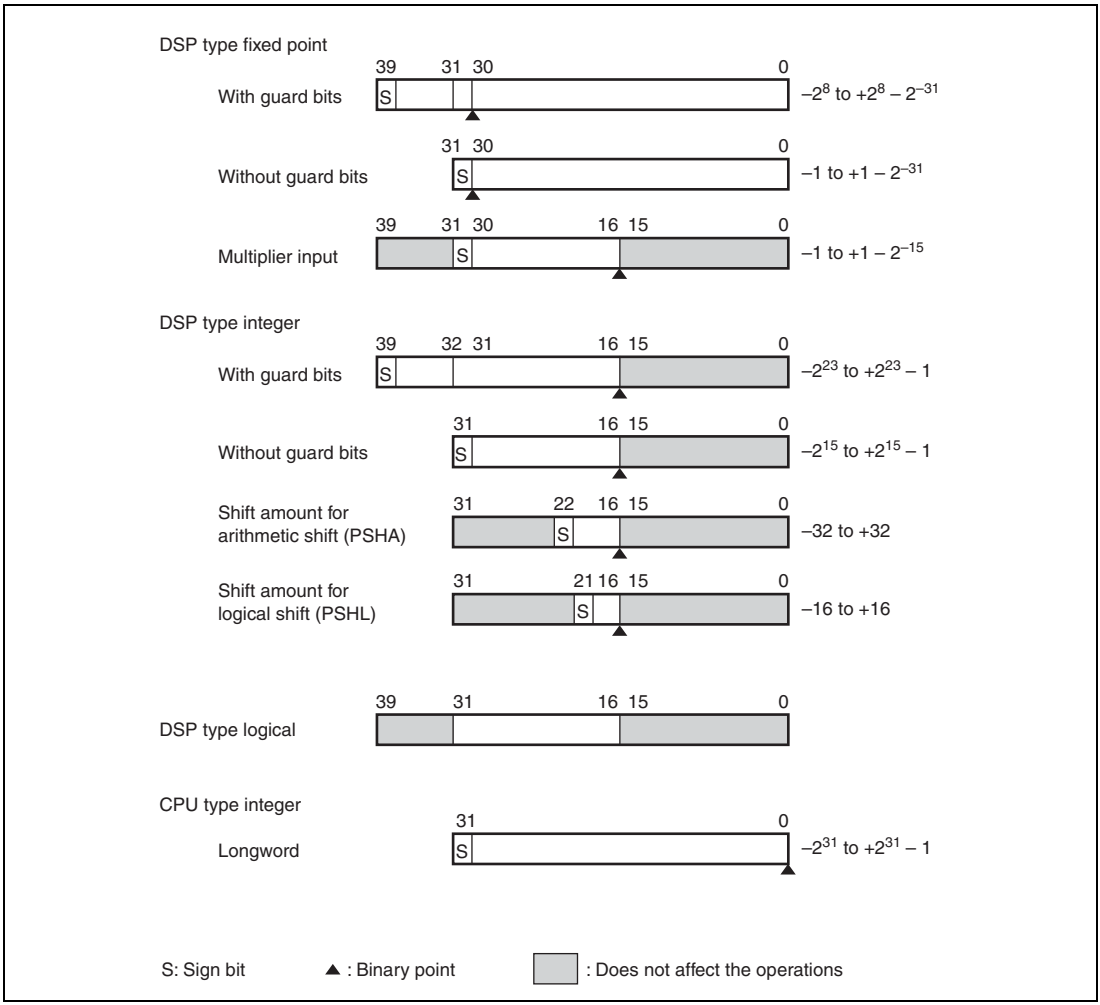


Figure 3.8 Data Formats

The shift amount for the arithmetic shift (PSHA) instruction has a 7-bit field that can represent values from –64 to +63, but –32 to +32 are valid numbers for the instruction. Also the shift amount for a logical shift operation has a 6-bit field, but –16 to +16 are valid numbers for the instruction. The results when an invalid shift amount is specified cannot be guaranteed.

3.5.4 ALU Fixed-Point Operations

Figure 3.9 shows the ALU arithmetic operation flow. Table 3.21 shows the variation of this type of operation and table 3.22 shows the correspondence between each operand and registers.

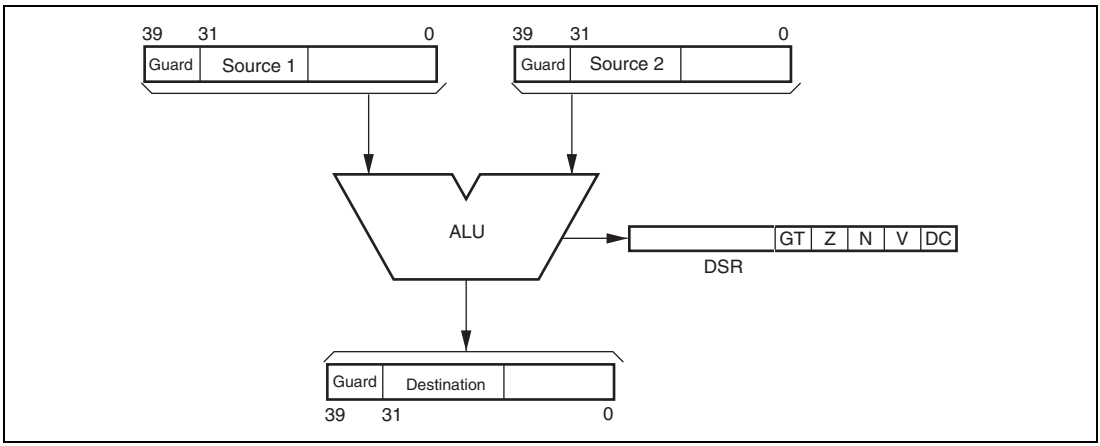


Figure 3.9 ALU Fixed-Point Arithmetic Operation Flow

Note: The ALU fixed-point arithmetic operations are basically 40-bit operation; 32 bits of the base precision and 8 bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

ALU fixed-point operations are executed between registers. Each source and destination operand are selected independently from one of the DSP registers. When a register providing guard bits is specified as an operand, the guard bits are activated for this type of operation. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Table 3.21 Variation of ALU Fixed-Point Operations

Mnemonic	Function	Source 1	Source 2	Destination
PADD	Addition	Sx	Sy	Dz (Du)
PSUB	Subtraction	Sx	Sy	Dz (Du)
PADDC	Addition with carry	Sx	Sy	Dz
PSUBC	Subtraction with borrow	Sx	Sy	Dz
PCMP	Comparison	Sx	Sy	—
PCOPY	Data copy	Sx	All 0	Dz
		All 0	Sy	Dz
PABS	Absolute	Sx	All 0	Dz
		All 0	Sy	Dz
PNEG	Negation	Sx	All 0	Dz
		All 0	Sy	Dz
PCLR	Clear	All 0	All 0	Dz

Table 3.22 Correspondence between Operands and Registers

Register	Sx	Sy	Dz	Du
A0	Yes		Yes	Yes
A1	Yes		Yes	Yes
M0		Yes	Yes	
M1		Yes	Yes	
X0	Yes		Yes	Yes
X1	Yes		Yes	
Y0		Yes	Yes	Yes
Y1		Yes	Yes	

As shown in figure 3.10, data loaded from the memory at the MA stage, which is programmed at the same line as the ALU operation, is not used as a source operand for this operation, even though the destination operand of the data load operation is identical to the source operand of the ALU operation. In this case, previous operation results are used as the source operands for the ALU operation, and then updated as the destination operand of the data load operation.

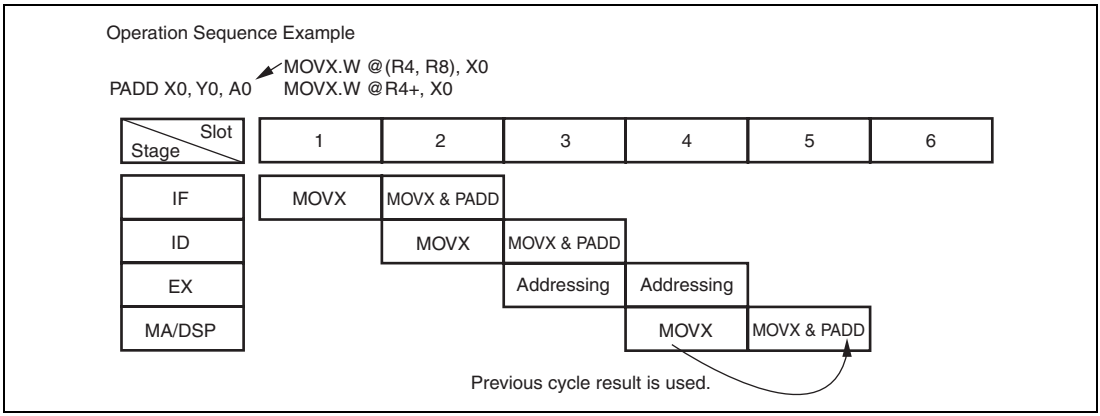


Figure 3.10 Operation Sequence Example

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. However, in case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of a DC bit is selected by CS[2:0] (condition selection) bits in DSR. The DC bit result is as follows:

Carry or Borrow Mode: CS[2:0] = 000: The DC bit indicates that carry or borrow is generated from the most significant bit of the operation result, except the guard-bit parts. Some examples are shown in figure 3.11. This mode is the default condition. When the input data is negative in a PABS or PNEG instruction, carry is generated.

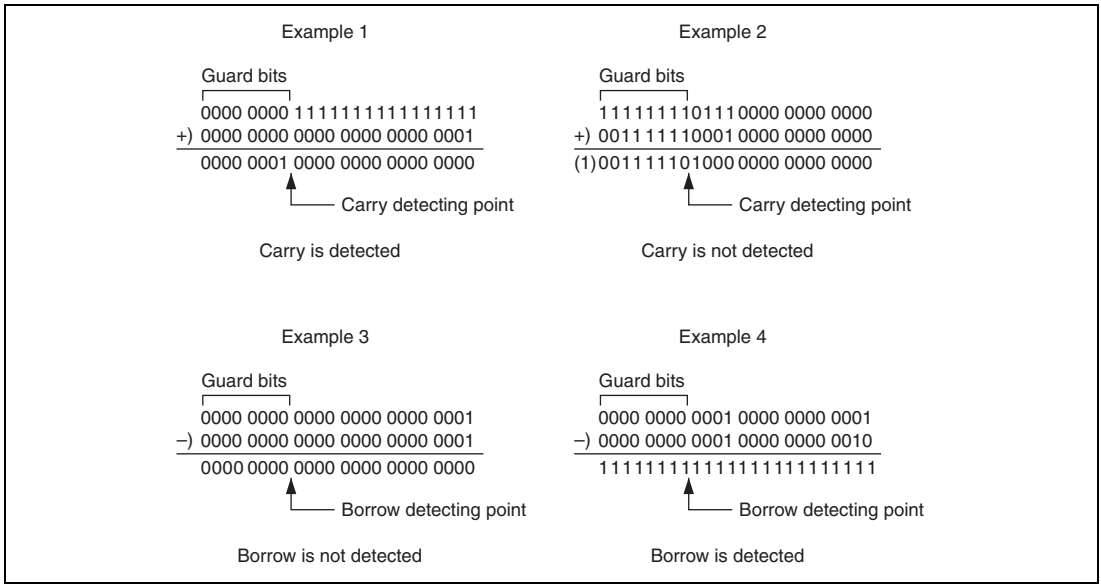


Figure 3.11 DC Bit Generation Examples in Carry or Borrow Mode

Negative Value Mode: CS[2:0] = 001: The DC flag indicates the same value as the MSB of the operation result. When the result is a negative number, the DC bit shows 1. When it is a positive number, the DC bit shows 0. The ALU always executes 40-bit arithmetic operation, so the sign bit to detect whether positive or negative is always got from the MSB of the operation result regardless of the destination operand. Some examples are shown in figure 3.12.

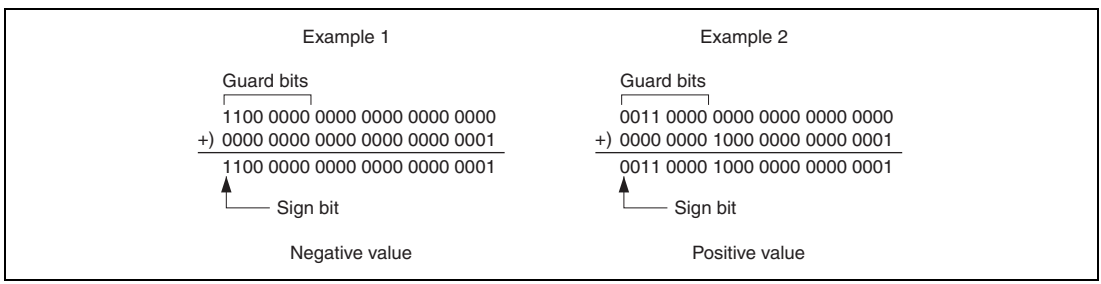


Figure 3.12 DC Bit Generation Examples in Negative Value Mode

Zero Value Mode: CS[2:0] = 010: The DC flag indicates whether the operation result is 0 or not. When the result is 0, the DC bit shows 1. When it is not 0, the DC bit shows 0.

Overflow Mode: CS[2:0] = 011: The DC bit indicates whether or not overflow occurs in the result. When an operation yields a result beyond the range of the destination register, except the

guard-bit parts, the DC bit is set. Even though guard bits are provided in the destination register, the DC bit always indicates the result of when no guard bits are provided. So, the DC bit is always set if the guard-bit parts are used for large number representation. Some DC bit generation examples in overflow mode are shown in figure 3.13.

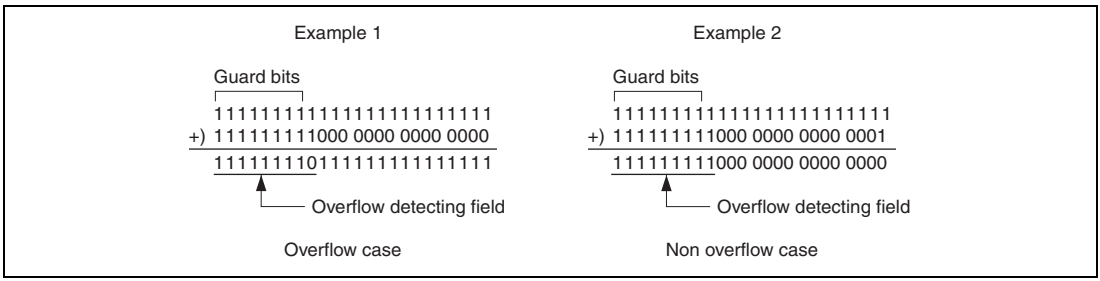


Figure 3.13 DC Bit Generation Examples in Overflow Mode

Signed Greater Than Mode: CS[2:0] = 100: The DC bit indicates whether or not the source 1 data (signed) is greater than the source 2 data (signed) as the result of compare operation PCMP. This mode is similar to the Negative Value Mode described before, because the result of a compare operation is a positive value if the source 1 data is greater than the source 2 data. However, the signed bit of the result shows a negative value if the compare operation yields a result beyond the range of the destination operand, including the guard-bit parts (called “Over-range”), even though the source 1 data is greater than the source 2 data. The DC bit is updated concerning this type of special case in this condition mode. The equation below shows the definition of getting this condition:

DC = ~ {(Negative ^ Over-range) | Zero}

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit’s result of the CMP/GT operation of the CPU instruction.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit indicates whether the source 1 data (signed) is greater than or equal to the source 2 data (signed) as the result of compare operation PCMP. This mode is similar to the Signed Greater Than Mode described before but the equal case is also included in this mode. The equation below shows the definition of getting this condition:

DC = ~ (Negative ^ Over-range)

When the PCMP operation is executed under this condition mode, the result of the DC bit is the same as the T bit’s result of a CMP/GE operation of the SH core instruction.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

Note: The DC bit is always updated as the carry/borrow flag for ‘PADDC’ and ‘PSUBC’ regardless of the CS[2:0] state.

- Overflow Protection

  The S bit in SR is effective for any ALU fixed-point arithmetic operations in the DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.5 ALU Integer Operations

Figure 3.14 shows the ALU integer arithmetic operation flow. Table 3.23 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.22.

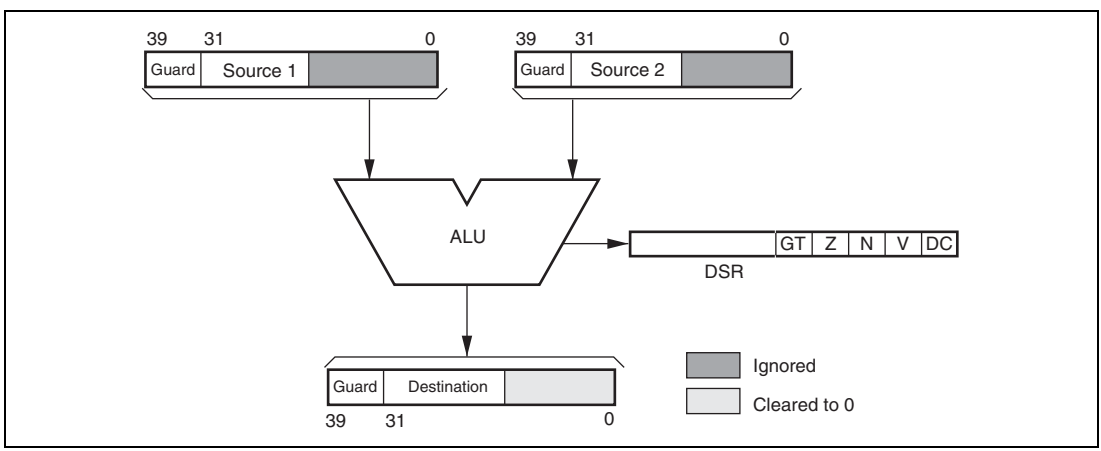


Figure 3.14 ALU Integer Arithmetic Operation Flow

Table 3.23 Variation of ALU Integer Operations

Mnemonic	Function	Source 1	Source 2	Destination
PINC	Increment by 1	Sx	+1	Dz
		+1	Sy	Dz
PDEC	Decrement by 1	Sx	–1	Dz
		–1	Sy	Dz

Note: The ALU integer operations are basically 24-bit operation, the upper 16 bits of the base precision and 8 bits of the guard-bits parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the upper word excluding the guard bits of the operation result are input into the destination register.

In ALU integer arithmetic operations, the lower word of the source operand is ignored and the lower word of the destination operand is automatically cleared. The guard-bit parts are effective in integer arithmetic operations if they are supported. Others are basically the same operation as ALU fixed-point arithmetic operations. As shown in table 3.23, however, this type of operation provides two kinds of instructions only, so that the second operand is actually either +1 or –1. When a word data is loaded into one of the DSP unit's registers, it is input as an upper word data. When a register providing guard bits is specified as an operand, the guard bits are also activated. These operations, as well as fixed-point operations, are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an ALU arithmetic operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. This is the same as fixed-point operations but the lower word of each source and destination operand is not used in order to generate them. See section 3.5.4, ALU Fixed-Point Operations, for details.

In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. See section 3.5.4, ALU Fixed-Point Operations, for details.

- Overflow Protection

  The S bit in SR is effective for any ALU integer arithmetic operations in DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.6 ALU Logical Operations

Figure 3.15 shows the ALU logical operation flow. Table 3.24 shows the variation of this type of operation. The correspondence between each operand and registers is the same as the ALU fixed-point operations as shown in table 3.21.

The ALU logical operation is executed between registers. Each source and destination operand is selected independently from one of the DSP registers. As shown in figure 3.15, this type of operation uses only the upper word of each operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared. These operations are also executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

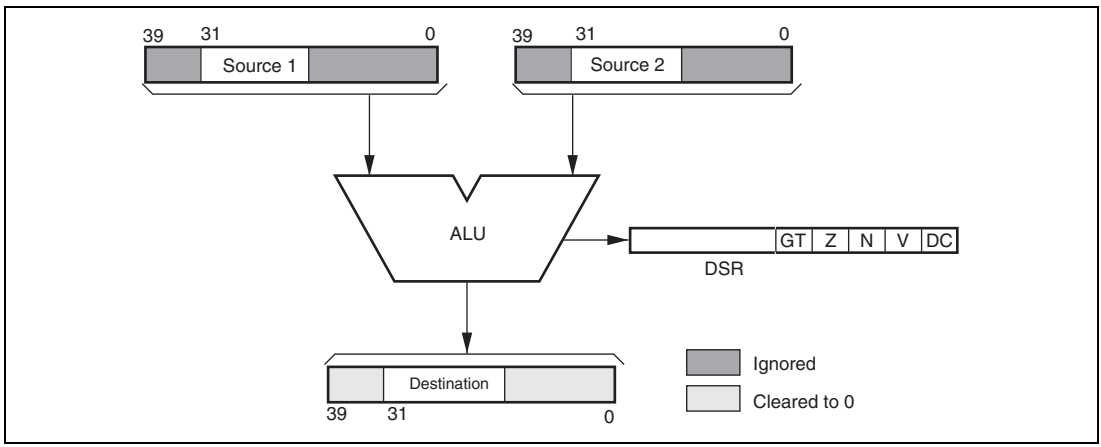


Figure 3.15 ALU Logical Operation Flow

Table 3.24 Variation of ALU Logical Operations

Mnemonic	Function	Source 1	Source 2	Destination
PAND	Logical AND	Sx	Sy	Dz
POR	Logical OR	Sx	Sy	Dz
PXOR	Logical exclusive OR	Sx	Sy	Dz

Every time an ALU logical operation is executed, the DC, N, Z, V, and GT bits in the DSR register are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the

operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

Carry or Borrow Mode: CS[2:0] = 000: The DC bit is always cleared.

Negative Value Mode: CS[2:0] = 001: Bit 31 of the operation result is loaded into the DC bit.

Zero Value Mode: CS[2:0] = 010: The DC bit is set when the operation result is zero; otherwise it is cleared.

Overflow Mode: CS[2:0] = 011: The DC bit is always cleared.

Signed Greater Than Mode: CS[2:0] = 100: The DC bit is always cleared.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.5.7 Fixed-Point Multiply Operation

Figure 3.16 shows the multiply operation flow. Table 3.25 shows the variation of this type of operation and table 3.26 shows the correspondence between each operand and registers. The multiply operation of the DSP unit is single-word signed single-precision multiplication. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

If a double-precision multiply operation is needed, the CPU standard double-word multiply instructions can be made of use.

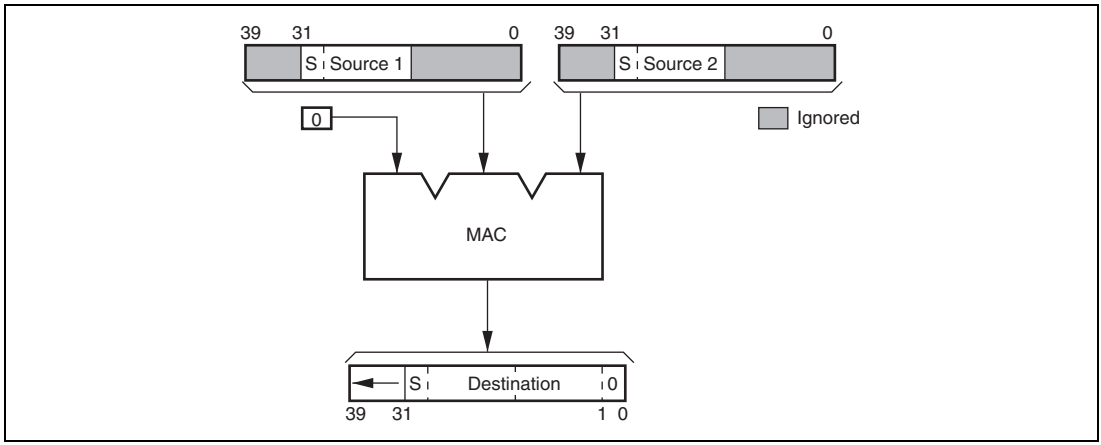


Figure 3.16 Fixed-Point Multiply Operation Flow

Table 3.25 Variation of Fixed-Point Multiply Operation

Mnemonic	Function	Source 1	Source 2	Destination
PMULS	Signed multiplication	Se	Sf	Dg

Table 3.26 Correspondence between Operands and Registers

Register	Se	Sf	Dg
A0	—	—	Yes
A1	Yes	Yes	Yes
M0	—	—	Yes
M1	—	—	Yes
X0	Yes	Yes	—
X1	Yes	—	—
Y0	Yes	Yes	—
Y1	—	Yes	—

Note: The multiply operations basically generate 32-bit operation results. So when a register providing the guard-bit parts are specified as a destination operand, the guard-bit parts will copy bit 31 of the operation result.

The multiply operation of the DSP unit side is not integer but fixed-point arithmetic. So, the upper words of each multiplier and multiplicand are input into a MAC unit as shown in figure 3.16. In the SH's standard multiply operations, the lower words of both source operands are input into a MAC unit. The operation result is also different from the SH's case. The SH's multiply operation

result is aligned to the LSB of the destination, but the fixed-point multiply operation result is aligned to the MSB, so that the LSB of the fixed-point multiply operation result is always 0.

Multiply is always unconditional, but does not affect any condition code bits, DC, N, Z, V, and GT , in DSR.

- Overflow Protection

  The S bit in SR is effective for this multiply operation in the DSP unit. See section 3.5.11, Overflow Protection, for details.

  If the S bit is 0, overflow occurs only when `H'8000*H'8000 ((-1.0)*(-1.0))` operation is executed as signed fixed-point multiply. The result is `H'00 8000 0000` but it does not mean (`+1.0`). If the S bit is 1, overflow is prevented and the result is `H'00 7FFF FFFF`.

3.5.8 Shift Operations

Shift operations can use either register or immediate value as the shift amount operand. Other source and destination operands are specified by the register. There are two kinds of shift operations of arithmetic and logical shifts. Table 3.27 shows the variation of this type of operation. The correspondence between each operand and registers, except for immediate operands, is the same as the ALU fixed-point operations as shown in table 3.21.

Table 3.27 Variation of Shift Operations

Mnemonic	Function	Source 1	Source 2	Destination
PSHA Sx, Sy, Dz	Arithmetic shift	Sx	Sy	Dz
PSHL Sx, Sy, Dz	Logical shift	Sx	Sy	Dz
PSHA #Imm1, Dz	Arithmetic shift with immediate.	Dz	Imm1	Dz
PSHL #Imm2, Dz	Logical shift with immediate.	Dz	Imm2	Dz

–32 <= Imm1 <= +32, –16 <= Imm2 <= +16

Arithmetic Shift: Figure 3.17 shows the arithmetic shift operation flow.

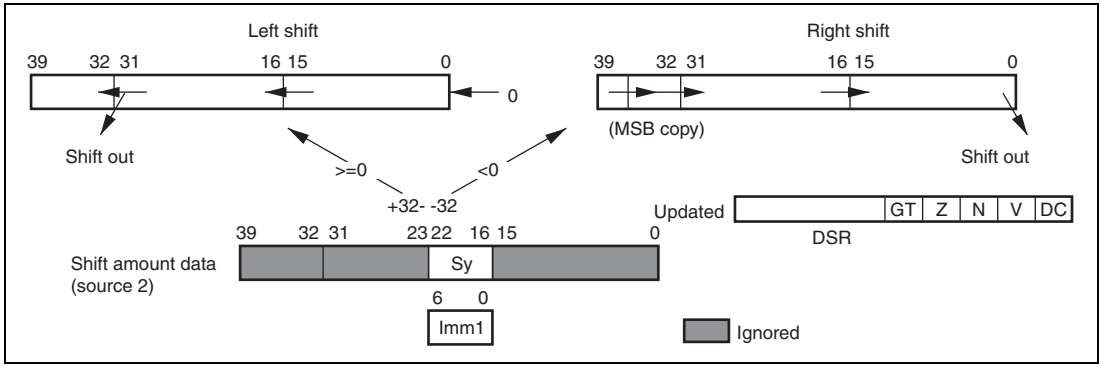


Figure 3.17 Arithmetic Shift Operation Flow

Note: The arithmetic shift operations are basically 40-bit operation, that is, the 32 bits of the base precision and eight bits of the guard-bit parts. So the signed bit is copied to the guard-bit parts when a register not providing the guard-bit parts is specified as the source operand. When a register not providing the guard-bit parts is specified as a destination operand, the lower 32 bits of the operation result are input into the destination register.

In this arithmetic shift operation, all bits of the source 1 and destination operands are activated. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either a register or immediate operand. The available shift range is from –32 to +32. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –64 to +63 but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. This operation is executed in the DSP stage, as shown in figure 3.10 as well as in fixed-point operations. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time an arithmetic shift operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001

The DC bit is set when the operation result is a negative value, and cleared when the operation result is zero or a positive value.

3. Zero Value Mode: CS[2:0] = 010

   The DC bit is set when the operation result is zero; otherwise it is cleared.

4. Overflow Mode: CS[2:0] = 011

   The DC bit is set when an overflow occurs.

5. Signed Greater Than Mode: CS[2:0] = 100

   The DC bit is always cleared.

6. Signed Greater Than or Equal Mode: CS[2:0] = 101

   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits. See the overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

- Overflow Protection

  The S bit in SR is also effective for arithmetic shift operation in the DSP unit. See section 3.5.11, Overflow Protection, for details.

Logical Shift: Figure 3.18 shows the logical shift operation flow.

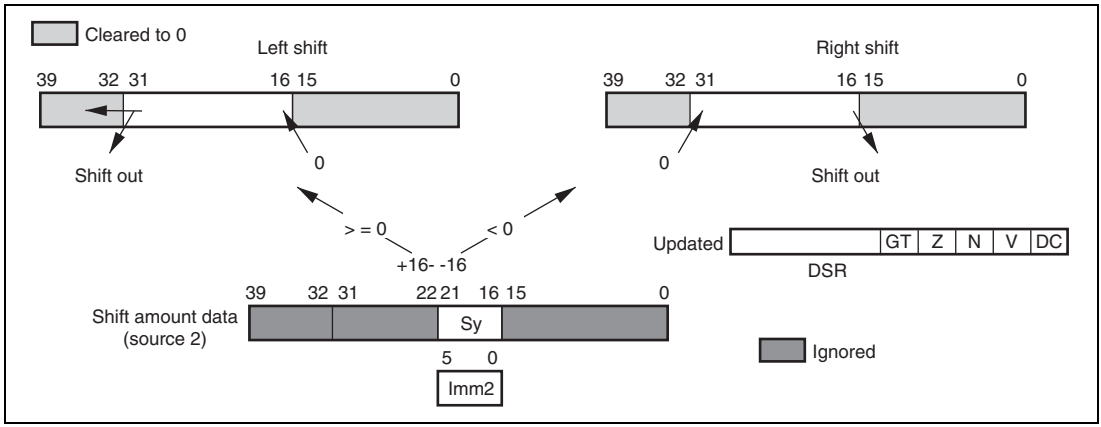


Figure 3.18 Logical Shift Operation Flow

As shown in figure 3.18, the logical shift operation uses the upper word of the source 1 operand and the destination operand. The lower word and guard-bit parts are ignored for the source operand and those of the destination operand are automatically cleared as in the ALU logical operations. The shift amount is specified by the source 2 operand as an integer data. The source 2 operand can be specified by either the register or immediate operand. The available shift range is from –16 to +16. Here, a negative value means the right shift, and a positive value means the left shift. It is possible for any source 2 operand to specify from –32 to +31, but the result is unknown if an invalid shift value is specified. In case of a shift with an immediate operand instruction, the source 1 operand must be the same register as the destination's. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a logical shift operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated even though the specified condition is true and the operation is executed. In case of an unconditional operation, they are always updated in accordance with the operation result. The definition of the DC bit is selected by the CS[2:0] (condition selection) bits in DSR. The DC bit result is:

1. Carry or Borrow Mode: CS[2:0] = 000
   The DC bit indicates the last shifted out data as the operation result.
2. Negative Value Mode: CS[2:0] = 001
   Bit 31 of the operation result is loaded into the DC bit.
3. Zero Value Mode: CS[2:0] = 010

The DC bit is set when the operation result is zero; otherwise it is cleared.

4. Overflow Mode: CS[2:0] = 011
   The DC bit is always cleared.
5. Signed Greater Than Mode: CS[2:0] = 100
   The DC bit is always cleared.
6. Signed Greater Than or Equal Mode: CS[2:0] = 101
   The DC bit is always cleared.

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in overflow mode by the CS[2:0] bits, but it is always cleared in this operation. So is the GT bit.

3.5.9 Most Significant Bit Detection Operation

The PDMSB, most significant bit detection operation, is used to calculate the shift amount for normalization. Figure 3.19 shows the PDMSB operation flow and table 3.28 shows the operation definition. Table 3.29 shows the possible variations of this type of operation. The correspondence between each operand and registers is the same as for ALU fixed-point operations, as shown in table 3.21.

Note: The result of the MSB detection operation is basically 24 bits as well as ALU integer operation, the upper 16 bits of the base precision and eight bits of the guard-bit parts. When a register not providing the guard-bit parts is specified as a destination operand, the upper word of the operation result is input into the destination register.

As shown in figure 3.19, the PDMSB operation uses all bits as a source operand, but the destination operand is treated as an integer operation result because shift amount data for normalization should be integer data as described in section 3.5.8, Shift Operations. These operations are executed in the DSP stage, as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time a PDMSB operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated, even though the specified condition is true, and the operation is executed. In case of an unconditional operation, they are always updated with the operation result.

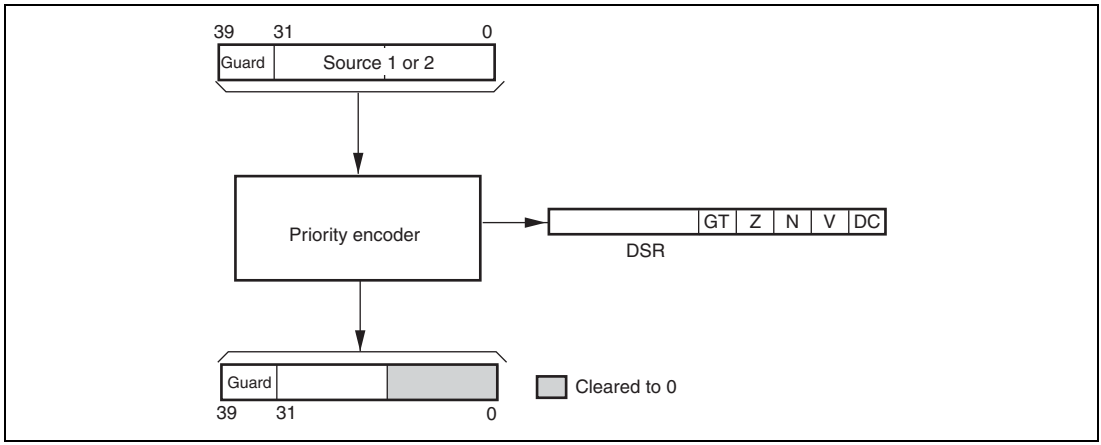


Figure 3.19 PDMSB Operation Flow

The definition of the DC bit is selected by the CS0–CS2 (condition selection) bits in DSR. The DC bit result is

Carry or Borrow Mode: CS[2:0] = 000: The DC bit is always cleared.

Negative Value Mode: CS[2:0] = 001: The DC bit is set when the operation result is a negative value, and cleared when the operation result is zero or a positive value.

Zero Value Mode: CS[2:0] = 010: The DC bit is set when the operation result is zero; otherwise it is cleared.

Overflow Mode: CS[2:0] = 011: The DC bit is always cleared.

Signed Greater Than Mode: CS[2:0] = 100: The DC bit is set when the operation result is a positive value; otherwise it is cleared.

Signed Greater Than or Equal Mode: CS[2:0] = 101: The DC bit is set when the operation result is zero or a positive value; otherwise it is cleared.

Table 3.28 Operation Definition of PDMSB

Source Data														Result for DST							
Guard Bit					Upper Word					Lower Word				Guard Bit	Upper Word							
39	38	…	33	32	31	30	29	28	…	3	2	1	0	39–32	31–22	21	20	19	18	17	16	Decimal
0	0	…	0	0	0	0	0	0	…	0	0	0	0	All 0	All 0	0	1	1	1	1	1	+31
0	0	…	0	0	0	0	0	0	…	0	0	0	1	All 0	All 0	0	1	1	1	1	0	+30
0	0	…	0	0	0	0	0	0	…	0	0	1	*	All 0	All 0	0	1	1	1	0	1	+29
0	0	…	0	0	0	0	0	0	…	0	1	*	*	All 0	All 0	0	1	1	1	0	0	+28
		:							:									:				
0	0	…	0	0	0	0	0	1	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
0	0	…	0	0	0	0	1	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
0	0	…	0	0	0	1	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
0	0	…	0	0	1	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
0	0	…	0	1	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
		:							:									:				
0	1	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
1	0	…	*	*	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	0	0	0	–8
																		:				
1	1	…	1	0	*	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	0	–2
1	1	…	1	1	0	*	*	*	…	*	*	*	*	All 1	All 1	1	1	1	1	1	1	–1
1	1	…	1	1	1	0	*	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	0	0
1	1	…	1	1	1	1	0	*	…	*	*	*	*	All 0	All 0	0	0	0	0	0	1	+1
1	1	…	1	1	1	1	1	0	…	*	*	*	*	All 0	All 0	0	0	0	0	1	0	+2
		:							:									:				
1	1	…	1	1	1	1	1	1	…	1	0	*	*	All 0	All 0	0	1	1	1	0	0	+28
1	1	…	1	1	1	1	1	1	…	1	1	0	*	All 0	All 0	0	1	1	1	0	1	+29
1	1	…	1	1	1	1	1	1	…	1	1	1	0	All 0	All 0	0	1	1	1	1	0	+30
1	1	…	1	1	1	1	1	1	…	1	1	1	1	All 0	All 0	0	1	1	1	1	1	+31

Note: * means don't care.

Table 3.29 Variation of PDMSB Operation

Mnemonic	Function	Source	Source 2	Destination
PDMSB	MSB detection	Sx	—	Dz
		—	Sy	Dz

The N bit always indicates the same state as the DC bit set in negative value mode by the CS[2:0] bits. See the negative value mode part above. The Z bit always indicates the same state as the DC bit set in zero value mode by the CS[2:0] bits. See the zero value mode part above. The V bit always indicates the same state as the DC bit set in Overflow mode by the CS[2:0] bit. See the Overflow mode part above. The GT bit always indicates the same state as the DC bit set in signed greater than mode by the CS[2:0] bits. See the signed greater than mode part above.

3.5.10 Rounding Operation

The DSP unit provides the function that rounds from 32 bits to 16 bits. In case of providing guard-bit parts, it rounds from 40 bits to 24 bits. When a round instruction is executed, H'00008000 is added to the source operand data and then, the lower word is cleared. Figure 3.20 shows the rounding operation flow and figure 3.21 shows the operation definition. Table 3.30 shows the variation of this type of operation. The correspondence between each operand and registers is the same as ALU fixed-point operations as shown in table 3.21.

As shown in figure 3.21, the rounding operation uses full-size data for both source and destination operands. These operations are executed in the DSP stage as shown in figure 3.10. The DSP stage is the same stage as the MA stage in which memory access is performed.

Every time rounding operation is executed, the DC, N, Z, V, and GT bits in DSR are basically updated in accordance with the operation result. In case of a conditional operation, they are not updated, even though the specified condition is true, and the operation is executed. In case of an unconditional operation, they are always updated with the operation results. The definition of the DC bit is selected by the CS0–CS2 (condition selection) bits in DSR. The result of these condition code bits is the same as the ALU-fixed point arithmetic operations.

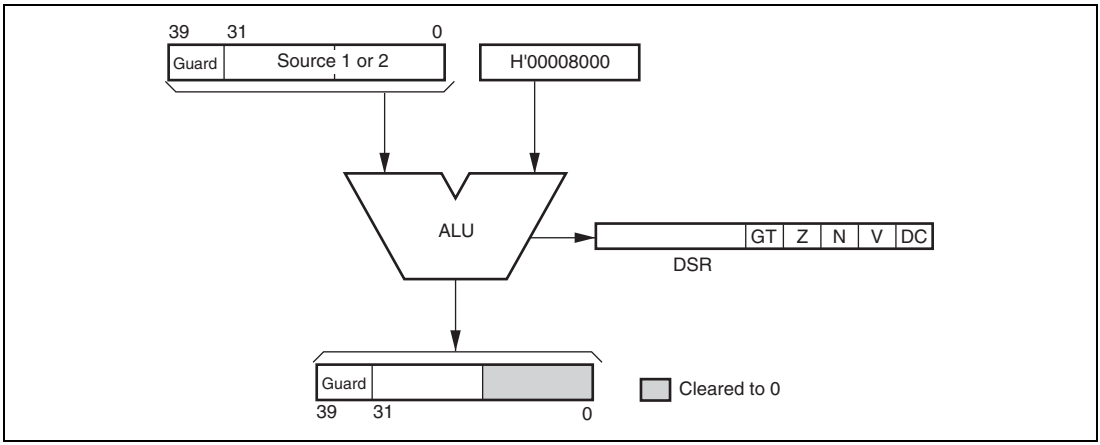


Figure 3.20 Rounding Operation Flow

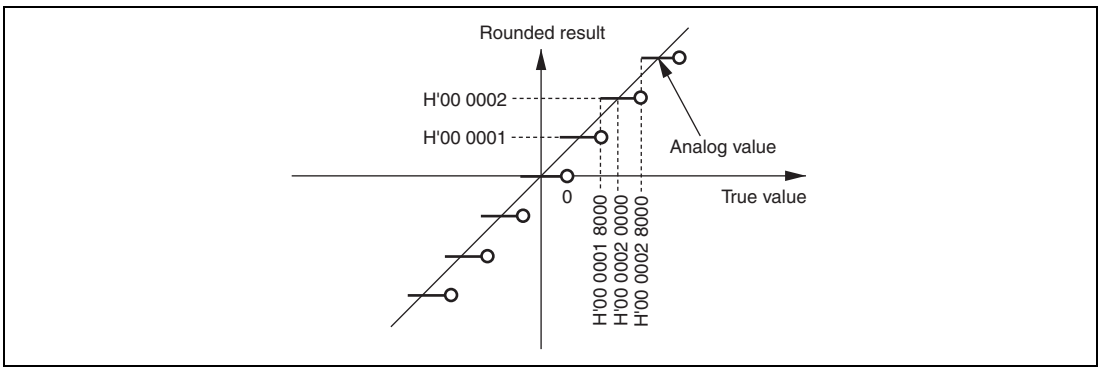


Figure 3.21 Definition of Rounding Operation

Table 3.30 Variation of Rounding Operation

Mnemonic	Function	Source 1	Source 2	Destination
PRND	Rounding	Sx	—	Dz
		—	Sy	Dz

- Overflow Protection

  The S bit in SR is effective for any rounding operations in the DSP unit. See section 3.5.11, Overflow Protection, for details.

3.5.11 Overflow Protection

The S bit in SR is effective for any arithmetic operations executed in the DSP unit, including the SH's standard multiply and MAC operations. The S bit in SR is used as the overflow protection enable bit. The arithmetic operation overflows when the operation result exceeds the range of two's complement representation without guard-bit parts. Table 3.31 shows the definition of overflow protection for fixed-point arithmetic operations, including fixed-point signed by signed multiplication described in section 3.5.7, Fixed-Point Multiply Operation. Table 3.32 shows the definition of overflow protection for integer arithmetic operations. The lower word of the saturation value of the integer arithmetic operation is don't care. Lower word value cannot be guaranteed.

When the overflow protection is effective, overflow never occurs. So, the V bit is cleared, and the DC bit is also cleared when the overflow mode is selected by the CS[2:0] bits.

Table 3.31 Definition of Overflow Protection for Fixed-Point Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $1 - 2^{-31}$	$1 - 2^{-31}$	00 7FFF FFFF
Negative	Result < –1	–1	FF 8000 0000

Table 3.32 Definition of Overflow Protection for Integer Arithmetic Operations

Sign	Overflow Condition	Fixed Value	Hex Representation
Positive	Result > $2^{15} - 1$	$2^{15} - 1$	00 7FFF ****
Negative	Result < -2^{15}	-2^{15}	FF 8000 ****

Note: * means don't care.

3.5.12 Local Data Move Instruction

The DSP unit of this LSI provides additional two independent registers, MACL and MACH, in order to support CPU standard multiply/MAC operations. They can be also used as temporary storage registers by local data move instructions between MACH/L and other DSP registers. Figure 3.22 shows the flow of seven local data move instructions. Table 3.33 shows the variation of this type of instruction.

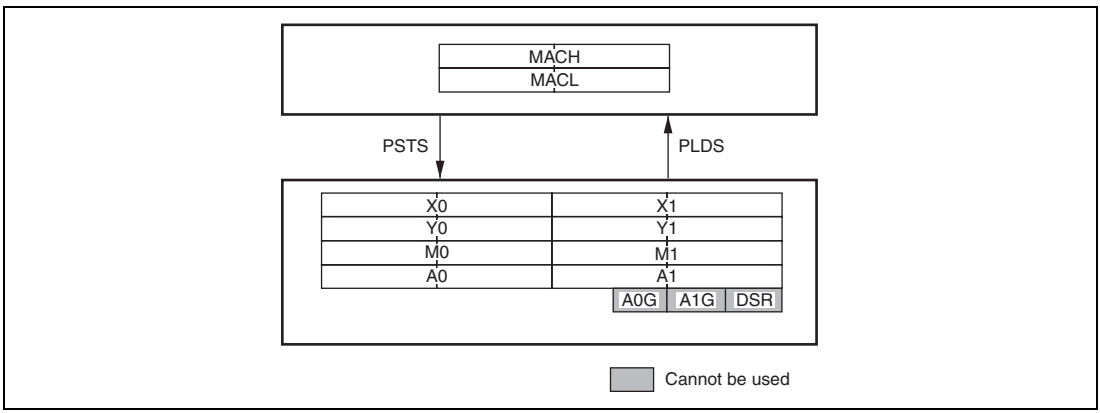


Figure 3.22 Local Data Move Instruction Flow

Table 3.33 Variation of Local Data Move Operations

Mnemonic	Function	Operand
PLDS	Data move from DSP register to MACL/MACH	Dz
PSTS	Data move from MACL/MACH to DSP register	Dz

This instruction is very similar to other transfer instructions. If either the A0 or A1 register is specified as the destination operand of PSTS, the signed bit is sign-extended and copied into the corresponding guard-bit parts, A0G or A1G. The DC bit in DSR and other condition code bits are not updated regardless of the instruction result. This instruction can operate as a conditional. This instruction can operate with MOVX and MOVY in parallel.

3.5.13 Operand Conflict

When an identical destination operand is specified with multiple parallel instructions, data conflict occurs. Table 3.34 shows the correspondence between each operand and registers.

Table 3.34 Correspondence between Operands and Registers

		X-Memory Load			Y-Memory Load			6-operand ALU			3-operand Multiply			3-operand ALU		
		Ax	Ix	Dx	Ay	Iy	Dy	Sx	Sy	Du	Se	Sf	Dg	Sx	Sy	Dz
DSP Registers	A0							*1		*2			*2	*1		*1
	A1							*1		*2	*1	*1	*2	*1		*1
	M0								*1				*1		*1	*1
	M1								*1				*1		*1	*1
	X0			*2				*1		*2	*1	*1		*1		*2
	X1			*2				*1			*1			*1		*2
	Y0						*2		*1	*2	*1	*1			*1	*2
	Y1						*2		*1			*1			*1	*2

Notes: 1. Registers available for operands
2. Registers available for operands (when there is operand conflict)

There are three cases of operand conflict problems.

- When ALU operation and multiply instructions specify the same destination operand (Du and Dg)
- When X-memory load and ALU operation specify the same destination operand (Dx and Du, or Dz)
- When Y-memory load and ALU operation specify the same destination operand (Dy and Du, or Dz)

In these cases above, the result is not guaranteed.

3.6 DSP Extended Function Instruction Set

3.6.1 CPU Extended Instructions

Table 3.35 DSP Mode Extended System Control Instructions

Instruction	Instruction Code	Operation	Execution States	T Bit	Category
SETRC #imm	10000010iiiiiiii	imm→RC (of SR)	1	—	
SETRC Rn	0100nnnn00010100	Rn[11:0] →RC(of SR)	1	—	
LDRS @(disp,PC)	10001100dddddddd	(disp x 2 + PC) →RS	1	—	
LDRE @(disp,PC)	10001110dddddddd	(disp x 2 + PC) →RE	1	—	
STC MOD,Rn	0000nnnn01010010	MOD→Rn	1	—	
STC RS,Rn	0000nnnn01100010	RS→Rn	1	—	
STC RE,Rn	0000nnnn01110010	RE→Rn	1	—	
STS DSR,Rn	0000nnnn01101010	DSR→Rn	1	—	
STS A0,Rn	0000nnnn01111010	A0→Rn	1	—	
STS X0,Rn	0000nnnn10001010	X0→Rn	1	—	
STS X1,Rn	0000nnnn10011010	X1→Rn	1	—	
STS Y0,Rn	0000nnnn10101010	Y0→Rn	1	—	
STS Y1,Rn	0000nnnn10111010	Y1→Rn	1	—	
STS.L DSR,@-Rn	0100nnnn01100010	Rn-4→Rn, DSR→(Rn)	1	—	
STS.L A0,@-Rn	0100nnnn01110010	Rn-4→Rn, A0→(Rn)	1	—	
STS.L X0,@-Rn	0100nnnn10000010	Rn-4→Rn, X0→(Rn)	1	—	
STS.L X1,@-Rn	0100nnnn10010010	Rn-4→Rn, X1→(Rn)	1	—	
STS.L Y0,@-Rn	0100nnnn10100010	Rn-4→Rn, Y0→(Rn)	1	—	
STS.L Y1,@-Rn	0100nnnn10110010	Rn-4→Rn, Y1→(Rn)	1	—	
STC.L MOD,@-Rn	0100nnnn01010011	Rn-4→Rn, MOD→(Rn)	1	—	
STC.L RS,@-Rn	0100nnnn01100011	Rn-4→Rn, RS→(Rn)	1	—	
STC.L RE,@-Rn	0100nnnn01110011	Rn-4→Rn, RE→(Rn)	1	—	
LDS.L @Rn + ,DSR	0100nnnn01100110	(Rn) →DSR, Rn + 4→Rn	1	—	
LDS.L @Rn + ,A0	0100nnnn01110110	(Rn) →A0, Rn + 4→Rn	1	—	
LDS.L @Rn + ,X0	0100nnnn10000110	(Rn) →X0, Rn + 4→Rn	1	—	
LDS.L @Rn + ,X1	0100nnnn10010110	(Rn) →X1, Rn + 4→Rn	1	—	

Instruction	Instruction Code	Operation	Execution States	T Bit	Category
LDS.L @Rn + ,Y0	0100nnnn10100110	(Rn) →Y0, Rn + 4→Rn	1	—	
LDS.L @Rn + ,Y1	0100nnnn10110110	(Rn) →Y1, Rn + 4→Rn	1	—	
LDC.L @Rn + ,MOD	0100nnnn01010111	(Rn) →MOD, Rn + 4→Rn	4	—	
LDC.L @Rn + ,RS	0100nnnn01100111	(Rn) →RS, Rn + 4→Rn	4	—	
LDC.L @Rn + ,RE	0100nnnn01110111	(Rn) →RE, Rn + 4→Rn	4	—	
LDS Rn,DSR	0100nnnn01101010	Rn→DSR	1	—	
LDS Rn,A0	0100nnnn01111010	Rn→A0	1	—	
LDS Rn,X0	0100nnnn10001010	Rn→X0	1	—	
LDS Rn,X1	0100nnnn10011010	Rn→X1	1	—	
LDS Rn,Y0	0100nnnn10101010	Rn→Y0	1	—	
LDS Rn,Y1	0100nnnn10111010	Rn→Y1	1	—	
LDC Rn,MOD	0100nnnn01011110	Rn→MOD	4	—	
LDC Rn,RS	0100nnnn01101110	Rn→RS	4	—	
LDC Rn,RE	0100nnnn01111110	Rn→RE	4	—	

3.6.2 Double-Data Transfer Instructions

Table 3.36 Double Data Transfer Instruction

Instruction		Instruction Code	Operation	Execution States	DC
X memory data transfer	NOPX	1111000*0*0*00**	X memory no access	1	—
	MOVX.W @Ax,Dx	111100A*D*0*01**	(Ax) → MSW of Dx, 0 →LSW of Dx	1	—
	MOVX.W @Ax + ,Dx	111100A*D*0*10**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + 2 → Ax	1	—
	MOVX.W @Ax + Ix,Dx	111100A*D*0*11**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix → Ax	1	—
	MOVX.W Da,@Ax	111100A*D*1*01**	MSW of Da → (Ax)	1	—
	MOVX.W Da,@Ax +	111100A*D*1*10**	MSW of Da → (Ax), Ax + 2 → Ax	1	—
	MOVX.W Da,@Ax + Ix	111100A*D*1*11**	MSW of Da → (Ax), Ax + Ix → Ax	1	—
Y memory data transfer	NOPY	111100*0*0*0**00	Y memory no access	1	—
	MOVY.W @Ay,Dy	111100*A*D*0**01	(Ay) → MSW of Dy, 0 → LSW of Dy	1	—
	MOVY.W @Ay + ,Dy	111100*A*D*0**10	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	1	—
	MOVY.W @Ay + Iy,Dy	111100*A*D*0**11	(Ay) → MSW of Dy, 0 →LSW of Dy, Ay + Iy →Ay	1	—
	MOVY.W Da,@Ay	111100*A*D*1**01	MSW of Da → (Ay)	1	—
	MOVY.W Da,@Ay +	111100*A*D*1**10	MSW of Da → (Ay), Ay + 2 → Ay	1	—
	MOVY.W Da,@Ay + Iy	111100*A*D*1**11	MSW of Da → (Ay), Ay + Iy → Ay	1	—

3.6.3 Single-Data Transfer Instructions

Table 3.37 Single Data Transfer Instructions

Instruction	Instruction Code	Operation	Execution States	DC	Category
MOVS.W @-As,Ds	111101AADDDD0000	As-2 → As, (As) → MSW of Ds, 0 → LSW of Ds	1	—	
MOVS.W @As,Ds	111101AADDDD0100	(As) → MSW of Ds, 0 → LSW of Ds	1	—	
MOVS.W @As + ,Ds	111101AADDDD1000	(As) → MSW of Ds, 0 → LSW of Ds, As + 2 → As	1	—	
MOVS.W @As + Ix,Ds	111101AADDDD1100	(As) → MSW of Ds, 0 → LSW of Ds, As + Ix → As	1	—	
MOVS.W Ds,@-As	111101AADDDD0001	As-2 → As, MSW of Ds → (As)	1	—	*
MOVS.W Ds,@As	111101AADDDD0101	MSW of Ds → (As)	1	—	*
MOVS.W Ds,@As +	111101AADDDD1001	MSW of Ds → (As), As + 2 →As	1	—	*
MOVS.W Ds,@As + Ix	111101AADDDD1101	MSW of Ds → (As), As + Ix → As	1	—	*
MOVS.L @-As,Ds	111101AADDDD0010	As-4 → As, (As) → Ds	1	—	
MOVS.L @As,Ds	111101AADDDD0110	(As) → Ds	1	—	
MOVS.L @As + ,Ds	111101AADDDD1010	(As) → Ds, As + 4 → As	1	—	
MOVS.L @As + Ix,Ds	111101AADDDD1110	(As) → Ds, As + Ix → As	1	—	
MOVS.L Ds,@-As	111101AADDDD0011	As-4 → As, Ds → (As)	1	—	
MOVS.L Ds,@As	111101AADDDD0111	Ds → (As)	1	—	
MOVS.L Ds,@As +	111101AADDDD1011	Ds → (As), As + 4 → As	1	—	
MOVS.L Ds,@As + Ix	111101AADDDD1111	Ds → (As), As + Ix → As	1	—	

Note: * If guard bit registers A0G and A1G are specified in source operand Ds, the data is output to the LDB[7:0] bus and the sign bit is copied into the upper bits, [31:8].

The correspondence between DSP data transfer operands and registers is shown in table 3.38.

Table 3.38 Correspondence between DSP Data Transfer Operands and Registers

Register		Ax	Ix	Dx	Ay	Iy	Dy	Da	As	Ds
SH	R0									
register	R1									
	R2 (As2)								Yes	
	R3 (As3)								Yes	
	R4 (Ax0)	Yes							Yes	
	R5 (Ax1)	Yes							Yes	
	R6 (Ay0)				Yes					
	R7 (Ay1)				Yes					
	R8 (Ix)		Yes							
	R9 (Iy)					Yes				
DSP	A0							Yes		Yes
register	A1							Yes		Yes
	M0									Yes
	M1									Yes
	X0			Yes						Yes
	X1			Yes						Yes
	Y0						Yes			Yes
	Y1						Yes			Yes
	A0G									Yes
	A1G									Yes

Note: Yes: The register which can be set.

3.6.4 DSP Operation Instructions

Table 3.39 DSP Operation Instructions

Instruction	Instruction Code	Operation	Execution States	DC
PMULS Se,Sf,Dg	111110**********	Se*Sf ->Dg (Signed)	1	—
	0100eeff0000gg00			
PADD Sx,Sy,Du	111110**********	Sx + Sy ->Du Se*Sf ->Dg (Signed)	1	*
PMULS Se,Sf,Dg	0111eeffxxyygguu			
PSUB Sx,Sy,Du	111110**********	Sx-Sy ->Du Se*Sf ->Dg (Signed)	1	*
PMULS Se,Sf,Dg	0110eeffxxyygguu			
PADD Sx,Sy,Dz	111110**********	Sx + Sy ->Dz	1	*
	10110001xxyyzzzz			
DCT PADD Sx,Sy,Dz	111110**********	If DC=1, Sx + Sy ->Dz If DC=0, nop	1	—
	10110010xxyyzzzz			
DCF PADD Sx,Sy,Dz	111110**********	If DC=0, Sx + Sy ->Dz If DC=1, nop	1	—
	10110011xxyyzzzz			
PSUB Sx,Sy,Dz	111110**********	Sx-Sy ->Dz	1	*
	10100001xxyyzzzz			
DCT PSUB Sx,Sy,Dz	111110**********	If DC=1, Sx-Sy ->Dz If DC=0, nop	1	—
	10100010xxyyzzzz			
DCF PSUB Sx,Sy,Dz	111110**********	If DC=0, Sx-Sy ->Dz If DC=1, nop	1	—
	10100011xxyyzzzz			
PSHA Sx,Sy,Dz	111110**********	If Sy>=0, Sx<<Sy ->Dz (arithmetic shift)	1	*
	10010001xxyyzzzz	If Sy<0, Sx>>Sy ->Dz		
DCT PSHA Sx,Sy,Dz	111110**********	If DC=1 & Sy>=0, Sx<<Sy ->Dz (arithmetic shift)	1	—
	10010010xxyyzzzz	If DC=1 & Sy<0, Sx>>Sy ->Dz If DC=0, nop		
DCF PSHA Sx,Sy,Dz	111110**********	If DC=0 & Sy>=0, Sx<<Sy ->Dz (arithmetic shift)	1	—
	10010011xxyyzzzz	If DC=0 & Sy<0, Sx>>Sy ->Dz If DC=1, nop		

Instruction	Instruction Code	Operation	Execution States	DC
PSHL Sx,Sy,Dz	111110**********	If Sy>=0, Sx<<Sy ->Dz (logical shift)	1	*
	10000001xxyyzzzz	If Sy<0, Sx>>Sy ->Dz		
DCT PSHL Sx,Sy,Dz	111110**********	If DC=1 & Sy>=0, Sx<<Sy ->Dz (logical shift)	1	—
	10000010xxyyzzzz	If DC=1 & Sy<0, Sx>>Sy ->Dz If DC=0, nop		
DCF PSHL Sx,Sy,Dz	111110**********	If DC=0 & Sy>=0, Sx<<Sy ->Dz (logical shift)	1	—
	10000011xxyyzzzz	If DC=0 & Sy<0, Sx>>Sy ->Dz If DC=1, nop		
PCOPY Sx,Dz	111110**********	Sx ->Dz	1	*
	11011001xx00zzzz			
PCOPY Sy,Dz	111110**********	Sy ->Dz	1	*
	1111100100yyzzzz			
DCT PCOPY Sx,Dz	111110**********	If DC=1, Sx ->Dz If DC=0, nop	1	—
	11011010xx00zzzz			
DCT PCOPY Sy,Dz	111110**********	If DC=1, Sy ->Dz If DC=0, nop	1	—
	1111101000yyzzzz			
DCF PCOPY Sx,Dz	111110**********	If DC=0, Sx ->Dz If DC=1, nop	1	—
	11011011xx00zzzz			
DCF PCOPY Sy,Dz	111110**********	If DC=0, Sy ->Dz If DC=1, nop	1	—
	1111101100yyzzzz			
PDMSB Sx,Dz	111110**********	Sx ->Dz normalization count shift value	1	*
	10011101xx00zzzz			
PDMSB Sy,Dz	111110**********	Sy ->Dz normalization count shift value	1	*
	1011110100yyzzzz			
DCT PDMSB Sx,Dz	111110**********	If DC=1, normalization count shift value Sx ->Dz If DC=0, nop	1	—
	10011110xx00zzzz			
DCT PDMSB Sy,Dz	111110**********	If DC=1, normalization count shift value Sy ->Dz If DC=0, nop	1	—
	1011111000yyzzzz			
DCF PDMSB Sx,Dz	111110**********	If DC=0, normalization count shift value Sx ->Dz If DC=1, nop	1	—
	10011111xx00zzzz			
DCF PDMSB Sy,Dz	111110**********	If DC=0, normalization count shift value Sy ->Dz If DC=1, nop	1	—
	1011111100yyzzzz			

Instruction	Instruction Code	Operation	Execution States	DC
PINC Sx,Dz	111110**********	MSW of Sx + 1 ->Dz	1	*
	10011001xx00zzzz			
PINC Sy,Dz	111110**********	MSW of Sy + 1 ->Dz	1	*
	1011100100yyzzzz			
DCT PINC Sx,Dz	111110**********	If DC=1, MSW of Sx + 1 ->Dz If DC=0, nop	1	—
	10011010xx00zzzz			
DCT PINC Sy,Dz	111110**********	If DC=1, MSW of Sy + 1 ->Dz If DC=0, nop	1	—
	1011101000yyzzzz			
DCF PINC Sx,Dz	111110**********	If DC=0, MSW of Sx + 1 ->Dz If DC=1, nop	1	—
	10011011xx00zzzz			
DCF PINC Sy,Dz	111110**********	If DC=0, MSW of Sy+ 1 ->Dz If DC=1, nop	1	—
	1011101100yyzzzz			
PNEG Sx,Dz	111110**********	0-Sx ->Dz	1	*
	11001001xx00zzzz			
PNEG Sy,Dz	111110**********	0-Sy ->Dz	1	*
	1110100100yyzzzz			
DCT PNEG Sx,Dz	111110**********	If DC=1, 0-Sx ->Dz If DC=0, nop	1	—
	11001010xx00zzzz			
DCT PNEG Sy,Dz	111110**********	If DC=1, 0-Sy ->Dz If DC=0, nop	1	—
	1110101000yyzzzz			
DCF PNEG Sx,Dz	111110**********	If DC=0, 0-Sx ->Dz If DC=1, nop	1	—
	11001011xx00zzzz			
DCF PNEG Sy,Dz	111110**********	If DC=0, 0-Sy ->Dz If DC=1, nop	1	—
	1110101100yyzzzz			
POR Sx,Sy,Dz	111110**********	Sx | Sy ->Dz	1	*
	10110101xxyyzzzz			
DCT POR Sx,Sy,Dz	111110**********	If DC=1, Sx | Sy ->Dz If DC=0, nop	1	—
	10110110xxyyzzzz			
DCF POR Sx,Sy,Dz	111110**********	If DC=0, Sx | Sy ->Dz If DC=1, nop	1	—
	10110111xxyyzzzz			

Instruction	Instruction Code	Operation	Execution States	DC
PAND Sx,Sy,Dz	111110**********	Sx & Sy ->Dz	1	*
	10010101xxyyzzzz			
DCT PAND Sx,Sy,Dz	111110**********	If DC=1, Sx & Sy ->Dz If DC=0, nop	1	—
	10010110xxyyzzzz			
DCF PAND Sx,Sy,Dz	111110**********	If DC=0, Sx & Sy ->Dz If DC=1, nop	1	—
	10010111xxyyzzzz			
PXOR Sx,Sy,Dz	111110**********	Sx ^ Sy ->Dz	1	*
	10100101xxyyzzzz			
DCT PXOR Sx,Sy,Dz	111110**********	If DC=1, Sx ^ Sy ->Dz If DC=0, nop	1	—
	10100110xxyyzzzz			
DCF PXOR Sx,Sy,Dz	111110**********	If DC=0, Sx ^ Sy ->Dz If DC=1, nop	1	—
	10100111xxyyzzzz			
PDEC Sx,Dz	111110**********	Sx [39:16]-1 ->Dz	1	*
	10001001xx00zzzz			
DCT PDEC Sx,Dz	111110**********	If DC=1, Sx [39:16]-1 ->Dz If DC=0, nop	1	—
	10001010xx00zzzz			
DCF PDEC Sx,Dz	111110**********	If DC=0, Sx [39:16]-1 ->Dz If DC=1, nop	1	—
	10001011xx00zzzz			
PDEC Sy,Dz	111110**********	Sy [31:16]-1 ->Dz	1	*
	1010100100yyzzzz			
DCT PDEC Sy,Dz	111110**********	If DC=1, Sy [31:16]-1 ->Dz If DC=0, nop	1	—
	1010101000yyzzzz			
DCF PDEC Sy,Dz	111110**********	If DC=0, Sy [31:16]-1 ->Dz If DC=1, nop	1	—
	1010101100yyzzzz			
PCLR Dz	111110**********	h'00000000 ->Dz	1	*
	100011010000zzzz			
DCT PCLR Dz	111110**********	If DC=1, h'00000000 ->Dz If DC=0, nop	1	—
	100011100000zzzz			
DCF PCLR Dz	111110**********	If DC=0, h'00000000 ->Dz If DC=1, nop	1	—
	100011110000zzzz			

Instruction	Instruction Code	Operation	Execution States	DC
PSHA #imm,Dz	111110**********	If imm>=0, Dz<<imm ->Dz (arithmetic shift)	1	*
	00010iiiiiiizzzz	If imm<0, Dz>>imm ->Dz		
PSHL #imm,Dz	111110**********	If imm>=0, Dz<<imm ->Dz (logical shift)	1	*
	00000iiiiiiizzzz	If imm<0, Dz>>imm ->Dz		
PSTS MACH,Dz	111110**********	MACH ->Dz	1	—
	1100110100000zzzz			
DCT PSTS MACH,Dz	111110**********	If DC=1, MACH ->Dz	1	—
	1100111000000zzzz			
DCF PSTS MACH,Dz	111110**********	If DC=0, MACH ->Dz	1	—
	1100111100000zzzz			
PSTS MACL,Dz	111110**********	MACL ->Dz	1	—
	1101110100000zzzz			
DCT PSTS MACL,Dz	111110**********	If DC=1, MACL ->Dz	1	—
	1101111000000zzzz			
DCF PSTS MACL,Dz	111110**********	If DC=0, MACL ->Dz	1	—
	1101111100000zzzz			
PLDS Dz,MACH	111110**********	Dz ->MACH	1	—
	1110110100000zzzz			
DCT PLDS Dz,MACH	111110**********	If DC=1, Dz ->MACH	1	—
	1110111000000zzzz			
DCF PLDS Dz,MACH	111110**********	If DC=0, Dz ->MACH	1	—
	1110111100000zzzz			
PLDS Dz,MACL	111110**********	Dz ->MACL	1	—
	1111110100000zzzz			
DCT PLDS Dz,MACL	111110**********	If DC=1, Dz ->MACL	1	—
	1111111000000zzzz			
DCF PLDS Dz,MACL	111110**********	If DC=0, Dz ->MACL	1	—
	1111111100000zzzz			
PADDC Sx,Sy,Dz	111110**********	Sx + Sy + DC ->Dz Carry ->DC	1	Carry
	10110000xxyyzzzz			

Instruction	Instruction Code	Operation	Execution States	DC
PSUBC Sx,Sy, Dz	111110**********	Sx-Sy-DC ->Dz Borrow ->DC	1	Borrow
	10100000xxyyzzzz			
PCMP Sx,Sy	111110**********	Sx-Sy ->DC update	1	*
	10000100xxyy0000			
PABS Sx,Dz	111110**********	If Sx<0, 0-Sx ->Dz If Sx>=0, Sx->Dz	1	*
	10001000xx00zzzz			
PABS Sy,Dz	111110**********	If Sy<0, 0-Sy ->Dz If Sy>=0, Sy ->Dz	1	*
	1010100000yyzzzz			
PRND Sx,Dz	111110**********	Sx + h'00008000 ->Dz h'0000 ->LSW of Dz	1	*
	10011000xx00zzzz			
PRND Sy,Dz	111110**********	Sy + h'00008000 ->Dz h'0000 ->LSW of Dz	1	*
	1011100000yyzzzz			

Note: * See table 3.19.

3.6.5 Operation Code Map in DSP Mode

Table 3.40 shows the operation code map including an instruction codes extended in the DSP mode.

Table 3.40 Operation Code Map

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0000	Rn	Fx	0000				
0000	Rn	Fx	0001				
0000	Rn	00MD	0010	STC SR, Rn	STC GBR, Rn	STC VBR, Rn	STC SSR, Rn
0000	Rn	01MD	0010	STC SPC, Rn	STC MOD, Rn	STC RS, Rn	STC RE, Rn
0000	Rn	10MD	0010	STC R0_BANK, Rn	STC R1_BANK, Rn	STC R2_BANK, Rn	STC R3_BANK, Rn
0000	Rn	11MD	0010	STC R4_BANK, Rn	STC R5_BANK, Rn	STC R6_BANK, Rn	STC R7_BANK, Rn
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rm	10MD	0011	PREF @Rm			
0000	Rn	Rm	01MD	MOV.B Rm, @(R0, Rn)	MOV.W Rm, @(R0, Rn)	MOV.L Rm, @(R0, Rn)	MUL.L Rm, Rn
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	10MD	1000				
0000	0000	11MD	1000				
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	00MD	1010	STS MACH, Rn	STS MACL, Rn	STS PR, Rn	
0000	Rn	01MD	1010			STS DSR, Rn	STS A0, Rn
0000	Rn	10MD	1010	STS X0, Rn	STS X1, Rn	STS Y0, Rn	STS Y1, Rn
0000	Rn	Fx	1011				

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0000	Rn	Rm	11MD	MOV. B @(R0, Rm), Rn	MOV.W @(R0, Rm), Rn	MOV.L @(R0, Rm), Rn	MAC.L @Rm+,@Rn+
0001	Rn	Rm	disp	MOV.L Rm, @(disp:4, Rn)			
0010	Rn	Rm	00MD	MOV.B Rm, @Rn	MOV.W Rm, @Rn	MOV.L Rm, @Rn	
0010	Rn	Rm	01MD	MOV.B Rm, @–Rn	MOV.W Rm, @–Rn	MOV.L Rm, @–Rn	DIV0S Rm, Rn
0010	Rn	Rm	10MD	TST Rm, Rn	AND Rm, Rn	XOR Rm, Rn	OR Rm, Rn
0010	Rn	Rm	11MD	CMP/STR Rm, Rn	XTRCT Rm, Rn	MULU.W Rm, Rn	MULSW Rm, Rn
0011	Rn	Rm	00MD	CMP/EQ Rm, Rn		CMP/HS Rm, Rn	CMP/GE Rm, Rn
0011	Rn	Rm	01MD	DIV1 Rm, Rn	DMULU.L Rm,Rn	CMP/HI Rm, Rn	CMP/GT Rm, Rn
0011	Rn	Rm	10MD	SUB Rm, Rn		SUBC Rm, Rn	SUBV Rm, Rn
0011	Rn	Rm	11MD	ADD Rm, Rn	DMULS.L Rm,Rn	ADDC Rm, Rn	ADDV Rm, Rn
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn	
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn	
0100	Rn	Fx	0010	STS.L MACH, @–Rn	STS.L MACL, @–Rn	STS.L PR, @–Rn	
0100	Rn	00MD	0011	STC.L SR, @–Rn	STC.L GBR, @–Rn	STC.L VBR, @–Rn	STC.L SSR, @–Rn
0100	Rn	01MD	0011	STC.L SPC, @–Rn	STC.L MOD, @–Rn	STC.L RS, @–Rn	STC.L RE, @–Rn
0100	Rn	10MD	0011	STC.L R0_BANK, @–Rn	STC.L R1_BANK, @–Rn	STC.L R2_BANK, @–Rn	STC.L R3_BANK, @–Rn
0100	Rn	11MD	0011	STC.L R4_BANK, @–Rn	STC.L R5_BANK, @–Rn	STC.L R6_BANK, @–Rn	STC.L R7_BANK, @–Rn
0100	Rn	Fx	0100	ROTL Rn	SETRC Rn	ROTCL Rn	
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn	
0100	Rm	00MD	0110	LDS.L @Rm+, MACH	LDS.L @Rm+, MACL	LDS.L @Rm+, PR	
0100	Rm	01MD	0110			LDS.L @Rm+, DSR	LDS.L @Rm+, A0
0100	Rm	10MD	0110	LDS.L @Rm+, X0	LDS.L @Rm+, X1	LDS.L @Rm+, Y0	LDS.L @Rm+, Y1

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0100	Rm	00MD	0111	LDC.L @Rm+, SR	LDC.L @Rm+, GBR	LDC.L @Rm+, VBR	LDC.L @Rm+, SSR
0100	Rm	01MD	0111	LDC.L @Rm+, SPC	LDC.L @Rm+, MOD	LDC.L @Rm+, RS	LDC.L @Rm+, RE
0100	Rm	10MD	0111	LDC.L @Rm+, R0_BANK	LDC.L @Rm+, R1_BANK	LDC.L @Rm+, R2_BANK	LDC.L @Rm+, R3_BANK
0100	Rm	11MD	0111	LDC.L @Rm+, R4_BANK	LDC.L @Rm+, R5_BANK	LDC.L @Rm+, R6_BANK	LDC.L @Rm+, R7_BANK
0100	Rn	Fx	1000	SHLL2 Rn	SHLL8 Rn	SHLL16 Rn	
0100	Rn	Fx	1001	SHLR2 Rn	SHLR8 Rn	SHLR16 Rn	
0100	Rm	00MD	1010	LDS Rm, MACH	LDS Rm, MACL	LDS Rm, PR	
0100	Rm	01MD	1010			LDS Rm, DSR	LDS Rm, A0
0100	Rm	10MD	1010	LDS Rm, X0	LDS Rm, X1	LDS Rm, Y0	LDS Rm, Y1
0100	Rm/ Rn	Fx	1011	JSR @Rm	TAS.B @Rn	JMP @Rm	
0100	Rn	Rm	1100	SHAD Rm, Rn			
0100	Rn	Rm	1101	SHLD Rm, Rn			
0100	Rm	00MD	1110	LDC Rm, SR	LDC Rm, GBR	LDC Rm, VBR	LDC Rm, SSR
0100	Rm	01MD	1110	LDC Rm, SPC	LDC Rm, MOD	LDC Rm, RS	LDC Rm, RE
0100	Rm	10MD	1110	LDC Rm, R0_BANK	LDC Rm, R1_BANK	LDC Rm, R2_BANK	LDC Rm, R3_BANK
0100	Rm	11MD	1110	LDC Rm, R4_BANK	LDC Rm, R5_BANK	LDC Rm, R6_BANK	LDC Rm, R7_BANK
0100	Rn	Rm	1111	MAC.W @Rm+, Rn+			
0101	Rn	Rm	disp	MOV.L @(disp:4, Rm), Rn			
0110	Rn	Rm	00MD	MOV.B @Rm, Rn	MOV.W @Rm, Rn	MOV.L @Rm, Rn	MOV Rm, Rn
0110	Rn	Rm	01MD	MOV.B @Rm+, Rn	MOV.W @Rm+, Rn	MOV.L @Rm+, Rn	NOT Rm, Rn
0110	Rn	Rm	10MD	SWAP.B Rm, Rn	SWAP.W Rm, Rn	NEGC Rm, Rn	NEG Rm, Rn
0110	Rn	Rm	11MD	EXTU.B Rm, Rn	EXTU.W Rm, Rn	EXTS.B Rm, Rn	EXTS.W Rm, Rn
0111	Rn	imm		ADD #imm : 8, Rn			
1000	00MD	Rn / imm	disp	MOV.B R0, @(disp: 4, Rn)	MOV.W R0, @(disp: 4, Rn)	SETRC #imm	

Instruction Code			Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB		LSB	MD: 00	MD: 01	MD: 10	MD: 11
1000	01MD	Rm disp	MOV.B @(disp:4, Rm), R0	MOV.W @(disp: 4, Rm), R0		
1000	10MD	imm/disp	CMP/EQ #imm:8, R0	BT disp: 8		BF disp: 8
1000	11MD	imm/disp	LDRS @(disp:8,PC)	BT/S disp: 8	LDRE @(disp:8,PC)	BF/S disp: 8
1001	Rn	disp	MOV.W @(disp : 8, PC), Rn			
1010	disp		BRA disp : 12			
1011	disp		BSR disp: 12			
1100	00MD	imm/disp	MOV.B R0, @(disp: 8, GBR)	MOV.W R0, @(disp: 8, GBR)	MOV.L R0, @(disp: 8, GBR)	TRAPA #imm: 8
1100	01MD	disp	MOV.B @(disp: 8, GBR), R0	MOV.W @(disp: 8, GBR), R0	MOV.L @(disp: 8, GBR), R0	MOVA @(disp: 8, PC), R0
1100	10MD	imm	TST #imm: 8, R0	AND #imm: 8, R0	XOR #imm: 8, R0	OR #imm: 8, R0
1100	11MD	imm	TST.B #imm: 8, @(R0, GBR)	AND.B #imm: 8, @(R0, GBR)	XOR.B #imm: 8, @(R0, GBR)	OR.B #imm: 8, @(R0, GBR)
1101	Rn	disp	MOV.L @(disp: 8, PC), Rn			
1110	Rn	imm	MOV #imm:8, Rn			
1111	00**	********	MOVX.W, MOVY.W Double data transfer instruction			
1111	01**	********	MOVS.W, MOVS.L Single data transfer instruction			
1111	10**	********	MOVX.W, MOVY.W Double data transfer instruction, with DSP parallel operation instruction (32-bit instruction)			
1111	11**	********				

Notes: 1. For details, refer to the SH-3/SH-3E/SH3-DSP Programming Manual.

2. Instructions in the hatched areas are DSP extended instructions. These instructions can be executed only when the DSP bit of the SR register is set to 1.

Section 4 Exception Handling

Exception handling is separate from normal program processing, and is performed by a routine separate from the normal program. For example, if an attempt is made to execute an undefined instruction code or an instruction protected by the CPU processing mode, a control function may be required to return to the source program by executing the appropriate operation or to report an abnormality and carry out end processing. In addition, a function to control processing requested by LSI on-chip modules or an LSI external module to the CPU may also be required.

Transferring control to a user-defined exception processing routine and executing the process to support the above functions are called exception handling. This LSI has two types of exceptions: general exceptions and interrupts. The user can execute the required processing by assigning exception handling routines corresponding to the required exception processing and then return to the source program.

A reset input can terminate the normal program execution and pass control to the reset vector after register initialization. This reset operation can also be regarded as an exception handling. This section describes an overview of the exception handling operation. Here, general exceptions and interrupts are referred to as exception handling. For interrupts, this section describes only the process executed for interrupt requests. For details on how to generate an interrupt request, refer to section 8, Interrupt Controller (INTC).

4.1 Register Descriptions

There are five registers for exception handling. A register with an undefined initial value should be initialized by the software. Refer to section 24, List of Registers, for the addresses and access sizes of these registers.

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

Figure 4.1 shows the bit configuration of each register.

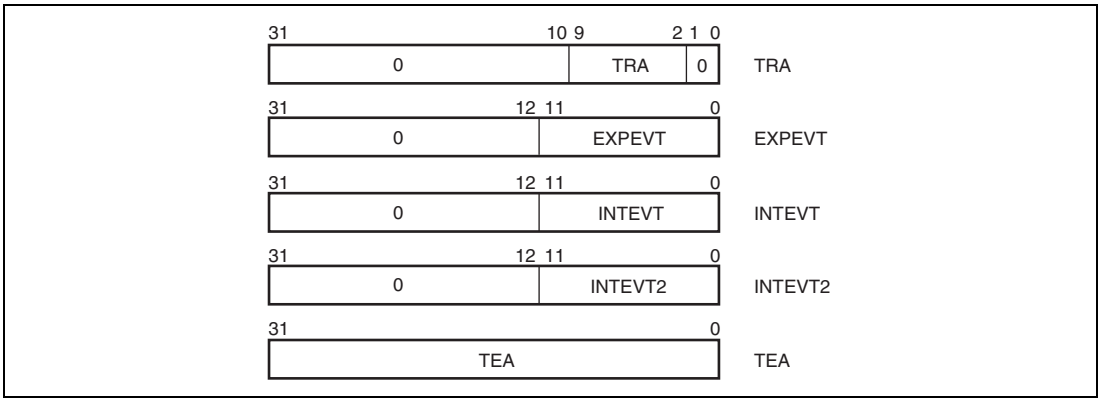


Figure 4.1 Register Bit Configuration

4.1.1 TRAPA Exception Register (TRA)

TRA is assigned to address H′FFFFFFD0 and consists of the 8-bit immediate data (imm) of the TRAPA instruction. TRA is automatically specified by the hardware when the TRAPA instruction is executed. Only bits 9 to 2 of the TRA can be re-written using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	TRA	—	R/W	8-bit Immediate Data
1, 0	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.

4.1.2 Exception Event Register (EXPEVT)

EXPEVT is assigned to address H′FFFFFFD4 and consists of a 12-bit exception code. Exception codes to be specified in EXPEVT are those for resets and general exceptions. These exception codes are automatically specified by the hardware when an exception occurs. Only bits 11 to 0 of EXPEVT can be re-written using the software.

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	EXPEVT	*	R/W	12-bit Exception Code

Note: * Initialized to H′000 at power-on reset and H′020 at manual reset.

4.1.3 Interrupt Event Register (INTEVT)

INTEVT is assigned to address H′FFFFFFD8 and consists of the exception code or the interrupt priority code. Whether the occurrence of an interrupt sets the exception code or the interrupt priority code depends on the interrupt sources. (See section 8.3.5, Interrupt Exception Handling and Priority, for details.) These exception codes and interrupt priority codes are automatically specified by the hardware when an exception occurs. Only bits 11 to 0 in INTEVT can be re-written using the software.

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	INTEVT	—	R/W	12-bit Exception Code

4.1.4 Interrupt Event Register 2 (INTEVT2)

INTEVT2 is assigned to address H′A4000000 and consists of the exception code. Exception codes to be specified in INTEVT2 are those for interrupt requests. These exception codes are automatically specified by the hardware when an exception occurs. INTEVT2 cannot be modified using the software.

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	INTEVT2	—	R	12-bit Exception Code

4.1.5 Exception Address Register (TEA)

TEA is assigned to address H′FFFFFFFC and the logical address for an exception occurrence is stored in this register when an exception related to memory accesses occurs. TEA can be modified using the software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TEA	0	R/W	Logical address for Exception Occurrence

4.2 Exception Handling Function

4.2.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC) and status register (SR) are saved in the saved program counter (SPC) and saved status register (SSR), respectively, and execution of the exception handler is invoked from a vector address. By executing the return from exception handler (RTE) in the exception handler routine, it restores the contents of PC and SR, and returns to the processor state at the point of interruption and the address where the exception occurred.

A basic exception handling sequence consists of the following operations. If an exception occurs and the CPU accepts it, operations 1 to 8 are executed.

1. The contents of PC is saved in SPC.
2. The contents of SR is saved in SSR.
3. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
4. The mode (MD) bit in SR is set to 1 to place the privileged mode.
5. The register bank (RB) bit in SR is set to 1.
6. An exception code identifying the exception event is written to bits 11–0 of the exception event (EXPEVT) or interrupt event (INTEVT or INTEVT2) register.
7. If a TRAPA instruction is executed, an 8-bit immediate data specified by the TRAPA instruction is set to TRA. For an exception related to memory accesses, the logic address where the exception occurred is written to TEA. *[1]
8. Instruction execution jumps to the designated exception vector address to invoke the handler routine.

The above operations from 1 to 8 are executed in sequence. During these operations, no other exceptions may be accepted unless multiple exception acceptance is enabled.

In an exception handling routine for a general exception, the appropriate exception handling must be executed based on an exception source determined by the EXPEVP. In an interrupt exception handling routine, the appropriate exception handling must be executed based on an exception source determined by the INTEVT or INTEVT2. After the exception handling routine has been completed, program execution can be resumed by executing an RTE instruction. The RTE instruction causes the following operations to be executed.

1. The contents of the SSR are restored into the SR to return to the processing state in effect before the exception handling took place.
2. A delay slot instruction of the RTE instruction is executed.*[2]

3. Control is passed to the address stored in the SPC.

The above operations from 1 to 3 are executed in sequence. During these operations, no other exceptions may be accepted. By changing the SPC and SSR before executing the RTE instruction, a status different from that in effect before the exception handling can also be specified.

Notes: 1. The MMU registers are modified if an MMU exception occurs.
2. For details on the CPU processing mode in which RTE delay slot instructions are executed, please refer to section 4.5, Usage Notes.

4.2.2 Exception Vector Addresses

A vector address for general exceptions is determined by adding a vector offset to a vector base address. The vector offset for general exceptions other than the TLB error exception is H′00000100. The vector offset for interrupts is H′00000600. The vector base address is loaded into the vector base register (VBR) using the software. The vector base address should reside in the P1 or P2 fixed physical address space.

4.2.3 Exception Codes

The exception codes are written to bits 11 to 0 in EXPEVT (for reset or general exceptions) or INTEVT2 (for interrupt requests) to identify each specific exception event. See section 8, Interrupt Controller (INTC), for details on the exception codes for interrupt requests. Table 4.1 lists exception codes for resets and general exceptions.

4.2.4 Exception Request and BL Bit (Multiple Exception Prevention)

The BL bit in SR is set to 1 when a reset or exception is accepted. While the BL bit is set to 1, acceptance of general exceptions is restricted as described below, making it possible to effectively prevent multiple exceptions from being accepted.

If the BL bit is set to 1, an interrupt request is not accepted and is retained. The interrupt request is accepted when the BL bit is cleared to 0. If the CPU is in low power consumption mode, an interrupt is accepted even if the BL bit is set to 1 and the CPU returns from the low power consumption mode.

A DMA error is not accepted and is retained if the BL bit is set to 1 and accepted when the BL bit is cleared to 0. User break requests generated while the BL bit is set are ignored and are not retained. Accordingly, user breaks are not accepted even if the BL bit is cleared to 0.

If a general exception other than a DMA address error or user break occurs while the BL bit is set to 1, the CPU enters a state similar to that in effect immediately after a reset, and passes control to the reset vector (H′A0000000) (multiple exception). In this case, unlike a normal reset, modules other than the CPU are not initialized, the contents of EXPEVT, SPC, and SSR are undefined, and this status is not detected by an external device.

To enable acceptance of multiple exceptions, the contents of SPC and SSR must be saved while the BL bit is set to 1 after an exception has been accepted, and then the BL bit must be cleared to 0. Before restoring the SPC and SSR, the BL bit must be set to 1.

4.2.5 Exception Source Acceptance Timing and Priority

Exception Request of Instruction Synchronous Type and Instruction Asynchronous Type: Resets and interrupts are requested asynchronously regardless of the program flow. In general exceptions, a DMA address error and a user break under the specific condition are also requested asynchronously. The user cannot expect on which instruction an exception is requested. For general exceptions other than a DMA address error and a user break under a specific condition, each general exception corresponds to a specific instruction.

Re-Execution Type and Processing-Completion Type Exceptions: All exceptions are classified into two types: a re-execution type and a processing-completion type. If a re-execution type exception is accepted, the current instruction executed when the exception is accepted is terminated and the instruction address is saved to the SPC. After returning from the exception processing, program execution resumes from the instruction where the exception was accepted. In a processing-completion type exception, the current instruction executed when the exception is accepted is completed, the next instruction address is saved to the SPC, and then the exception processing is executed.

During a delayed branch instruction and delay slot, the following operations are executed. A re-execution type exception detected in a delay slot is accepted before executing the delayed branch instruction. A processing-completion type exception detected in a delayed branch instruction or a delay slot is accepted when the delayed branch instruction has been executed. In this case, the acceptance of delayed branch instruction or a delay slot precedes the execution of the branch destination instruction. In the above description, a delay slot indicates an instruction following an unconditional delayed branch instruction or an instruction following a conditional delayed branch instruction whose branch condition is satisfied. If a branch does not occur in a conditional delayed branch, the normal processing is executed.

Acceptance Priority and Test Priority: Acceptance priorities are determined for all exception requests. The priority of resets, general exceptions, and interrupts are determined in this order: a reset is always accepted regardless of the CPU status. Interrupts are accepted only when resets or general exceptions are not requested.

If multiple general exceptions occur simultaneously in the same instruction, the priority is determined as follows.

1. A processing-completion type exception generated at the previous instruction*
2. A user break before instruction execution (re-execution type)
3. An exception related to an instruction fetch (CPU address error and MMU related exceptions: re-execution type)
4. An exception caused by an instruction decode (General illegal instruction exceptions and slot illegal instruction exceptions: re-execution type, unconditional trap: processing-completion type)
5. An exception related to data access (CPU address error and MMU related exceptions: re-execution type)
6. Unconditional trap (processing-completion type)
7. A user break other than one before instruction execution (processing-completion type)
8. DMA address error (processing-completion type)

Note: * If a processing-completion type exception is accepted at an instruction, exception processing starts before the next instruction is executed. This exception processing executed before an exception generated at the next instruction is detected.

Only one exception is accepted at a time. Accepting multiple exceptions sequentially results in all exception requests being processed.

Table 4.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Process at BL=1	Vector Code	Vector Offset
Reset (asynchronous)	Aborted	Power-on reset	1	—	Reset	H'000	—
		Manual reset	1	—	Reset	H'020	—
		H-UDI reset	1	1	Reset	H'000	—
General exception events (synchronous)	Re-executed	User break(before instruction execution)	2	0	Ignored	H'1E0	H'00000100
		CPU address error (instruction access) *[4]	2	1	Reset	H'0E0	H'00000100
		*[5] TLB miss (instruction access) *[4]	2	1-1	Reset	H'040	H'00000400
		TLB invalid (instruction access)*[4]	2	1-2	Reset	H'040	H'00000100
		TLB protection violation (instruction access)*[4]	2	1-3	Reset	H'0A0	H'00000100
		Illegal general instruction exception	2	2	Reset	H'180	H'00000100
		Illegal slot instruction exception	2	2	Reset	H'1A0	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	4	Reset	H'160	H'00000100
	Re-executed	CPU address error (data read/write)*[4]	2	3	Reset	H'0E0/ H'100	H'00000100
		[5] TLB miss (data read/write)[4]	2	3-1	Reset	H'040/ H'060	H'00000400
		TLB invalid (data read/write)*[4]	2	3-2	Reset	H'040/ H'060	H'00000100
		TLB protection violation (data read/write)*[4]	2	3-3	Reset	H'0A0/ H'0C0	H'00000100
		Initial page write (data write)*[4]	2	3-4	Reset	H'080	H'00000100
	Completed	User breakpoint (After instruction execution, address)	2	5	Ignored	H'1E0	H'00000100

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Process at BL=1	Vector Code	Vector Offset
General exception events (asynchronous)	Completed	User breakpoint (Data break, I-BUS break)	2	5	Ignored	H'1E0	H'00000100
		DMA address error	2	6	Retained	H'5C0	H'00000100
General interrupt requests (asynchronous)	Completed	Interrupt requests	3	—*[2]	Retained	—*[3]	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being the highest and 3 the lowest.
A reset has the highest priority. An interrupt is accepted only when general exceptions are not requested.
2. For details on priorities in multiple interrupt sources, refer to section 8, Interrupt Controller (INTC).
3. If an interrupt is accepted, the interrupt source register (EXPEVT) is not changed. The interrupt source code is specified in interrupt source register 2 (EXPEVT2). For details, refer to section 8, Interrupt Controller (INTC).
4. If one of these exceptions occurs in a specific part of the repeat loop, a specific code and vector offset are specified.
5. These exception codes are valid when the MMU is used.

4.3 Individual Exception Operations

This section describes the conditions for specific exception handling, and the processor operations. Resets and general exceptions are described in particular. For details on interrupt operations, refer to section 8, Interrupt Controller (INTC).

4.3.1 Resets

Power-On Reset:

- Conditions
  Power-on reset is request
- Operations
  Set EXPEVT to H'000, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H′A0000000. For details, refer to the register descriptions in the relevant sections.
  Be sure to perform power-On Reset at the time of a power supply injection.

Manual Reset:

- Conditions
  Manual reset is request
- Operations
  Set EXPEVT to H'020, initialize the CPU and on-chip peripheral modules, and branch to the reset vector H′A0000000. For details, refer to the register descriptions in the relevant sections.

H-UDI Reset:

- Conditions
  An H-UDI reset command is input (see section 23.4.4 H-UDI Reset.)
- Operations
  EXPEVT is set to H'000, vector base register (VBR) and status register (SR) are initialized, and branched to the reset vector (H'A0000000). VBR is cleared to H'00000000 by initialization. In SR, the MD, RB, and BL bits are set to 1, the DSP bit is cleared to 0, and the interrupt mask bits (I3 to I0) are set to B'1111. Then, the CPU and on-chip peripheral modules are initialized. For details, see the Register Description in each section.

4.3.2 General Exceptions

CPU address error:

- Conditions
  - — Instruction is fetched from odd address (4n + 1, 4n + 3)
  - — Word data is accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - — Longword is accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
  - — The area ranging from H'80000000 to H'FFFFFFFF in logical space is accessed in user mode
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H′0E0

  An exception occurred during write: H′100
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA.

Illegal general instruction exception:

- Conditions
  - — When undefined code not in a delay slot is decoded

    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S

Note: For details on undefined code, refer to table 2.12 in section 2, Operation Code Map. When an undefined code other than H′FC00 to H′FFFF is decoded, operation cannot be guaranteed.

  - — When a privileged instruction not in a delay slot is decoded in user mode

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

- Types
  Instruction synchronous, re-execution type
- Save address
  An instruction address where an exception occurs
- Exception code
  H′180
- Remarks
  None

Illegal slot instruction:

- Conditions
  — When undefined code in a delay slot is decoded
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
  — When a privileged instruction in a delay slot is decoded in user mode
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions.
  — When an instruction that rewrites PC in a delay slot is decoded
    Instructions that rewrite PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L @Rm+, SR
- Types
  Instruction synchronous, re-execution type
- Save address
  A delayed branch instruction address
- Exception code
  H′1A0
- Remarks
  None

Unconditional trap:

- Conditions
  TRAPA instruction executed
- Types
  Instruction synchronous, processing-completion type
- Save address
  An address of an instruction following TRAPA
- Exception code
  H′160
- Remarks
  The exception is a processing-completion type, so PC of the instruction after the TRAPA instruction is saved to SPC. The 8-bit immediate value in the TRAPA instruction is quadrupled and set in TRA[9:2].

User break point trap:

- Conditions
  When a break condition set in the user break controller is satisfied
- Types
  Break (L bus) before instruction execution: Instruction synchronous, re-execution type
  Operand break (L bus): Instruction synchronous, processing-completion type
  Data break (L bus): Instruction asynchronous, processing-completion type
  I bus break: Instruction asynchronous, processing-completion type
- Save address
  Re-execution type: An address of the instruction where a break occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
  Processing-completion type: An address of the instruction following the instruction where a break occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
- Exception code
  H′1E0
- Remarks
  For details on the user break controller, refer to section 9, User Break Controller.

DMA address error:

- Conditions
  - Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
  - Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
- Types
  Instruction asynchronous, processing-completion type
- Save address
  An address of the instruction following the instruction where an exception occurs (a delayed branch instruction destination address if an instruction is assigned to a delay slot)
- Exception code
  H′5C0
- Remarks
  An exception occurs when a DMA transfer is executed while an illegal instruction address described above is specified in the DMAC. Since the DMA transfer is performed asynchronously with the CPU instruction operation, an exception is also requested asynchronously with the instruction execution. For details on the DMAC, refer to section 13, Direct Memory Access Controller (DMAC).

4.3.3 General Exceptions (MMU Exceptions)

When the address translation unit of the memory management unit (MMU) is valid, MMU exceptions are checked after a CPU address error has been checked. Four types of MMU exceptions are defined: TLB miss exception, TLB invalid exception, TLB protection exception, initial page write exception. These exceptions are checked in this order.

A vector offset for a TLB miss exception is defined as H′00000400 to simplify exception source determination. For details on MMU exception operations, refer to section 5, Memory Management Unit (MMU).

TLB miss exception:

- Conditions
  Comparison of TLB addresses shows no address match.
- Types
  Instruction synchronous, re-execution type

- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H′040

  An exception occurred during write: H′060
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU registers are updated. The vector address of the TLB miss exception becomes VBR + H'0400. To speed up TLB miss processing, the offset differs from other exceptions.

TLB invalid exception:

- Conditions

  Comparison of TLB addresses shows address match but V = 0.
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H′040

  An exception occurred during write: H′060
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU registers are updated.

TLB protection exception:

- Conditions

  When a hit access violates the TLB protection information (PR bits).
- Types

  Instruction synchronous, re-execution type

- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  An exception occurred during read: H′0A0

  An exception occurred during write: H′0C0
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU registers are updated.

Initial page write exception:

- Conditions

  A hit occurred to the TLB for a data write access, but D = 0.
- Types

  Instruction synchronous, re-execution type
- Save address

  Instruction fetch: An instruction address to be fetched when an exception occurred

  Data access: An instruction address where an exception occurs (a delayed branch instruction address if an instruction is assigned to a delay slot)
- Exception code

  H′080
- Remarks

  The logical address (32 bits) that caused the exception is set in TEA and the MMU registers are updated.

4.4 Exception Processing while DSP Extension Function is Valid

When the DSP extension function is valid (the DSP bit of SR is set to 1), some exception processing acceptance conditions or exception processing may be changed.

4.4.1 Illegal Instruction Exception and Slot Illegal Instruction Exception

In the DSP mode, a DSP extension instruction can be executed. If a DSP extension instruction is executed when the DSP bit of SR is cleared to 0 (in a mode other than the DSP mode), an illegal instruction exception occurs.

In the DSP mode, STC and LDC instructions for the SR register can be executed even in user mode. (Note, however, that only the RC[11:0], DMX, DMY, and RF[1:0] bits in the DSP extension bits can be changed.)

4.4.2 CPU Address Error

In the DSP mode, a part of the space P2 (Uxy area: H′A5000000 to H′A5FFFFFF) can be accessed in user mode and no CPU address error will occur even if the area is accessed.

4.4.3 Exception in Repeat Control Period

If an exception is requested or an exception is accepted during repeat control, the exception may not be accepted correctly or a program execution may not be returned correctly from exception processing that is different from the normal state. These restrictions may occur from repeat detection instruction to repeat end instruction while the repeat counter is 1 or more. In this section, this period is called the repeat control period.

The following shows program examples where the number of instructions in the repeat loop are 4 or more, 3, 2, and 1, respectively. In this section, a repeat detection instruction and its instruction address are described as RptDtct. The first, second, and third instructions following the repeat detection instruction are described as RptDtct1, RptDtct2, and RptDtct3. In addition, [A], [B], [C1], and [C2] in the following examples indicate instructions where a restriction occurs. Table 4.2 summarizes the instruction positions and restriction types.

Table 4.2 Instruction Positions and Restriction Types

Instruction Position	SPC*[1]	Illegal Instruction*[2]	Interrupt, Break*[3]	CPU Address Error*[4]
[A]				
[B]			Retained	
[C1]		Added	Retained	Instruction/data
[C2]	Illegal	Added	Retained	Instruction/data

Notes: 1. A specific address is specified in the SPC if an exception occurs while SR.RC[11:0] ≥2.
2. There are a greater number of instructions that can be illegal instructions while SR.RC[11:0] ≥1.
3. An interrupt, break or DMA address error request is retained while SR.RC[11:0] ≥1.
4. A specific exception code is specified while SR.RC[11:0] ≥1.

- Example 1: Repeat loop consisting of four or greater instructions

```
            LDRS RptStart  ; [A]
            LDRS RptDtct + 4      ; [A]
            SETRC #4       ; [A]
            instr0         ; [A]
RptStart:   instr1         ; [A][Repeat start instruction]
            ……             ; [A]
            ……             ; [A]
RptDtct:    RptDtct        ; [B] A repeat detection
                                 instruction is an
                                 instruction three
                                 instructions before a
                                 repeat end instruction
            RptDtct1       ; [C1]
            RptDtct2       ; [C2]
RptEnd:     RptDtct3       ; [C2][Repeat end instruction]
            InstrNext      ; [A]
```

- Example 2: Repeat loop consisting of three instructions

```
            LDRS      RptDtct + 4    ; [A]
            LDRS      RptDtct + 4    ; [A]
            SETRC #4                 ; [A]
  RptDtct:  RptDtct                  ; [B] A repeat detection
                                           instruction is an
                                           instruction prior to a
                                           repeat start instruction
  RptStart: RptDtct1                 ; [C1][Repeat start instruction]
            RptDtct2                 ; [C2]
  RptEnd:   RptDtct3                 ; [C2][Repeat end instruction]
            InstrNext                ; [A]
```

- Example 3: Repeat loop consisting of two instructions

```
            LDRS      RptDtct + 6    ; [A]
            LDRS      RptDtct + 4    ; [A]
            SETRC #4                 ; [A]
  RptDtct:  RptDtct                  ; [B] A repeat detection
                                           instruction is an
                                           instruction prior to a
                                           repeat start instruction
  RptStart: RptDtct1                 ; [C1][Repeat start instruction]
  RptEnd:   RptDtct2                 ; [C2][Repeat end instruction]
            InstrNext                ; [A]
```

- Example 4: Repeat loop consisting of one instruction

```
            LDRS      RptDtct + 8    ; [A]
            LDRS      RptDtct + 4    ; [A]
            SETRC #4                 ; [A]
RptDtct:    RptDtct                  ; [B] A repeat detection
                                           instruction is an
                                           instruction prior to a
                                           repeat start instruction
RptStart:
RptEnd:     RptDtct1                 ; [C1][Repeat start
                                           instruction]== [Repeat end
                                           instruction]
            InstrNext                ; [A]
```

SPC Saved by Exception in Repeat Control Period: If an exception is accepted in the repeat control period while the repeat counter (RC[11:0]) in the SR register is two or greater, the program counter to be saved may not indicate the value to be returned correctly. To execute the repeat control after returning from an exception processing, the return address must indicate an instruction prior to a repeat detection instruction. Accordingly, if an exception is accepted in repeat control period, an exception other than re-execution type exception by a repeat detection instruction cannot return to the repeat control correctly.

Table 4.3 SPC Value when Re-Execution Type Exception Occurs in Repeat Control (RC[11:0] ≥ 2)

Instruction where Exception Occurs	Number of Instructions in Repeat Loop			
	1	2	3	4 or Greater
RptDtct	RptDtct	RptDtct	RptDtct	RptDtct
RptDtct1	RptDtct1	RptDtct1	RptDtct1	RptDtct1
RptDtct2	—	RptDtct1	RptDtct1	RS-4
RptDtct3	—	—	RptDtct1	RS-2

Note: The following labels are used here.
RptDtct: Repeat detection instruction address
RptDtct1: An instruction address one instruction following the repeat detection instruction
RptDtct2: An instruction address two instruction following the repeat detection instruction
RptDtct3: An instruction address three instruction following the repeat detection instruction
RS: Repeat start instruction address
If a re-execution type exception is accepted at an instruction in the hatched areas above, a return address to be saved in the SPC is incorrect. If RC[11:0] is 1 or 0, a correct return address is saved in the SPC.

Illegal Instruction Exception in Repeat Control Period: If one of the following instructions is executed at the address following RptDtct1, a general illegal instruction exception occurs. For details on an address to be saved in the SPC, refer to the description in section 4.4.3, SPC Saved by Exception in Repeat Control Period.

- Branch instructions
  BRA, BSR, BT, BF, BT/S, BF/S, BSRF, RTS, BRAF, RTE, JSR, JMP, TRAPA
- Repeat control instructions
  SETRC, LDRS, LDRE
- Load instructions for SR, RS, and RE
  LDC Rn, SR, LDC @Rn+, SR, LDC Rn, RE, LDC @Rn+, RE, LDC Rn,RS, LDC @Rn+, Rs

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the three instructions that include a repeat end instruction.

Exception Retained in Repeat Control Period: In the repeat control period, an interrupt or some exception will be retained to prevent an exception acceptance at an instruction where returning from the exception cannot be performed correctly. For details, refer to repeat loop program examples 1 to 4. In the examples, exceptions generated at instructions indicated as [B], [C], [C1], or [C2], the following processing is executed.

- Interrupt, DMA address error

  An exception request is not accepted and retained at instructions [B] and [C]. If an instruction indicates as [A] is executed at the next time, an exception request is accepted.* As shown in program examples 1 to 4, any interrupt or DMA address error cannot be accepted in a repeat loop consisting of four instructions or less.

Note: * An interrupt request or a DMA address error exception request is retained in the interrupt controller (INTC) and the direct memory access controller (DMAC) until the CPU can accept a request.

- User break before instruction execution

  A user break before instruction execution is accepted at instruction [B], and an address of instruction [B] is saved in the SPC. This exception cannot be accepted at instruction [C] but the exception request is retained until an instruction [A] or [B] is executed at the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.

- User break after instruction execution

  A user break after instruction execution cannot be accepted at instructions [B] and [C] but the exception request is retained until an instruction [A] or [B] is executed at the next time. Then, the exception request is accepted before an instruction [A] or [B] is executed. In this case, an address of instruction [A] or [B] is saved in the SPC.

Table 4.4 Exception Acceptance in Repeat Loop

Exception Type	Instruction [B]	Instruction [C]
Interrupt	Not accepted	Not accepted
DMA address error	Not accepted	Not accepted
User break before instruction execution	Accepted	Not accepted
User break after instruction execution	Not accepted	Not accepted

CPU Address Error in Repeat Control Period: If a CPU address error occurs in the repeat control period, the exception is accepted but an exception code (H'070) indicating the repeat loop period is specified in the EXPEVT. If a CPU address error occurs in instructions following a repeat detection instruction to repeat end instruction, an exception code for instruction access or data access is specified in the EXPEVT.

The SPC is saved according to the description in section 4.4.3, SPC Saved by Exception in Repeat Control Period.

After the CPU address error exception processing, the repeat control cannot be returned correctly. To execute a repeat loop correctly, care must be taken not to generate a CPU address error in the repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the four instructions that include a repeat end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

Table 4.5 Instruction Where a Specific Exception Occurs when Memory Access Exception Occurs in Repeat Control (SR.RC[11:0] ≥ 1)

Instruction where Exception Occurs	Number of Instructions in Repeat Loop			
	1	**2**	**3**	**4 or Greater**
RptDtct				
RptDtct1	Instruction/data access	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct2	—	Instruction/data access	Instruction/data access	Instruction/data access
RptDtct3	—	—	Instruction/data access	Instruction/data access

Note: The following labels are used here.
RptDtct: Repeat detection instruction
RptDtct1: An instruction of one instruction following the repeat detection instruction
RptDtct2: An instruction of two instruction following the repeat detection instruction
RptDtct3: An instruction of three instruction following the repeat detection instruction

MMU Exception in Repeat Control Period: If an MMU exception occurs in the repeat control period, a specific exception code is generated as well as a CPU address error. For a TLB miss exception, TLB invalid exception, and initial page write exception, an exception code (H′070) is specified in the EXPEVT. For a TLB protection exception, an exception code (H′0D0) is specified in the EXPEVT. In a TLB miss exception, vector offset is specified as H′00000100.

An instruction where an exception occurs and the SPC value to be saved are the same as those for the CPU address error.

After this exception processing, the repeat control cannot be returned correctly. To execute a repeat loop correctly, care must be taken not to generate an MMU related exception in the repeat control period.

Note: In a repeat loop consisting of one to three instructions, some restrictions apply to repeat detection instructions and all the remaining instructions. In a repeat loop consisting of four or more instructions, restrictions apply to only the four instructions that include a repeat end instruction. The restriction occurs when SR.RC[11:0] ≥ 1.

4.5 Usage Notes

1. An instruction assigned at a delay slot of the RTE instruction is executed after the contents of the SSR is restored into the SR. An acceptance of an exception related to instruction access is determined according to the SR before restore. An acceptance of other exceptions is determined by processing mode of the SR after restore, and BL bit value. A processing-completion type exception is accepted before an instruction at the RTE branch destination address is executed. However, note that the correct operation cannot be guaranteed if a re-execution type exception occurs.
2. In an instruction assigned at a delay slot of the RTE instruction, a user break cannot be accepted.
3. If the MD and BL bits of the SR register are changed by the LDC instruction, an exception is accepted according to the changed SR value from the next instruction.* A processing-completion type exception is accepted after the next instruction is executed. An interrupt and DMA address error in re-execution type exceptions are accepted before the next instruction is executed.

Note: * If an LDC instruction is executed for the SR, the following instructions are re-fetched and an instruction fetch exception is accepted according to the modified SR value.

Section 5 Memory Management Unit (MMU)

This LSI has an on-chip memory management unit (MMU) that supports a virtual memory system. The on-chip translation look-aside buffer (TLB) caches information for user-created address translation tables located in external memory. It enables high-speed translation of virtual addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 kbyte or 4 kbytes). The access rights to virtual address space can be set for each of the privileged and user modes to provide memory protection.

5.1 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown in figure 5.1, if a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory. However, if the process increases in size to the extent that it no longer fits into physical memory, it becomes necessary to partition the process and to map those parts requiring execution onto memory as occasion demands (figure 5.1 (1)). Having the process itself consider this mapping onto physical memory would impose a large burden on the process. To lighten this burden, the idea of virtual memory was born as a means of performing en bloc mapping onto physical memory (figure 5.1 (2)). In a virtual memory system, substantially more virtual memory than physical memory is provided, and the process is mapped onto this virtual memory. Thus a process only has to consider operation in virtual memory. Mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally controlled by the operating system, switching physical memory to allow the virtual memory required by a process to be mapped onto physical memory in a smooth fashion. Switching of physical memory is performed via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in a time-sharing system (TSS) in which a number of processes are running simultaneously (figure 5.1 (3)). If processes running in a TSS had to take mapping onto virtual memory into consideration while running, it would not be possible to increase efficiency. Virtual memory is thus used to reduce this load on the individual processes and so improve efficiency (figure 5.1 (4)). In the virtual memory system, virtual memory is allocated to each process. The task of the MMU is to perform efficient mapping of these virtual memory areas onto physical memory. It also has a memory protection feature that prevents one 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 occur that the relevant translation information is not recorded in the MMU, with the result that one process may inadvertently access the virtual memory allocated to another process. In this case, 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 also be implemented by software alone, the need for translation to be performed by software each time a process accesses physical memory would result in poor efficiency. For this reason, a buffer for address translation (translation look-aside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails, that is, if an exception is generated, switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

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

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

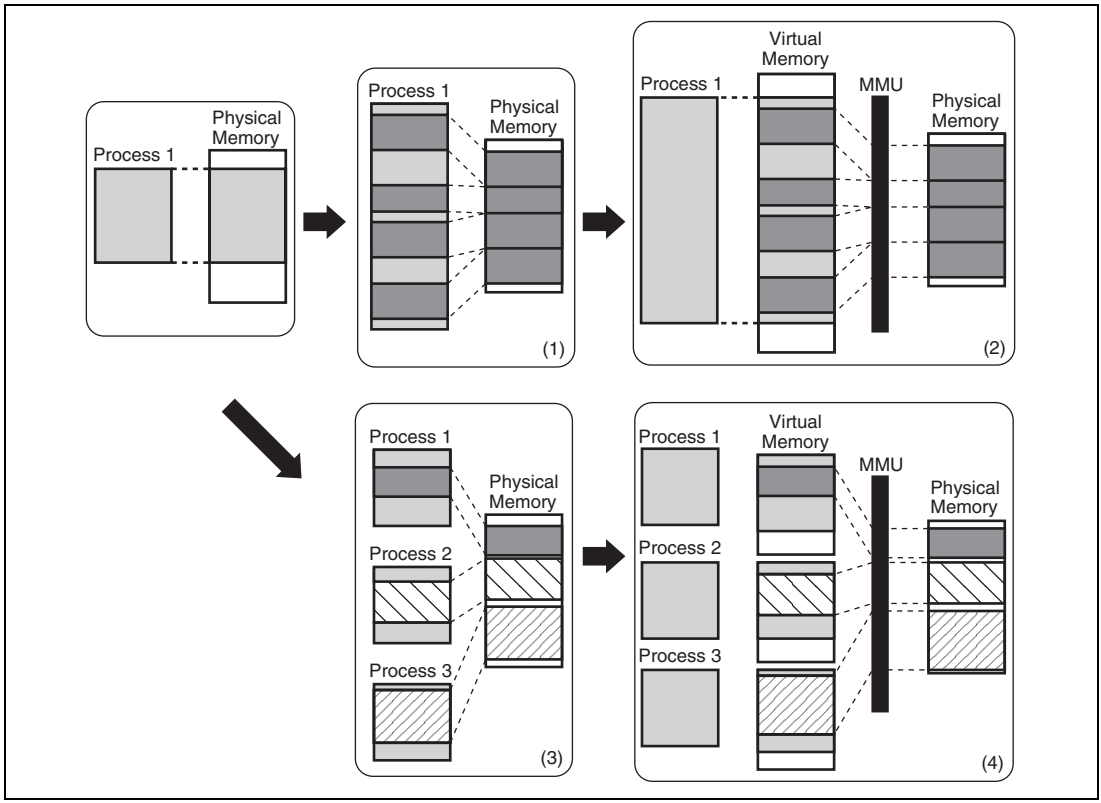


Figure 5.1 MMU Functions

5.1.1 MMU of This LSI

Virtual Address Space: This LSI supports a 32-bit virtual address space that enables access to a 4-Gbyte address space. As shown in figures 5.2 and 5.3, the virtual address space is divided into several areas. In privileged mode, a 4-Gbyte space comprising areas P0 to P4 are accessible. In user mode, a 2-Gbyte space of U0 area is accessible, and a 16-Mbyte space of Uxy area is also accessible if the DSP bit of the SR register is set to 1. Access to any area (excluding the U0 area and Uxy area) in user mode will result in an address error.

If the MMU is enabled by setting the AT bit of the MMUCR register to 1, P0, P3, and U0 areas can be used as any physical address area in 1- or 4-kbyte page units. By using an 8-bit address space identifier, P0, P2, and U0 areas can be increased to up to 256 areas. Mapping from virtual address to 29-bit physical address can be achieved by the TLB.

1. P0, P3, and U0 Areas

   The P0, P3, and U0 areas can be address translated by the TLB and can be accessed through the cache. If the MMU is enabled, these areas can be mapped to any physical address space in 1- or 4-kbyte page units via the TLB. If the CE bit in the cache control register (CCR1) is set to 1 and if the corresponding cache enable bit (C bit) of the TLB entry is set to 1, access via the cache is enabled. If the MMU is disabled, replacing the upper three bits of an address in these areas with 0s creates the address in the corresponding physical address space. If the CE bit of the CCR1 register is set to 1, access via the cache is enabled. When the cache is used, either the copy-back or write-through mode is selected for write access via the WT bit in CCR1.

   If these areas are mapped to the on-chip module control register area or on-chip memory area in area 1 in the physical address space via the TLB, the C bit of the corresponding page must be cleared to 0.

2. P1 Area

   The P1 area can be accessed via the cache and cannot be address-translated by the TLB. Whether the MMU is enabled or not, replacing the upper three bits of an address in these areas with 0s creates the address in the corresponding physical address space. Use of the cache is determined by the CE bit in the cache control register (CCR1). When the cache is used, either the copy-back or write-through mode is selected for write access by the CB bit in the CCR1 register.

3. P2 Area

   The P2 area cannot be accessed via the cache and cannot be address-translated by the TLB. Whether the MMU is enabled or not, replacing the upper three bits of an address in this area with 0s creates the address in the corresponding physical address space.

4. P4 Area

   The P4 area is mapped to the on-chip I/O of this LSI. This area cannot be accessed via the cache and cannot be address-translated by the TLB. Figure 5.4 shows the configuration of the P4 area.

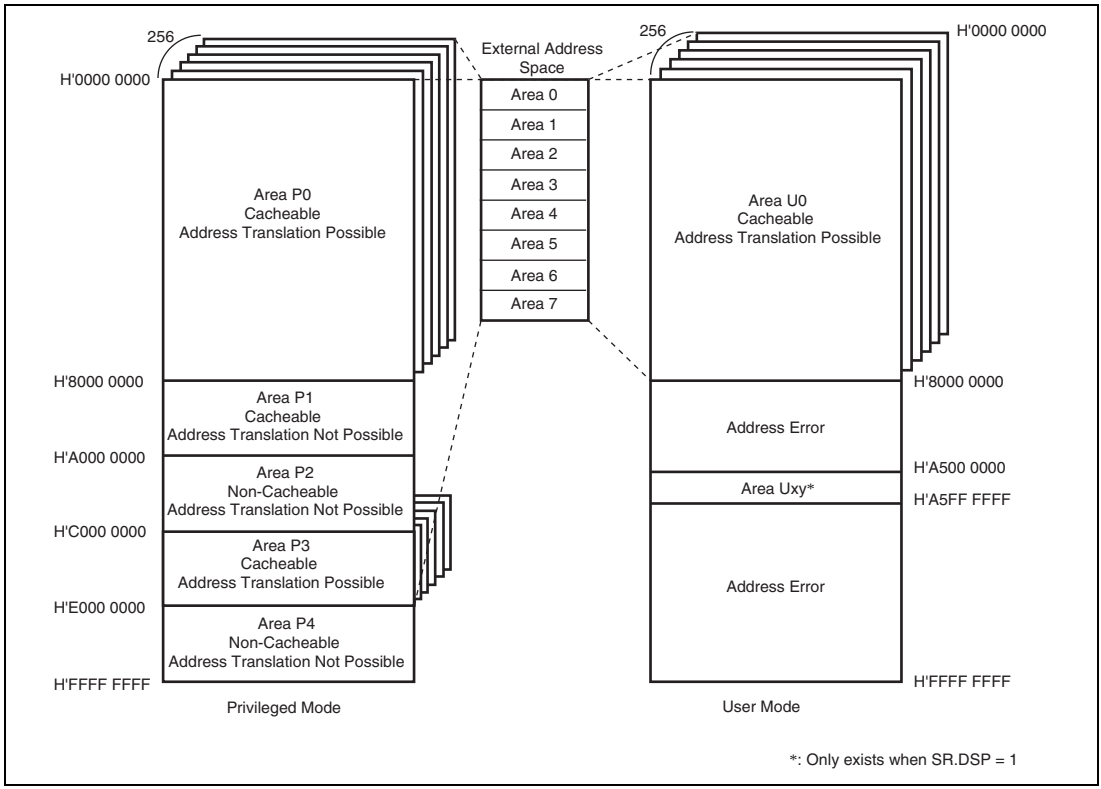


Figure 5.2 Virtual Address Space (MMUCR.AT = 1)

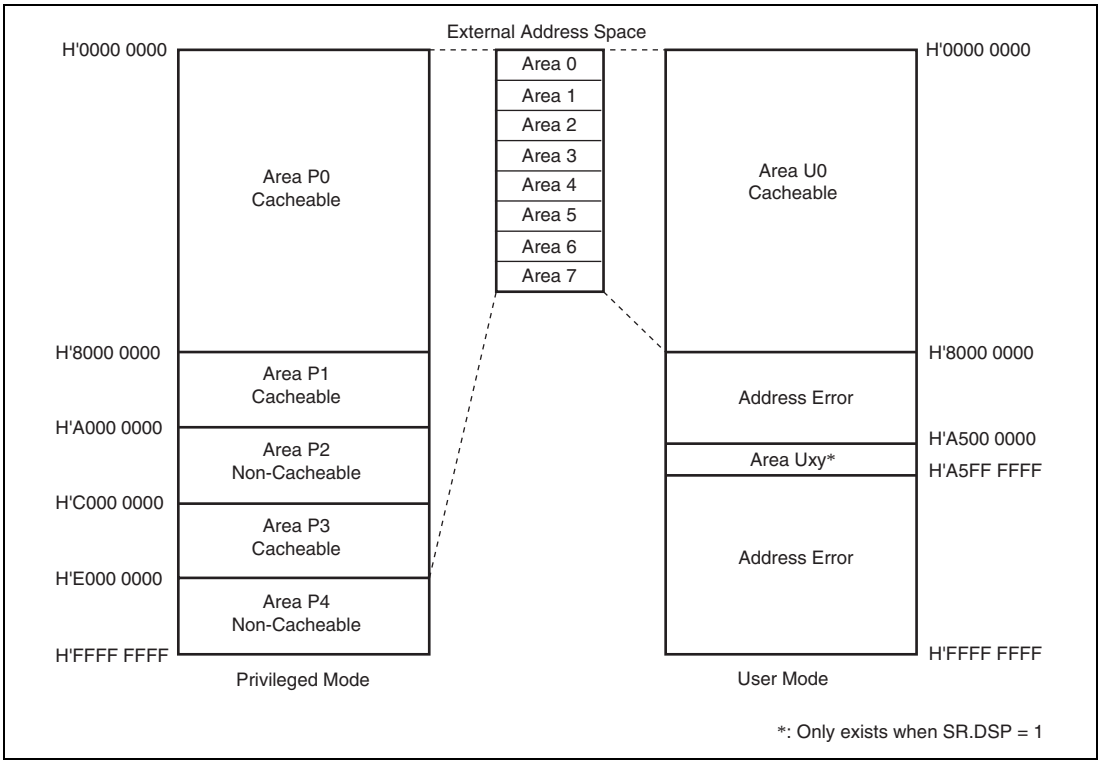


Figure 5.3 Virtual Address Space (MMUCR.AT = 0)

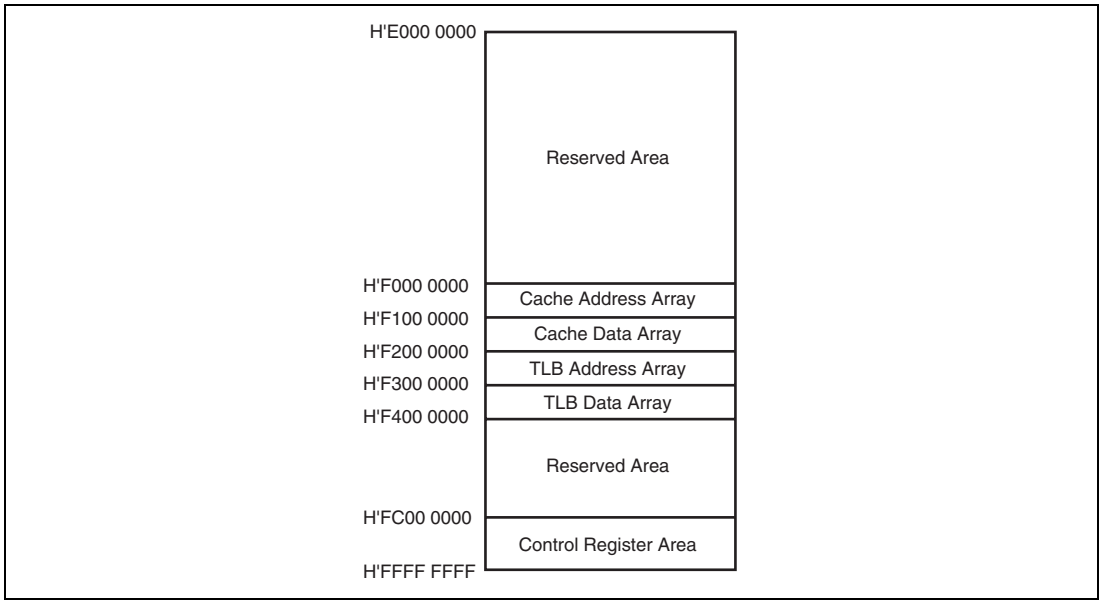


Figure 5.4 P4 Area

The area from H'F000 0000 to H'F0FF FFFF is for direct access to the cache address array. For more information, see section 6.4, Memory-Mapped Cache.

The area from H'F100 0000 to H'F1FF FFFF is for direct access to the cache data array. For more information, see section 6.4, Memory-Mapped Cache.

The area from H'F200 0000 to H'F2FF FFFF is for direct access to the TLB address array. For more information, see section 5.6, Memory-Mapped TLB.

The area from H'F300 0000 to H'F3FF FFFF is for direct access to the TLB data array. For more information, see section 5.6, Memory-Mapped TLB.

The area from H'FC00 0000 to H'FFFF FFFF is reserved for registers of the on-chip peripheral modules. For more information, see section 24, List of Registers.

5. Uxy Area

   The Uxy area is mapped to the on-chip memory of this LSI. This area is made usable in user mode when the DSP bit in the SR register is set to 1. In user mode, accessing this area when the DSP bit is 0 will result in an address error. This area cannot be accessed via the cache and cannot be address-translated by the TLB. For more information on the Uxy area, see section 7, X/Y Memory.

Physical Address Space: This LSI supports a 29-bit physical address space. As shown in figure 5.5, the physical address space is divided into eight areas. Area 1 is mapped to the on-chip module control register area and on-chip memory area. Area 7 is reserved.

For details on physical address space, refer to section 12, Bus State Controller (BSC).

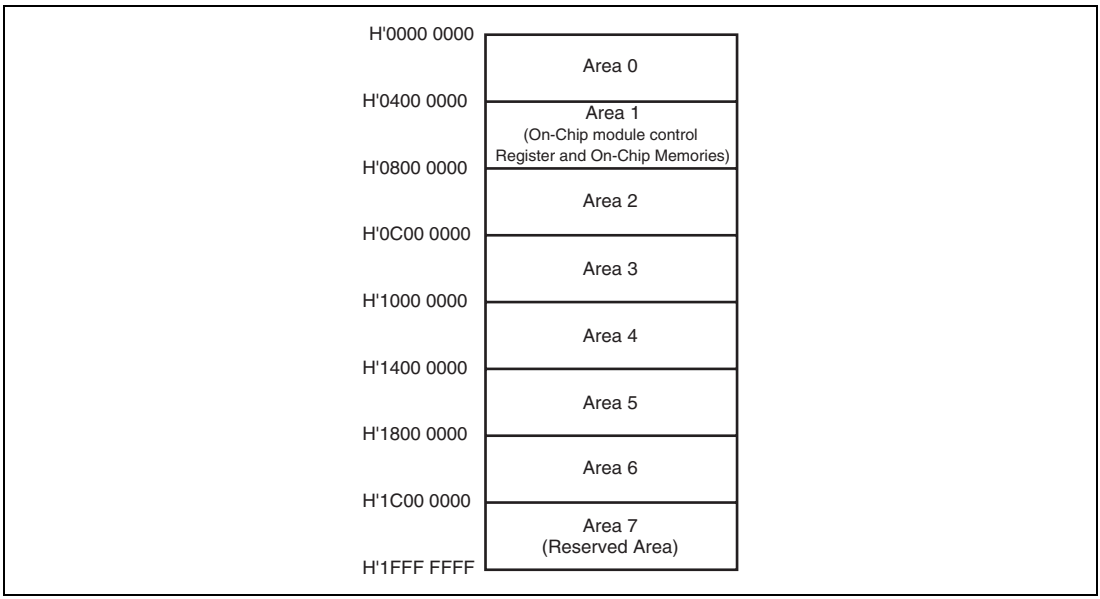


Figure 5.5 External Memory Space

Address Transition: When the MMU is enabled, the virtual address space is divided into units called pages. Physical addresses are translated in page units. Address translation tables in external memory hold information such as the physical address that corresponds to the virtual address and memory protection codes. When an access to area P1 or P2 occurs, there is no TLB access and the physical address is defined uniquely by hardware. If it belongs to area P0, P3 or U0, the TLB is searched by virtual address and, if that virtual address is registered in the TLB, the access hits the TLB. The corresponding physical address and the page control information are read from the TLB and the physical address is determined.

If the virtual address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that

results in a physical address space of H'20000000 to H'FFFFFFFF should not be registered in the TLB.

When the MMU is disabled, masking the upper three bits of the virtual address to 0s creates the address in the corresponding physical address space. Since this LSI supports 29-bit address space as physical address space, the upper three bits of the virtual address are ignored as shadow areas. For details, refer to section 12, Bus State Controller (BSC). For example, address H'00001000 in the P0 area, address H'80001000 in the P1 area, address H'A0001000 in the P2 area, and address H'C0001000 in the P3 area are all mapped to the same physical memory. If these addresses are accessed while the cache is enabled, the upper three bits are always cleared to 0 to guarantee the continuity of addresses stored in the address array of the cache.

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

In terms of operation, the only difference between single virtual memory mode and multiple virtual memory mode is in the TLB address comparison method (see section 5.3.3, TLB Address Comparison).

Address Space Identifier (ASID): In multiple virtual memory mode, the address space identifier (ASID) is used to differentiate between processes running in parallel and sharing virtual address space. The ASID is eight bits in length and can be set by software setting of the ASID of the currently running process in page table entry register high (PTEH) within the MMU. When the process is switched using the ASID, the TLB does not have to be purged.

In single virtual memory mode, the ASID is used to provide memory protection for processes running simultaneously and using the virtual address space exclusively (see section 5.3.3, TLB Address Comparison).

5.2 Register Descriptions

There are four registers for MMU processing. These are all peripheral module registers, so they are located in address space area P4 and can only be accessed from privileged mode by specifying the address.

The MMU has the following registers. Refer the section 24, List of Registers, for the addresses and access size for these registers.

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

5.2.1 Page Table Entry Register High (PTEH)

The page table entry register high (PTEH) register residing at address H'FFFFFFF0, which consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the virtual address at which the exception is generated in case of an MMU exception or address error exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtual address, but in this case the upper 22 bits of the virtual address are set. The VPN can also be modified by software. As the ASID, software sets the number of the currently executing process. The VPN and ASID are recorded in the TLB by the LDTLB instruction.

A program that modifies the ASID in PTEH should be allocated in the P1 or P2 areas.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	—	R/W	Number of Virtual Page
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

5.2.2 Page Table Entry Register Low (PTEL)

The page table entry register low (PTEL) register residing at address H'FFFFFFF4, and used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified in response to a software command.

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	PPN	—	R	Number of Physical Page
9	—	0	R/W	Page management information
8	V	—		For more details, see section 5.3, TLB Functions
7	—	0		
6, 5	PR	—		
4	SZ	—		
3	C	—		
2	D	—		
1	SH	—		
0	—	0		

5.2.3 Translation Table Base Register (TTB)

The translation table base register (TTB) residing at address H'FFFFFFF8, which points to the base address of the current page table. The hardware does not set any value in TTB automatically. TTB is available to software for general purposes. The initial value is undefined.

5.2.4 MMU Control Register (MMUCR)

The MMU control register (MMUCR) residing at address H'FFFFFFE0, which makes the MMU settings described in figure 5.3. Any program that modifies MMUCR should reside in the P1 or P2 area.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SV	0	R/W	Single virtual memory mode 0: Multiple virtual memory mode 1: Single virtual memory mode
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	RC	All 0	R/W	Random counter A 2-bit random counter that is automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all of TLB entry way corresponding to the virtual address at which the exception occurred are checked. If all ways are valid, 1 is added to RC; if there is one or more invalid way, they are set by priority from way 0, in the order way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TF	0	R/W	TLB flush Write 1 to flush the TLB (clear all valid bits of the TLB to 0). When they are read, 0 is always returned.
1	IX	0	R/W	Index mode 0: VPN bits 16 to 12 are used as the TLB index number. 1: The value obtained by EX-ORing ASID bits 4 to 0 in PTEH and VPN bits 16 to 12 is used as the TLB index number.

Bit	Bit Name	Initial Value	R/W	Description
0	AT	0	R/W	Address translation Enables/disables the MMU. 0: MMU disabled 1: MMU enabled

5.3 TLB Functions

5.3.1 Configuration of the TLB

The TLB caches address translation table information located in the external memory. The address translation table stores the logical page number and the corresponding physical number, the address space identifier, and the control information for the page, which is the unit of address translation. Figure 5.6 shows the overall TLB configuration. The TLB is 4-way set associative with 128 entries. There are 32 entries for each way. Figure 5.7 shows the configuration of virtual addresses and TLB entries.

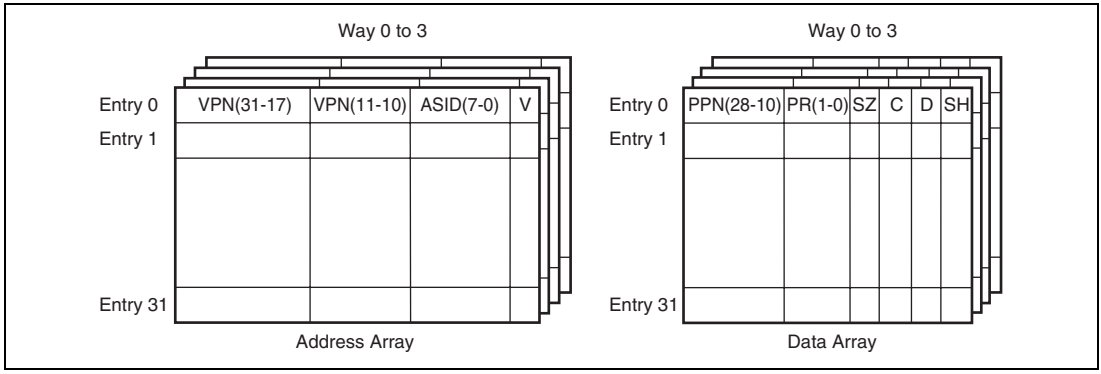
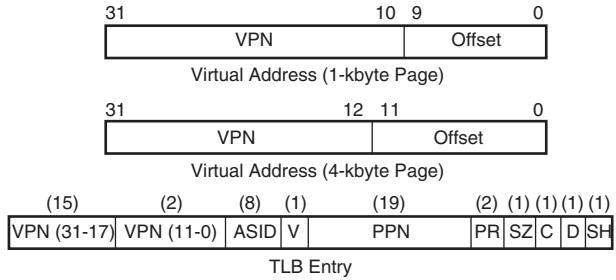


Figure 5.6 Overall Configuration of the TLB



Legend

VPN: Virtual page number
Upper 22 bits of virtual address for a 1-kbyte page, or upper 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16 to 12 are used as the index number, they are not stored in the TLB entry. Attention must be paid to the synonym problem (see section 5.4.4, Avoiding Synonym Problems).

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

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

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

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

PPN: Physical page number
Upper 22 bits of physical address. PPN bits 11 to10 are not used in case of a 4-kbyte page.

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

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

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

Figure 5.7 Virtual Address and TLB Structure

5.3.2 TLB Indexing

The TLB uses a 4-way set associative scheme, so entries must be selected by index. VPN bits 16 to 12 are used as the index number regardless of the page size. The index number can be generated in two different ways depending on the setting of the IX bit in MMUCR.

1. When IX = 0, VPN bits 16 to 12 alone are used as the index number
2. When IX = 1, VPN bits 16 to 12 are EX-ORed with ASID bits 4 to 0 to generate a 5-bit index number

The first method is used to prevent lowered TLB efficiency that results when multiple processes run simultaneously in the same virtual address space (multiple virtual memory) and a specific entry is selected by indexing of each process. In single virtual memory mode (MMUCR.SV = 1), IX bit should be set to 0. Figures 5.8 and 5.9 show the indexing schemes.

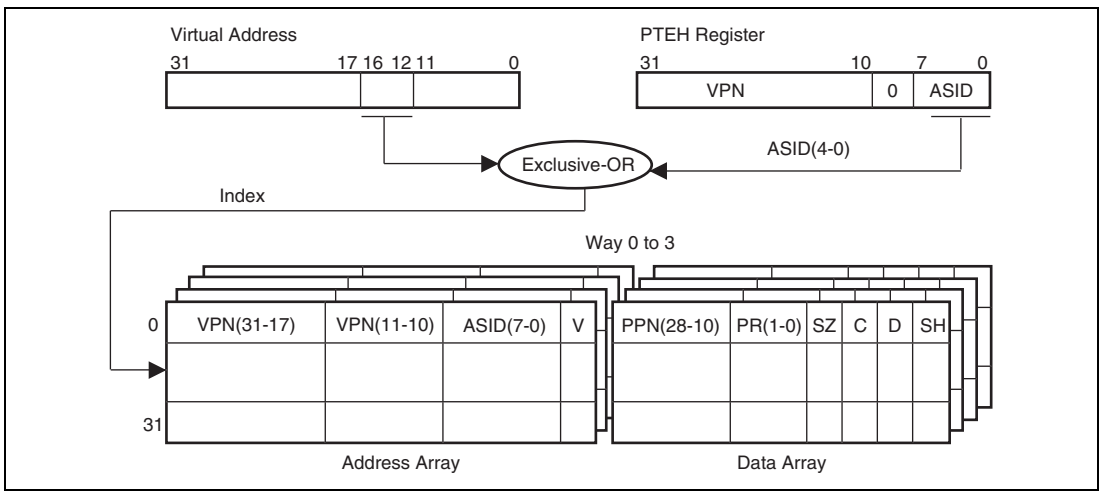


Figure 5.8 TLB Indexing (IX = 1)

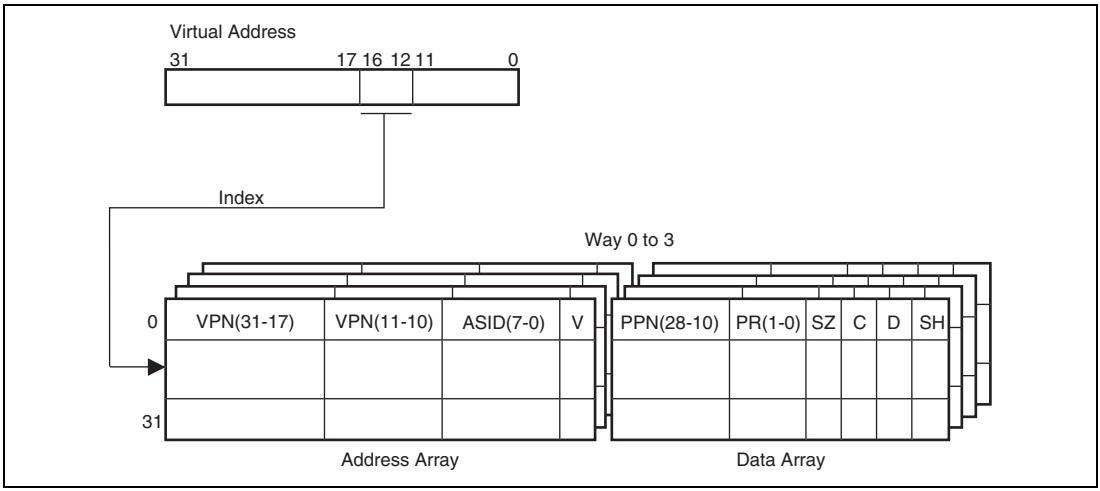


Figure 5.9 TLB Indexing (IX = 0)

5.3.3 TLB Address Comparison

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

It is necessary to have software ensure that TLB hits do not occur simultaneously in more than one way, as hardware operation is not guaranteed if this occurs. An example of setting which causes TLB hits to occur simultaneously in more than one way is described below. It is necessary to ensure that this kind of setting is not made by software.

1. If there are two identical TLB entries with the same VPN and a setting is made such that a TLB hit is made only by a process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the non-shared state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simultaneous TLB hits in both these ways.
2. If several entries which have different ASID with the same VPN are registered in single virtual memory mode, there is the possibility of simultaneous TLB hits in more than one way when accessing the corresponding page in privileged mode. Several entries with the same VPN must not be registered in single virtual memory mode.
3. There is the possibility of simultaneous TLB hits in more than one way. These hits may occur depending on the contents of ASID in PTEH when a page to which SH is set 1 is registered in the TLB in index mode (MMUCR.IX = 1). Therefore a page to which SH is set 1 must not be

registered in index mode. When memory is shared by several processings, different pages must be registered in each ASID.

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

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

The sharing information (SH) determines whether the PTEH.ASID and the ASID in the TLB entry are compared. ASIDs are compared when there is no sharing between processes (SH = 0) but not when there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is engaged (SR.MD = 1), all process resources can be accessed. This means that ASIDs are not compared when single virtual memory is supported and privileged mode is engaged. The objects of address comparison are shown in figure 5.10.

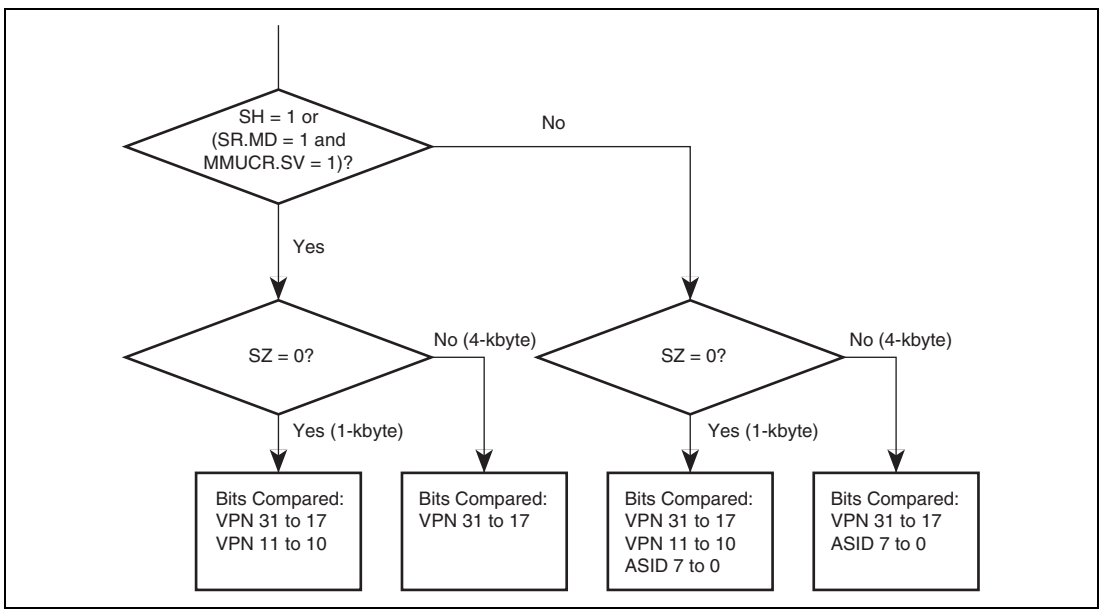


Figure 5.10 Objects of Address Comparison

5.3.4 Page Management Information

In addition to the SH and SZ bits, the page management information of TLB entries also includes D, C, and PR bits.

The D bit of a TLB entry indicates whether the page is dirty (i.e., has been written to). If the D bit is 0, an attempt to write to the page results in an initial page write exception. For physical page swapping between secondary memory and main memory, for example, pages are controlled so that a dirty page is paged out of main memory only after that page is written back to secondary memory. To record that there has been a write to a given page in the address translation table in memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or non-cacheable area of memory. When the control registers and on-chip memory in area 1 are mapped, set the C bit to 0. The PR field specifies the access rights for the page in privileged and user modes and is used to protect memory. Attempts at non-permitted accesses result in TLB protection violation exceptions.

Access states designated by the D, C, and PR bits are shown in table 5.1.

Table 5.1 Access States Designated by D, C, and PR Bits

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

5.4 MMU Functions

5.4.1 MMU Hardware Management

There are two kinds of MMU hardware management as follows.

1. The MMU decodes the virtual address accessed by a process and performs address translation by controlling the TLB in accordance with the MMUCR settings.
2. In address translation, the MMU receives page management information from the TLB, and determines the MMU exception and whether the cache is to be accessed (using the C bit). For details of the determination method and the hardware processing, see section 5.5, MMU Exceptions.

5.4.2 MMU Software Management

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

1. MMU register setting

   MMUCR setting, in particular, should be performed in areas P1 and P2 for which address translation is not performed. Also, since SV and IX bit changes constitute address translation system changes, in this case, TLB flushing should be performed by simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generated in the MMU disabled state with the AT bit cleared to 0, use in the disabled state must be avoided with software that does not use the MMU.
2. TLB entry recording, deletion, and reading

   TLB entry recording can be done in two ways by using the LDTLB instruction, or by writing directly to the memory-mapped TLB. For TLB entry deletion and reading, the memory allocation TLB can be accessed. See section 5.4.3, MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 5.6, Memory-Mapped TLB, for details of the memory-mapped TLB.
3. MMU exception processing

   When an MMU exception is generated, it is handled on the basis of information set from the hardware side. See section 5.5, MMU Exceptions, for details.

When single virtual memory mode is used, it is possible to create a state in which physical memory access is enabled in the privileged mode only by clearing the share status bit (SH) to 0 to specify recording of all TLB entries. This strengthens inter-process memory protection, and enables special access levels to be created in the privileged mode only.

Recording a 1- or 4- kbyte page TLB entry may result in a synonym problem. See section 5.4.4, Avoiding Synonym Problems.

5.4.3 MMU Instruction (LDTLB)

The load TLB instruction (LDTLB) is used to record TLB entries. When the IX bit in MMUCR is 0, the LDTLB instruction changes the TLB entry in the way specified by the RC bit in MMUCR to the value specified by PTEH and PTEL, using VPN bits 16 to 12 specified in PTEH as the index number. When the IX bit in MMUCR is 1, the EX-OR of VPN bits 16 to 12 specified in PTEH and ASID bits 4 to 0 in PTEH are used as the index number.

Figure 5.11 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the virtual address that caused the exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for each exception according to the rules (see section 5.2.4, MMU Control Register (MMUCR)). Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU exception processing routine, TLB entry recording is possible. Any TLB entry can be updated by software rewriting of PTEH and the RC bits in MMUCR.

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

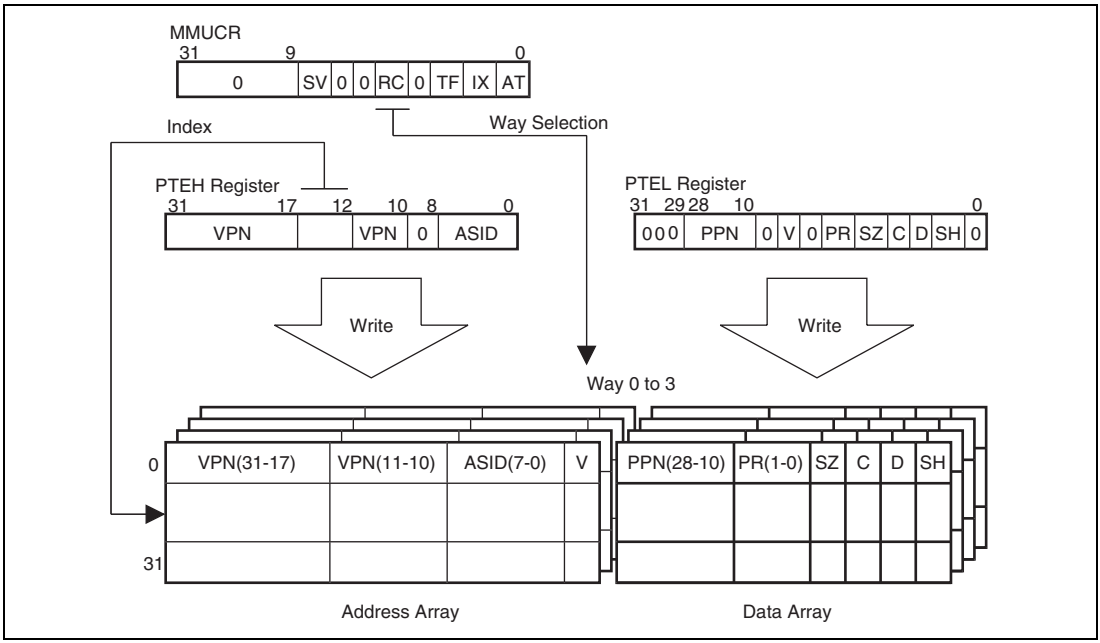


Figure 5.11 Operation of LDTLB Instruction

5.4.4 Avoiding Synonym Problems

When a 1- or 4-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a number of virtual addresses are mapped onto a single physical address, the same physical address data will be recorded in a number of cache entries, and it will not be possible to guarantee data congruity. The reason that this problem occurs is explained below with reference to figure 5.12. The relationship between bit n of the virtual address and cache size is shown in the following table.

Cache Size	Bit n of Virtual Address
16 kbytes	11
32 kbytes	12

To achieve high-speed operation of this LSI's cache, an index number is created using virtual address [n:4]. When a 1-kbyte page is used, virtual address [n:10] is subject to address translation and when a 4-kbyte page is used, a virtual address [n:12] is subject to address translation. Therefore, the physical address [n:10] may not be the same as the virtual address [n:10].

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

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

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

Virtual address 1 is recorded in cache entry H'000, and virtual address 2 in cache entry H'0C0. Since two virtual addresses are recorded in different cache entries despite the fact that the physical addresses are the same, memory inconsistency will occur as soon as a write is performed to either virtual address.

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

1. When address translation information whereby a number of 1-kbyte page TLB entries are translated into the same physical address is recorded in the TLB, ensure that the VPN [n:10] are the same.
2. When address translation information whereby a number of 4-kbyte page TLB entries are translated into the same physical address is recorded in the TLB, ensure that the VPN [n:12] is the same.
3. Do not use the same physical addresses for address translation information of different page sizes.

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 SuperH RISC engine family expansion, ensure that the VPN bits 20 to 10 are the same.

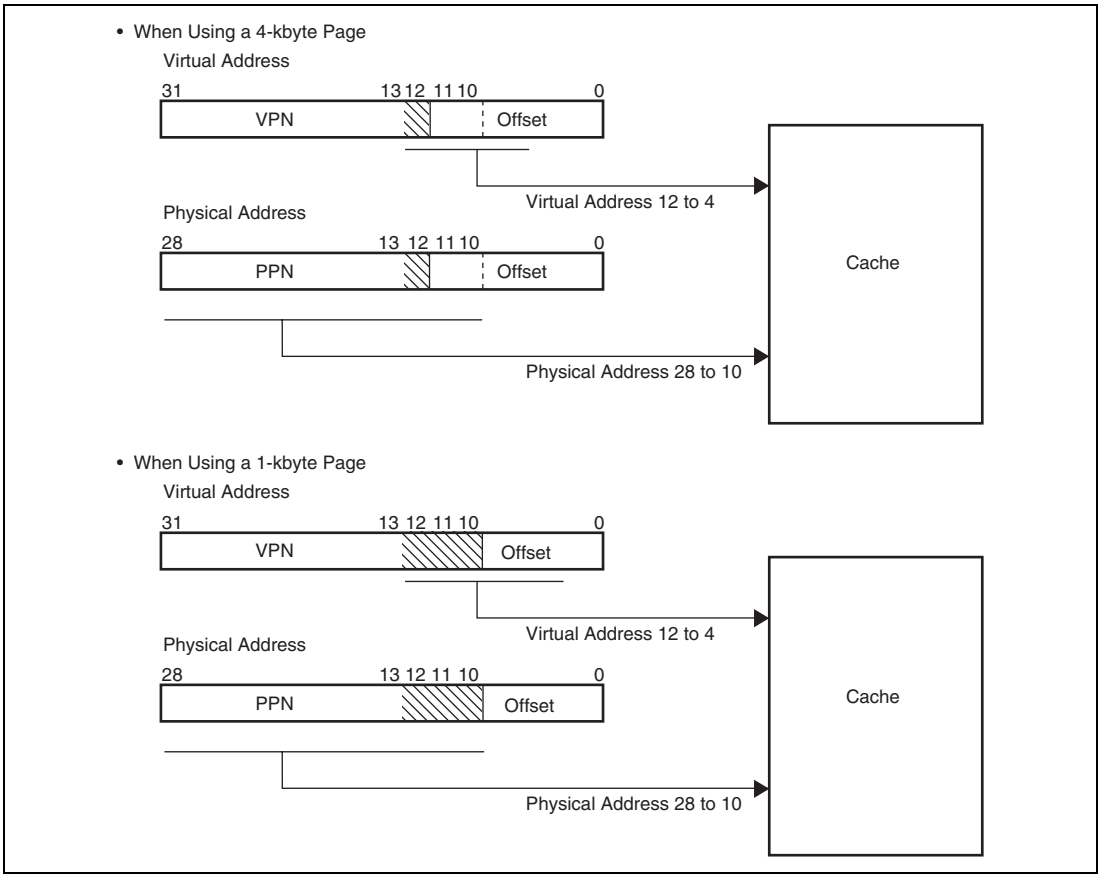


Figure 5.12 Synonym Problem (32-kbyte Cache)

5.5 MMU Exceptions

When the address translation unit of the MMU is enabled, occurrence of the MMU exception is checked following the CPU address error check. There are four MMU exceptions: TLB miss, TLB protection violation, TLB invalid, and initial page write, and these MMU exceptions are checked in this order.

5.5.1 TLB Miss Exception

A TLB miss results when the virtual address and the address array of the selected TLB entry are compared and no match is found. TLB miss exception processing includes both hardware and software operations.

Hardware Operations: In a TLB miss, this hardware executes a set of prescribed operations, as follows:

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

Software (TLB Miss Handler) Operations: The software searches the page tables in external memory and allocates the required page table entry. Upon retrieving the required page table entry, software must execute the following operations:

1. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the address translation table in the external memory into the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the return from exception handler (RTE) instruction to terminate the handler routine and return to the instruction stream.

5.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the virtual address and the address array of the selected TLB entry are compared and a valid entry is found to match, but the type of access is not permitted by the access rights specified in the PR field. TLB protection violation exception processing includes both hardware and software operations.

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

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

Software (TLB Protection Violation Handler) Operations: Software resolves the TLB protection violation and issues the RTE (return from exception handler) instruction to terminate the handler and return to the instruction stream. Issue the RTE instruction after issuing two instructions from the LDTLB instruction.

5.5.3 TLB Invalid Exception

A TLB invalid exception results when the virtual address is compared to a selected TLB entry address array and a match is found but the entry is not valid (the V bit is 0). TLB invalid exception processing includes both hardware and software operations.

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

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

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

1. Write the values of the physical page number (PPN) field and the values of the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

5.5.4 Initial Page Write Exception

An initial page write exception results in a write access when the virtual address and the address array of the selected TLB entry are compared and a valid entry with the appropriate access rights is found to match, but the D (dirty) bit of the entry is 0 (the page has not been written to). Initial page write exception processing includes both hardware and software operations.

Hardware Operations: In an initial page write exception, this hardware executes a set of prescribed operations, as follows:

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

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

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

5.5.5 MMU Exception in Repeat Loop

If a CPU address error or MMU exception occurs in a specific instruction in the repeat loop, the SPC may indicate an illegal address or the repeat loop cannot be reexecuted correctly even if the SPC is correct. Accordingly, if a CPU address error or MMU exception occurs in a specific instruction in the repeat loop, this LSI generates a specific exception code to set the EXPEVT to H'070 for a TLB miss exception, TLB invalid exception, initial page write exception, and CPU address error and to H'0D0 for a TLB protection violation exception. In addition, a vector offset for TLB miss exception is H'100. For details, refer to section 4.4.3,Exception in Repeat Control Period.

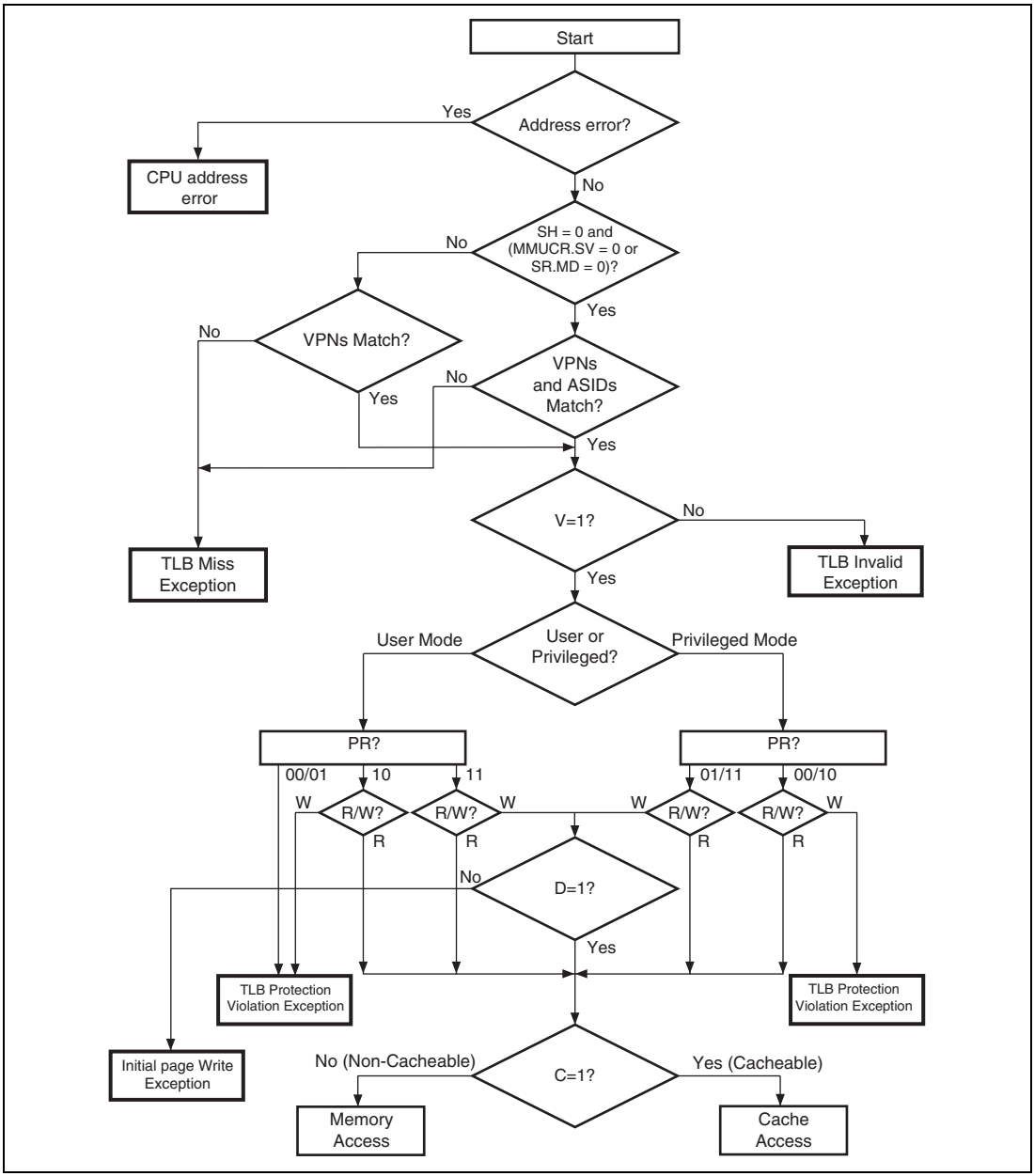


Figure 5.13 MMU Exception Generation Flowchart

5.6 Memory-Mapped TLB

In order for TLB operations to be managed by software, TLB contents can be read or written to in the privileged mode using the MOV instruction. The TLB is assigned to the P4 area in the virtual address space. The TLB address array (VPN, V bit, and ASID) is assigned to H'F2000000 to H'F2FFFFFF, and the data array (PPN, PR, SZ, C, D, and SH bits) to H'F3000000 to H'F3FFFFFF. The V bit in the address array can also be accessed from the data array. Only longword access is possible for both the address array and the data array. However, the instruction data cannot be fetched from both arrays.

5.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address array, the 32-bit address field (for read/write operations) and 32-bit data field (for write operations) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the VPN, V bit and ASID to be written to the address array (figure 5.14 (1)).

In the address field, specify the entry address for selecting the entry (bits 16 to 12), W for selecting the way (bits 9 to 8) and H'F2 to indicate address array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

The following two operations can be used on the address array:

1. Address array read

   VPN, V, and ASID are read from the TLB entry corresponding to the entry address and way set in the address field.
2. TLB address array write

   The data specified in the data field are written to the TLB entry corresponding to the entry address and way set in the address field.

5.6.2 Data Array

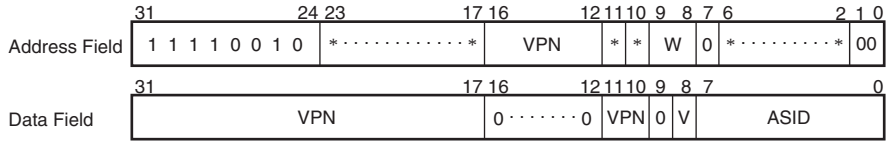
The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. The address section specifies information for selecting the entry to be accessed; the data section specifies the longword data to be written to the data array (figure 5.14 (2)).

In the address section, specify the entry address for selecting the entry (bits 16 to 12), W for selecting the way (bits 9 to 8), and H'F3 to indicate data array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

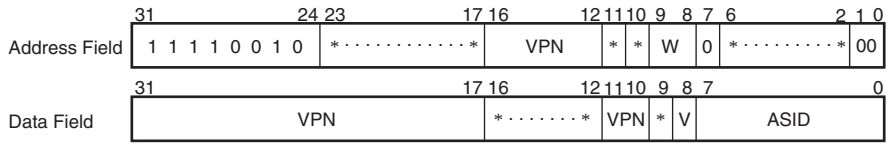
Both reading and writing use the longword of the data array specified by the entry address and way number. The access size of the data array is fixed at longword.

(1) TLB Address Array Access

• Read Access



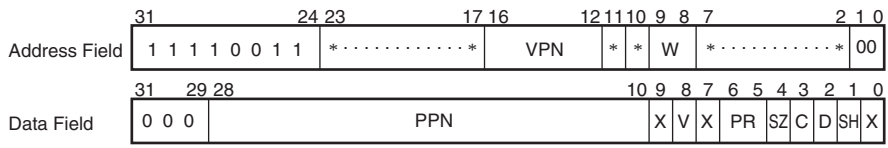
• Write Access



VPN: Virtual Page Number
V: Valid Bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
ASID: Address Space Identifier
*: Don't Care Bit

(2) TLB Data Array Access

• Read/Write Access



PPN: Physical Page Number
PR: Protection Key Field
C: Cacheable Bit
SH: Share Status Bit
VPN: Virtual Page Number
X: 0 for Read, Don't Care Bit for Write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
V: Valid Bit
SZ: Page-Size Bit
D: Dirty bit
*: Don't Care Bit

Figure 5.14 Specifying Address and Data for Memory-Mapped TLB Access

5.6.3 Usage Examples

Invalidating Specific Entries: Specific TLB entries can be invalidated by writing 0 to the entry's V bit. R0 specifies the write data and R1 specifies the address.

```
; R0=H'1547 381C  R1=H'F201 3000
; MMUCR.IX=0
; the V bit of way 0 of the entry selected by the VPN(16-12)=B'1 0011
; index is cleared to0,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a specific TLB entry. The bit order indicated in the data field in figure 5.14 (2) is read. R0 specifies the address and the data section of a selected entry is read to R1.

```
; R0=H'F300 4300   VPN(16-12)=B'00100   Way 3
; MOV.L @R0,R1
```

5.7 Usage Note

The following operations should be performed in the P1 or P2 areas. In addition, when the P0, P3, or U0 areas are accessed consecutively (this access includes instruction fetching), the instruction code should be placed at least two instructions after the instruction that executes the following operations.

1. Modification of SR.MD or SR.BL
2. Execution of the LDTLB instruction
3. Write to the memory-mapped TLB
4. Modification of MMUCR
5. Modification of PTEH.ASID

Section 6 Cache

6.1 Features

- Capacity: 16 or 32 kbytes
- Structure: Instructions/data mixed, 4-way set associative
- Locking: Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 256 entries/way in 16-kbyte mode to 512 entries/way in 32-kbyte mode
- Write system: Write-back/write-through is selectable for spaces P0, P1, P3, and U0
- Replacement method: Least-recently used (LRU) algorithm

Note: After power-on reset or manual reset, initialized as 16-kbyte mode (256 entries/way).

6.1.1 Cache Structure

The cache mixes instructions and data and uses a 4-way set associative system. It is composed of four ways (banks), and each of which is divided into an address section and a data section.

Each of the address and data sections is divided into 512 entries. The entry data is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 8 kbytes (16 bytes × 512 entries) in the cache as a whole (4 ways). The cache capacity is 32 kbytes as a whole. Figure 6.1 shows the cache structure.

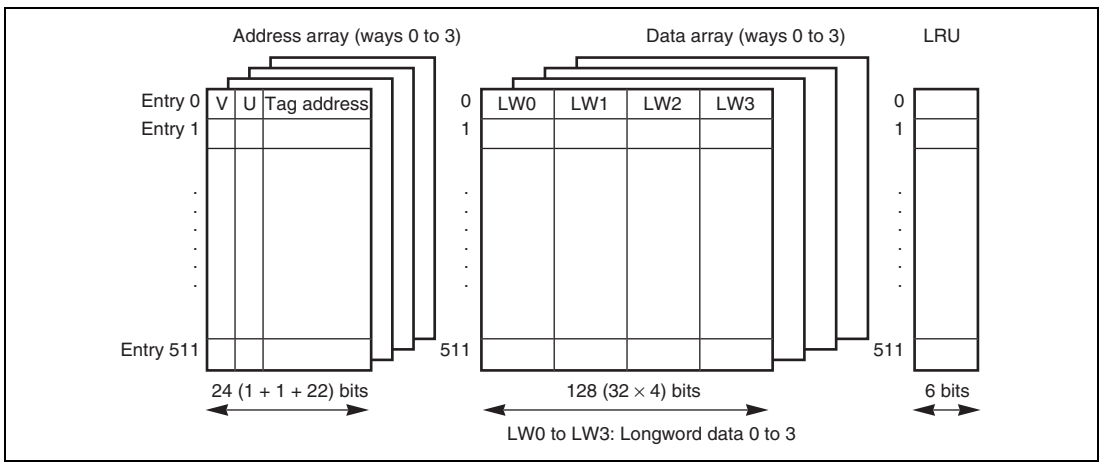


Figure 6.1 Cache Structure

Address Array: The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid. The U bit indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not. The tag address holds the physical address used in the external memory access. It is composed of 22 bits (address bits 31–10) used for comparison during cache searches.

In this LSI, the top three of 32 physical address bits are used as shadow bits (see section 12, Bus State Controller (BSC)), and therefore the top three bits of the tag address are cleared to 0.

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

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

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

Six LRU bits indicate the way to be replaced, when a cache miss occurs. Table 6.1 shows the relationship between the LRU bits and the way to be replaced when the cache locking mechanism is disabled. (For the relationship when the cache locking mechanism is enabled, refer to section 6.2.2, Cache Control Register 2 (CCR2).) If a bit pattern other than those listed in table 6.1 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 6.1.

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

Table 6.1 LRU and Way Replacement (when Cache Locking Mechanism is Disabled)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

6.2 Register Descriptions

The cache has the following registers. For details on register addresses and register states during each process, refer to section 24, List of Registers.

- Cache control register 1 (CCR1)
- Cache control register 2 (CCR2)
- Cache control register 3 (CCR3)

6.2.1 Cache Control Register 1 (CCR1)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries (clears the V, U, and LRU bits of all cache entries to 0). This bit is always read as 0. Write-back to external memory is not performed when the cache is flushed.
2	CB	0	R/W	Write-Back Indicates the cache's operating mode for space P1. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write-Through Indicates the cache's operating mode for spaces P0, U0, and P3. 0: Write-back mode 1: Write-through mode

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Cache Enable Indicates whether the cache function is used. 0: The cache function is not used. 1: The cache function is used.

6.2.2 Cache Control Register 2 (CCR2)

The CCR2 register controls the cache locking mechanism in cache lock mode only. The CPU enters the cache lock mode when the DSP bit (bit 12) in the status register (SR) is set to 1 or the lock enable bit (bit 16) in the cache control register 2 (CCR2) is set to 1. The cache locking mechanism is disabled in non-cache lock mode (DSP bit = 0).

When a prefetch instruction (PREF@Rn) is issued in cache lock mode and a cache miss occurs, the line of data pointed to by Rn will be loaded into the cache, according to the setting of bits 9 and 8 (W3LOAD, W3LOCK) and bits 1 and 0 (W2LOAD, W2LOCK in CCR2).

Table 6.2 shows the relationship between the settings of bits and the way that is to be replaced when the cache is missed by a prefetch instruction.

On the other hand, when the cache is hit by a prefetch instruction, new data is not loaded into the cache and the valid entry is held. For example, a prefetch instruction is issued while bits W3LOAD and W3LOCK are set to 1 and the line of data to which Rn points is already in way 0, the cache is hit and new data is not loaded into way 3.

In cache lock mode, bits W3LOCK and W2LOCK restrict the way that is to be replaced, when instructions other than the prefetch instruction are issued. Table 6.3 shows the relationship between the settings of bits in CCR2 and the way that is to be replaced when the cache is missed by instructions other than the prefetch instruction.

Programs that change the contents of the CCR2 register should be placed in address space that is not cached.

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	LE	0	R/W	Lock enable (LE) Controls cache lock mode. 0: Enters cache lock mode when the DSP bit of the SR register is set to 1. 1: Enters cache lock mode regardless of the DSP bit value.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD	0	R/W	Way 3 Load (W3LOAD)
8	W3LOCK	0	R/W	Way 3 Lock (W3LOCK) When the cache is missed by a prefetch instruction while in cache lock mode and when bits W3LOAD and W3LOCK in CCR2 are set to 1, the data is always loaded into way 3. Under any other condition, the prefetched data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD	0	R/W	Way 2 Load (W2LOAD)
0	W2LOCK	0	R/W	Way 2 Lock (W2LOCK) When the cache is missed by a prefetch instruction while in cache lock mode and when bits W2LOAD and W2LOCK in CCR2 are set to 1, the data is always loaded into way 2. Under any other condition, the prefetched data is loaded into the way to which LRU points.

Note: W2LOAD and W3LOAD should not be set to 1 at the same time.

Table 6.2 Way Replacement when a PREF Instruction Misses the Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Determined by LRU (table 6.1)
1	*	0	*	0	Determined by LRU (table 6.1)
1	*	0	0	1	Determined by LRU (table 6.4)
1	0	1	*	0	Determined by LRU (table 6.5)
1	0	1	0	1	Determined by LRU (table 6.6)
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

Note: * Don't care
W3LOAD and W2LOAD should not be set to 1 at the same time.

Table 6.3 Way Replacement when Instructions other than the PREF Instruction Miss the Cache

DSP Bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	Determined by LRU (table 6.1)
1	*	0	*	0	Determined by LRU (table 6.1)
1	*	0	*	1	Determined by LRU (table 6.4)
1	*	1	*	0	Determined by LRU (table 6.5)
1	*	1	*	1	Determined by LRU (table 6.6)

Note: * Don't care
W3LOAD and W2LOAD should not be set to 1 at the same time.

Table 6.4 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =0)

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

Table 6.5 LRU and Way Replacement (when W2LOCK = 0 and W3LOCK =1)

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

Table 6.6 LRU and Way Replacement (when W2LOCK = 1 and W3LOCK =1)

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

6.2.3 Cache Control Register 3 (CCR3)

The CCR3 register controls the cache size to be used. The cache size must be specified according to the LSI to be selected. If the specified cache size exceeds the size of cache incorporated in the LSI, correct operation cannot be guaranteed. Note that programs that change the contents of the CCR3 register should be placed in un-cached address space. In addition, note that all cache entries must be invalidated by setting the CF bit of the CCR1 to 1 before accessing the cache after the CCR3 is modified.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CSIZE7 to CSIZE0	H'01	R/W	Cache Size Specify the cache size as shown below. 0000 0001: 16-kbyte cache 0000 0010: 32-kbyte cache Settings other than above are prohibited.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3 Operation

6.3.1 Searching the Cache

If the cache is enabled (the CE bit in CCR1 = 1), whenever instructions or data in spaces P0, P1, P3, and U0 are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 6.2 illustrates the method by which the cache is searched. The cache is a physical cache and holds physical addresses in its address section.

Entries are selected using bits 12 to 4 of the address (virtual) of the access to memory and the tag address of that entry is read. The virtual address (bits 31 to 10) of the access to memory and the physical address (tag address) read from the address array are compared. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 6.2 shows a hit on way 1.

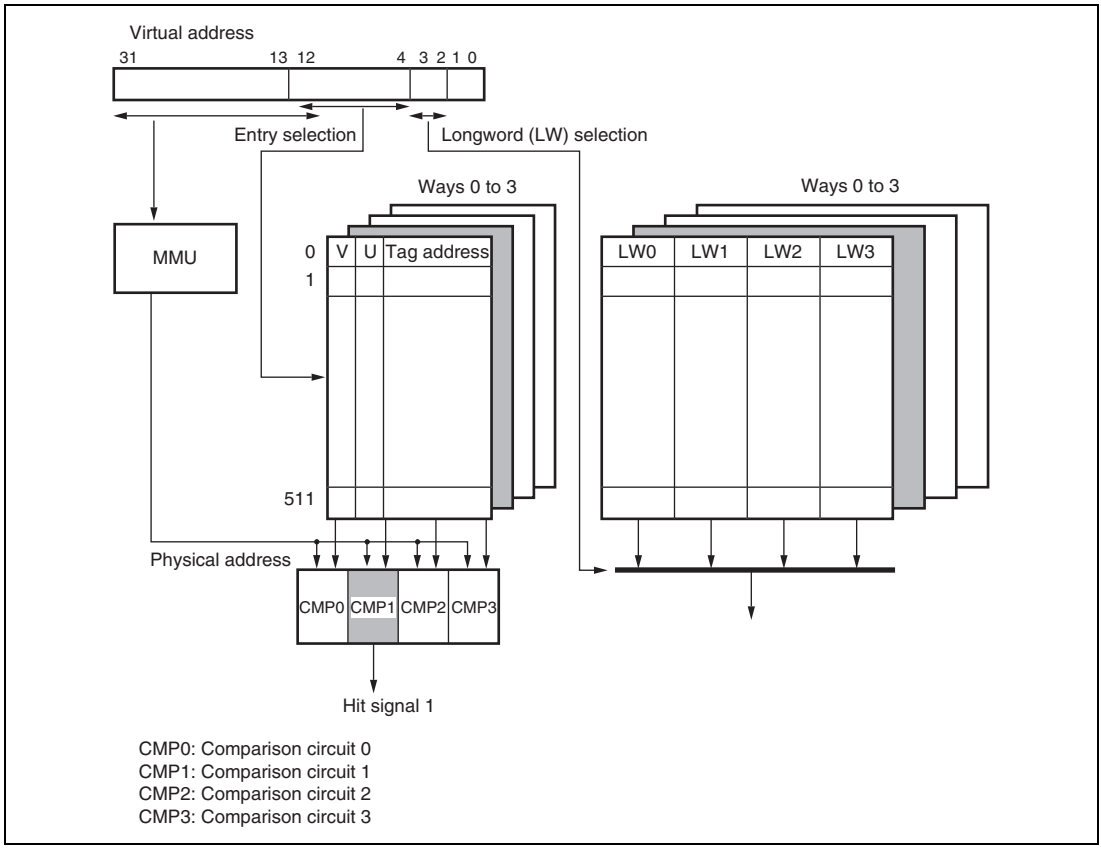


Figure 6.2 Cache Search Scheme

6.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. The LRU is updated to indicate that the hit way is the most recently hit way.

Read Miss: An external bus cycle starts and the entry is updated. The way to be replaced is shown in table 6.3. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded to the cache, the U bit is cleared to 0 and the V bit is set to 1 to indicate that the hit way is the most recently hit way. When the U bit for the entry which is to be replaced by entry updating in write-back mode is 1, the cache-update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 16-byte units.

6.3.3 Prefetch Operation

Prefetch Hit: The LRU is updated to indicate that the hit way is the most recently hit way. The other contents of the cache are not changed. Instructions and data are not transferred from the cache to the CPU.

Prefetch Miss: Instructions and data are not transferred from the cache to the CPU. The way that is to be replaced is shown in table 6.2. The other operations are the same as those for a read miss.

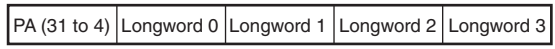
6.3.4 Write Access

Write Hit: In a write access in write-back mode, the data is written to the cache and no external memory write cycle is issued. The U bit of the entry that has been written to is set to 1, and the LRU is updated to indicate that the hit way is the most recently hit way. In write-through mode, the data is written to the cache and an external memory write cycle is issued. The U bit of the entry that has been written to is not updated, and the LRU is updated to indicate that the hit way is the most recently hit way.

Write Miss: In write-back mode, an external write cycle starts when a write miss occurs, and the entry is updated. The way to be replaced is shown in table 6.3. When the U bit of the entry which is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has been transferred to the write-back buffer. Data is written to the cache and the U bit and the V bit are set to 1. The LRU is updated to indicate that the replaced way is the most recently updated way. After the cache has completed its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

6.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the fetching of new entries to the cache completes, the write-back buffer writes the entry back to the external memory. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 6.3 shows the configuration of the write-back buffer.



PA (31 to 4): Physical address written to external memory
Longword 0 to 3: One line of cache data to be written to external memory

Figure 6.3 Write-Back Buffer Configuration

6.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is placed in an address space to which caching applies, use the memory-mapped cache to make the data invalid and written back, as required. Memory that is shared by this LSI's CPU and DMAC should also be handled in this way.

6.4 Memory-Mapped Cache

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

6.4.1 Address Array

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

In the address field, specify the entry address for selecting the entry, W for selecting the way, A for enabling or disabling the associative operation, and H'F0 for indicating address array access. As for W, 00 indicates way 0, 01 indicates way 1, 10 indicates way 2, and 11 indicates way 3).

In the data field, specify the tag address, LRU bits, U bit, and V bit. Figures 6.4 and 6.5 show the address and data formats. The following three operations are available in the address array.

Address-Array Read: Read the tag address, LRU bits, U bit, and V bit for the entry that corresponds to the entry address and way specified by the address field of the read instruction. In reading, the associative operation is not performed, regardless of whether the associative bit (A bit) specified in the address is 1 or 0.

Address-Array Write (non-Associative Operation): Write the tag address, LRU bits, U bit, and V bit, specified by the data field of the write instruction, to the entry that corresponds to the entry address and way as specified by the address field of the write instruction. Ensure that the associative bit (A bit) in the address field is set to 0. When writing to a cache line for which the U bit = 1 and the V bit =1, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field of the write instruction. When 0 is written to the V bit, 0 must also be written to the U bit for that entry.

Address-Array Write (Associative Operation): When writing with the associative bit (A bit) of the address = 1, the addresses in the four ways for the entry specified by the address field of the write instruction are compared with the tag address that is specified by the data field of the write instruction. If the MMU is enabled in this case, a logical address specified by data is translated into a physical address via the TLB before comparison. Write the U bit and the V bit specified by the data field of the write instruction to the entry of the way that has a hit. However, the tag

address and LRU bits remain unchanged. When there is no way that receives a hit, nothing is written and there is no operation. This function is used to invalidate a specific entry in the cache. When the U bit of the entry that has received a hit is 1 at this point, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

6.4.2 Data Array

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

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

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

Figures 6.4 and 6.5 show the address and data formats.

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

Data-Array Read: Read the data specified by L of the address filed, from the entry that corresponds to the entry address and the way that is specified by the address filed.

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

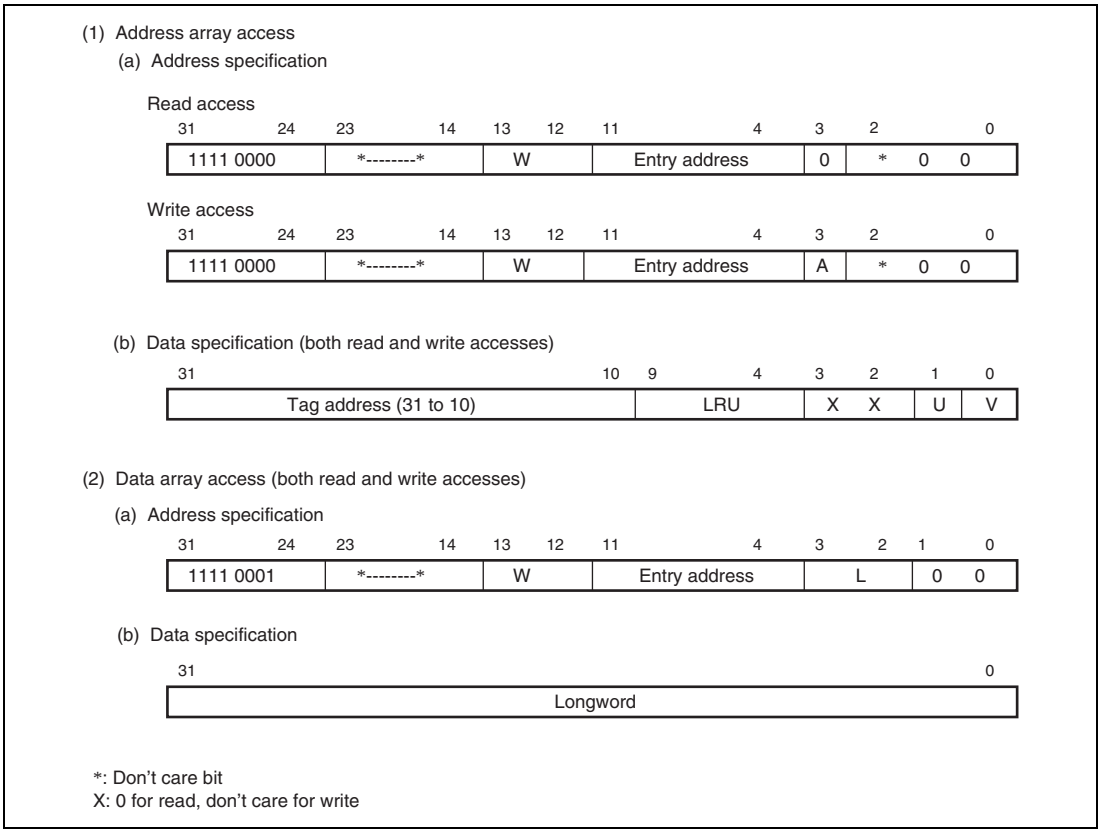
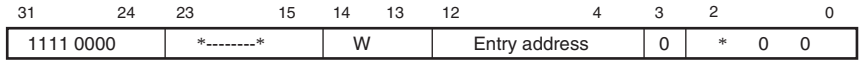


Figure 6.4 Specifying Address and Data for Memory-Mapped Cache Access (16 kbytes mode)

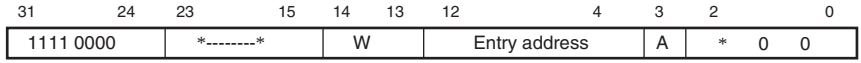
(1) Address array access

(a) Address specification

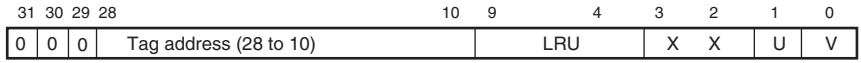
Read access



Write access

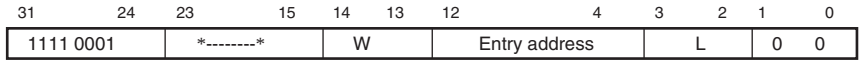


(b) Data specification (both read and write accesses)

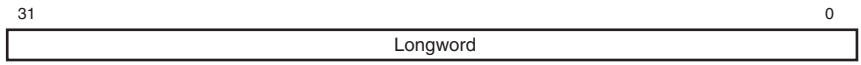


(2) Data array access (both read and write accesses)

(a) Address specification



(b) Data specification



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

Figure 6.5 Specifying Address and Data for Memory-Mapped Cache Access (32 kbytes mode)

6.4.3 Usage Examples

Invalidating Specific Entries: Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory-mapped cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address within the cache selected by the entry address, and a match is found, the entry is written back if the entry's U bit is 1 and the V bit and U bit specified by the write data are written. If no match is found, there is no operation. In the example shown below, R0 specifies the write data and R1 specifies the address.

```
; R0=H'01100010; VPN=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F0000088; address array access, entry=B'00001000, A=1
;
MOV.L R0,@R1
```

Reading the Data of a Specific Entry: To read the data field of a specific entry is enabled by the memory-mapped cache access. The longword indicated in the data field of the data array in figure 6.4 or 6.5 is read into the register. In the example shown below, R0 specifies the address and R1 shows what is read.

```
; R0=H'F100 004C; data array access, entry=B'00000100
; Way = 0, longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 7 X/Y Memory

This LSI has on-chip X-memory and Y-memory which can be used to store instructions or data.

7.1 Features

- Page

  There are four pages. The X memory is divided into two pages (pages 0 and 1) and the Y memory is divided into two pages (pages 0 and 1).
- Memory map

  The X/Y memory is located in the logical address space, physical address space, and X-bus and Y-bus address spaces.

  In the logical address space, this memory is located in the addresses shown in table 7.1. These addresses are included in space P2 (when SR.MD = 1) or Uxy (when SR.MD = 0 and SR.DSP = 1) according to the CPU operating mode.

Table 7.1 X/Y Memory Logical Addresses

	Memory Size (Total Four Pages)
Page	**16 kbytes**
Page 0 of X memory	H′A5007000 to H′A5007FFF
Page 1 of X memory	H′A5008000 to H′A5008FFF
Page 0 of Y memory	H′A5017000 to H′A5017FFF
Page 1 of Y memory	H′A5018000 to H′A5018FFF

On the other hand, this memory is located in a part of area 1 in the physical address space. When this memory is accessed from the physical address space, addresses in which the upper three bits are 0 in addresses shown in table 7.1 are used. In the X-bus and Y-bus address spaces, addresses in which the upper 16 bits are ignored in addresses of X memory and Y memory shown in table 7.1 are used.

- Ports

  Each page has three independent read/write ports and is connected to each bus. The X memory is connected to the I bus, X bus, and L bus. The Y memory is connected to the I bus, Y bus, and L bus. The L bus is used when this memory is accessed from the logical address space. The I bus is used when this memory is accessed from the physical address space. The X bus and Y bus are used when this memory is accessed from the X-bus and Y-bus address spaces.

- Priority order

  In the event of simultaneous accesses to the same page from different buses, the accesses are processed according to the priority order. The priority order is: I bus > X bus > L bus in the X memory and I bus > Y bus > L bus in the Y memory.

7.2 Operation

7.2.1 Access from CPU

Methods for accessing by the CPU are directly via the L bus from the logical addresses, and via the I bus after the logical addresses are converted to be the physical addresses using the MMU. As long as a conflict on the page does not occur, access via the L bus is performed in one cycle. Several cycles are necessary for accessing via the I bus. According to the CPU operating mode, access from the CPU is as follows:

Privileged mode and privileged DSP mode (SR. MD = 1): The X/Y memory can be accessed by the CPU directly from space P2. The MMU can be used to map the logical addresses in spaces P0 and P3 to this memory.

User DSP mode (SR.MD = 0 and SR.DSP = 1): The X/Y memory can be accessed by the CPU directly from space Uxy. The MMU can be used to map the logical addresses in space U0 to this memory.

User mode (SR.MD = 0 and SR.DSP = 0): The MMU can be used to map the logical addresses in space U0 to this memory.

7.2.2 Access from DSP

Methods for accessing from the DSP differ according to instructions.

With a X data transfer instruction and a Y data transfer instruction, the X/Y memory is always accessed via the X bus or Y bus. As long as a conflict on the page does not occur, access via the X bus or Y bus is performed in one cycle. The X memory access via the X bus and the Y memory access via the Y bus can be performed simultaneously.

In the case of a single data transfer instruction, methods for accessing from the DSP are directly via the L bus from the logical addresses, and via the I bus after the logical addresses are converted to be the physical addresses using the MMU. As long as a conflict on the page does not occur, access via the L bus is performed in one cycle. Several cycles are necessary for accessing via the I bus. According to the CPU operating mode, access from the CPU is as follows:

Privileged DSP mode (SR. MD = 1 and SR.DSP = 1): The X/Y memory can be accessed by the DSP directly from space P2. The MMU can be used to map the logical addresses in spaces P0 and P3 to this memory.

User DSP mode (SR.MD = 0 and SR.DSP = 1): The X/Y memory can be accessed by the DSP directly from space Uxy. The MMU can be used to map the logical addresses in space U0 to this memory.

7.2.3 Access from DMAC, E-DMAC, and IPSEC

The X/Y memory is always accessed by the DMAC, E-DMAC, and IPSEC via the I bus, which is a physical address bus. Addresses in which the upper three bits are 0 in addresses shown in table 7.1 must be used.

7.3 Usage Notes

7.3.1 Page Conflict

In the event of simultaneous accesses to the same page from different buses, the conflict on the pages occurs. Although each access is completed correctly, this kind of conflict tends to lower X/Y memory accessibility. Therefore it is advisable to provide software measures to prevent such conflict as far as possible. For example, conflict will not arise if different memory or different pages are accessed by each bus.

7.3.2 Bus Conflict

The I bus is shared by several bus master modules. When the X/Y memory is accessed via the I bus, a conflict between the other I-bus master modules may occur on the I bus. This kind of conflict tends to lower X/Y memory accessibility. Therefore it is advisable to provide software measures to prevent such conflict as far as possible. For example, by accessing the X/Y memory by the CPU not via the I bus but directly from space P2 or Uxy, conflict on the I bus can be prevented.

7.3.3 MMU and Cache Settings

When the X/Y memory is accessed via the I bus using the cache from the CPU and DSP, correct operation cannot be guaranteed. If the X/Y memory is accessed while the cache is enabled (CCR1.CE = 1), it is advisable to access the X/Y memory via the L bus from space P2 or Uxy. If the X/Y memory is accessed from space P0, P3, or U0, it is advisable to access the X/Y memory

via the I bus, which does not use the cache, with MMU setting enabled (MMUCR.AT = 1) and cache disabled (C bit = 0) as page attributes. Since access using the MMU occurs via the I bus, several cycles are necessary (the number of necessary cycles varies according to the ratio between the internal clock (Iϕ) and bus clock (Bϕ) or the operation state of the DMAC, E-DMAC and IPSEC). In a program that requires high performance, it is advisable to access the X/Y memory from space P2 or Uxy.

The relationship described above is summarized in table 7.2.

Table 7.2 MMU and Cache Settings

Setting		Logical Address Space and Access Enabled or Disabled			
CCR1.CE	**MMUCR.AT**	**P0, U0**	**P1**	**P2, Uxy**	**P3**
0	0	B	B	A	B
0	1	B	B	A	B
1	0	X	X	A	X
1	1	C	X	A	C

[Legend]
A: Accessible (recommended)
B: Accessible
C: Accessible (Note that MMU page attribute must be specified as cache disabled by clearing the C bit to 0.)
X: Not Accessible

7.3.4 Sleep Mode

In sleep mode, I bus master modules such as the DMAC, E-DMAC and IPSEC cannot access the X/Y memory.

7.3.5 Address Error

If writing that may causes an address error is performed on the X/Y memory, the contents of the X/Y memory are not guaranteed.

Section 8 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

8.1 Features

The INTC has the following features:

- 16 levels of interrupt priority can be set

  By setting the interrupt-priority registers, the priorities of on-chip peripheral modules, and IRQ interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceler function

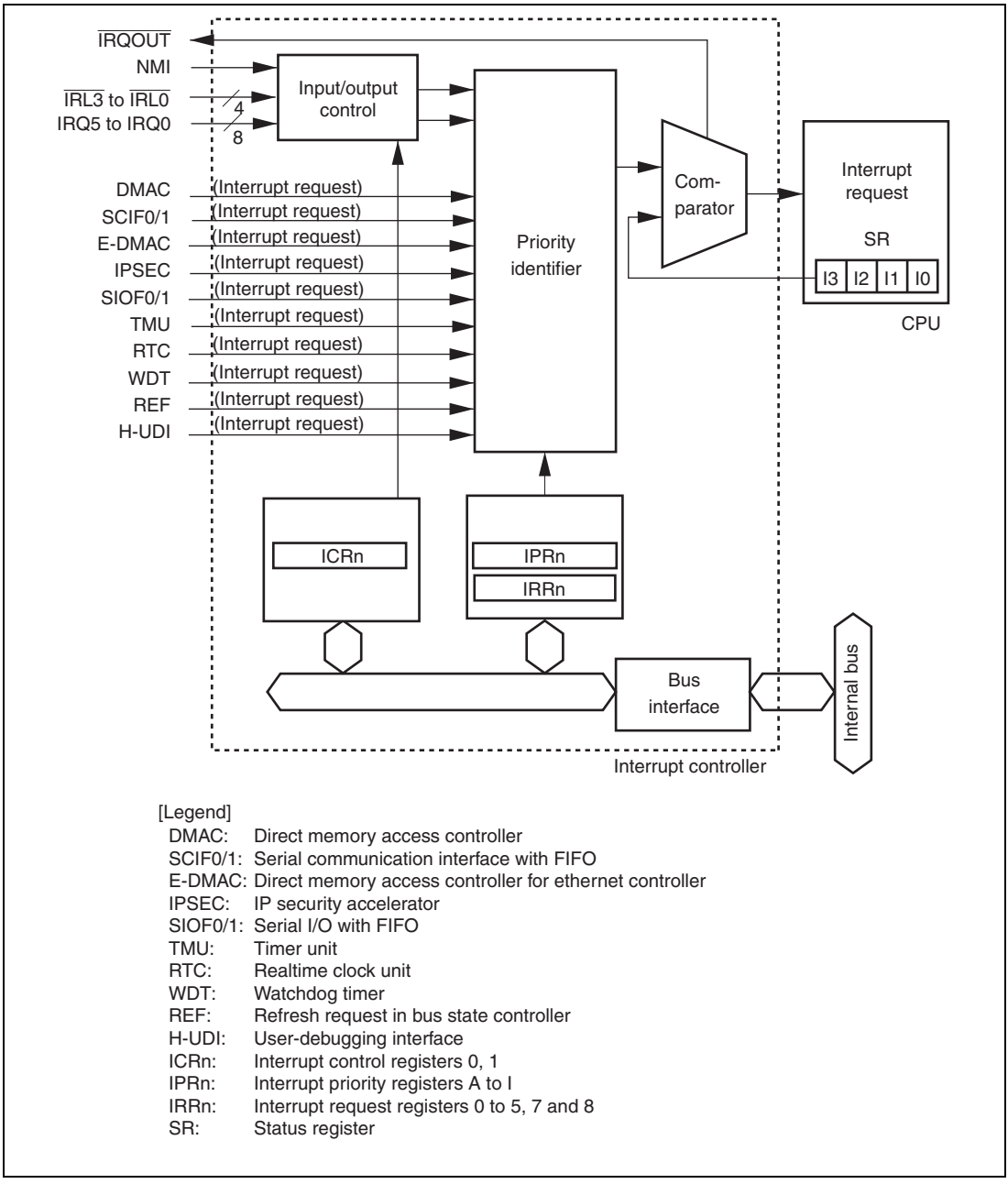
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as a noise canceler.
- IRQ interrupts can be set

  Detection of low level, rising edge, falling edge, or high level
- Interrupt request signal can be externally output ($\overline{\text{IRQOUT}}$ pin)

  The bus mastership can be requested by notifying the external bus master that the external interrupt and on-chip peripheral module interrupt requests have been generated.

8.1.1 Block Diagram

Figure 8.1 shows a block diagram of the INTC.



[Legend]

DMAC:	Direct memory access controller
SCIF0/1:	Serial communication interface with FIFO
E-DMAC:	Direct memory access controller for ethernet controller
IPSEC:	IP security accelerator
SIOF0/1:	Serial I/O with FIFO
TMU:	Timer unit
RTC:	Realtime clock unit
WDT:	Watchdog timer
REF:	Refresh request in bus state controller
H-UDI:	User-debugging interface
ICRn:	Interrupt control registers 0, 1
IPRn:	Interrupt priority registers A to I
IRRn:	Interrupt request registers 0 to 5, 7 and 8
SR:	Status register

Figure 8.1 Block Diagram of INTC

8.2 Input/Output Pins

Table 8.1 shows the INTC pin configuration.

Table 8.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	Input	Input of interrupt request signal, not maskable by the interrupt mask bits in SR
Interrupt input pins	IRQ5 to IRQ0 $\overline{IRL3}$ to $\overline{IRL0}$*[1]	Input	Input of interrupt request signals
Bus mastership request output pin*[2]	$\overline{IRQOUT}$	Output	Notifies that an interrupt request has generated

Notes: 1. The $\overline{IRL3}$ to $\overline{IRL0}$ pins and the IRQ3 to IRQ0 cannot be used simultaneously because these pins are multiplexed with the IRQ3 to IRQ0 pins.
2. When the NMI or H-UDI interrupt requests are generated and the response time of the CPU is short, this pin may not be asserted.

8.3 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 1 the lowest and 16 the highest. Priority level 0 masks an interrupt, so the interrupt request is ignored.

8.3.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. When the BLMSK bit in the interrupt control register 1 (ICR1) is set to 1 or the BL bit in the status register (SR) is 0 and the MAI bit in ICR1 is 0, NMI interrupts are accepted. NMI interrupts are edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL setting. The NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is used to select either rising or falling edge detection.

When using edge-input detection for NMI interrupts, a pulse width of at least two Pϕ cycles (peripheral clock) is necessary. NMI interrupt exception handling does not affect the interrupt mask level bits (I3 to I0) in the status register (SR). When the BL bit is 1 and the BLMSK bit in ICR1 is set to 1, only the NMI interrupt is accepted.

It is possible to wake the chip up from sleep mode or standby mode with the NMI interrupt.

8.3.2 IRQ Interrupts

IRQ interrupts are input by level or edge from pins IRQ0 to IRQ5. The priority level can be set by interrupt priority registers C and D (IPRC and IPRD) in a range from 0 to 15.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having software read 1 from the corresponding bit in IRR0, then write 0 to the bit.

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

Edge input interrupt detection requires input of a pulse width of more than two cycles on a P clock basis.

When using level-sensing for IRQ interrupts, the pin levels must be retained until the CPU samples the pins. Therefore, the interrupt source must be cleared by the interrupt handler.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRQ interrupt handling. IRQ interrupts can be used for recovering from standby when the corresponding interrupt level is higher than that of bits I3 to I0 in SR. (However, only when the RTC is used, recovering from standby by using the clock for the RTC is enabled.)

8.3.3 IRL Interrupts

IRL interrupts are input by the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins as level. The priority level is the higher level that is indicated by the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins. When the values of the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ pins are 0 (B′0000), the highest level interrupt request (interrupt priority level 15) is indicated. When the values of the pins are 15 (B′1111), no interrupt is requested (interrupt priority level 0). Figure 8.2 shows an example of connection for an IRL interrupt. Table 8.3 lists the $\overline{\text{IRL}}$ pins and interrupt level.

IRL interrupts are included with a noise canceler function and detected when the sampled levels of each peripheral module clock keep the same value for 2 cycles. This prevents sampling error level in $\overline{\text{IRL}}$ pin changing. In standby mode, a noise canceler is handled by the RTC clock (32 kHz) because the peripheral module clocks are halted. Therefore, when the RTC is not used, recovering from standby by IRL interrupts cannot be executed in standby mode.

The priority level of IRL interrupts should be kept until an interrupt is accepted and its handling is started. However, changing to higher level is enabled.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by the IRL interrupt handling.

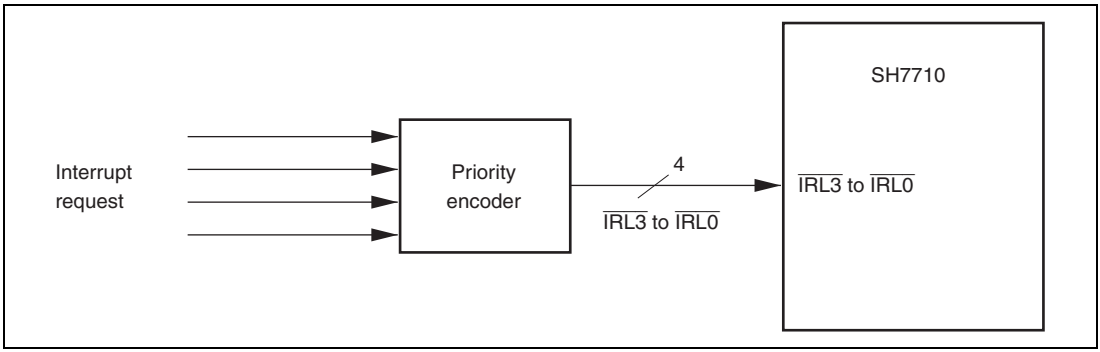


Figure 8.2 Example of IRL Interrupt Connection

8.3.4 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following 14 modules:

- Direct memory access controller (DMAC)
- Serial communication interface with FIFO (SCIF0 and SCIF1)
- Direct memory access controller for ethernet controller (E-DMAC) (includes an EtherC interrupt)
- IP security accelerator (IPSEC)
- Serial I/O with FIFO (SIOF0 and SIOF1)
- Timer unit (TMU0 to TMU2)
- Realtime clock (RTC)
- Watchdog timer (WDT)
- Bus state controller (BSC)
- User-debugging interface (H-UDI)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected in the interrupt event registers (INTEVT and INTEVT2). It is easy to identify sources by using the value of INTEVT or INTEVT2 as a branch offset.

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

The interrupt mask bits (I3 to I0) in the status register are not affected by on-chip peripheral module interrupt handling.

8.3.5 Interrupt Exception Handling and Priority

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral modules. The priority of each interrupt source is set within priority levels 0 to 16; level 16 is the highest and level 1 is the lowest. When the priority is set to level 0, that interrupt is masked and the interrupt request is ignored.

Tables 8.2 and 8.3 list the codes for the interrupt event registers (INTEVT and INTEVT2) and the order of interrupt priority.

Each interrupt source is assigned a unique code by INTEVT or INTEVT2. The start address of the interrupt service routine is common for each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as an offset at the start of the interrupt service routine and branched to in order to identify the interrupt source.

IRQ interrupt and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each module by setting interrupt priority registers A to I (IPRA to IPRI). A reset assigns priority level 0 to IRQ and on-chip peripheral module interrupts.

If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in tables 8.2 and 8.3.

Table 8.2 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		Interrupt Code*1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0*2	16	—	—	High
H-UDI		H'5E0*2	15	—	—	↕
IRQ	IRQ0	H'600*3	0 to 15 (0)	IPRC (3 to 0)	—	↕
	IRQ1	H'620*3	0 to 15 (0)	IPRC (7 to 4)	—	↕
	IRQ2	H'640*3	0 to 15 (0)	IPRC (11 to 8)	—	↕
	IRQ3	H'660*3	0 to 15 (0)	IPRC (15 to 12)	—	↕
	IRQ4	H'680*3	0 to 15 (0)	IPRD (3 to 0)	—	↕
	IRQ5	H'6A0*3	0 to 15 (0)	IPRD (7 to 4)	—	Low

Interrupt Source		Interrupt Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
DMAC (1)	DEI0	H'800*[3]	0 to 15 (0)	IPRE (15 to12)	High	High
	DEI1	H'820*[3]	0 to 15 (0)			
	DEI2	H'840*[3]	0 to 15 (0)			
	DEI3	H'860*[3]	0 to 15 (0)		Low	
SCIF0	ERI0	H'880*[3]	0 to 15 (0)	IPRE (11 to 8)	High	
	RXI0	H'8A0*[3]				
	BRI0	H'8C0*[3]				
	TXI0	H'8E0*[3]			Low	
SCIF1	ERI1	H'900*[3]	0 to 15 (0)	IPRE (7 to 4)	High	
	RXI1	H'920*[3]				
	BRI1	H'940*[3]				
	TXI1	H'960*[3]			Low	
DMAC (2)	DEI4	H'B80*[3]	0 to 15 (0)	IPRF (11 to 8)	High	
	DEI5	H'BA0*[3]			Low	
IPSEC	IPSECI	H'BE0*[3]	0 to 15 (0)	IPRF (15 to 12)	—	
E-DMAC	EINT0	H'C00*[3]	0 to 15 (0)	IPRG (15 to 12)	—	
	EINT1	H'C20*[3]	0 to 15 (0)	IPRG (11 to 8)	—	
	EINT2	H'C40*[3]	0 to 15 (0)	IPRG (7 to 4)	—	
SIOF0	ERI0	H'E00*[3]	0 to 15 (0)	IPRH (3 to 0)	High	
	TXI0	H'E20*[3]				
	RXI0	H'E40*[3]				
	CCI0	H'E60*[3]			Low	
SIOF1	ERI1	H'E80*[3]	0 to 15 (0)	IPRI (7 to 4)	High	
	TXI1	H'EA0*[3]				
	RXI1	H'EC0*[3]				
	CCI1	H'EE0*[3]			Low	
TMU0	TUNI0	H'400*[2]	0 to 15 (0)	IPRA (15 to 12)	—	
TMU1	TUNI1	H'420*[2]	0 to 15 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440*[2]	0 to 15 (0)	IPRA (7 to 4)	—	Low

Interrupt Source		Interrupt Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
RTC	ATI	H′480*[2]	0 to 15 (0)	IPRA (3 to 0)	High	High
	PRI	H′4A0*[2]			↕	↕
	CUI	H′4C0*[2]			Low	
WDT	ITI	H′560*[2]	0 to 15 (0)	IPRB (15 to 12)	—	
REF	RCMI	H′580*[2]	0 to 15 (0)	IPRB (11 to 8)	—	Low

Notes: 1. INTEVT2 code

2. The code set in INTEVT is as same as INTEVT2.

3. The code that indicates the interrupt level (H′200 to H′3C0) is set in INTEVT. For details on correspondence between the interrupt level and INTEVT, see table 8.4.

Table 8.3 Interrupt Exception Handling Sources and Priority (IRL Mode)

Interrupt Source		Interrupt Code*1	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H′1C0*2	16	—	—	High
H-UDI		H′5E0*2	15	—	—	▲
IRL	IRL[3:0] = B′0000	H′200*3	15	—	—	▲
	IRL[3:0] = B′0001	H′220*3	14	—	—	▲
	IRL[3:0] = B′0010	H′240*3	13	—	—	▲
	IRL[3:0] = B′0011	H′260*3	12	—	—	▲
	IRL[3:0] = B′0100	H′280*3	11	—	—	▲
	IRL[3:0] = B′0101	H′2A0*3	10	—	—	▲
	IRL[3:0] = B′0110	H′2C0*3	9	—	—	▲
	IRL[3:0] = B′0111	H′2E0*3	8	—	—	▲
	IRL[3:0] = B′1000	H′300*3	7	—	—	▲
	IRL[3:0] = B′1001	H′320*3	6	—	—	▲
	IRL[3:0] = B′1010	H′340*3	5	—	—	▲
	IRL[3:0] = B′1011	H′360*3	4	—	—	▲
	IRL[3:0] = B′1100	H′380*3	3	—	—	▲
	IRL[3:0] = B′1101	H′3A0*3	2	—	—	▲
	IRL[3:0] = B′1110	H′3C0*3	1	—	—	▲
IRQ	IRQ4	H′680*3	0 to 15 (0)	IPRD (3 to 0)	—	▲
	IRQ5	H′6A0*3	0 to 15 (0)	IPRD (7 to 4)	—	Low

Interrupt Source		Interrupt Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
DMAC (1)	DEI0	H'800*[3]	0 to 15 (0)	IPRE (15 to12)	High	High
	DEI1	H'820*[3]	0 to 15 (0)			
	DEI2	H'840*[3]	0 to 15 (0)			
	DEI3	H'860*[3]	0 to 15 (0)		Low	
SCIF0	ERI0	H'880*[3]	0 to 15 (0)	IPRE (11 to 8)	High	
	RXI0	H'8A0*[3]				
	BRI0	H'8C0*[3]				
	TXI0	H'8E0*[3]			Low	
SCIF1	ERI1	H'900*[3]	0 to 15 (0)	IPRE (7 to 4)	High	
	RXI1	H'920*[3]				
	BRI1	H'940*[3]				
	TXI1	H'960*[3]			Low	
DMAC (2)	DEI4	H'B80*[3]	0 to 15 (0)	IPRF (11 to 8)	High	
	DEI5	H'BA0*[3]			Low	
IPSEC	IPSECI	H'BE0*[3]	0 to 15 (0)	IPRF (15 to 12)	—	
E-DMAC	EINT0	H'C00*[3]	0 to 15 (0)	IPRG (15 to 12)	—	
	EINT1	H'C20*[3]	0 to 15 (0)	IPRG (11 to 8)	—	
	EINT2	H'C40*[3]	0 to 15 (0)	IPRG (7 to 4)	—	
SIOF0	ERI0	H'E00*[3]	0 to 15 (0)	IPRH (3 to 0)	High	
	TXI0	H'E20*[3]				
	RXI0	H'E40*[3]				
	CCI0	H'E60*[3]			Low	
SIOF1	ERI1	H'E80*[3]	0 to 15 (0)	IPRI (7 to 4)	High	
	TXI1	H'EA0*[3]				
	RXI1	H'EC0*[3]				
	CCI1	H'EE0*[3]			Low	
TMU0	TUNI0	H'400*[2]	0 to 15 (0)	IPRA (15 to 12)	—	
TMU1	TUNI1	H'420*[2]	0 to 15 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440*[2]	0 to 15 (0)	IPRA (7 to 4)	—	Low

Interrupt Source		Interrupt Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
RTC	ATI	H′480*[2]	0 to 15 (0)	IPRA (3 to 0)	High	High
	PRI	H′4A0*[2]			↕	↕
	CUI	H′4C0*[2]			Low	
WDT	ITI	H′560*[2]	0 to 15 (0)	IPRB (15 to 12)	—	
REF	RCMI	H′580*[2]	0 to 15 (0)	IPRB (11 to 8)	—	Low

Notes: 1. INTEVT2 code
2. The code set in INTEVT is as same as INTEVT2.
3. The code that indicates the interrupt level (H′200 to H′3C0) is set in INTEVT. For details on correspondence between the interrupt level and INTEVT, see table 8.4.

Table 8.4 Interrupt Level and INTEVT Code

Interrupt Level	INTEVT Code
15	H′200
14	H′220
13	H′240
12	H′260
11	H′280
10	H′2A0
9	H′2C0
8	H′2E0
7	H′300
6	H′320
5	H′340
4	H′360
3	H′380
2	H′3A0
1	H′3C0

8.4 Register Descriptions

The INTC has the following registers. For details on register addresses and register access size, refer to section 24, List of Registers.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)
- Interrupt request register 3 (IRR3)
- Interrupt request register 4 (IRR4)
- Interrupt request register 5 (IRR5)
- Interrupt request register 7 (IRR7)
- Interrupt request register 8 (IRR8)

8.4.1 Interrupt Priority Registers A to I (IPRA to IPRI)

IPRA to IPRI are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for on-chip peripheral module and IRQ interrupts. These registers are initialized to H'0000 by a power-on reset or manual reset, but are not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 8.5, Interrupt Sources and IPRA to IPRI.
14	—	0	R/W	
13	—	0	R/W	
12	—	0	R/W	
11	—	0	R/W	
10	—	0	R/W	
9	—	0	R/W	
8	—	0	R/W	
7	—	0	R/W	
6	—	0	R/W	
5	—	0	R/W	
4	—	0	R/W	
3	—	0	R/W	
2	—	0	R/W	
1	—	0	R/W	
0	—	0	R/W	

Table 8.5 Interrupt Sources and IPRA to IPRI

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	Reserved*	Reserved*
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	Reserved*	Reserved*	IRQ5	IRQ4
IPRE	DMAC (1)	SCIF0	SCIF1	Reserved*
IPRF	IPSEC	DMAC (2)	Reserved*	Reserved*
IPRG	E-DMAC (1)	E-DMAC (2)	E-DMAC (3)	Reserved*
IPRH	Reserved*	Reserved*	Reserved*	SIOF0
IPRI	Reserved*	Reserved*	SIOF1	Reserved*

Note: * Always read as 0. The write value should always be 0.

As shown in table 8.5, on-chip peripheral module or IRQ interrupts are assigned to four 4-bit groups in each register. These 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested); H'F means priority level 15 (the highest level).

8.4.2 Interrupt Control Register 0 (ICR0)

ICR0 is a register that sets the input signal detection mode of external interrupt input pin NMI, and indicates the input signal level at the NMI pin. This register is initialized to H'0000 or H'8000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the NMI pin. This bit can be read from to determine the NMI pin level. This bit cannot be modified. 0: NMI input level is low 1: NMI input level is high
14	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	—	0	R	
12	—	0	R	
11	—	0	R	
10	—	0	R	
9	—	0	R	
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal at the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI pin input 1: Interrupt request is detected on rising edge of NMI pin input

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value should always be 0.
5	—	0	R	
4	—	0	R	
3	—	0	R	
2	—	0	R	
1	—	0	R	
0	—	0	R	

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

8.4.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ0 to IRQ5 individually: rising edge, falling edge, high level, or low level. This register is initialized to H′4000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	MAI	0	R/W	All Interrupt Mask When this bit is set to 1, all interrupt requests are masked while low level is input to the NMI pin. In standby mode, an NMI interrupt is masked. 0: When the NMI pin is low, all interrupt requests are not masked 1: When the NMI pin is low, all interrupt requests are masked
14	IRQLVL	1	R/W	Interrupt Request Level Detection Enables or disables the use of pins IRQ3 to IRQ0 as four independent interrupt pins. The IRQ4 and IRQ5 pins are not affected. 0: Use of pins IRQ3 to IRQ0 as four independent interrupt pins enabled 1: Use of pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ as encoded 15 level interrupt pins

Bit	Bit Name	Initial Value	R/W	Description
13	BLMSK	0	R/W	BL Bit Mask When the BL bit in SR is set to 1, this bit specifies whether an NMI interrupt is masked or not. 0: When the BL bit is set to 1, an NMI interrupt is masked 1: An NMI interrupt is accepted regardless of the BL bit setting
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	IRQ51S	0	R/W	IRQn Sense Select These bits select whether interrupt request signals corresponding to pins IRQ5 to IRQ0 are detected by a rising edge, falling edge, high level, or low level.
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

Bit 2n+1 IRQn1S	Bit 2n IRQn0S	
0	0	Interrupt request is detected on falling edge of IRQn input
0	1	Interrupt request is detected on rising edge of IRQn input
1	0	Interrupt request is detected on low level of IRQn input
1	1	Interrupt request is detected on high level of IRQn input

Legend n=0 to 5

8.4.4 Interrupt Request Register 0 (IRR0)

IRR0 is an 8-bit register that indicates interrupt requests from external input pins IRQ5 to IRQ0. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	—	0	R	These bit are always read as 0. The write value should always be 0.
5	IRQ5R	0	R/W	IRQn Interrupt Request
4	IRQ4R	0	R/W	Indicates whether there is interrupt request input to the IRQn pin. When edge-detection mode is set for IRQn, an interrupt request is cleared by writing 0 to the IRQnR bit after reading IRQnR = 1.
3	IRQ3R	0	R/W	
2	IRQ2R	0	R/W	
1	IRQ1R	0	R/W	
0	IRQ0R	0	R/W	When level-detection mode is set for IRQn, an interrupt request is set/cleared by only 1/0 input to the IRQn pin. IRQnR 0: No interrupt request input to IRQn pin 1: Interrupt request input to IRQn pin Legend: n = 0 to 5

8.4.5 Interrupt Request Register 1 (IRR1)

IRR1 is an 8-bit register that indicates whether interrupt requests from the DMAC and the SCIF0 are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TXI0R	0	R	TXI0 Interrupt Request Indicates whether the TXI0 (SCIF0) interrupt request is generated. 0: TXI0 interrupt request is not generated 1: TXI0 interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
6	BRI0R	0	R	BRI0 Interrupt Request Indicates whether the BRI0 (SCIF0) interrupt request is generated. 0: BRI0 interrupt request is not generated 1: BRI0 interrupt request is generated
5	RXI0R	0	R	RXI0 Interrupt Request Indicates whether the RXI0 (SCIF0) interrupt request is generated. 0: RXI0 interrupt request is not generated 1: RXI0 interrupt request is generated
4	ERI0R	0	R	ERI0 Interrupt Request Indicates whether the ERI0 (SCIF0) interrupt request is generated. 0: ERI0 interrupt request is not generated 1: ERI0 interrupt request is generated
3	DEI3R	0	R	DEI3 Interrupt Request Indicates whether the DEI3 (DMAC) interrupt request is generated. 0: DEI3 interrupt request is not generated 1: DEI3 interrupt request is generated
2	DEI2R	0	R	DEI2 Interrupt Request Indicates whether the DEI2 (DMAC) interrupt request is generated. 0: DEI2 interrupt request is not generated 1: DEI2 interrupt request is generated
1	DEI1R	0	R	DEI1 Interrupt Request Indicates whether the DEI1 (DMAC) interrupt request is generated. 0: DEI1 interrupt request is not generated 1: DEI1 interrupt request is generated
0	DEI0R	0	R	DEI0 Interrupt Request Indicates whether the DEI0 (DMAC) interrupt request is generated. 0: DEI0 interrupt request is not generated 1: DEI0 interrupt request is generated

8.4.6 Interrupt Request Register 2 (IRR2)

IRR2 is an 8-bit register that indicates whether interrupt requests from the SCIF1 are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

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	TXI1R	0	R	TXI1 Interrupt Request Indicates whether the TXI1 (SCIF1) interrupt request is generated. 0: TXI1 interrupt request is not generated 1: TXI1 interrupt request is generated
2	BRI1R	0	R	BRI1 Interrupt Request Indicates whether the BRI1 (SCIF1) interrupt request is generated. 0: BRI1 interrupt request is not generated 1: BRI1 interrupt request is generated
1	RXI1R	0	R	RXI1 Interrupt Request Indicates whether the RXI1 (SCIF1) interrupt request is generated. 0: RXI1 interrupt request is not generated 1: RXI1 interrupt request is generated
0	ERI1R	0	R	ERI1 Interrupt Request Indicates whether the ERI1 (SCIF1) interrupt request is generated. 0: ERI1 interrupt request is not generated 1: ERI1 interrupt request is generated

8.4.7 Interrupt Request Register 3 (IRR3)

IRR3 is an 8-bit register that indicates whether interrupt requests from the RTC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

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	CUIR	0	R	CUI Interrupt Request Indicates whether the CUI (RTC) interrupt request is generated. 0: CUI interrupt request is not generated 1: CUI interrupt request is generated
1	PRIR	0	R	PRI Interrupt Request Indicates whether the PRI (RTC) interrupt request is generated. 0: PRI interrupt request is not generated 1: PRI interrupt request is generated
0	ATIR	0	R	ATI Interrupt Request Indicates whether the ATI (RTC) interrupt request is generated. 0: ATI interrupt request is not generated 1: ATI interrupt request is generated

8.4.8 Interrupt Request Register 4 (IRR4)

IRR4 is an 8-bit register that indicates whether interrupt requests from the TMU2, TMU1, TMU0, WDT, and REF are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit always read as 0. The write value should always be 0.
6	TUNI2R	0	R	TUNI2 Interrupt Request Indicates whether the TUNI2 (TMU2) interrupt request is generated. 0: TUNI2 interrupt request is not generated 1: TUNI2 interrupt request is generated
5	TUNI1R	0	R	TUNI1Interrupt Request Indicates whether the TUNI1 (TMU1) interrupt request is generated. 0: TUNI1 interrupt request is not generated 1: TUNI1 interrupt request is generated
4	TUNI0R	0	R	TUNI0 Interrupt Request Indicates whether the TUNI0 (TMU0) interrupt request is generated. 0: TUNI0 interrupt request is not generated 1: TUNI0 interrupt request is generated
3	ITIR	0	R	ITI Interrupt Request Indicates whether the ITI (WDT) interrupt request is generated. 0: ITI interrupt request is not generated 1: ITI interrupt request is generated
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	RCMIR	0	R	RCMI Interrupt Request Indicates whether the RCMI (REF) interrupt request is generated. 0: RCMI interrupt request is not generated 1: RCMI interrupt request is generated

8.4.9 Interrupt Request Register 5 (IRR5)

IRR5 is an 8-bit register that indicates whether interrupt requests from the IPSEC, DMAC, and E-DMAC are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	IPSECIR	0	R	IPSECI Interrupt Request Indicates whether the IPSECI (IPSEC) interrupt request is generated. 0: IPSECI interrupt request is not generated 1: IPSECI interrupt request is generated
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	DEI5R	0	R	DEI5 Interrupt Request Indicates whether the DEI5 (DMAC) interrupt request is generated. 0: DEI5 interrupt request is not generated 1: DEI5 interrupt request is generated
4	DEI4R	0	R	DEI4 Interrupt Request Indicates whether the DEI4 (DMAC) interrupt request is generated. 0: DEI4 interrupt request is not generated 1: DEI4 interrupt request is generated
3	—	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
2	EINT2R	0	R	EINT2 Interrupt Request Indicates whether the EINT2 (E-DMAC) interrupt request is generated. 0: EINT2 interrupt request is not generated 1: EINT2 interrupt request is generated
1	EINT1R	0	R	EINT1 Interrupt Request Indicates whether the EINT1 (E-DMAC) interrupt request is generated. 0: EINT1 interrupt request is not generated 1: EINT1 interrupt request is generated
0	EINT0R	0	R	EINT0 Interrupt Request Indicates whether the EINT0 (E-DMAC) interrupt request is generated. 0: EINT0 interrupt request is not generated 1: EINT0 interrupt request is generated

8.4.10 Interrupt Request Register 7 (IRR7)

IRR7 is an 8-bit register that indicates whether an interrupt request from the SIOF0 is generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CCI0R	0	R	CCI0 Interrupt Request Indicates whether the CCI0 (SIOF0) interrupt request is generated. 0: CCI0 interrupt request is not generated 1: CCI0 interrupt request is generated
6	RXI0R	0	R	RXI0 Interrupt Request Indicates whether the RXI0 (SIOF0) interrupt request is generated. 0: RXI0 interrupt request is not generated 1: RXI0 interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
5	TXI0R	0	R	TXI0 Interrupt Request Indicates whether the TXI0 (SIOF0) interrupt request is generated. 0: TXI0 interrupt request is not generated 1: TXI0 interrupt request is generated
4	ERI0R	0	R	ERI0 Interrupt Request Indicates whether the ERI0 (SIOF0) interrupt request is generated. 0: ERI0 interrupt request is not generated 1: ERI0 interrupt request is generated
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.4.11 Interrupt Request Register 8 (IRR8)

IRR8 is an 8-bit register that indicates whether interrupt requests from the SIOF1 are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CCI1R	0	R	CCI1 Interrupt Request Indicates whether the CCI1 (SIOF1) interrupt request is generated. 0: CCI1 interrupt request is not generated 1: CCI1 interrupt request is generated
6	RXI1R	0	R	RXI1 Interrupt Request Indicates whether the RXI1 (SIOF1) interrupt request is generated. 0: RXI1 interrupt request is not generated 1: RXI1 interrupt request is generated

Bit	Bit Name	Initial Value	R/W	Description
5	TXI1R	0	R	TXI1 Interrupt Request Indicates whether the TXI1 (SIOF1) interrupt request is generated. 0: TXI1 interrupt request is not generated 1: TXI1 interrupt request is generated
4	ERI1R	0	R	ERI1 Interrupt Request Indicates whether the ERI1 (SIOF1) interrupt request is generated. 0: ERI1 interrupt request is not generated 1: ERI1 interrupt request is generated
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.5 Operation

8.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 8.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in the interrupt priority registers A to I (IPRA to IPRI). Lower priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected, according to tables 8.2 and 8.3, Interrupt Exception Handling Sources and Priority.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTEVT2).
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, respectively.
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). This jump is not a delayed branch. The interrupt handler may branch with INTEVT or INTEVT2 value as its offset in order to identify the interrupt source. This enables it to branch to the handling routine for the individual interrupt source.

Notes: 1. The interrupt mask bits (I3 to I0) in the status register (SR) are not changed by acceptance of an interrupt in this LSI.
2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then clear the BL bit or execute an RTE instruction.

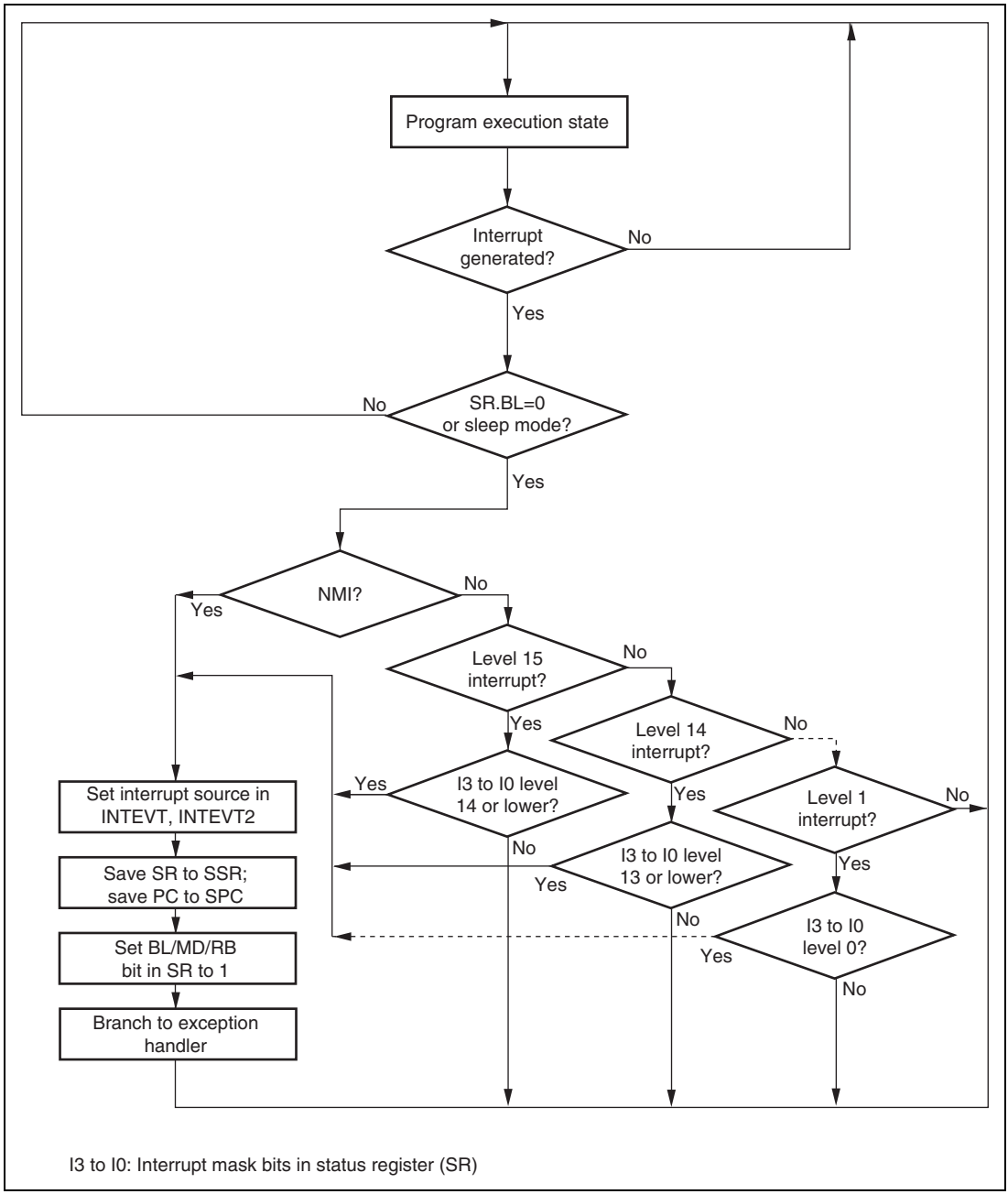


Figure 8.3 Interrupt Operation Flowchart

8.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handler should include the following procedures:

1. Branch to a specific interrupt handler corresponding to a code set in INTEVT or INTEVT2. The code in INTEVT or INTEVT2 can be used as an offset for branching to the specific handler.
2. Clear the interrupt source in each specific handler.
3. Save SSR and SPC to memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bits in SR.
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing the BL bit in step 4. See figure 8.3 on a sample interrupt operation flowchart.

Section 9 User Break Controller

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and stop timing in the case of instruction fetch.

9.1 Features

The UBC has the following features:

1. The following break comparison conditions can be set.

   Number of break channels: two channels (channels A and B)

   User break can be requested as either the independent or sequential condition on channels A and B (sequential break setting: channel A and then channel B match with break conditions, but not in the same bus cycle).

- Address

  Compares 40 bits configured of the ASID and addresses 32 bits: the ASID can be selected either all-bit comparison or all-bit mask. Comparison bits are maskable in 1-bit units; user can mask addresses at lower 12 bits (4-k page), lower 10 bits (1-k page), or any size of page, etc.

  One of the four address buses (logic address bus (LAB), internal address bus (IAB),

  X-memory address bus (XAB), and Y-memory address bus (YAB)) can be selected.

- Data

  Only on channel B, 32-bit maskable.

  One of the four data buses (L-bus data (LDB), I-bus data (IDB), X-memory data bus (XDB) and Y-memory data bus (YDB)) can be selected.

- Bus cycle

  Instruction fetch or data access

- Read/write
- Operand size

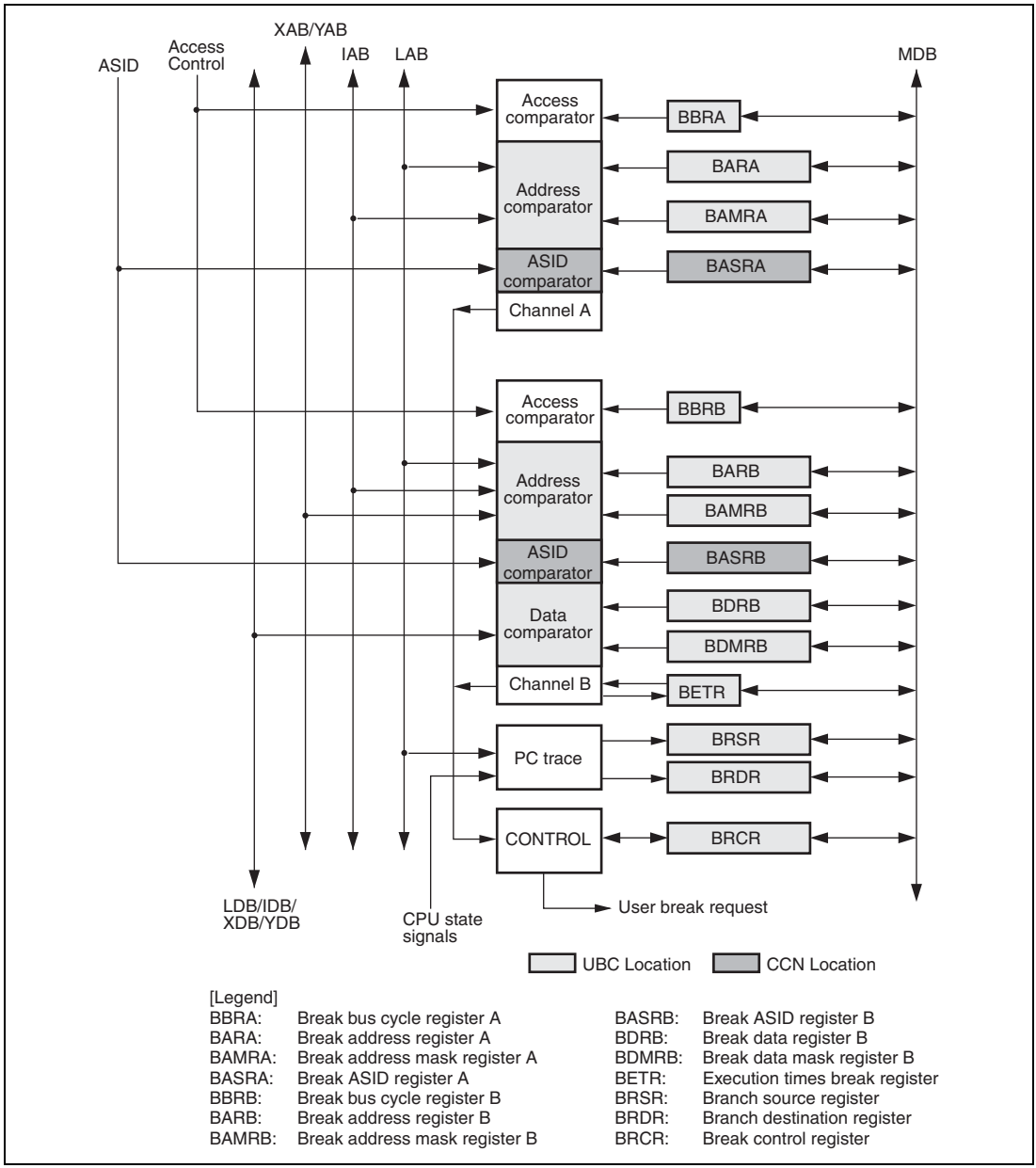
  Byte, word, and longword

2. A user-designed user-break condition exception processing routine can be run.

3. In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.

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

Figure 9.1 shows a block diagram of the UBC.



[Legend]

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

Figure 9.1 Block Diagram of User Break Controller

9.2 Register Descriptions

The user break controller has the following registers. For details on register addresses and access sizes, refer to section 24, List of Registers.

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

9.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used as a break condition in channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA 0	All 0	R/W	Break Address A Store the address on the LAB or IAB specifying break conditions of channel A.

9.2.2 Break Address Mask Register A (BAMRA)

BAMRA is a 32-bit readable/writable register. BAMRA specifies bits masked in the break address specified by BARA.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMA31 to BAMA 0	All 0	R/W	Break Address Mask A Specify bits masked in the channel A break address bits specified by BARA (BAA31–BAA0). 0: Break address bit BAAn of channel A is included in the break condition 1: Break address bit BAAn of channel A is masked and is not included in the break condition Note: n = 31 to 0

9.2.3 Break Bus Cycle Register A (BBRA)

BBRA is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of channel A.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CDA1	0	R/W	L Bus Cycle/I Bus Cycle Select A
6	CDA0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDA1	0	R/W	Instruction Fetch/Data Access Select A
4	IDA0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for the channel A break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

9.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used as a break condition in channel B. Control bits CDB1, CDB0, XYE, and XYS in BBRB select one of the four address buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB 0	All 0	R/W	Break Address B Stores an address which specifies a break condition in channel B. If the I bus or L bus is selected in BBRB, an IAB or LAB address is set in BAB31 to BAB0. If the X memory is selected in BBRB, the values in bits 15 to 1 in XAB are set in BAB31 to BAB17. In this case, the values in BAB16 to BAB0 are arbitrary. If the Y memory is selected in BBRB, the values in bits 15 to 1 in YAB are set in BAB15 to BAB1. In this case, the values in BAB31 to BAB16, and BABO are arbitrary.

Table 9.1 Specifying Break Address Register

Bus Selection in BBRB	BAB31 to BAB17	BAB16	BAB15 to BAB1	BAB0
L bus	LAB31 to LAB0			
I bus	IAB31 to IAB0			
X bus	XAB15 to XAB1	Don't care	Don't care	Don't care
Y bus	Don't care	Don't care	YAB15 to YAB1	Don't care

9.2.5 Break Address Mask Register B (BAMRB)

BAMRB is a 32-bit readable/writable register. BAMRB specifies bits masked in the break address specified by BARB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMB31 to BAMB 0	All 0	R/W	Break Address Mask B Specifies bits masked in the break address of channel B specified by BARB (BAB31 to BAB0). 0: Break address BABn of channel B is included in the break condition 1: Break address BABn of channel B is masked and is not included in the break condition Note: n = 31 to 0

9.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. The control bits CDB1, CDB0, XYE, and XYS in BBRB select one of the four data buses for break condition B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	All 0	R/W	Break Data Bit B Stores data which specifies a break condition in channel B. If the I bus is selected in BBRB, the break data on IDB is set in BDB31 to BDB0. If the L bus is selected in BBRB, the break data on LDB is set in BDB31 to BDB0. If the X memory is selected in BBRB, the break data in bits 15 to 0 in XDB is set in BDB31 to BDB16. In this case, the values in BDB15 to BDB0 are arbitrary. If the Y memory is selected in BBRB, the break data in bits 15 to 0 in YDB are set in BDB15 to BDB0. In this case, the values in BDB31 to BDB16 are arbitrary.

Table 9.2 Specifying Break Data Register

Bus Selection in BBRB	BDB31 to BDB16	BDB15 to BDB0
L bus	LDB31 to LDB0	
I bus	IDB31 to IDB0	
X bus	XDB15 to XDB0	Don't care
Y bus	Don't care	YDB15 to YDB0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.
3. Set the data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As, Ds, MOVS.W @As, Ds, MOVS.W @As+, Ds, or MOVS.W @As+Ix, Ds instruction.

9.2.7 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit readable/writable register. BDMRB specifies bits masked in the break data specified by BDRB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB 0	All 0	R/W	Break Data Mask B Specifies bits masked in the break data of channel B specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is included in the break condition 1: Break data BDBn of channel B is masked and is not included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break mask data in BDMRB.
3. Set the mask data in bits 31 to 16 when including the value of the data bus as an L-bus break condition for the MOVS.W @-As, Ds, MOVS.W @As, Ds, MOVS.W @As+, Ds, or MOVS.W @As+Ix, Ds instruction.

9.2.8 Break Bus Cycle Register B (BBRB)

BBRB is a 16-bit readable/writable register, which specifies (1) X bus or Y bus, (2) L bus cycle or I bus cycle, (3) instruction fetch or data access, (4) read or write, and (5) operand size in the break conditions of channel B.

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	XYE	0	R/W	Selects the X memory bus or Y memory bus as the channel B break condition. Note that this bit setting is enabled only when the L bus is selected with the CDB1 and CDB0 bits. Selection between the X memory bus and Y memory bus is done by the XYS bit. 0: Selects L bus for the channel B break condition 1: Selects X/Y memory bus for the channel B break condition
8	XYS	0	R/W	Selects the X bus or the Y bus as the bus of the channel B break condition. 0: Selects the X bus for the channel B break condition 1: Selects the Y bus for the channel B break condition
7	CDB1	0	R/W	L Bus Cycle/I Bus Cycle Select B
6	CDB0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the channel B break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

9.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Channels A and B are used in two independent channel conditions or under the sequential condition.
2. A break is set before or after instruction execution.
3. Specify whether to include the number of execution times on channel B in comparison conditions.
4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable ASID check.

BRCR is a 32-bit readable/writable register that has break conditions match flags and bits for setting a variety of break conditions.

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	BASMA	0	R/W	Break ASID Mask A Specifies whether bits in channel A break ASID7 to ASID0 (BASA7 to BASA0) which are set in BASRA are masked or not. 0: All BASRA bits are included in the break conditions and the ASID is checked 1: All BASRA bits are not included in the break conditions and the ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether bits in channel B break ASID7 to ASID0 (BASB7 to BASB0) which are set in BASRB are masked or not. 0: All BASRB bits are included in the break conditions and the ASID is checked 1: All BASRB bits are not included in the break conditions and the ASID is not checked

Bit	Bit Name	Initial Value	R/W	Description
19 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	SCMFCA	0	R/W	L Bus Cycle Condition Match Flag A When the L bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel A does not match 1: The L bus cycle condition for channel A matches
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel B does not match 1: The L bus cycle condition for channel B matches
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does not match 1: The I bus cycle condition for channel A matches
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does not match 1: The I bus cycle condition for channel B matches

Bit	Bit Name	Initial Value	R/W	Description
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9	—	0	R	Reserved
8	—	0	R	These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution. 0: PC break of channel B is set before instruction execution 1: PC break of channel B is set after instruction execution
5	—	0	R	Reserved
4	—	0	R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential conditions. 0: Channels A and B are compared under independent conditions 1: Channels A and B are compared under sequential conditions (channel A, then channel B)
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The write value should always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by BETR. 0: The execution-times break condition is disabled on channel B 1: The execution-times break condition is enabled on channel B

9.2.10 Execution Times Break Register (BETR)

BETR is a 16-bit readable/writable register. When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. When a break condition is satisfied, it decreases BETR. A break is issued when the break condition is satisfied after BETR becomes H'0001.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	BET11 to BET0	All 0	R/W	Number of Execution Times

Note: When the instruction fetch cycle is specified as the break condition of the channel B, and its break condition is triggered by the following instructions, the BETR is decremented by the following value (not by one).

Instruction	Decrement value	Instruction	Decrement value
RTE	4	LDC.L @Rm+,SR	6
DMULS.L Rm,Rn	2	LDC.L @Rm+,GBR	4
DMULU.L Rm,Rn	2	LDC.L @Rm+,VBR	4
MAC.L @Rm+,@Rn	2	LDC.L @Rm+,SSR	4
MAC.W @Rm+,@Rn	2	LDC.L @Rm+,SPC	4
MUL.L Rm,Rn	3	LDC.L @Rm+,R0_BANK	4
AND.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R1_BANK	4
OR.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R2_BANK	4
TAS.B @Rn	3	LDC.L @Rm+,R3_BANK	4
TST.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R4_BANK	4
XOR.B #imm,@(R0,GBR)	3	LDC.L @Rm+,R5_BANK	4
LDC Rm,SR	4	LDC.L @Rm+,R6_BANK	4
LDC Rm,GBR	4	LDC.L @Rm+,R7_BANK	4
LDC Rm,VBR	4	LDC.L @Rn+,MOD	4
LDC Rm,SSR	4	LDC.L @Rn+,RS	4
LDC Rm,SPC	4	LDC.L @Rn+,RE	4
LDC Rm,R0_BANK	4	LDC Rn,MOD	4
LDC Rm,R1_BANK	4	LDC Rn,RS	4
LDC Rm,R2_BANK	4	LDC Rn,RE	4
LDC Rm,R3_BANK	4	BSR label	4
LDC Rm,R4_BANK	4	BSRF Rm	2
LDC Rm,R5_BANK	4	JSR @Rm	2
LDC Rm,R6_BANK	4		
LDC Rm,R7_BANK	4		

9.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch source instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRSR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the branch source address is stored. When a branch source address is fetched, this flag is set to 1. This flag is cleared to 0 by reading from BRSR. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BSA27 to BSA0	—	R	Branch Source Address Store bits 27 to 0 of the branch source address.

9.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the branch destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The eight BRDR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	DVF	0	R	BRDR Valid Flag Indicates whether a branch destination address is stored. When a branch destination address is fetched, this flag is set to 1. This flag is cleared to 0 by reading BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BDA27 to BDA0	—	R	Branch Destination Address Store bits 27 to 0 of the branch destination address.

9.2.13 Break ASID Register A (BASRA)

BASRA is an 8-bit readable/writable register that specifies ASID which becomes the break condition for channel A. BASRA is in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	—	R/W	Break ASID A Store ASID (bits 7 to 0) which is the break condition for channel A.

9.2.14 Break ASID Register B (BASRB)

BASRB is an 8-bit readable/writable register that specifies ASID which becomes the break condition for channel B. BASRB is in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASB7 to BASB0	—	R/W	Break ASID B Store ASID (bits 7 to 0) which is the break condition for channel B.

9.3 Operation

9.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is described below:

1. The break addresses and corresponding ASID are set in the break address registers (BARA or BARB) and break ASID registers (BASRA or BASRB in CCN). The masked addresses are set in the break address mask registers (BAMRA or BAMRB). The break data is set in the break data register (BDRB). The masked data is set in the break data mask register (BDMRB). The bus break conditions are set in the break bus cycle registers (BBRA or BBRB). Three groups of BBRA or BBRB (L bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set with 00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA or BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition match flag (SCMFDA or SCMFDB) for the appropriate channel. When the X/Y memory bus is specified for channel B, SCMFCB is used for the condition match flag.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, but they are not reset. 0 must first be written to them before they can be used again.
4. There is a chance that the break set in channel A and the break set in channel B occur around the same time. In this case, there will be only one break request to the CPU, but these two break channel match flags could be both set.
5. When selecting the I bus as the break condition, note the following:

— Several bus masters, including the CPU, DMAC and E-DMAC, are connected to the I bus. The UBC monitors bus cycles generated by all bus masters, and determines the condition match.

— Physical addresses are used for the I bus. Set a physical address in break address registers (BARA and BARB). The bus cycles for logical addresses issued on the L bus by the CPU are converted to physical addresses before being output to the I bus. (If the address translation function is enabled, address translation by the MMU is carried out.)

— For data access cycles issued on the L bus by the CPU, if their logical addresses are not to be cached, they are issued with the data size specified on the L bus and their addresses are not rounded.

— For instruction fetch cycles issued on the L bus by the CPU, even though their logical addresses are not to be cached, they are issued in longwords and their addresses are rounded to match longword boundaries.

— If a logical address issued on the L bus by the CPU is an address to be cached and a cache miss occurs, its bus cycle is issued as a cache fill cycle on the I bus. In this case, it is issued in longwords and its address is rounded to match longword boundaries. However note that cache fill is not performed for a write miss in write through mode. In this case, the bus cycle is issued with the data size specified on the L bus and its address is not rounded. In write back mode, a write back cycle may be issued in addition to a read fill cycle. It is a longword bus cycle whose address is rounded to match longword boundaries.

— I bus cycles (including read fill cycles) resulting from instruction fetches on the L bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.

— The DMAC and E-DMAC only issues data access cycles for I bus cycles.

— If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

6. While the block bit (BL) in the CPU status register (SR) is set to 1, no breaks can be accepted. However, condition determination will be carried out, and if the condition matches, the corresponding condition match flag is set to 1.

9.3.2 Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBRA or BBRB), the break condition becomes the L bus instruction fetch cycle. Whether it breaks before or after the execution of the instruction can then be selected with the PCBA or PCBB bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BARA or BARB) to 0. A break cannot be generated as long as this bit is set to 1.
2. An instruction set for a break before execution breaks when it is confirmed that the instruction has been fetched and will be executed. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the break is generated prior to execution of the delayed branch instruction.

   Note: If a branch does not occur at a delay condition branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When the condition is specified to be occurred after execution, the instruction set with the break condition is executed and then the break is generated prior to the execution of the next instruction. As with pre-execution breaks, this cannot be used with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, a break is not generated until the first instruction at the branch destination.
4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB) is ignored. Therefore, break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the condition is determined for the instruction fetch cycles on the I bus. For details, see 5 in section 9.3.1, Flow of the User Break Operation.

9.3.3 Break on Data Access Cycle

1. If the L bus is specified as a break condition for data access break, condition comparison is performed for the logical addresses (and data) accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical addresses (and data) of the data access cycles that are issued on the I bus by all bus masters including the CPU, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 5 in section 9.3.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 9.3.

Table 9.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

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

This means that when address H'00001003 is set in the break address register (BARA or BARB), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on channel B:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size of the break bus cycle register B (BBRB). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in two bytes at bits 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask register B (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ignored. Set the word data in bits 31 to 16 in BDRB and BDMRB when including the value of the data bus as a break condition for the MOVS.W @-As, Ds, MOVS.W @As, Ds, MOVS.W @As+, Ds, or MOVS.W @As+Ix, Ds instruction (bits 15 to 0 are ignored).

4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if including the value of the data bus when a PREF instruction is specified as a break condition, a break will not occur.

5. If the L bus is selected, a break occurs on ending execution of the instruction that matches the break condition, and immediately before the next instruction is executed. However, when data is also specified as the break condition, the break may occur on ending execution of the instruction following the instruction that matches the break condition. If the I bus is selected, the instruction at which the break will occur cannot be determined. When this kind of break occurs at a delayed branch instruction or its delay slot, the break may not actually take place until the first instruction at the branch destination.

9.3.4 Break on X/Y-Memory Bus Cycle

1. The break condition on an X/Y-memory bus cycle is specified only in channel B. If the XYE bit in BBRB is set to 1, the break address and break data on X/Y-memory bus are selected. At this time, select the X-memory bus or Y-memory bus by specifying the XYS bit in BBRB. The break condition cannot include both X-memory and Y-memory at the same time. The break condition is applied to an X/Y-memory bus cycle by specifying L bus/data access/read or write/word or no specified operand size in bits 7 to 0 in the break bus cycle register B (BBRB).
2. When an X-memory address is selected as the break condition, specify an X-memory address in the upper 16 bits in BARB and BAMRB. When a Y-memory address is selected, specify a Y-memory address in the lower 16 bits. Specification of X/Y-memory data is the same for BDRB and BDMRB.
3. The timing of a data access break for the X memory or Y memory bus to occur is the same as a data access break of the L bus. For details, see 5 in section 9.3.3, Break on Data Access Cycle.

9.3.5 Sequential Break

1. By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel B break condition matches after a channel A break condition matches. A user break is not generated even if a channel B break condition matches before a channel A break condition matches. When channels A and B conditions match at the same time, the sequential break is not issued. To clear the channel A condition match when a channel A condition match has occurred but a channel B condition match has not yet occurred in a sequential break specification, clear the SEQ bit in BRCR to 0.
2. In sequential break specification, the L/I/X/Y bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

9.3.6 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resumed is saved in the SPC, and the exception handling state is entered. If the L bus is specified as a break condition, the instruction at which the break should occur can be clearly determined (except for when data is included in the break condition). If the I bus is specified as a break condition, the instruction at which the break should occur cannot be clearly determined.

1. When instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved in the SPC. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the address of the delayed branch instruction is saved in the SPC.
2. When instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, these instructions are executed, and the branch destination address is saved in the SPC.
3. When data access (address only) is specified as a break condition:

   The address of the instruction immediately after the instruction that matched the break condition is saved in the SPC. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delay slot instruction matches the condition, the branch destination address is saved in the SPC.
4. When data access (address + data) is specified as a break condition:

   When a data value is added to the break conditions, the address of an instruction that is within two instructions of the instruction that matched the break condition is saved in the SPC. At which instruction the break occurs cannot be determined accurately.

   When a delay slot instruction matches the condition, the branch destination address is saved in the SPC. If the instruction following the instruction that matches the break condition is a branch instruction, the break may occur after the branch instruction or delay slot has finished. In this case, the branch destination address is saved in the SPC.

9.3.7 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt exception) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
2. The values stored in BRSR and BRDR are as given below due to the kind of branch.
   - If a branch occurs due to a branch instruction, the address of the branch instruction is saved in BRSR and the address of the branch destination instruction is saved in BRDR.
   - If a branch occurs due to an interrupt or exception, the value saved in SPC due to exception occurrence is saved in BRSR and the start address of the exception handling routine is saved in BRDR.

   When a repeat loop of the DSP extended function is used, control being transferred from the repeat end instruction to the repeat start instruction is not recognized as a branch, and the values are not stored in BRSR and BRDR.
3. BRSR and BRDR have eight pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid.

9.3.8 Usage Examples

Break Condition Specified for L Bus Instruction Fetch Cycle:

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300400

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  The ASID check is not included.

  <Channel B>

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

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

Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

The ASID check is not included.

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

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

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

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

  <Channel B>

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

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

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

  After an instruction with ASID = H’80 and address H'00037226 is executed, a user break occurs before an instruction with ASID = H’70 and address H'0003722E is executed.

- Register specifications

  BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300000

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00027128, Address mask: H'00000000

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

  The ASID check is not included.

  <Channel B>

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

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

  The ASID check is not included.

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

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

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B sequential mode

  <Channel A>

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

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

  <Channel B>

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

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

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

  Since instruction fetch is not a write cycle on channel A, a sequential condition does not match. Therefore, no user break occurs.

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00000500, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The ASID check is not included.

  <Channel B>

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

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

  The number of execution-times break enable (5 times)

  The ASID check is not included.

  On channel A, a user break occurs before an instruction of address H'00000500 is executed. On channel B, a user break occurs after the instruction of address H'00001000 are executed four times and before the fifth time.

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel B>

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

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction with ASID = H'80 and addresses H'00008000 to H'00008FFE is executed or before an instruction with ASID = H’70 and addresses H'00008010 to H'00008016 are executed.

Break Condition Specified for L Bus Data Access Cycle:

- Register specifications

  BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'00123456, Address mask: H'00000000, ASID = H'80

  Bus cycle: L bus/data access/read (operand size is not included in the condition)

  <Channel B>

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

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

  Bus cycle: L bus/data access/write/word

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

- Register specifications

  BARA = H'01000000, BAMRA = H'00000000, BBRA = H'0066, BARB = H'0000F000, BAMRB = H'FFFF0000, BBRB = H'036A, BDRB = H'00004567, BDMRB = H'00000000, BRCR = H'00300080

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

  Address: H'01000000, Address mask: H'00000000

  Bus cycle: L bus/data access/read/word

  The ASID check is not included.

  <Channel B>

  Y Address: H'0000F000, Address mask: H'FFFF0000

  Data: H'00004567, Data mask: H'00000000

  Bus cycle: Y bus/data access/write/word

  The ASID check is not included.

  On channel A, a user break occurs during word read from address H'01000000 in the memory space. On channel B, a user break occurs when word data H'4567 is written in address H'0000F000 in the Y memory space. The X/Y-memory space is changed by a mode setting.

Break Condition Specified for I Bus Data Access Cycle:

- Register specifications

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

  Specified conditions: Channel A/channel B independent mode

  <Channel A>

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

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)

  <Channel B>

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

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

  Bus cycle: I bus/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for ASID = H’80 and address H'00314156 in the memory space.

  On channel B, a user break occurs when ASID = H’70 and byte data H'7* is written in address H'00055555 on the I bus.

9.4 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. UBC cannot monitor access to the L bus and I bus in the same channel.
3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no break occurs even if a bus cycle, in which an A-channel match and a channel B match occur simultaneously, is set.
4. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 4.1 in section 4, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re-execution-type exception (including pre-execution break) that has higher priority, the re-execution-type exception is accepted, and the condition match flag is not set (see the exception in the following note). The break will occur and the condition match flag will be set only after the exception source of the re-execution-type exception has been cleared by the exception handling routine and re-execution of the same instruction has ended.
   - When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, though a break does not occur, the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that generates a CPU address error (or TLB related exception) by data access, the CPU address error (or TLB related exception) is given priority to the break. Note that the UBC condition match flag is set in this case.

6. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, the break does not occur until the branch destination of the RTE instruction.

7. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.

8. When the repeat loop of the DSP extended function is used, even though a break condition is satisfied during execution of the entire repeat loop or several instructions in the repeat loop, the break may be held. For details, see section 4, Exception Handling.

Section 10 Power-Down Modes

With the power-down modes, the operation of the CPU and same of on-chip peripheral modules are halted to reduce power consumption. The power-down modes are canceled by interrupts or a reset.

10.1 Overview

10.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Module standby function

Table 10.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

Table 10.1 States of Power-Down Modes

		State							
Mode	Transition Conditions	CPG EtherC E-DMAC	CPU	CPU Reg-ister	On-Chip Memory	On-Chip Peripheral Modules	Pins	External Memory	Canceling Procedure
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Run	Halt	Held	Held	Run	Held	Refreshed	1. Interrupt 2. Reset
Software Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halt	Halt	Held	Held	Halt*[1]	Held	Self-refreshed	1. Interrupt 2. Reset
Module standby function	Set MSTP bit to 1 in STBCR, STBCR2, and STBCR3	Run	Run	Held	Held	Specified module halts	*[2]	Refreshed	1. Clear MSTP bit to 0 2. Power-on reset

Notes: 1. The RTC runs when the START bit in RCR2 is set to 1. For details, see section 15, Realtime Clock (RTC).
2. Depends on the on-chip peripheral modules. For details, see section 1, Overview and Pin Function.

10.1.2 Reset

A reset is used at power-on or to re-execute from the initial state. This LSI supports two types of reset: power-on reset and manual reset. In power-on reset, any processing to be currently executed is terminated and any events not executed are canceled to execute reset processing immediately. In manual reset, processing required to maintain external memory contents is continued. The following shows the conditions in which power-on reset or manual reset occurs.

- Power-on reset
  1. A low level signal is input to the $\overline{\text{RESETP}}$ pin.
  2. The WDT counter overflows if the WDT starts counting while the WT/$\overline{\text{IT}}$ and RSTS bits in WTCSR are set to 1 and cleared to 0, respectively.
  3. An H-UDI reset occurs. (For details on the H-UDI reset, refer to section 23, User Debugging Interface (H-UDI).)

- Manual reset
  1. A low signal is input to the $\overline{\text{RESETM}}$ pin.
  2. The WDT counter overflows if WDT starts counting while the WT/$\overline{\text{IT}}$ and RSTS bits of the WTCSR are set to 1.

Note: Immediately after a power-on reset or manual reset, be sure to execute the following routine:

```
    MOV.L   #H'FFFFFF40, R1
    MOV.L   #H'80000005, R0
    MOV.L   #H'A4FC0008, R2
    NOP
    NOP
TESTCR2_SET
    NOP
    MOV.B   R0, @R1
    MOV.B   R0, @R1
    MOV.L   R0, @R2
    NOP
    NOP
    NOP
    MOV.L   @R2, R3
    CMP/EQ  R3, R0
    BF      TESTCR2_SET
    NOP
    NOP
```

10.1.3 Input/Output Pins

Table 10.2 lists the pins used for the power-down modes.

Table 10.2 Pin Configuration

Pin Name	Symbol	I/O	Description
Processing state 1	STATUS1	O	Indicates the operating state of the processor.
Processing state 0	STATUS0		HH: Reset
			HL: Sleep mode
			LH: Standby mode
			LL: Normal operation
Power-on reset	$\overline{\text{RESETP}}$	I	Inputting low level signal to this pin cause a transition to power-on reset processing.
Manual reset	$\overline{\text{RESETM}}$	I	Inputting low level signal to this pin cause a transition to manual reset processing.

Note: H and L indicate high and low levels, respectively. The STATUS1 and STATUS0 pins indicate the pin status in this order.

10.2 Register Descriptions

The following registers are used for the power-down modes. Refer to section 24, List of Registers, for the addresses and access size for these registers.

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

10.2.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down mode. This register is initialized to H′00 at power-on reset but retains the previous value after manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction puts chip into sleep mode 1: Executing SLEEP instruction puts chip into software standby mode
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MSTP2	0	R/W	Module Stop Bit 2 When the MSTP2 bit is set to 1, the supply of the clock to the TMU is halted. 0: TMU runs 1: Clock supply to TMU halted
1	MSTP1	0	R/W	Module Stop Bit 1 When the MSTP1 bit is set to 1, the supply of the clock to the RTC is halted. 0: RTC runs 1: Clock supply to RTC halted
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

10.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in the power-down mode. This register is initialized to H′00 at power-on reset but retains the previous value after manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop Bit 10 When the MSTP10 bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI runs 1: Clock supply to H-UDI halted
6	MSTP9	0	R/W	Module Stop Bit 9 When the MSTP9 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC runs 1: Clock supply to UBC halted
5	MSTP8	0	R/W	Module Stop Bit 8 When the MSTP8 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC runs 1: Clock supply to DMAC halted
4	MSTP7	0	R/W	Module Stop Bit 7 When the MSTP7 bit is set to 1, the supply of the clock to the DSP is halted. 0: DSP runs 1: Clock supply to DSP halted
3	MSTP6	0	R/W	Module Stop Bit 6 When the MSTP6 bit is set to 1, the supply of the clock to the TLB is halted. 0: TLB runs 1: Clock supply to TLB halted

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP5	0	R/W	Module Stop Bit 5 When the MSTP5 bit is set to 1, the supply of the clock to the cache memory is halted. 0: The cache memory runs 1: Clock supply to the cache memory halted
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	MSTP3	0	R/W	Module Stop Bit 3 When the MSTP3 bit is set to 1, the supply of the clock to the X/Y memory is halted. 0: The X/Y memory runs 1: Clock supply to the X/Y memory halted

10.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of the peripheral modules in the power-down mode. This register is initialized to H′00 at power-on reset but retains the previous value after manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP37	0	R/W	Module Stop Bit 37 When the MSTP37 bit is set to 1, the supply of the clock to the IPSEC is halted. 0: IPSEC runs 1: Clock supply to IPSEC halted
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MSTP33	0	R/W	Module Stop Bit 33 When the MSTP33 bit is set to 1, the supply of the clock to the SIOF1 is halted. 0: SIOF1 runs 1: Clock supply to SIOF1 halted

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP32	0	R/W	Module Stop Bit 32 When the MSTP32 bit is set to 1, the supply of the clock to the SIOF0 is halted. 0: SIOF0 runs 1: Clock supply to SIOF0 halted
1	MSTP31	0	R/W	Module Stop Bit 31 When the MSTP31 bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: The SCIF1 runs 1: Clock supply to the SCIF1 halted
0	MSTP30	0	R/W	Module Stop Bit 30 When the MSTP30 bit is set to 1, the supply of the clock to the SCIF0 is halted. 0: The SCIF0 runs 1: Clock supply to the SCIF0 halted

10.3 Operation

10.3.1 Sleep Mode

Transition to Sleep Mode: Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run in sleep mode and the clock continues to be output to the CKIO pin. In sleep mode, a high signal and low signal are output from the STATUS1 and STATUS0 pins, respectively.

Canceling Sleep Mode: Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, or on-chip peripheral module) or reset. Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

- Canceling with an Interrupt

  When an NMI, IRQ, IRL, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in INTEVT and INTEVT2.

- Canceling with a Reset
  Sleep mode is canceled by a power-on reset or a manual reset.

10.3.2 Software Standby Mode

Transition to Software Standby Mode: The LSI switches from a program execution state to a software standby mode by executing the SLEEP instruction when the STBY bit is 1 in STBCR. In a software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CKIO pin also halts.

The contents of the CPU and cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 10.3 lists the states of on-chip peripheral modules registers in software standby mode.

Table 10.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt Controller (INTC)	—	All registers
On-Chip Oscillation Circuits	—	All registers
User Break Controller (UBC)	—	All registers
Bus State Controller (BSC)	—	All registers
Timer Unit (TMU)	TSTR	Registers other than TSTR
IPSEC	—	All registers
I/O ports	—	All registers
H-UDI	—	All registers
SCIF0/1	—	All registers
SIOF0/1	—	All registers
EtherC, E-DMAC	—	All registers
DMAC	—	All registers

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
2. Set the WDT's timer counter (WTCNT) to 0 and the CKS2 to CKS0 bits in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After the STBY bit in STBCR is set to 1, a SLEEP instruction is executed.
4. Software standby mode is entered and the clocks within the chip are halted. The STATUS1 and STATUS0 pins output low and high, respectively.

Canceling Software Standby Mode: Software standby mode is canceled by an interrupt (NMI, IRQ, IRL, or RTC) or a reset.

- Canceling with an Interrupt

The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRQ*[1], IRL*[1], or RTC*[1] interrupt, the clock will be supplied to the entire chip and software standby mode canceled after the time set in the WDT's timer control/status register has elapsed. The STATUS1 and STATUS0 pins go low. Interrupt exception handling then begins and a code indicating the interrupt source is set in INTEVT and INTEVT2. After the branch to the interrupt handling routine, clear the STBY bit in STBCR. The WDT stops automatically. If the STBY bit is not cleared, the WDT continues operation and a transition is made to software standby mode*[2] when WTCNT reaches H'80. A manual reset is not accepted until the STBY bit is cleared to 0.

Interrupts are accepted in software standby mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

Immediately after an interrupt is detected, the phase of the CKIO pin clock output may be unstable, until the software standby mode is canceled.

Notes: 1. Only when the RTC is used, software standby mode can be canceled by an IRQ, IRL, or RTC.
2. This standby mode can be canceled only by a power-on reset.

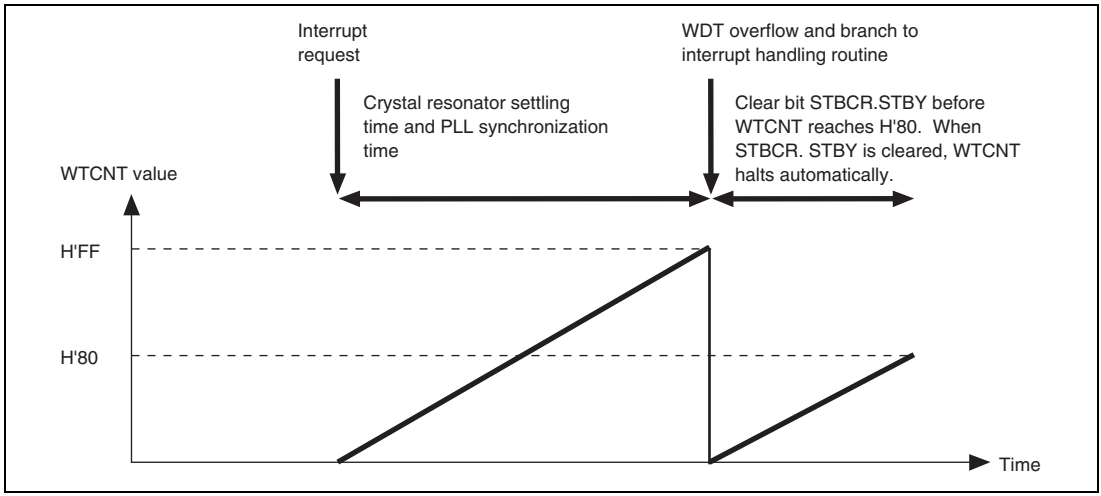


Figure 10.1 Canceling Standby Mode with STBCR.STBY

- Canceling with a Reset

Software standby mode is canceled by a reset (power-on or manual). Keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

10.3.3 Module Standby Function

Transition to Module Standby Function: Setting each MSTP bit in the standby control registers to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal or sleep mode. Before a transition is made, the module should be disabled.

In the module standby state, the functions of the external pins of the on-chip peripheral modules change depending on the on-chip peripheral module. For details, see section 1, Overview and Pin Function. All of the register states are the same as those in standby mode. For details, see table 10.3.

Canceling Module Standby Function: The module standby function can be canceled by clearing the MSTP bits to 0, or by a power-on reset.

To cancel the module standby function by clearing the corresponding MSTP bit to 0, read the MSTP bit to check the MSTP bit was cleared correctly.

10.3.4 STATUS Pin Change Timings

The STATUS1 and STATUS0 pin change timings are shown below.

Reset:

- Power-on reset

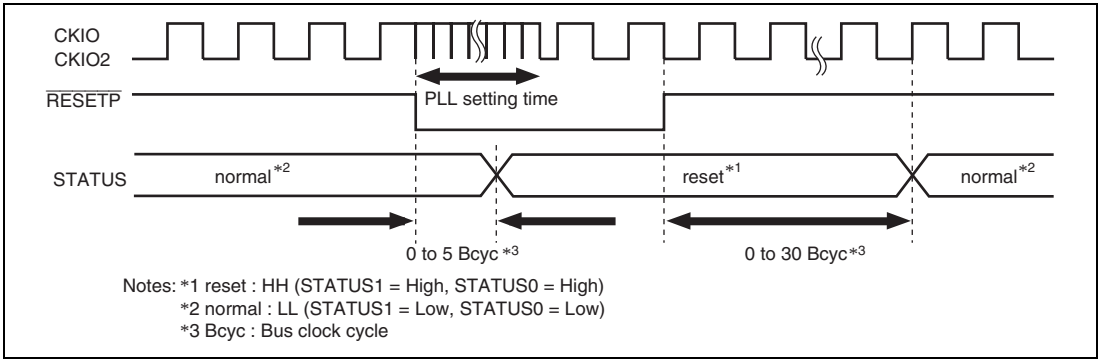


Figure 10.2 STATUS Output at Power-On Reset

- Manual reset

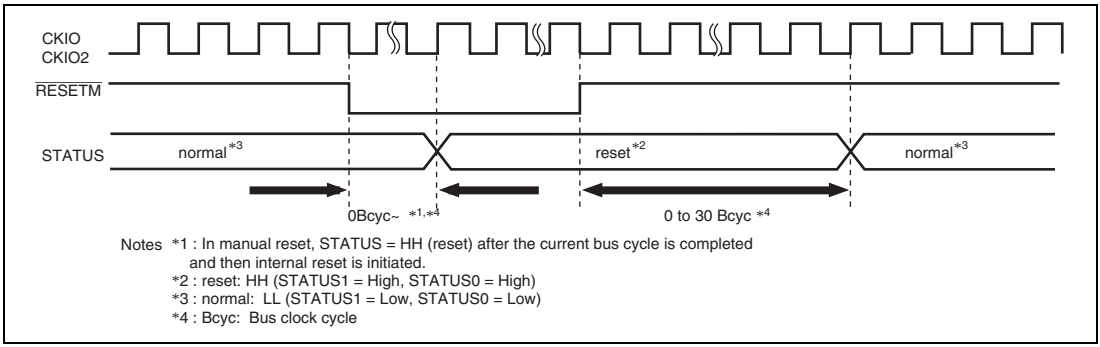


Figure 10.3 STATUS Output at Manual Reset

Software Standby Mode:

- Software standby mode is canceled by an interrupt

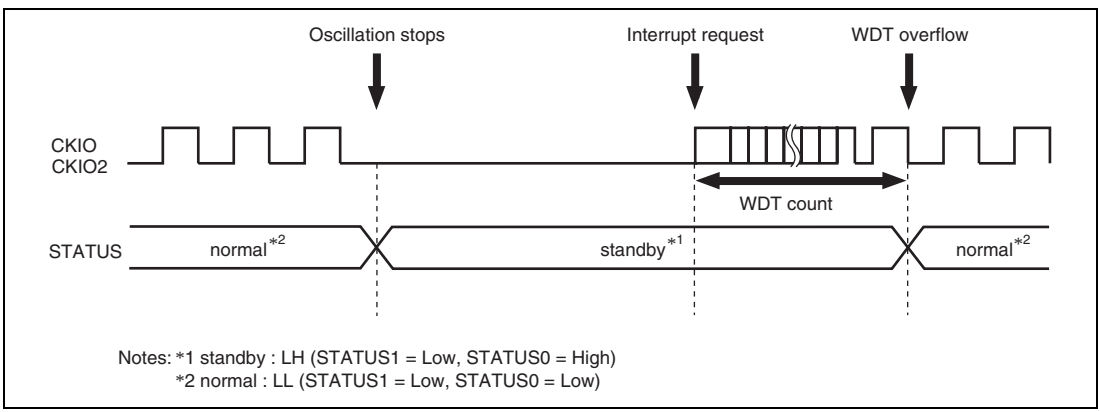
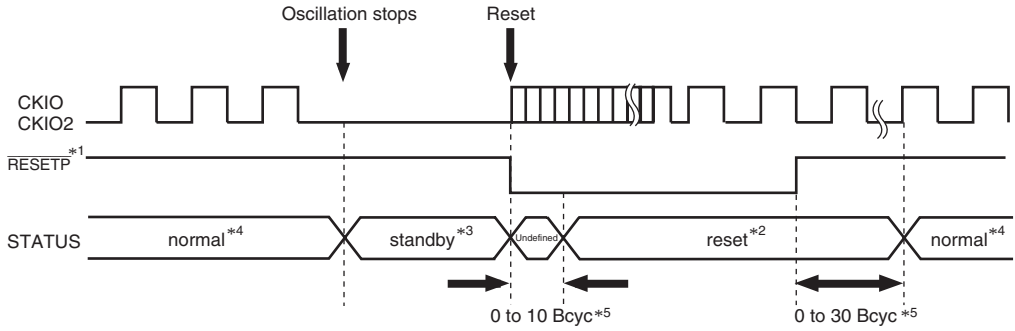


Figure 10.4 STATUS Output when Software Standby Mode is Canceled by Interrupt

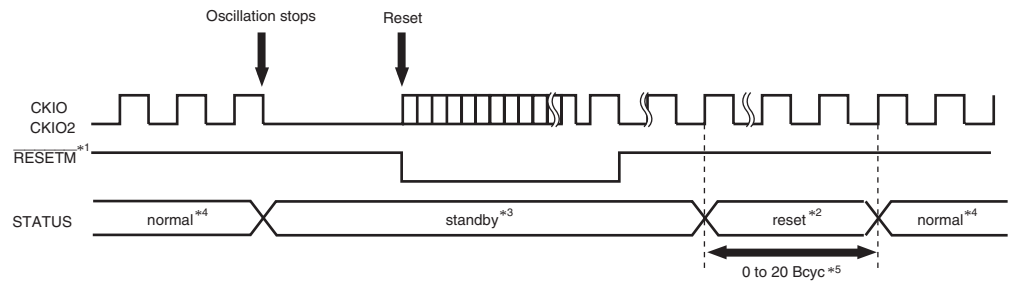
- Software standby mode is canceled by a power-on reset



Notes: *1 If a standby mode is canceled by a power on reset, the WDT stops counting. RESETP must be kept low for the PLL oscillation stabilization time.
*2 reset : HH (STATUS1 = High, STATUS0 = High)
*3 standby : LH (STATUS1 = Low, STATUS0 = High)
*4 normal : LL (STATUS1 = Low, STATUS0 = Low)
*5 Bcyc : Bus clock cycle

Figure 10.5 STATUS Output when Software Standby Mode is Canceled by Power-on Reset

- Software standby mode is canceled by a manual reset



Notes: *1 If a standby mode is canceled by a power on reset, the WDT stops counting. RESETM must be kept low for the PLL oscillation stabilization time.
*2 reset : HH (STATUS1 = High, STATUS0 = High)
*3 standby : LH (STATUS1 = Low, STATUS0 = High)
*4 normal : LL (STATUS1 = Low, STATUS0 = Low)
*5 Bcyc : Bus clock cycle

Figure 10.6 STATUS Output when Software Standby Mode is Canceled by Manual Reset

Sleep Mode:

- Sleep mode is canceled by an interrupt

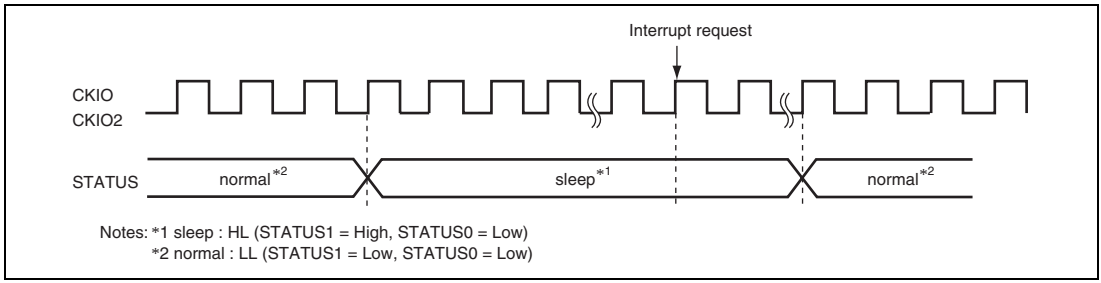


Figure 10.7 STATUS Output when Sleep Mode is Canceled by Interrupt

- Sleep mode is canceled by a power-on reset

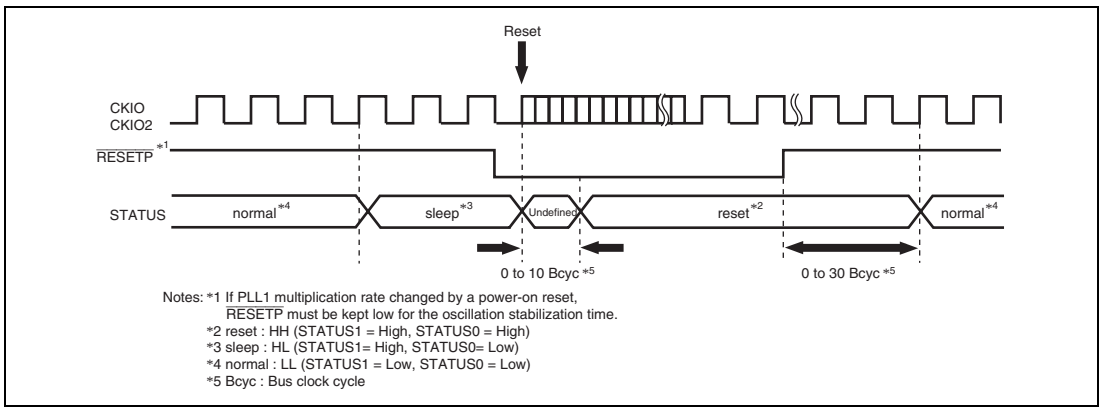


Figure 10.8 STATUS Output when Sleep Mode is Canceled by Power-on Reset

- Sleep mode is canceled by a manual reset

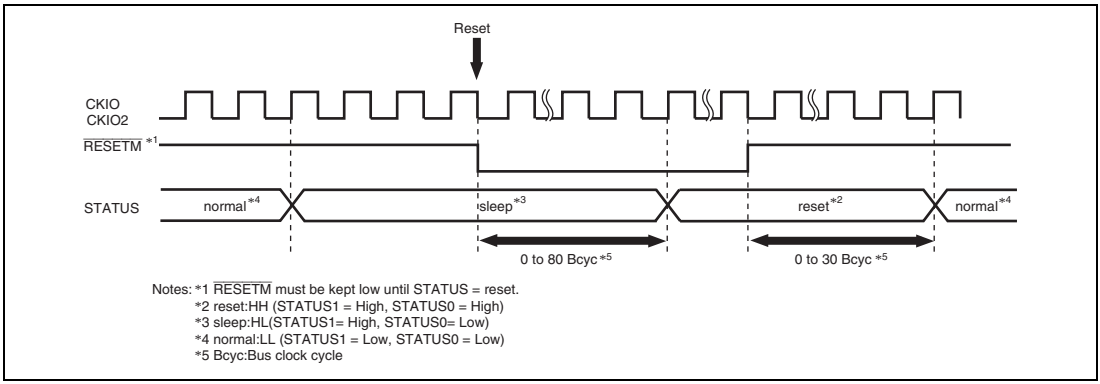


Figure 10.9 STATUS Output when Sleep Mode is Canceled by Manual Reset

Section 11 On-Chip Oscillation Circuits

11.1 Overview

The oscillator consists of a clock pulse generator (CPG) block and a watchdog timer (WDT) block.

The CPG generates clocks supplied to this LSI and controls the power-down modes.

The WDT is a single-channel timer that counts the clock settling time and is used when clearing standby mode and temporary standbys, such as frequency changes. It can also be used as an ordinary watchdog timer or interval timer.

11.1.1 Features

The CPG has the following features:

- Seven clock modes: Selection of seven clock modes according to the frequency range to be used and direct connection of crystal resonator or external clock input.
- Three clocks generated independently: An internal clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the peripheral modules; and a bus clock (Bϕ = CKIO) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and standby mode and specific modules can be stopped using the module standby function.

The WDT has the following features:

- Can be used to ensure the clock settling time:
  Use the WDT to cancel standby mode and the temporary standbys which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode:
  Internal resets occur after counter overflow.
  Selection of power-on reset or manual reset.

- Generates interrupts in interval timer mode:
  An interval timer interrupt is generated after counter overflow.
- Selection of eight counter input clocks
  Eight clocks in which peripheral clocks are divided ($\times 1$ to $\times 1/4096$) can be selected.

11.2 Overview of CPG

11.2.1 CPG Block Diagram

A block diagram of the on-chip clock pulse generator is shown in figure 11.1.

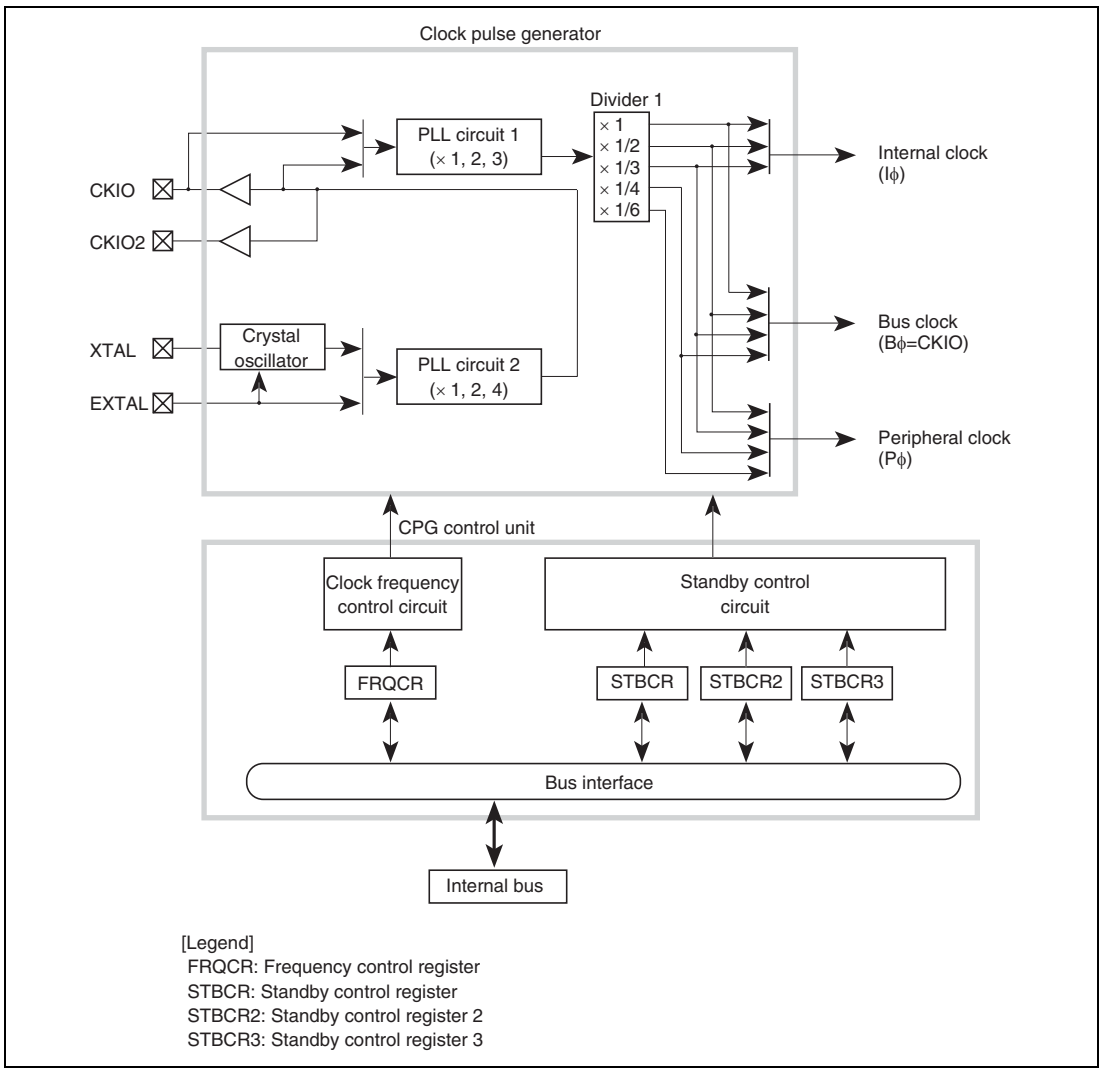


Figure 11.1 Block Diagram of CPG

The clock pulse generator blocks function as follows:

1. PLL Circuit 1: PLL circuit 1 doubles, triples, or leaves unchanged the input clock frequency from the CKIO terminal. The multiplication rate is set by the frequency control register. When this is done, the phase of the rising edge of the internal clock is controlled so that it will synchronize with the phase of the rising edge of the CKIO pin.
2. PLL Circuit 2: PLL circuit 2 doubles, quadruples, or leaves unchanged the input clock frequency from the crystal oscillator or EXTAL pin. The multiplication ratio is fixed by the clock operating modes. The clock operating modes is set by pins MD0, MD1, and MD2. See table 11.2 for more information on clock operating modes.
3. Crystal Oscillator: This oscillator is used when a crystal resonator is connected to the XTAL and EXTAL pins. This crystal oscillator operates according to the clock operating mode setting.
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal or peripheral clock. The operating frequency of the internal clock (Iϕ) can be 1, 1/2, or 1/3 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The operating frequency of the peripheral clock (Pϕ) can be 1/2, 1/3, 1/4, or 1/6 times the output frequency of PLL circuit 1 within 8.34 MHz ≤ Pϕ ≤ 33.34 MHz. The division ratio is set in the frequency control register.
5. Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD pins and the frequency control register.
6. Standby Control Circuit: The standby control circuit controls the state of the on-chip oscillator and other modules during clock switching or in sleep or standby mode.
7. Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.
8. Standby Control Register: The standby control register has bits for controlling the power-down modes. See section 10, Power-Down Modes, for more information.

11.2.2 Input/Output Pins

Table 11.1 lists the CPG pins and their functions.

Table 11.1 Pin Configuration

Pin Name	Abbreviation	I/O	Description
Mode control pins	MD0	I	Set the clock operating mode.
	MD1	I	Set the clock operating mode.
	MD2	I	Set the clock operating mode.
Crystal oscillator pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used to input an external clock.
Clock I/O pin	CKIO	IO	Inputs or outputs an external clock.
Clock output pin	CKIO2	O	Outputs an external clock.

Note: To prevent device malfunction, the value of the mode control pin is sampled only by a power-on reset.

11.3 Clock Operating Modes

Table 11.2 shows the relationship between the mode control pins (MD2 to MD0) combinations and the clock modes. Table 11.3 shows the available combinations of the values of the clock modes and frequency control register (FRQCR).

Table 11.2 Clock Operating Modes

	Pin Values			Clock I/O		PLL2	PLL1	CKIO
Mode	MD2	MD1	MD0	Source	Output	On/Off	On/Off	Frequency
0	0	0	0	EXTAL	CKIO	ON	ON	(EXTAL)
					CKIO2	(x 1)	(x 1, 2, 3)	
1	0	0	1	EXTAL	CKIO	ON	ON	(EXTAL) x 4
					CKIO2	(x 4)	(x 1, 2, 3)	
2	0	1	0	Crystal resonator	CKIO	ON	ON	(Crystal) x 4
					CKIO2	(x 4)	(x 1, 2, 3)	
4	1	0	0	Crystal resonator	CKIO	ON	ON	(Crystal)
					CKIO2	(x 1)	(x 1, 2, 3)	

Mode	Pin Values			Clock I/O		PLL2	PLL1	CKIO
	MD2	MD1	MD0	Source	Output	On/Off	On/Off	Frequency
5	1	0	1	EXTAL	CKIO	ON	ON	(EXTAL) x 2
					CKIO2	(x 2)	(x 1, 2, 3)	
6	1	1	0	Crystal resonator	CKIO	ON	ON	(Crystal) x 2
					CKIO2	(x 2)	(x 1, 2, 3)	
7	1	1	1	CKIO	—	OFF	ON	(CKIO)
							(x 1, 2, 3)	

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside this LSI. An input clock frequency of 33.3 MHz to 66.67 MHz can be used, and the CKIO frequency range is 33.3 MHz to 66.67 MHz.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency external clock to be used. An input clock frequency of 10.00 MHz to 16.67 MHz can be used, and the CKIO frequency range is 40.00 MHz to 66.67 MHz.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency external clock to be used. A crystal oscillation frequency of 10.00 MHz to 16.67 MHz can be used, and the CKIO frequency range is 40.00 MHz to 66.67 MHz.

Mode 4: The on-chip crystal oscillator operates and undergoes waveform shaping by PLL circuit 2 before being supplied inside this LSI. A crystal oscillation frequency of 33.34 MHz to 48.00 MHz can be used, and the CKIO frequency range is 33.34 MHz to 48.00 MHz.

Mode 5: An external clock is input from the EXTAL pin and its frequency is multiplied by 2 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency external clock to be used. An input clock frequency of 16.67 MHz to 33.34 MHz can be used, and the CKIO frequency range is 33.34 MHz to 66.67 MHz.

Mode 6: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 2 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency clock to be used. A crystal oscillation frequency of 10.00 MHz to 16.67 MHz can be used, and the CKIO frequency range is 40.00 MHz to 66.67 MHz

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and undergoes waveform shaping and also frequency multiplication according to the setting, by PLL circuit 1 before being supplied to this LSI. As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Table 11.3 Possible Combination of Clock Mode and FRQCR Values

Mode	FRQCR Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Frequency Range of Input Clock and Crystal Resonator	Frequency Range of CKIO Pin
0	1001	On (x1)	On (x1)	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	On (x1)	1:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	On (x1)	1:1:1/4	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1103	On (x2)	On (x1)	2:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	On (x1)	2:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	On (x1)	3:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
1, 2	1001	On (x1)	On (x4)	4:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1002	On (x1)	On (x4)	4:4:4/3	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1003	On (x1)	On (x4)	4:4:1	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1103	On (x2)	On (x4)	8:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1104	On (x2)	On (x4)	8:4:4/3	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
	1204	On (x3)	On (x4)	12:4:2	10.00 MHz to 16.67 MHz	40.00 MHz to 66.67 MHz
4	1001	On (x1)	On (x1)	1:1:1/2	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz
	1002	On (x1)	On (x1)	1:1:1/3	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz
	1003	On (x1)	On (x1)	1:1:1/4	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz
	1103	On (x2)	On (x1)	2:1:1/2	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz

Mode	FRQCR Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Frequency Range of Input Clock and Crystal Resonator	Frequency Range of CKIO Pin
4	1104	On (x2)	On (x1)	2:1:1/3	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz
	1204	On (x3)	On (x1)	3:1:1/2	33.34 MHz to 48.00 MHz	33.34 MHz to 48.00 MHz
5	1001	On (x1)	On (x2)	2:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	On (x2)	2:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	On (x2)	2:2:1/2	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1103	On (x2)	On (x2)	4:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	On (x2)	4:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	On (x2)	6:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
6	1001	On (x1)	On (x2)	2:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	On (x2)	2:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	On (x2)	2:2:1/2	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1103	On (x2)	On (x2)	4:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	On (x2)	4:2:2/3	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	On (x2)	6:2:1	16.67 MHz to 33.34 MHz	33.34 MHz to 66.67 MHz
7	1001	On (x1)	Off	1:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1002	On (x1)	Off	1:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1003	On (x1)	Off	1:1:1/4	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz

Mode	FRQCR Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Frequency Range of Input Clock and Crystal Resonator	Frequency Range of CKIO Pin
7	1103	On (x2)	Off	2:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1104	On (x2)	Off	2:1:1/3	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz
	1204	On (x3)	Off	3:1:1/2	33.34 MHz to 66.67 MHz	33.34 MHz to 66.67 MHz

Notes: * The input clock is 1.
Maximum frequency: Iϕ = 200.00 MHz, Bϕ (CKIO) = 66.67 MHz, Pϕ = 33.34 MHz

1. Use the CKIO frequency within 33.34 MHz ≤ CKIO ≤ 66.67 MHz.
2. The input to divider 1 is the output of PLL circuit 1.
3. Use the internal clock frequency within 33.34 MHz ≤ Iϕ ≤ 200.00 MHz.
   The internal clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQCR, and the division ratio selected by the IFC bit in FRQCR.
   Do not set the internal clock frequency lower than the CKIO pin frequency.
4. Use the peripheral clock frequency within 8.34 MHz ≤ Pϕ ≤ 33.34 MHz.
   The peripheral clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1 selected by the STC bit in FRQCR, and the division ratio selected by the PFC bit in FRQCR.
   Do not set the peripheral clock frequency higher than the frequency of the CKIO pin.
5. × 1, × 2, or × 3 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, or × 1/3 can be selected as the division ratio of an internal clock. × 1/2, × 1/3, × 1/4, or × 1/6 can be selected as the division ratio of a peripheral clock. Set the rate in FRQCR.
6. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1. Use the output frequency under 200.00 MHz.

11.4 Register Description

The CPG has the following register. For details on register addresses and register access size, refer to section 24, List of Registers.

- Frequency control register (FRQCR)

11.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register. FRQCR is initialized to H'1003 by a power-on reset, but retains its value in a manual reset and in standby mode.

The write values to bits 15 to 13, 11 to 10, 7 to 6, and 3 should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	CKOEN	1	R/W	Clock Output Enable CKOEN specifies whether a clock is output from the CKIO pin or the CKIO pin is placed in the level-fixed state in the standby mode, CKIO pin is fixed at low during STATUS 1 = L, and STATUSO = H, when CKOEN is set to 0. Therefore, the malfunction of an external circuit because of an unstable CKIO clock in releasing the standby mode can be prevented. The CKIO pin becomes to input pin regardless of the value of the CKOEN bit in clock operating mode 7. 0: CKIO pin goes to low level state in standby mode. 1: Clock is output from CKIO pin
11	—	0	R	Reserved
10	—	0	R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	STC1	0	R/W	Frequency Multiplication Ratio of PLL Circuit 1
8	STC0	0	R/W	00: × 1 time 01: × 2 times 10: × 3 times 11: Reserved (setting prohibited)
7	—	0	R	Reserved
6	—	0	R	These bits are always read as 0. The write value should always be 0.
5	IFC1	0	R/W	Internal Clock Frequency Division Ratio
4	IFC0	0	R/W	These bits specify the frequency division ratio of the internal clock (Iϕ) with respect to the output frequency of PLL circuit 1. 00: × 1 time 01: × 1/2 time 10: × 1/3 time 11: Reserved (setting prohibited)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	1	R/W	These bits specify the division ratio of the peripheral clock (Pϕ) frequency with respect to the output frequency of PLL circuit 1. 001: × 1/2 time 010: × 1/3 time 011: × 1/4 time 100: × 1/6 time Other than above: Reserved (setting prohibited)
0	PFC0	1	R/W	

11.5 Changing Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of divider 1. All of these are controlled by software through FRQCR. The methods are described below.

11.5.1 Changing Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:
   TME bit in WTCSR = 0: WDT stops
   CKS2 to CKS0 bits in WTCSR: Division ratio of WDT count clock
   WTCNT: Initial counter value
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in the IFC1 and IFC0 bits and PFC2 to PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. The internal and peripheral clocks both stop and the WDT is supplied with the clock. The clock will continue to be output at the CKIO pin.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

11.5.2 Changing Division Ratio

The WDT will not count unless the multiplication rate is changed simultaneously.

1. In the initial state, IFC1 and IFC0 = 00 and PFC2 to PFC0 = 011.
2. Set the IFC1, IFC0, and PFC2 to PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

11.6 Overview of WDT

11.6.1 Block Diagram of WDT

Figure 11.2 shows a block diagram of the WDT.

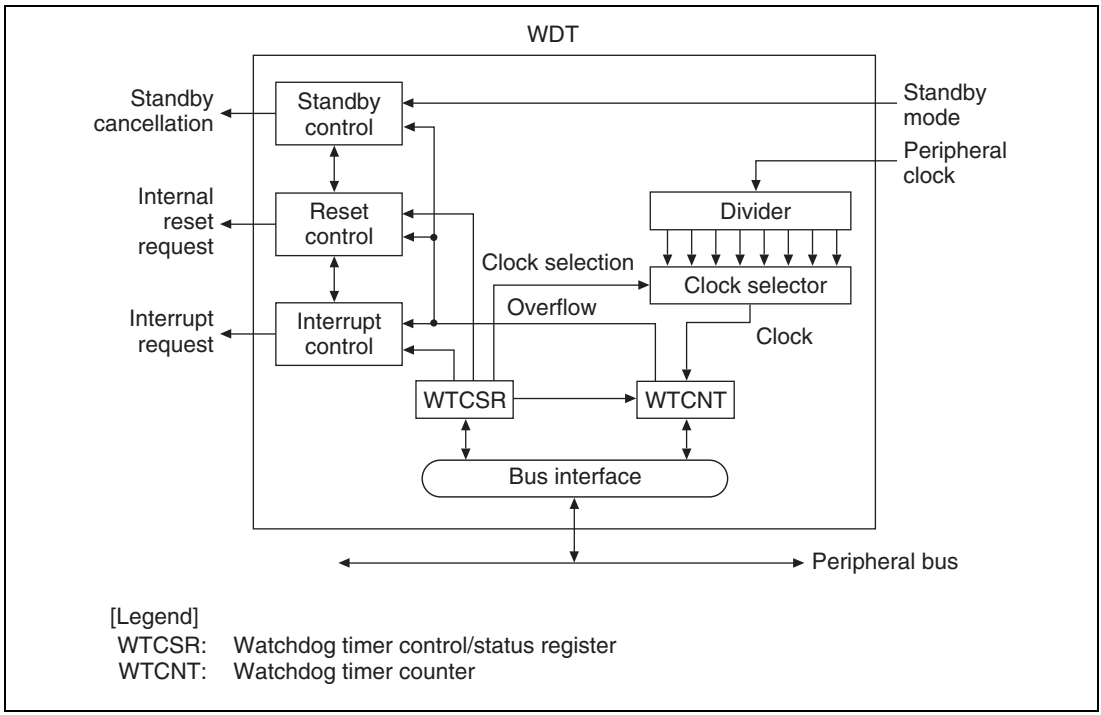


Figure 11.2 Block Diagram of WDT

11.7 Register Descriptions of WDT

The WDT has the following two registers that select the clock, switch the timer mode, and perform other functions. For details on register addresses and register access size, refer to section 24, List of Registers.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

11.7.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable counter. WTCNT increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. WTCNT is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. Use a word access to write to WTCNT, with H'5A in the upper byte. Use a byte access to read WTCNT.

Note: WTCNT differs from other registers in that it is more difficult to write to. See section 11.7.3, Notes on Register Access, for details.

11.7.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags.

WTCSR is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. When a WDT overflow causes an internal reset, WTCSR retains its value. When used to count the clock settling time for canceling a standby, it retains its value after counter overflow.

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

Note: WTCSR differs from other registers in that it is more difficult to write to. See section 11.7.3, Notes on Register Access, for details.

Bit	Bit Name	Initial Value	R/W	Description
7	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in standby mode or when changing the clock frequency. 0: Timer disabled: Count-up stops and WTCNT value is retained 1: Timer enabled
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{IT}$ is modified when the WDT is running, the up-count may not be performed correctly.
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock. The overflow period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.
0	CKS0	0	R/W	

Clock Select	Clock Division Ratio	Overflow Period (when Pϕ=15MHz)
000	1	17 μs
001	1/4	68 μs
010	1/16	273 μs
011	1/32	546 μs
100	1/64	1.09 ms
101	1/256	4.36 ms
110	1/1024	17.48 ms
111	1/4096	69.91 ms

Note: If bits CKS2 to CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

11.7.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 11.3. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

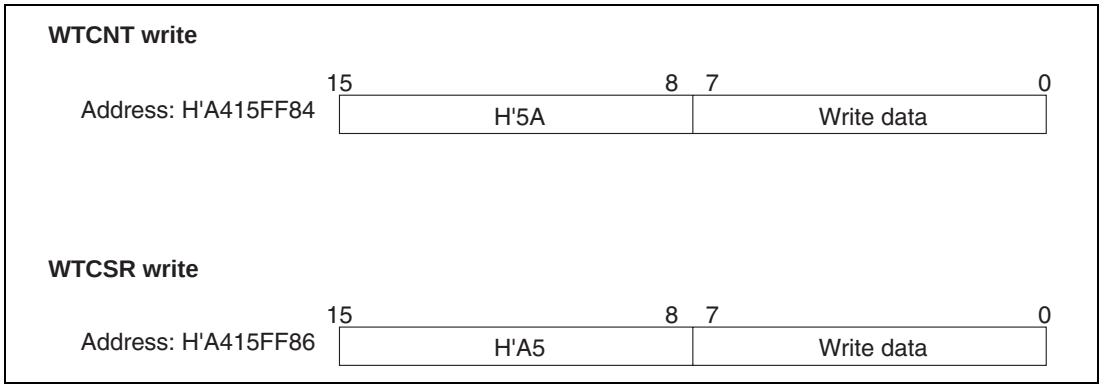


Figure 11.3 Writing to WTCNT and WTCSR

11.8 Using WDT

11.8.1 Canceling Standbys

The WDT can be used to cancel standby mode with an interrupt such as an NMI. The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RESETP}}$ or $\overline{\text{RESETM}}$ pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set at this time.
6. Since the WDT continues counting from H'00, clear the STBY bit in STBCR to 0 in the interrupt processing program and this will stop the WDT. When the STBY bit remains 1, the LSI again enters the standby mode when the WDT has counted up to H'80. This standby mode can be canceled by power-on resets.

11.8.2 Changing Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the processor stop temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set at this time.
5. The counter stops at the values H'00.
6. Before changing the WTCNT after the execution of the frequency change instruction, always confirm that the value of the WTCNT is H'00 by reading the WTCNT.

11.8.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

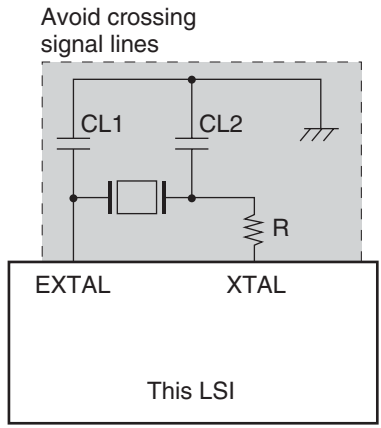
11.8.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{\text{IT}}$ bit in WTCSR to 0, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to INTC. The counter then resumes counting.

11.9 Notes on Board Design

When Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



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

Figure 11.4 Points for Attention when Using Crystal Resonator

Bypass Capacitors: Insert a laminated ceramic capacitor as a bypass capacitor for each Vss/VssQ and Vcc/VccQ pair. Mount the bypass capacitors to the power supply pins, and use components with a frequency characteristic suitable for the operating frequency of the LSI, as well as a suitable capacitance value.

Pin assignments of HQFP2828-256 (FP-256G/GV)

Vss/VssQ and Vcc/VccQ pair of digital circuitry
3 and 4, 13 and 14, 15 and 16, 25 and 26, 35 and 36, 43 and 44, 49 and 50, 57 and 58, 72 and 73, 81 and 82, 83 and 84, 92 and 93, 99 and 100, 107 and 108, 113 and 114, 121 and 122, 136 and 137, 146 and 147, 148 and 149, 158 and 159, 167 and 168, 176 and 177, 185 and 186, 191 and 192, 206 and 207, 208 and 209, 221 and 222, 227 and 228, 235 and 236, 241 and 242

Vss/VssQ and Vcc/VccQ pair of the on-chip oscillator
193 and 196, 251 and 252, 253 and 254

Pin assignments of P-LFBGA1717-256 (BP-256H/HV)

Vss/VssQ and Vcc/VccQ pair of digital circuitry
D2-B1, E1-F4, G2-G3, J2-J4, L3-L2, N3-N2, R4-P2, U2-W1, V4-Y4, Y6-U7, W8-V8, V10-W10, V11-W11, V13-W13, U15-W14, W17-Y19, U18-U20, P17-N19, N18-P20, L17-L20, J18-J19, E20-F17, D19-B20, C19-A20, D15-B14, C14-A15, B11-D11, C10-B10, C8-B8, D6-B7

Vss/VssQ and Vcc/VccQ pair of the on-chip oscillator
B19-A19, C3-B5, and C5-C4

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL Vcc and PLL Vss connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component.

Connect the EXTAL pin to VccQ or VssQ and make the XTAL pin open in clock mode 7.

The analog power supply system of the PLL is sensitive to a noise. Therefore the system malfunction may occur by the intervention with other power supply. Do not supply the analog power supply with the same resource as the digital power supply of Vcc and VccQ.

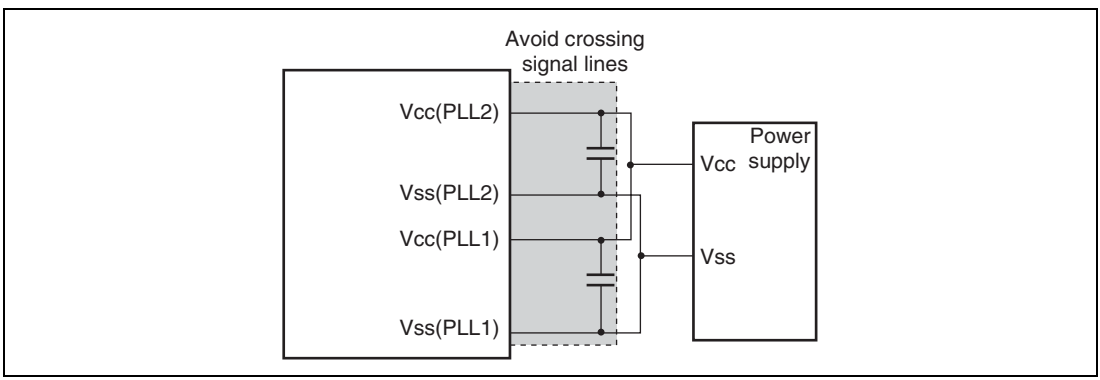


Figure 11.5 Points for Attention when Using PLL Oscillator Circuit

Section 12 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for various types of memory that is connected to the external address space and external devices. The BSC functions enable this LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

12.1 Features

The BSC has the following features:

1. External address space
- A maximum 32 or 64 Mbytes for each of the eight areas, CS0, CS2 to CS4, CS5A, CS5B, CS6A and CS6B, totally 384 Mbytes (divided into eight areas).
- A maximum 64 Mbytes for each of the six areas, CS0, CS2 to CS4, CS5, and CS6, totally a total of 384 Mbytes (divided into six areas).
- Can specify the normal space interface, byte-selection SRAM, burst ROM (clock synchronous or asynchronous), SDRAM, PCMCIA for each address space.
- Can select the data bus width (8, 16, or 32 bits) for each address space.
- Controls the insertion of the wait state for each address space.
- Controls the insertion of the wait state for each read access and write access.
- Can set the independent idling cycle in the continuous access for five cases: read-write (in same space/different space), read-read (in same space/different space), or the first cycle is a write access.

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

3. Burst ROM (clock asynchronous) interface
- High-speed access to the ROM that has the page mode function.

4. SDRAM interface
- Can set the SDRAM in up to 2 areas.
- Multiplex output for row address/column address.
- Efficient access by single read/single write.
- High-speed access by bank-active mode.
- Supports an auto-refresh and self-refresh.

5. Byte-selection SRAM interface
- Can connect directly to a byte-selection SRAM.

6. PCMCIA direct interface
- Supports IC memory cards and I/O card interfaces defined in the JEIDA specifications Ver 4.2 (PCMCIA2.1 Rev 2.1).
- Controls the insertion of the wait state using software.
- Supports the bus sizing function of the I/O bus width (only in little endian mode).

7. Burst ROM (clock synchronous) interface
- Can connect directly to a burst ROM of the clock synchronous type.

8. Bus arbitration
- Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.

9. Refresh function
- Supports the auto-refresh and self-refresh functions.
- Specifies the refresh interval using the refresh counter and clock selection.
- Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).

10. Interval timer using refresh counter
- Generates an interrupt request by a compare match.

The block diagram of the BSC is shown in figure 12.1.

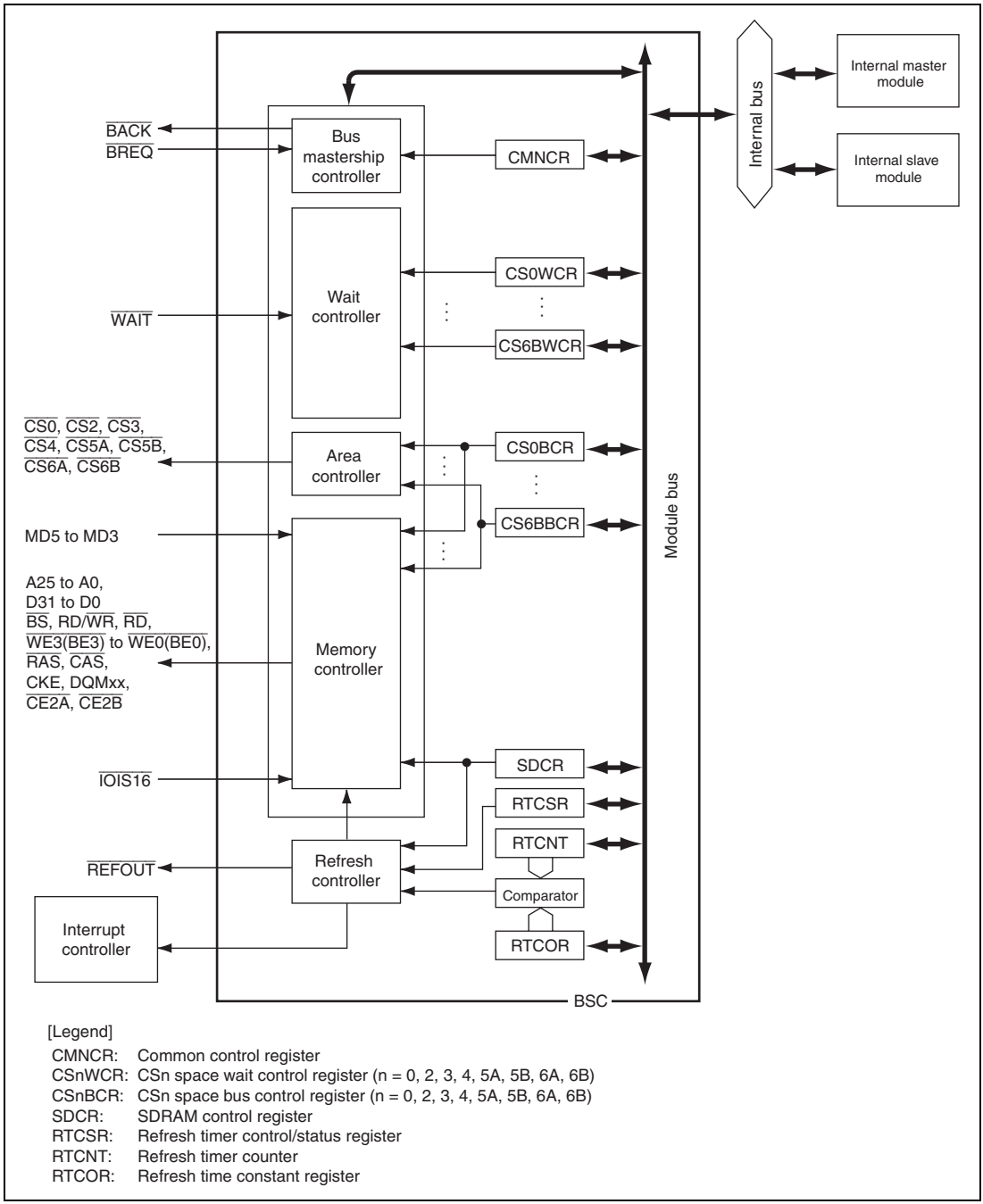


Figure 12.1 Block Diagram of BSC

12.2 Input/Output Pins

The configuration of pins in this module is shown in table 12.1.

Table 12.1 Pin Configuration

Name	I/O	Function
A25 to A0	O	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	O	Bus cycle start Asserted when a normal space, burst ROM (clock synchronous/asynchronous), or PCMCIA is accessed. Asserted by the same timing as $\overline{CAS}$ in SDRAM access.
$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$	O	Chip select
$\overline{CS5A}$	O	Chip select Active only for address map 1
$\overline{CS5B}$/$\overline{CE1A}$	O	Chip select Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
$\overline{CE2A}$	O	Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
$\overline{CS6A}$	O	Chip select Active only for address map 1
$\overline{CS6B}$/$\overline{CE1B}$	O	Chip select Corresponds to PCMCIA card select signals D7 to D0 when the PCMCIA is used.
$\overline{CE2B}$	O	Corresponds to PCMCIA card select signals D15 to D8 when the PCMCIA is used.
RD/$\overline{WR}$	O	Read/write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is connected.
$\overline{RD}$	O	Read pulse signal (read data output enable signal) A strobe signal to indicate the memory read cycle when the PCMCIA is used.

Name	I/O	Function
WE3(BE3)/ICIOWR	O	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the I/O write strobe signal when the PCMCIA is used.
WE2(BE2)/ICIORD	O	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the I/O read strobe signal when the PCMCIA is used.
WE1(BE1)/WE	O	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected. Functions as the memory write strobe signal when the PCMCIA is used.
WE0(BE0)	O	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is connected.
RAS	O	Connects to RAS pin when SDRAM is connected.
CAS	O	Connects to CAS pin when SDRAM is connected.
CKE	O	Connects to CKE pin when SDRAM is connected.
IOIS16	I	PCMCIA 16-bit I/O signal Valid only in little endian mode. Make it into low level at the time of big endian mode.
DQMUU DQMUL DQMLU DQMLL	O	Connected to the DQMxx when the SDRAM is connected. DQMUU: Selects D31 to D24 DQMUL: Selects D23 to D16 DQMLU: Selects D15 to D8 DQMLL: Selects D7 to D0
WAIT	I	External wait input
BREQ	I	Bus request input
BACK	O	Bus acknowledge output
MD5 to MD3	I	MD5: Selects data alignment (big endian or little endian) MD4 and MD3: Specify area 0 bus width (8/16/32 bits)
REFOUT	O	Refresh request output when a bus is released

12.3 Area Overview

12.3.1 Area Division

In the architecture of this LSI, both logical spaces and physical spaces have 32-bit address spaces. The upper three bits divide into the P0 to P4 areas, and specify the cache access method. For details see section 6, Cache. The remaining 29 bits are used for division of the space into ten areas (address map 1) or eight areas (address map 2) according to the MAP bit in CMNCR setting. The BSC performs control for this 29-bit space.

As listed in tables 12.2 and 12.3, this LSI can be connected directly to eight areas of memory, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$, $\overline{CS5A}$, $\overline{CS5B}$, $\overline{CS6A}$, and $\overline{CS6B}$) for each of them. $\overline{CS0}$ is asserted during area 0 access; $\overline{CS5A}$ is asserted during area 5A access when address map 1 is selected; and $\overline{CS5B}$ is asserted when address map 2 is selected.

12.3.2 Shadow Area

Areas 0, 2 to 4, 5A, 5B, 6A, and 6B are decoded by physical addresses A28 to A25, which correspond to areas 000 to 111. Address bits 31 to 29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space in P1 to P3 areas obtained by adding to it H'20000000 × n (n = 1 to 6).

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

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

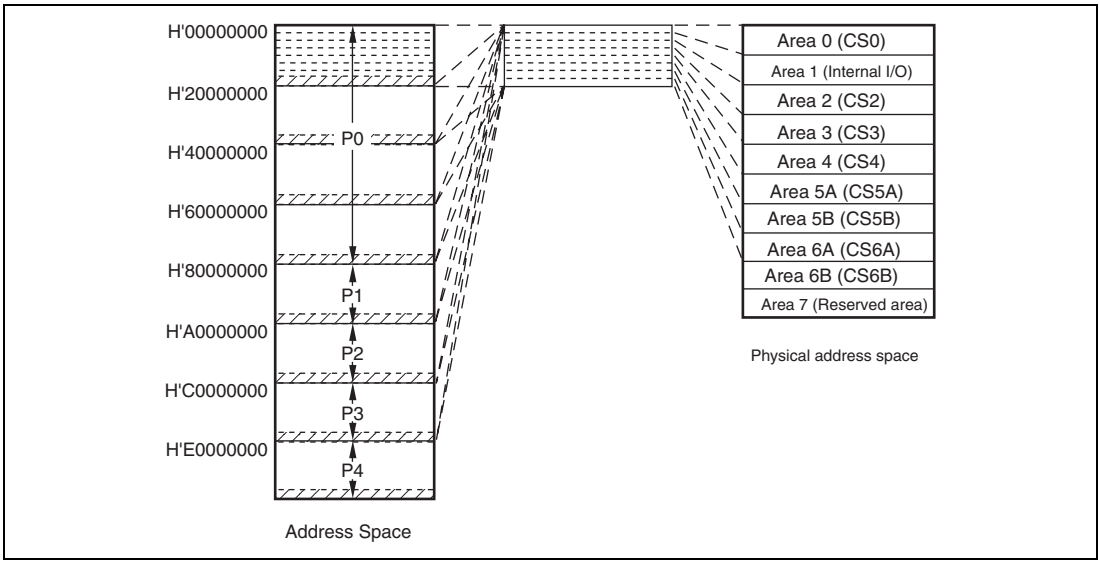


Figure 12.2 Address Space

12.3.3 Address Map

The external address space has a capacity of 384 Mbytes and is used by dividing 8 partial spaces (address map 1) or 6 partial spaces (address map 2). The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 12.2 Address Space Map 1 (CMNCR.MAP = 0)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory*3 Burst ROM (Asynchronous) Burst ROM (Synchronous)	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Internal I/O register area*2	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Normal memory*3 Byte-selection SRAM SDRAM	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory*3 Byte-selection SRAM SDRAM	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory*3 Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H'14000000 to H'15FFFFFF	Area 5A	Normal memory*3	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory*3 Byte-selection SRAM	32 Mbytes
H'18000000 to H'19FFFFFF	Area 6A	Normal memory*3	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory*3 Byte-selection SRAM	32 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*1	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. Set the top three bits of the address to 101 to allocate in the P2 space.
3. Memory that has an interface such as SRAM.

Table 12.3 Address Space Map 2 (CMNCR.MAP = 1)

Physical Address	Area	Memory to be Connected	Capacity
H′00000000 to H′03FFFFFF	Area 0	Normal memory*4 Burst ROM (Asynchronous) Burst ROM (Synchronous)	64 Mbytes
H′04000000 to H′07FFFFFF	Area 1	Internal I/O register area*3	64 Mbytes
H′08000000 to H′0BFFFFFF	Area 2	Normal memory*4 Byte-selection SRAM SDRAM	64 Mbytes
H′0C000000 to H′0FFFFFFF	Area 3	Normal memory*4 Byte-selection SRAM SDRAM	64 Mbytes
H′10000000 to H′13FFFFFF	Area 4	Normal memory*4 Byte-selection SRAM Burst ROM (Asynchronous)	64 Mbytes
H′14000000 to H′17FFFFFF	Area 5*2	Normal memory*4 Byte-selection SRAM PCMCIA	64 Mbytes
H′18000000 to H′1BFFFFFF	Area 6*2	Normal memory*4 Byte-selection SRAM PCMCIA	64 Mbytes
H′1C000000 to H′1FFFFFFF	Area 7	Reserved area*1	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. For area 5, CS5BBCR and CS5BWCR are valid.
For area 6, CS6BBCR and CS6BWCR are valid.
3. Set the top three bits of the address to 101 to allocate in the P2 space.
4. Memory that has an interface such as SRAM.

12.3.4 Area 0 Memory Type and Memory Bus Width

The memory bus width in this LSI can be set for each area. In area 0, external pins can be used to select byte (8 bits), word (16 bits), or longword (32 bits) on power-on reset. The memory bus width of the other area is set by the register. The correspondence between the memory type, external pins (MD3, MD4), and bus width is listed in the table below.

Table 12.4 Correspondence between External Pins (MD3 and MD4), Memory Type of CS0, and Memory Bus Width

MD4	MD3	Memory Type	Bus Width
0	0	Normal memory	Reserved (Setting prohibited)
	1		8 bits*
1	0		16 bits
	1		32 bits

Note: * The bus width must not be specified as eight bits if the burst ROM (clock synchronous) interface is selected.

12.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data alignment is specified using the external pin (MD5) at power-on reset as shown in table 12.5.

Table 12.5 Correspondence between External Pin (MD5) and Endians

MD5	Endian
0	Big endian
1	Little endian

12.4 Register Descriptions

The BSC has the following registers. Refer to section 24, List of Registers, for the addresses and access size for these registers.

Do not access spaces other than CS0 until the termination of the setting the memory interface.

- Common control register (CMNCR)
- Bus control register for area 0 (CS0BCR)
- Bus control register for area 2 (CS2BCR)
- Bus control register for area 3 (CS3BCR)
- Bus control register for area 4 (CS4BCR)
- Bus control register for area 5A (CS5ABCR)
- Bus control register for area 5B (CS5BBCR)
- Bus control register for area 6A (CS6ABCR)
- Bus control register for area 6B (CS6BBCR)
- Wait control register for area 0 (CS0WCR)
- Wait control register for area 2 (CS2WCR)
- Wait control register for area 3 (CS3WCR)
- Wait control register for area 4 (CS4WCR)
- Wait control register for area 5A (CS5AWCR)
- Wait control register for area 5B (CS5BWCR)
- Wait control register for area 6A (CS6AWCR)
- Wait control register for area 6B (CS6BWCR)
- SDRAM control register (SDCR)
- Refresh timer control/status register (RTCSR)*[1]
- Refresh timer counter (RTCNT)*[1]
- Refresh time constant register (RTCOR)*[1]
- SDRAM mode register for area 2 (SDMR2)*[2]
- SDRAM mode register for area 3 (SDMR3)*[2]

Notes: 1. This register only accepts 32-bit writing to prevent incorrect writing. In this case, the upper 16 bits of the data must be H'A55A. Otherwise, writing cannot be performed. In reading, the upper 16 bits are read as H'0000.

2. The contents of this register are stored in SDRAM. When this register space is accessed, the corresponding register in SDRAM is written to. For details, see description of Power-on Sequence in section 12.5.5, SDRAM Interface.

12.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. Do not access external memory other than area 0 until the CMNCR initialization is complete.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	BSD	0	R/W	Bus Access Start Timing Specification After Bus Acknowledge Specifies the bus access start timing after the external bus acknowledge signal is received. 0: Starts the external access at the same timing as the address drive start after the bus acknowledge signal is received. 1: Starts the external access one cycle following the address drive start after the bus acknowledge signal is received.
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	MAP	0	R/W	Space Specification Selects the address map for the external address space. The address maps to be selected are shown in tables 12.2 and 12.3. 0: Selects address map 1. 1: Selects address map 2.
11	BLOCK	0	R/W	Bus Lock Bit Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.

Bit	Bit Name	Initial Value	R/W	Description
10	DPRTY1	0	R/W	DMA Burst Transfer Priority
9	DPRTY0	0	R/W	Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer 11: Reserved (Setting prohibited)
8	DMAIW2	0	R/W	Wait States between Access Cycles when DMA Single Address is Transferred Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycled inserted 100: 6 idle cycled inserted 101: 8 idle cycle inserted 110: 10 idle cycles inserted 111: 12 idle cycled inserted
7	DMAIW1	0	R/W	
6	DMAIW0	0	R/W	
5	DMAIWA	0	R/W	Method of Inserting Wait States between Access Cycles when DMA Single Address is Transferred Specifies the method of inserting the idle cycles specified by the DMAIW1 and DMAIW0 bits. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. When the external device with DACK drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles even when the continuous accesses to an external device with DACK are performed.

Bit	Bit Name	Initial Value	R/W	Description
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	ENDIAN	0/1*	R	Endian Flag Samples the external pin for specifying endian on power-on reset (MD5). All address spaces are defined by this bit. This is a read-only bit. 0: The external pin for specifying endian (MD5) was low level on power-on reset. This LSI is being operated as big endian. 1: The external pin for specifying endian (MD5) was high level on power-on reset. This LSI is being operated as little endian.
2	CK2DRV	0	R/W	CKIO2 Drive Specifies whether the CKIO2 pin outputs a low level signal or clock (Bφ). 0: Outputs a low level signal 1: Outputs a clock (Bφ)
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$ ($\overline{BEn}$)/DQMxx, and $\overline{RD}$. When a bus is released, these pins enter the high-impedance state regardless of the setting of this bit. 0: High impedance in standby mode 1: Driven in standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in standby mode and bus released for CKIO, CKIO2, CKE, $\overline{RAS}$, and $\overline{CAS}$. 0: High impedance in standby mode and bus released for CKIO, CKIO2, CKE, $\overline{RAS}$, and $\overline{CAS}$. 1: Driven in standby mode and bus released for CKIO, CKIO2, CKE, $\overline{RAS}$, and $\overline{CAS}$. Note: If one of clock operating modes 4 to 6 is set, CKIO, CKIO2, CKE, $\overline{RAS}$, and $\overline{CAS}$ should be driven in standby mode and bus released.

Note: * The external pin (MD5) for specifying endian is sampled on power-on reset. When big endian is specified, this bit is read as 0 and when little endian is specified, this bit is read as 1.

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

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

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

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	IWW2	0	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
29	IWW1	1	R/W	
28	IWW0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
27	IWRWD2	0	R/W	Idle Cycles for Another Space Read-Write
26	IWRWD1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous accesses switch between different spaces.
25	IWRWD0	1	R/W	
				000: No idle cycle inserted
				001: 1 idle cycles inserted
				010: 2 idle cycles inserted
				011: 4 idle cycles inserted
				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted
24	IWRWS2	0	R/W	Idle Cycles for Read-Write in Same Space
23	IWRWS1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous accesses are for the same space.
22	IWRWS0	1	R/W	
				000: No idle cycle inserted
				001: 1 idle cycles inserted
				010: 2 idle cycles inserted
				011: 4 idle cycles inserted
				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
21	IWRRD2	0	R/W	Idle Cycles for Read-Read in Another Space
20	IWRRD1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses switch between different space.
19	IWRRD0	1	R/W	
				000: No idle cycle inserted
				001: 1 idle cycles inserted
				010: 2 idle cycles inserted
				011: 4 idle cycles inserted
				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted
18	IWRRS2	0	R/W	Idle Cycles for Read-Read in Same Space
17	IWRRS1	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous accesses are for the same space.
16	IWRRS0	1	R/W	
				000: No idle cycle inserted
				001: 1 idle cycles inserted
				010: 2 idle cycles inserted
				011: 4 idle cycles inserted
				100: 6 idle cycles inserted
				101: 8 idle cycles inserted
				110: 10 idle cycles inserted
				111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
15	TYPE3	0	R/W	Memory Type
14	TYPE2	0	R/W	Specify the type of memory connected to a space.
13	TYPE1	0	R/W	0000: Normal space
12	TYPE0	0	R/W	0001: Burst ROM (clock asynchronous)
				0010: Reserved (setting prohibited)
				0011: Byte-selection SRAM
				0100: SDRAM
				0101: PCMCIA
				0110: Reserved (setting prohibited)
				0111: Burst ROM (clock synchronous)*[2]
				1000: Reserved (setting prohibited)
				1001: Reserved (setting prohibited)
				1010: Reserved (setting prohibited)
				1011: Reserved (setting prohibited)
				1100: Reserved (setting prohibited)
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
				Note: Memory type for area 0 immediately after reset is normal space. The normal space, burst ROM (clock asynchronous), or burst ROM (clock synchronous) can be selected by these bits.
				For details on memory type in each area, see tables 12.2 and 12.3.
11	—	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
10	BSZ1	1*[1]	R/W	Data Bus Size
9	BSZ0	1*[1]	R/W	Specify the data bus sizes of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size Notes: 1. The data bus width for area 0 is specified by the external pin. The BSZ1 and BSZ0 bit settings in CS0BCR are ignored. 2. If area 5 or area 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 3. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as either 16 bits or 32 bits.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. CS0BCR samples the external pins (MD3 and MD4) that specify the bus width at a power-on reset.
2. The burst ROM (clock synchronous) must be accessed as a cacheable space.

12.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)

This register specifies various wait cycles for memory accesses. The bit configuration of this register varies as shown below according to the memory type (TYPE3, TYPE2, TYPE1, or TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

Normal Space, Byte-Selection SRAM:

- CS0WCR, CS6BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 11	SW1 SW0	0 0	R/W R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Reserved (Setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS2WCR, CS3WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{WEn}$ ($\overline{BEn}$) and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 11	—	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
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR3 to WR0 setting (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
15 to 13	—	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
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read/write access.
8	WR1	1	R/W	0000: 0 cycle
7	WR0	0	R/W	0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS5AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	WW2	0	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR3 to WR0 setting (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
17	WW1	0	R/W	
16	WW0	0	R/W	
15 to 13	—	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
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS5BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{WEn}$ ($\overline{BEn}$) and RD/$\overline{WR}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	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
18	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary for write access.
16	WW0	0	R/W	000: The same cycles as WR3 to WR0 setting (read access wait) 001: 0 cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read/write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0.
				0: External wait is valid
				1: External wait is ignored
5 to 2	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation
0	HW0	0	R/W	
				Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation.
				00: 0.5 cycle
				01: 1.5 cycles
				10: 2.5 cycles
				11: 3.5 cycles

- CS6AWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	
10	WR3	1	R/W	Number of Access Wait Cycles Specify the number of wait cycles that are necessary for read/write access. 0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
9	WR2	0	R/W	
8	WR1	1	R/W	
7	WR0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

Burst ROM (Clock Asynchronous):

- CS0WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16-burst access for an 8-bit bus width during 16-byte access. If this bit is set to 1, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	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
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first read/write access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: 0 cycle
				0001: 1 cycle
				0010: 2 cycles
				0011: 3 cycles
				0100: 4 cycles
				0101: 5 cycles
				0110: 6 cycles
				0111: 8 cycles
				1000: 10 cycles
				1001: 12 cycles
				1010: 14 cycles
				1011: 18 cycles
				1100: 24 cycles
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification
				Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0.
				0: External wait is valid
				1: External wait is ignored
5 to 0	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BEN	0	R/W	Burst Enable Specification Enables or disables 8-burst access for a 16-bit bus width or 16- burst access for an 8-bit bus width during 16-byte access. If this bit is set to 1, 2-burst access is performed four times when the bus width is 16 bits and 4-burst access is performed four times when the bus width is 8 bits. To use a device that does not support 8-burst access or 16-burst access, set this bit to 1. 0: Enables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width. 1: Disables 8-burst access for a 16-bit bus width and 16-burst access for an 8-bit bus width.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 13	—	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
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion. These bits can be specified only in area 4. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first read/write access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. These bits can be specified only in area 4. 00: 0.5 cycle 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

SDRAM*:

- CS2WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved These bits are always read as 1. The write value should always be 1.
9	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	A2CL1	1	R/W	CAS Latency for Area 2
7	A2CL0	0	R/W	Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

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

- CS3WCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	TRP1	0	R/W	Number of Wait Cycles Waiting Completion of Precharge Specify the number of minimum wait cycles to be inserted to wait the completion of precharge. The setting for areas 2 and 3 is common. (1) From starting auto-charge to issuing the ACTV command for the same bank (2) From issuing the PRE/PALL command to issuing the ACTV command for the same bank (3) To transiting to power-down mode/deep power-down mode (4) From issuing the PALL command at auto-refresh to issuing the REF command (5) From issuing the PALL command at self-refresh to issuing the SELF command 00: 0 cycle 01: 1 cycles 10: 2 cycles 11: 3 cycles
13	TRP0	0	R/W	
12	—	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
11	TRCD1	0	R/W	Number of Wait Cycles from ACTV Command to READ (A)/WRIT (A) Command
10	TRCD0	1	R/W	Specify the number of minimum wait cycles from issuing the ACTV command to issuing the READ (A)/WRIT (A) command. The setting for areas 2 and 3 is common. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	A3CL1	1	R/W	CAS Latency for Area 3.
7	A3CL0	0	R/W	Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	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
4	TRWL1	0	R/W	Number of Wait Cycles Waiting Start of Precharge
3	TRWL0	0	R/W	Specify the number of minimum wait cycles to be inserted to wait the start of precharge. The setting for areas 2 and 3 is common. (1) This LSI is in non-bank active mode from the issue of the WRITA command to the start of auto-precharge in SDRAM, and issues the ACTV command for the same bank after issuing the WRITA command. Confirm how many cycles are required from the reception of the WRITA command to the start of auto-precharge in each SDRAM data sheet. Set this bit so that the number of cycles is not above the cycles specified by this bit. (2) This LSI is in bank active mode from issuing the WRIT command to issuing the PRE command, and the access to different row address in the same bank is performed. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
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	TRC1	0	R/W	Number of Idle Cycles from REF Command/Self-refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles between the commands in the following cases. The setting for areas 2 and 3 is common. (1) From issuing the REF command to issuing the ACTV/REF/MSR command (2) From releasing self-refresh to issuing the ACTV/REF/MSR command 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles
0	TRC0	0	R/W	

Note: * If both areas 2 and 3 are specified as SDRAM, TRP1/0, TRCD0/1, TRWL1/0, and TRC1/0 bit settings are common. If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or byte-selection SRAM.

PCMCIA:

- CS5BWCR, CS6BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SA1	0	R/W	Space Attribute Specification
20	SA0	0	R/W	Specify memory card interface or I/C card interface when the PCMCIA interface is selected. SA1 0: Specifies memory card interface when A25 = 1 1: Specifies I/O card interface when A25 = 1 SA0 0: Specifies memory card interface when A25 = 0 1: Specifies I/O card interface when A25 = 0
19 to 15	—	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
14	TED3	0	R/W	Delay from Address to $\overline{RD}$ or $\overline{WE}$ Assert
13	TED2	0	R/W	Specify the delay time from address output to $\overline{RD}$ or $\overline{WE}$ assert in PCMCIA interface.
12	TED1	0	R/W	
11	TED0	0	R/W	0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10	PCW3	1	R/W	Number of Access Wait Cycles
9	PCW2	0	R/W	Specify the number of wait cycles to be inserted.
8	PCW1	1	R/W	0000: 3 cycles
7	PCW0	0	R/W	0001: 6 cycles
				0010: 9 cycles
				0011: 12 cycles
				0100: 15 cycles
				0101: 18 cycles
				0110: 22 cycles
				0111: 26 cycles
				1000: 30 cycles
				1001: 33 cycles
				1010: 36 cycles
				1011: 38 cycles
				1100: 52 cycles
				1101: 60 cycles
				1110: 64 cycles
				1111: 80 cycles
6	WM	0	R/W	External Wait Mask Specification
				Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0.
				0: External wait is valid
				1: External wait is ignored
5, 4	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TEH3	0	R/W	Delay from $\overline{RD}$ or $\overline{WE}$ Negate to Address
2	TEH2	0	R/W	Specify the address hold time from $\overline{RD}$ or $\overline{WE}$ negate in the PCMCIA interface.
1	TEH1	0	R/W	
0	TEH0	0	R/W	0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

Burst ROM (Clock Synchronous):

- CS0WCR

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	BW1	0	R/W	Number of Burst Wait Cycles
16	BW0	0	R/W	Specify the number of wait cycles to be inserted between the second or later access cycles in burst access. 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

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	W3	1	R/W	Number of Access Wait Cycles
9	W2	0	R/W	Specify the number of wait cycles to be inserted in the first read/write access cycle.
8	W1	1	R/W	
7	W0	0	R/W	0000: 0 cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specify whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

12.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	A2ROW1	0	R/W	Number of Bits of Row Address for Area 2
19	A2ROW0	0	R/W	Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	A2COL1	0	R/W	Number of Bits of Column Address for Area 2
16	A2COL0	0	R/W	Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters the deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode

Bit	Bit Name	Initial Value	R/W	Description
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CKIO). When this bit set to 1, the hold time for command, address, and write and read data can be extended. This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CKIO. Read data from SDRAM is latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM is output at the falling edge of CKIO. Read data from SDRAM is latched at the falling edge of CKIO.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R/W	Power-Down Mode Specify whether SDRAM is put in power-down mode or not after the access to memory other than SDRAM is completed. This bit, when set to 1, drives the CKE pin low and places SDRAM in power-down mode by using an access to a memory other than SDRAM as a trigger. 0: Does not place SDRAM in power-down mode after an access to a memory other than SDRAM. 1: Places SDRAM in power-down mode after an access to a memory other than SDRAM.

Bit	Bit Name	Initial Value	R/W	Description
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be used only in area 3. In this case, the bus width can be selected as 16 or 32 bits. When both areas 2 and 3 are set to SDRAM, specify auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	A3COL1	0	R/W	Number of Bits of Column Address for Area 3
0	A3COL0	0	R/W	Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

12.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables a CMF interrupt request when the CMF bit of RTCSR is set to 1. 0: Disables the CMF interrupt request 1: Enables the CMF interrupt request
5	CKS2	0	R/W	Clock Select
4	CKS1	0	R/W	Select the clock input to count-up the refresh timer counter (RTCNT).
3	CKS0	0	R/W	000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096

Bit	Bit Name	Initial Value	R/W	Description
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long.
0	RRC0	0	R/W	
				000: Once
				001: Twice
				010: 4 times
				011: 6 times
				100: 8 times
				101: Reserved (setting prohibited)
				110: Reserved (setting prohibited)
				111: Reserved (setting prohibited)

12.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CKS0 in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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

12.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

If the CMIE bit of the RTCSR is set to 1, an interrupt is requested by this matching signal. This request is maintained until the CMF bit in RTCSR is cleared to 0. Clearing the CMF bit in RTCSR affects only interrupts and does not affect refresh requests. This makes it possible to count the number of refresh requests during refresh by interrupts, and to specify the refresh and interval timer interrupts simultaneously. When the RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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

12.5 Operation

12.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSByte) in the byte data and little endian, in which the 0 address is the least significant byte (LSByte) in the byte data. Endian is specified on power-on reset by the external pin (MD5). When MD5 pin is low level on power-on reset, the endian will become big endian and when MD5 pin is high level on power-on reset, the endian will become little endian.

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

Tables 12.6 to 12.11 show the relationship between endian, device data width, and access unit.

Table 12.6 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, DQMLL
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—	—
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Word access at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 12.7 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3}}$(BE3), DQMUU	$\overline{\text{WE2}}$(BE2), DQMUL	$\overline{\text{WE1}}$(BE1), DQMLU	$\overline{\text{WE0}}$(BE0), DQMLL
Byte access at 0		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert

Table 12.8 8-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	1st time at 0	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 1	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	2nd time at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24	—	—	—	Assert
	2nd time at 1	—	—	—	Data 23 to 16	—	—	—	Assert
	3rd time at 2	—	—	—	Data 15 to 8	—	—	—	Assert
	4th time at 3	—	—	—	Data 7 to 0	—	—	—	Assert

Table 12.9 32-Bit External Device/Little Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3}}$(BE3), DQMUU	$\overline{\text{WE2}}$(BE2), DQMUL	$\overline{\text{WE1}}$(BE1), DQMLU	$\overline{\text{WE0}}$(BE0), DQMLL
Byte access at 0	—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 3	Data 7 to 0	—	—	—	Assert	—	—	—
Word access at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 12.10 16-Bit External Device/Little Endian Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	Data 7 to 0	—	—	—	Assert	—
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
	2nd time at 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert

Table 12.11 8-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WE3(BE3)}}$, DQMUU	$\overline{\text{WE2(BE2)}}$, DQMUL	$\overline{\text{WE1(BE1)}}$, DQMLU	$\overline{\text{WE0(BE0)}}$, DQMLL
Byte access at 0		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 2		—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 3	—	—	—	Data 15 to 8	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
	3rd time at 2	—	—	—	Data 23 to 16	—	—	—	Assert
	4th time at 3	—	—	—	Data 31 to 24	—	—	—	Assert

12.5.2 Normal Space Interface

Basic Timing: For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 12.5.7, Byte-Selection SRAM Interface. Figure 12.3 shows the basic timings of normal space access. 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.

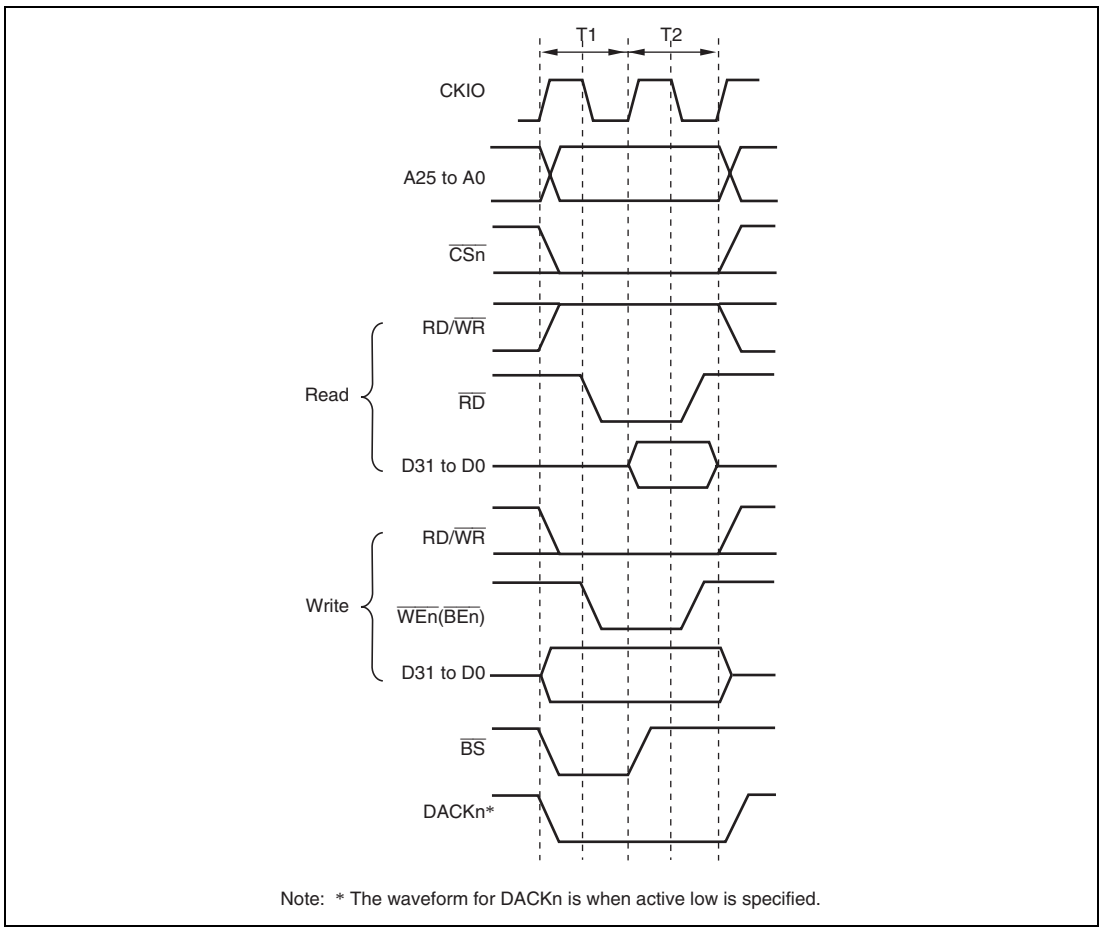


Figure 12.3 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always

read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, only the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{\text{RD}}$ when a buffer is established in the data bus. The RD/$\overline{\text{WR}}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 12.4 and 12.5 show the basic timings of normal space accesses. If the WM bit of the CSnWCR is cleared to 0, a Tnop cycle is inserted to evaluate the external wait (figure 12.4). If the WM bit of the CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 12.5).

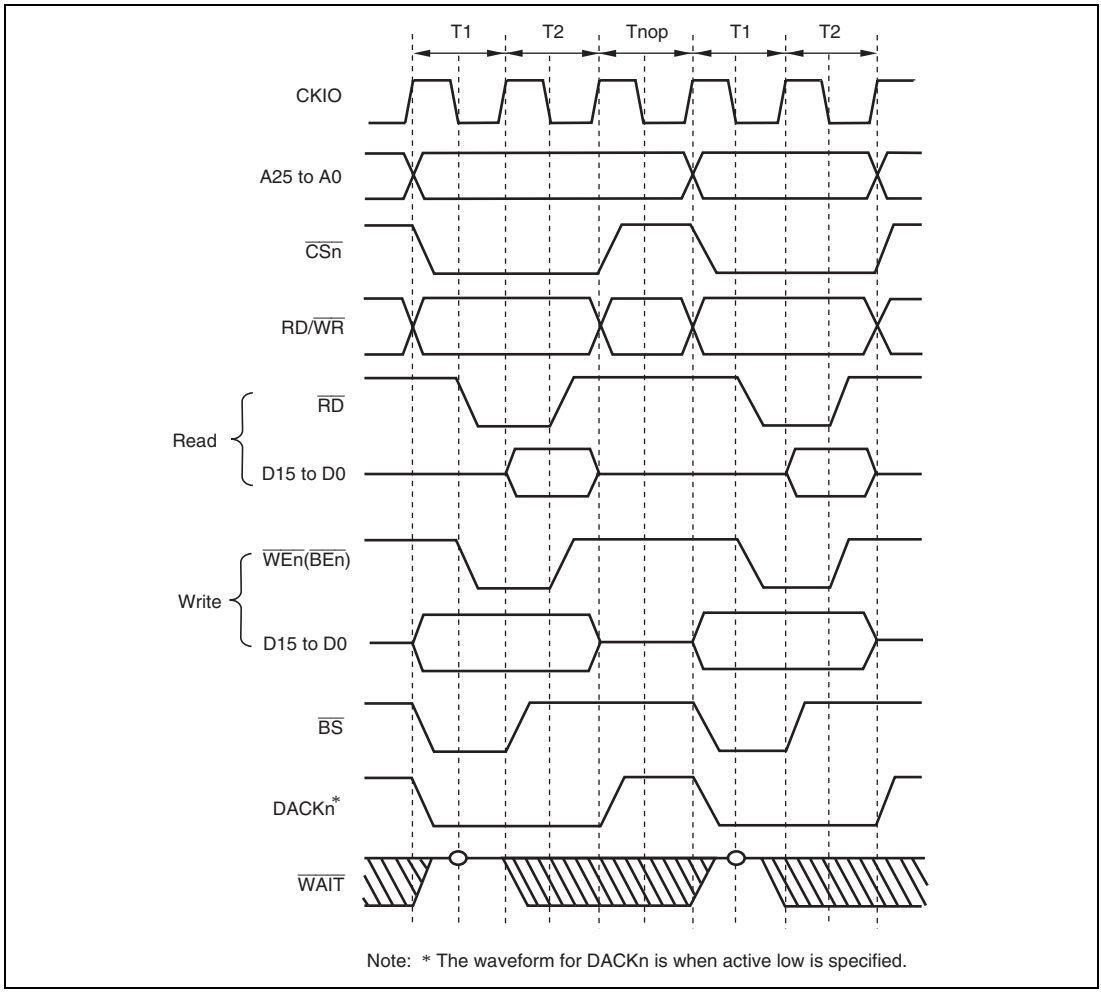


Figure 12.4 Continuous Access for Normal Space 1, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 0 (Access Wait = 0, Cycle Wait = 0)

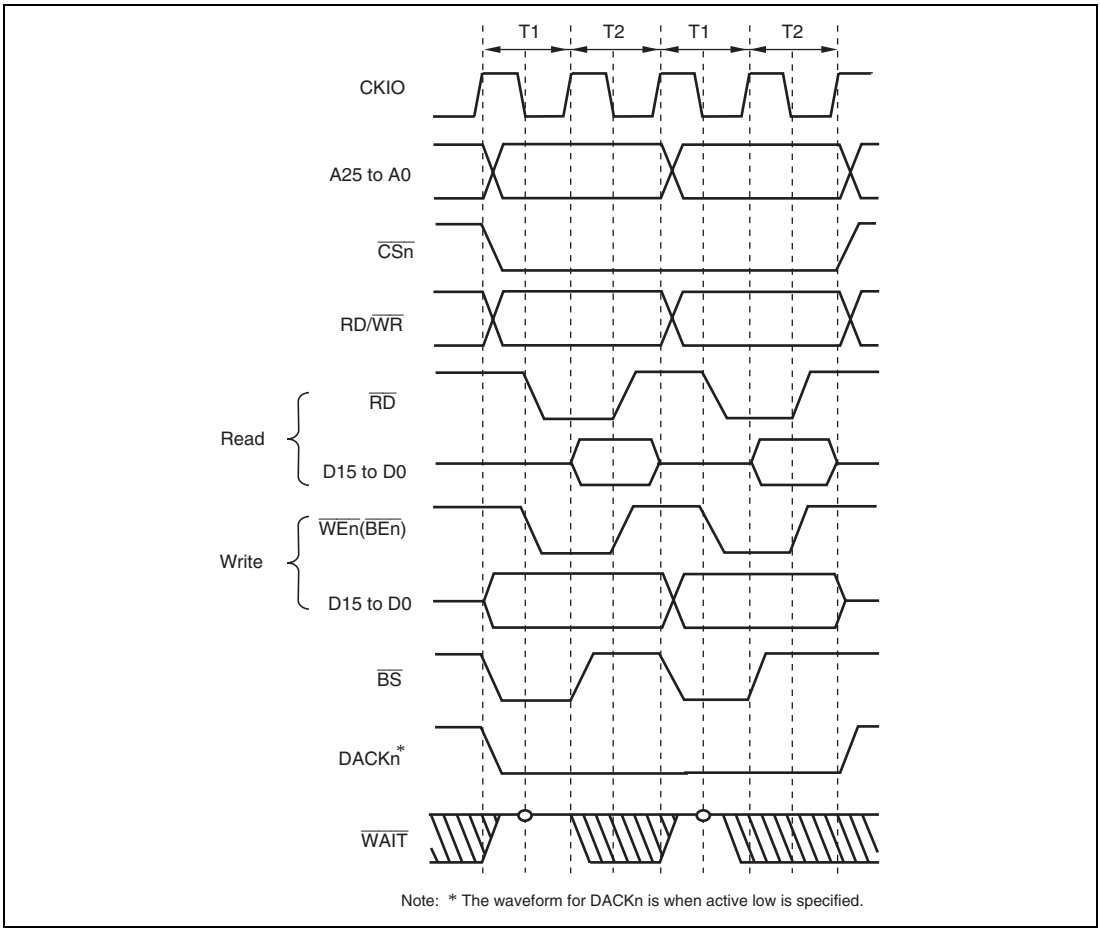


Figure 12.5 Continuous Access for Normal Space 2, Bus Width = 16 bits, Longword Access, CSnWCR.WM Bit = 1 (Access Wait = 0, Cycle Wait = 0)

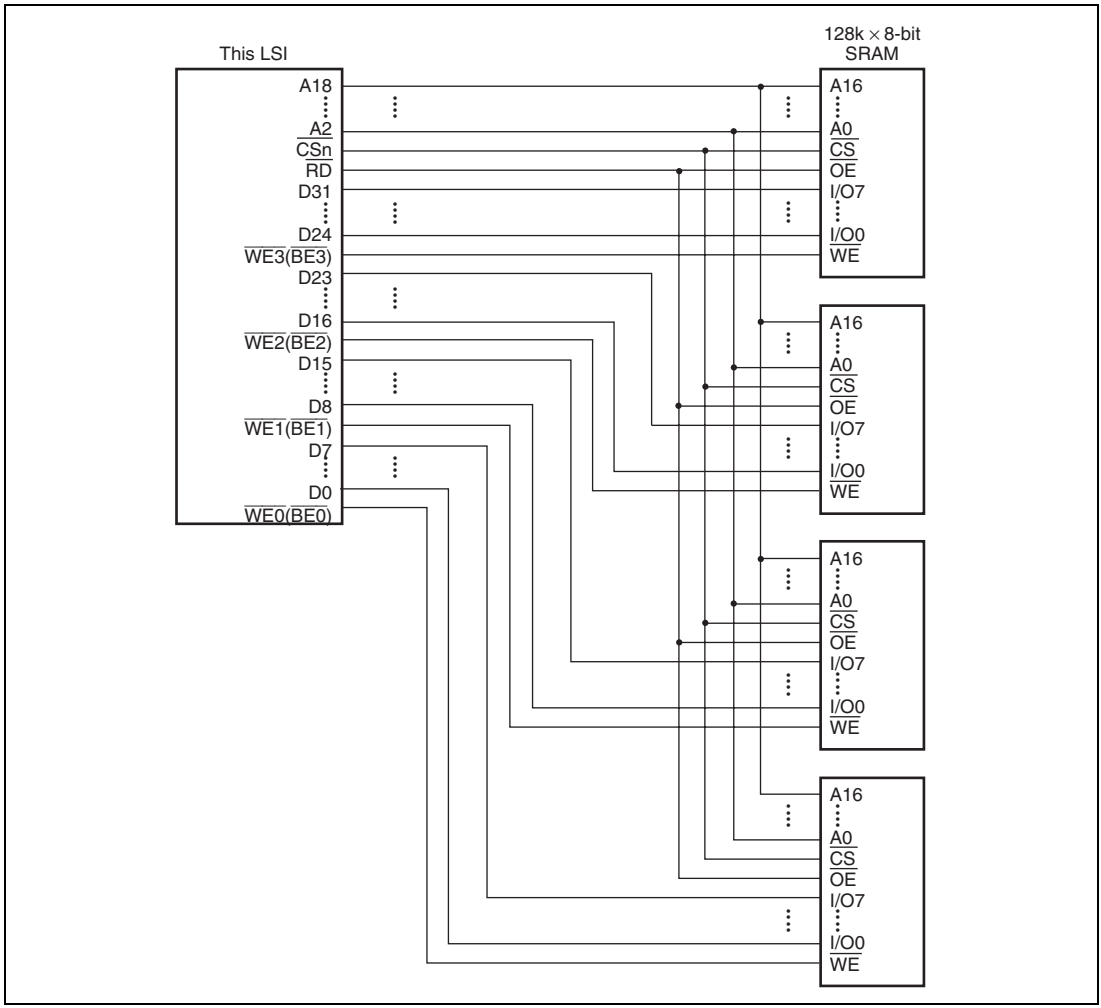


Figure 12.6 Example of 32-Bit Data-Width SRAM Connection

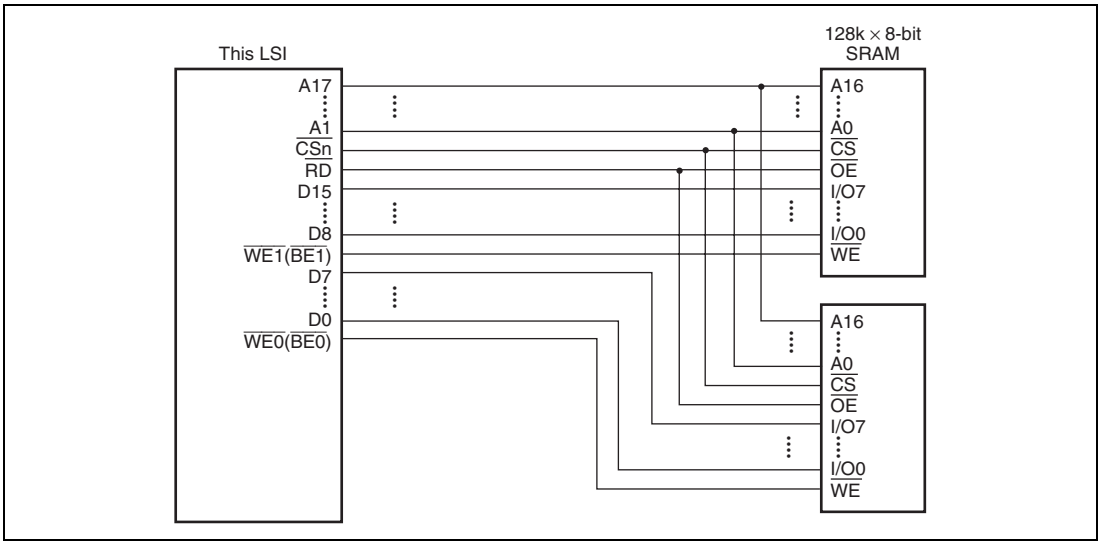


Figure 12.7 Example of 16-Bit Data-Width SRAM Connection

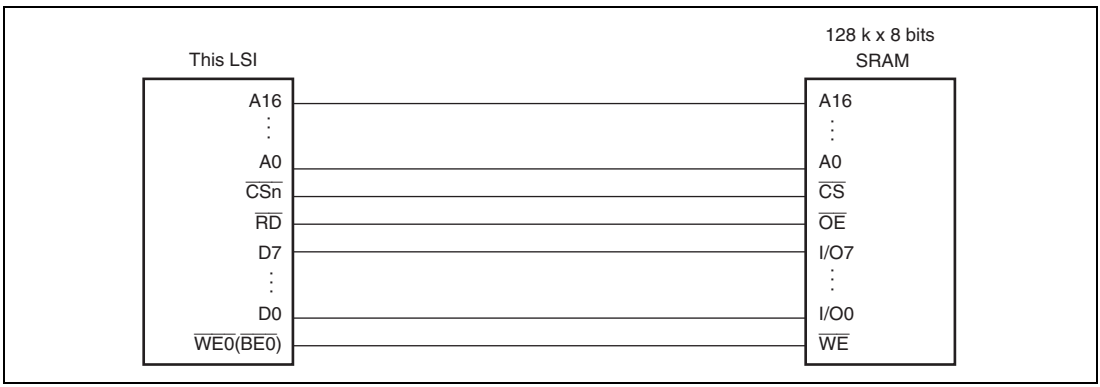
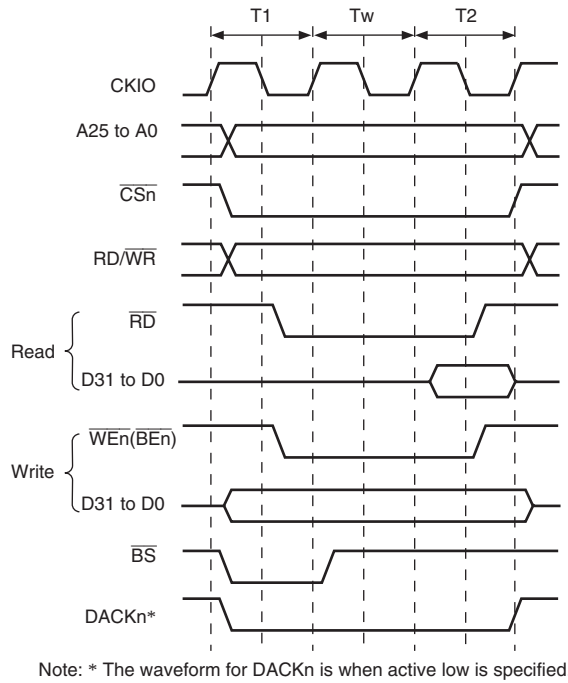


Figure 12.8 Example of 8-Bit Data-Width SRAM Connection

12.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 4, 5A, and 5B to insert wait cycles independently in read access and in write access. The areas other than 4, 5A, and 5B have common access wait for read cycle and write cycle. The specified number of Tw cycles is inserted as wait cycles in a normal space access shown in figure 12.9.



Note: * The waveform for DACKn is when active low is specified.

Figure 12.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input WAIT signal is also sampled. WAIT pin sampling is shown in figure 12.10. A 2-cycle wait is specified as a software wait. The WAIT signal is sampled on the falling edge of CKIO at the transition from the T1 or Tw cycle to the T2 cycle.

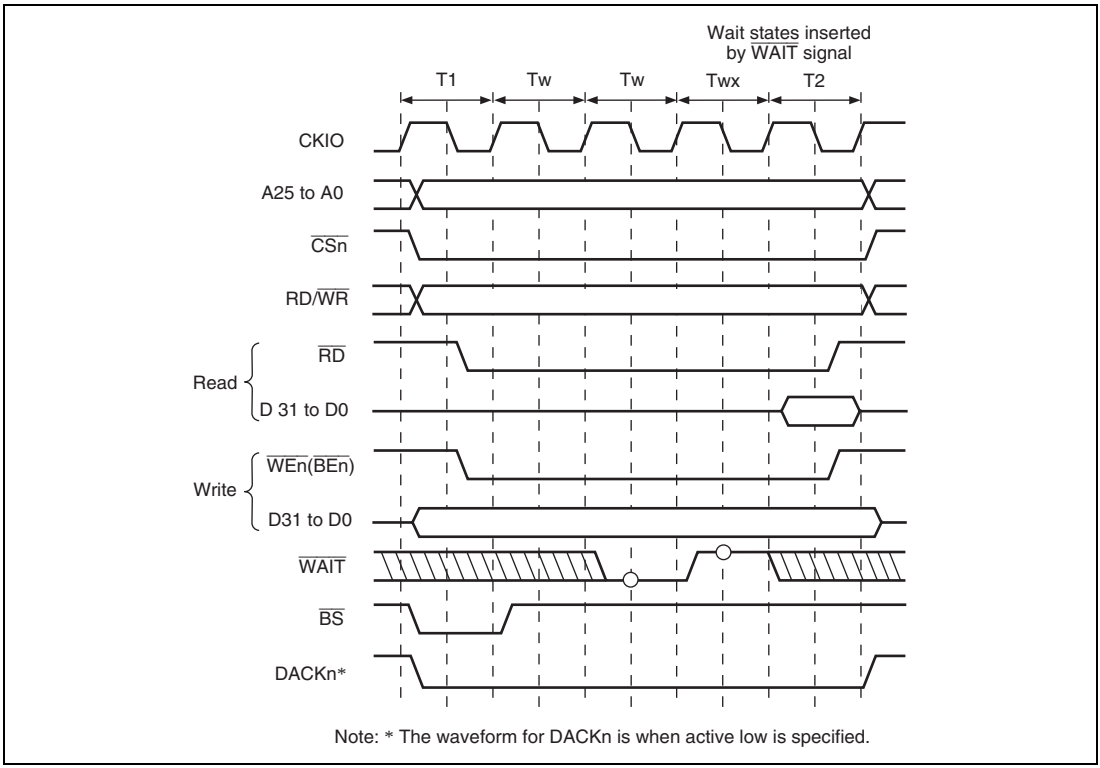


Figure 12.10 Wait State Timing for Normal Space Access (Wait State Insertion by WAIT Signal)

12.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 12.11 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

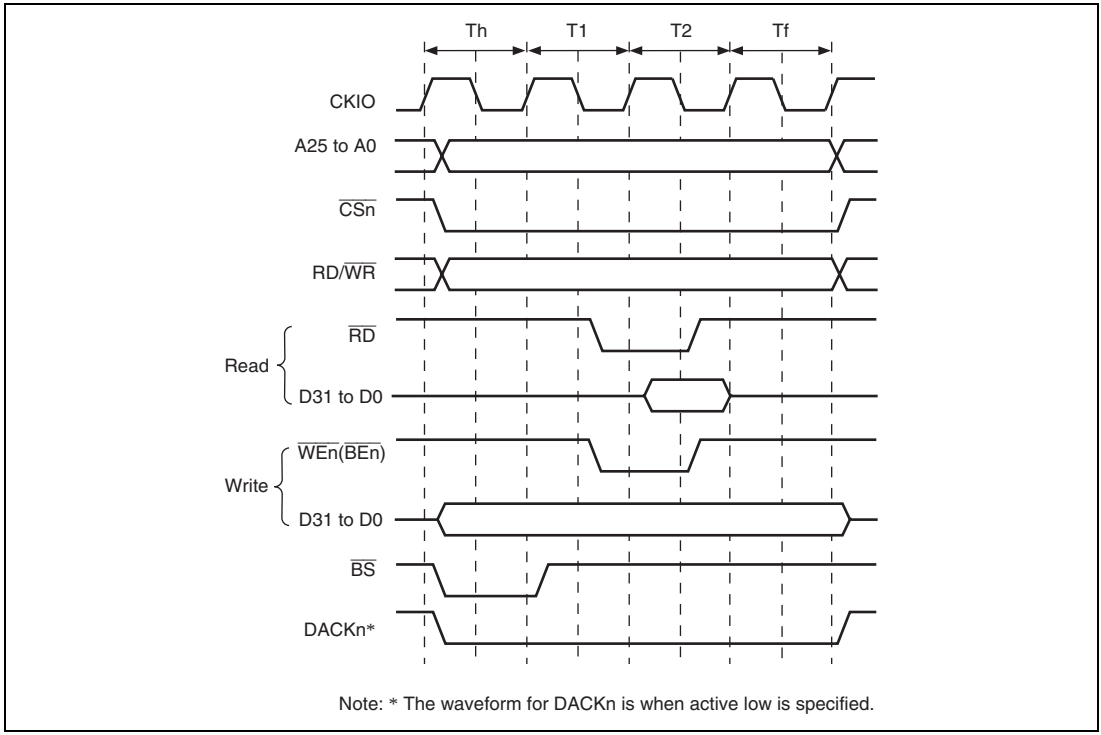


Figure 12.11 $\overline{\text{CSn}}$ Assert Period Expansion

12.5.5 SDRAM Interface

SDRAM Direct Connection: The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for direct connection of SDRAM are $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{CS2}$, and $\overline{CS3}$. All the signals other than $\overline{CS2}$ and $\overline{CS3}$ are common to all areas, and signals other than CKE are valid when $\overline{CS2}$ or $\overline{CS3}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 32 or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, and specific address signals. These commands are shown below.

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks precharge (PALL)
- Specified bank precharge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with precharge (READA)
- Write (WRIT)
- Write with precharge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, refer to section 12.5.1, Endian/Access Size and Data Alignment.

Figures 12.12 and 12.13 show examples of the connection of the SDRAM with the LSI.

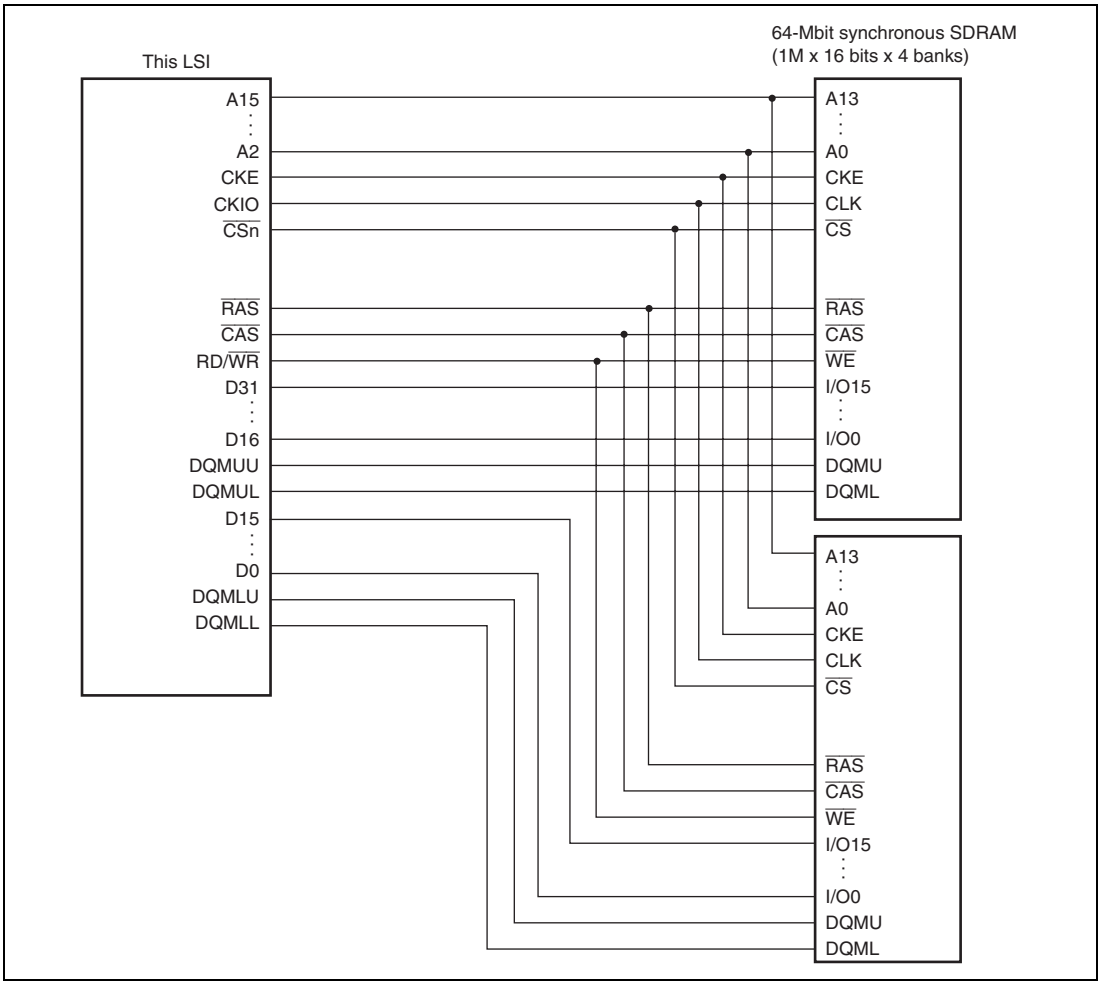


Figure 12.12 Example of 32-Bit Data-Width SDRAM Connection

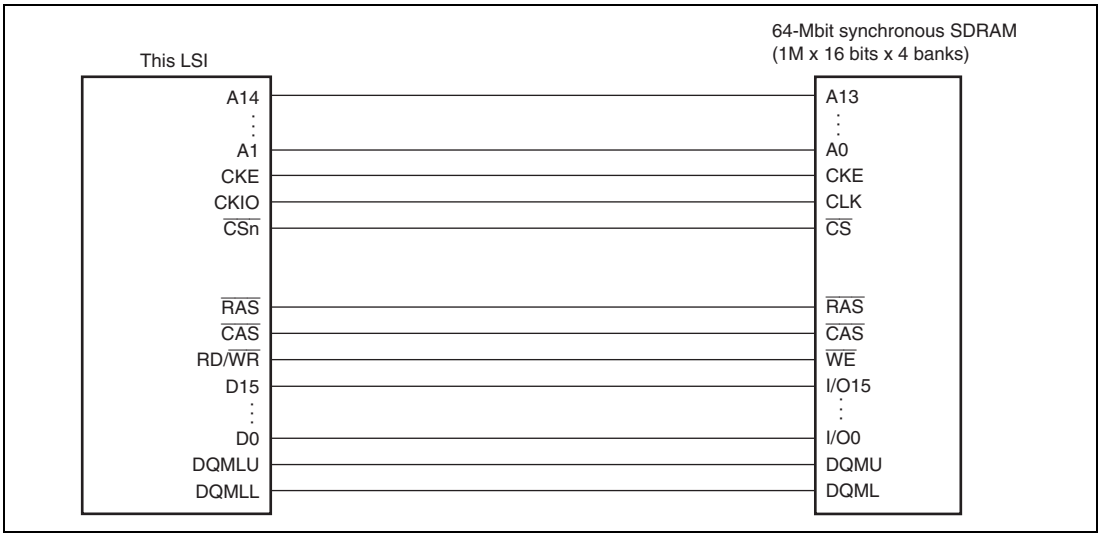


Figure 12.13 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0]in CSnBCR, AxROW[1:0] and AxCOL[1:0] in SDCR. Tables 12.12 to 12.17 show the relationship between the settings of bits BSZ[1:0], AxROW[1:0], and AxCOL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

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

Table 12.12 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A12 (BA1)*[3]	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 kwords x 32 bits x 4 banks, column 8 bits product): 1				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

3. If the number of 16-Mbit SDRAM (512 kwords × 16 bits × 2 banks: pin with 8-bit column) is two, the bank address specification is not required. Therefore, the bank address should be not used.

Table 12.12 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A24	A17		Unused
A16	A23	A16		
A15	A23*[2]	A23*[2]	A13 (BA1)	Specifies bank
A14	A22*[2]	A22*[2]	A12 (BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mword x 32 bits x 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 12.13 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 12.13 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]	A13 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A12 (BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*[1]	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 12.14 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A15	A24*[2]	A24*[2]	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords x 32 bits x 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 12.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*[2]	A20*[2]	A11 (BA0)	Specifies bank
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 kwords x 16 bits x 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 12.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]	A22*[2]	A13 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword x 16 bits x 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 12.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*[2]	A23*[2]	A13 (BA1)	Specifies bank
A13	A22*[2]	A22*[2]	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*[1]	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 12.16 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A13	A23*[2]	A23*[2]	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*[1]	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 12.17 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A14 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*[1]	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords x 16 bits x 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 12.17 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2

Setting				
A2/3 BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output**	**Column Address Output**	**Synchronous DRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*[1]	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords x 16 bits x 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

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

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

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 4 times to read 16-byte continuous data from the SDRAM that is connected to a 32-bit data bus.

Table 12.18 shows the relationship between the access size and the number of bursts.

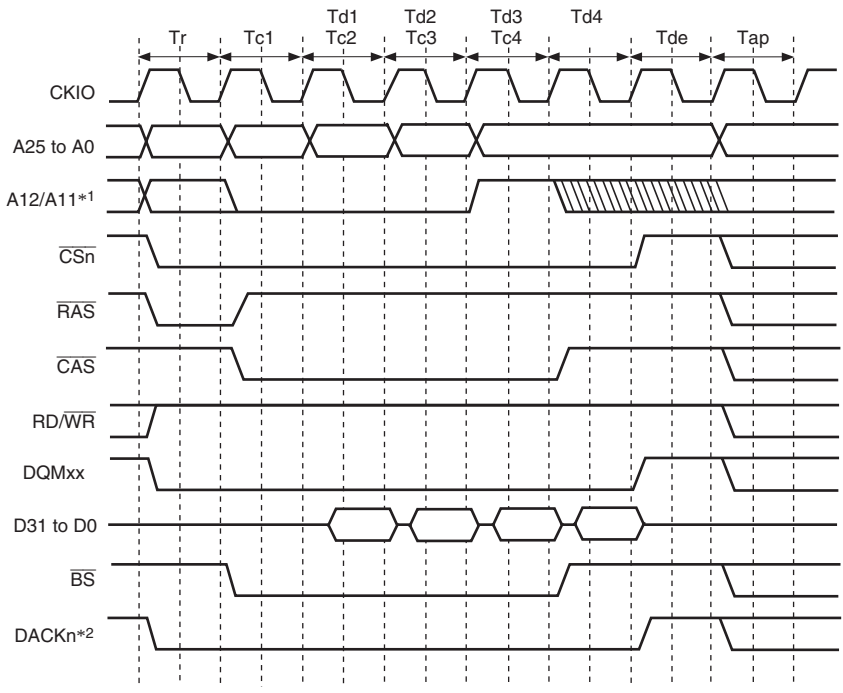
Table 12.18 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4

Figures 12.14 and 12.15 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the TRP1 and TRP0 bits in CS3WCR.

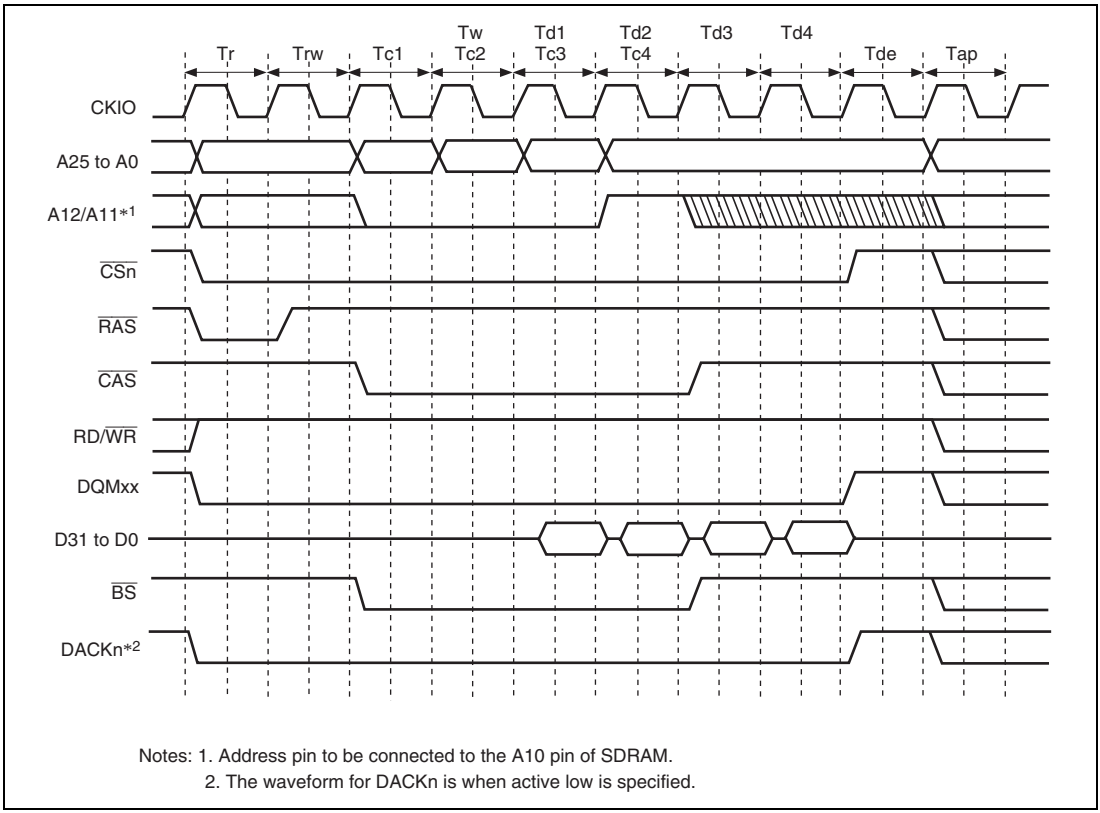
In this LSI, wait cycles can be inserted by specifying each bit in CSnWCR to connect the SDRAM in variable frequencies. Figure 12.15 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READA command is output can be specified using the TRCD1 and TRCD0 bits in CS3WCR. If the TRCD1 and TRCD0 bits specify one cycle or more, a Trw cycle where the NOP command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle

where the READA command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and TRCD0 bit in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the synchronous DRAM CAS latency. The CAS latency for the synchronous DRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the synchronous DRAM.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.14 Burst Read Basic Timing (Auto Precharge)

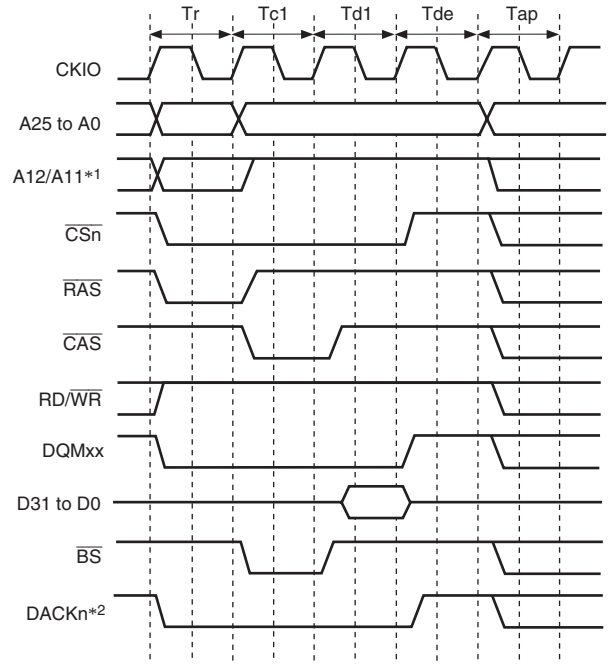


Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.15 Burst Read Wait Specification Timing (Auto Precharge)

Single Read: A read access ends in one cycle when data exists in non-cachable region and the data bus width is larger than or equal to access size. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

Figure 12.16 shows the single read basic timing.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

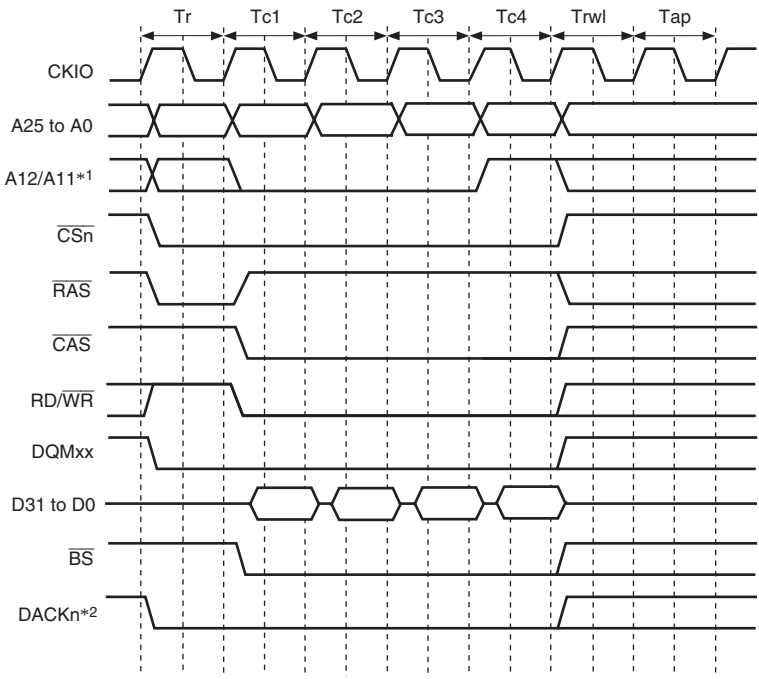
Figure 12.16 Basic Timing for Single Read (Auto Precharge)

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

1. Access size in writing is larger than data bus width.
2. Copyback of the cache
3. 16-byte transfer in DMAC or E-DMAC (access to non-cachable region)

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 4 times to write 16-byte continuous data to the SDRAM that is connected to a 32-bit data bus. The relationship between the access size and the number of bursts is shown in table 12.18.

Figure 12.17 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the TRP1 and TRP0 bits in CS3WCR.

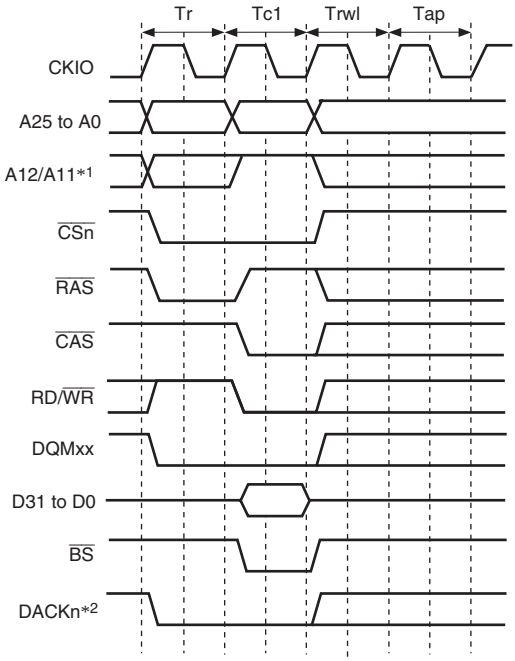


Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.17 Basic Timing for Burst Write (Auto Precharge)

Single Write: A write access ends in one cycle when data is written in non-cachable region and the data bus width is larger than or equal to access size.

Figure 12.18 shows the single write basic timing.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.18 Basic Timing for Single Write (Auto-Precharge)

Bank Active: The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the BACTV bit in SDCR is 1, accesses are performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or byte-selection SRAM. When areas 2 and 3 are both set to SDRAM, auto precharge mode must be set.

When a bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As synchronous DRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank,

then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the TRP[1:0] bits in CSnWCR.

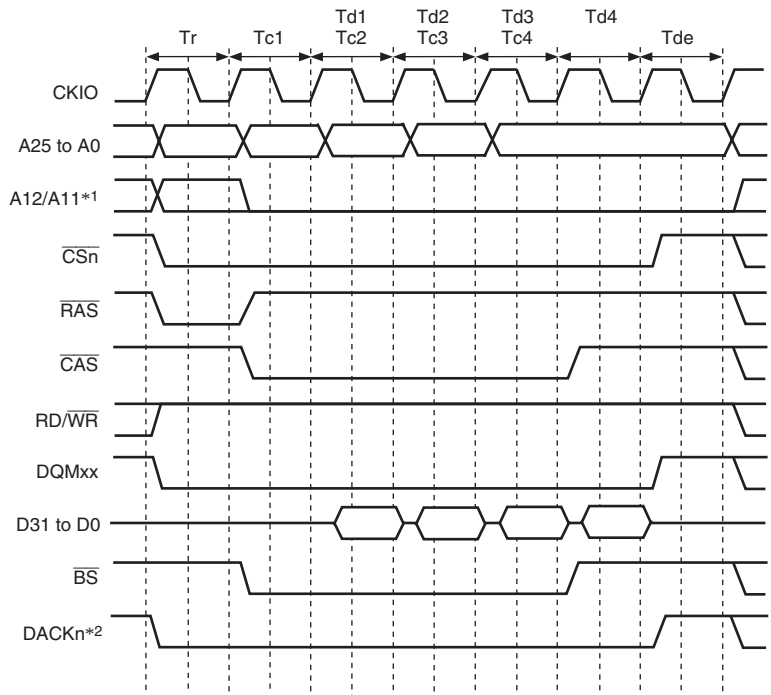
In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 12.19, a burst read cycle for the same row address in figure 12.20, and a burst read cycle for different row addresses in figure 12.21. Similarly, a single write cycle without auto-precharge is shown in figure 12.22, a single write cycle for the same row address in figure 12.23, and a single write cycle for different row addresses in figure 12.24.

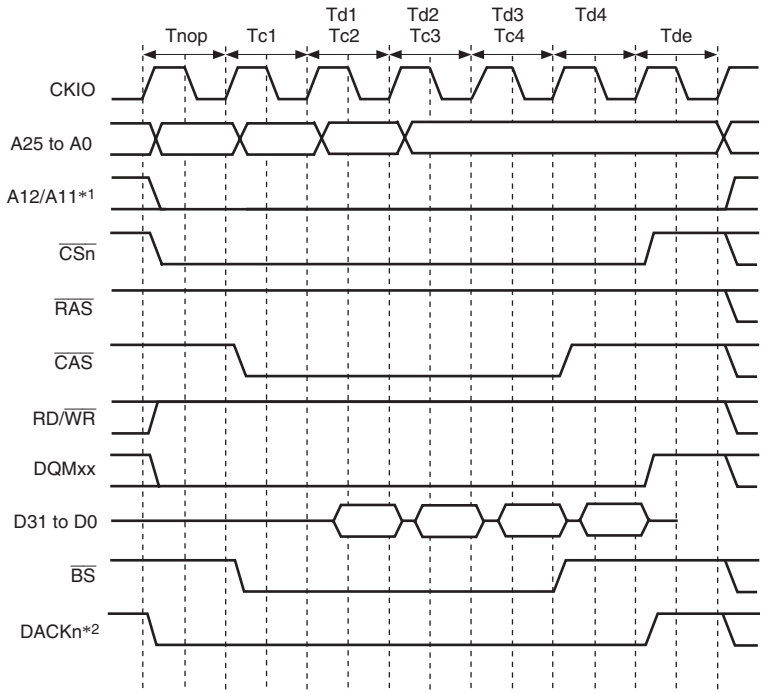
In figure 12.20, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 space are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 12.19 or 12.22, followed by repetition of the cycle in figure 12.20 or 12.23. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 12.21 or 12.24 is executed instead of that in figure 12.20 or 12.23. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



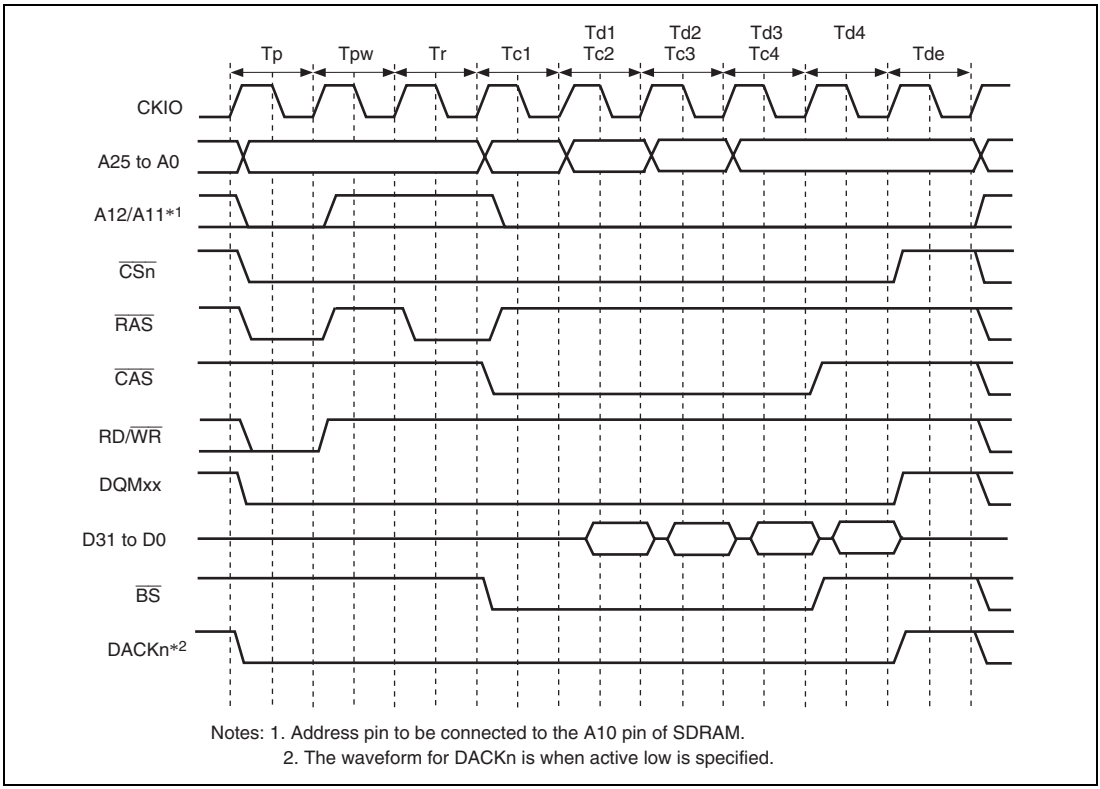
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.19 Burst Read Timing (No Auto Precharge)



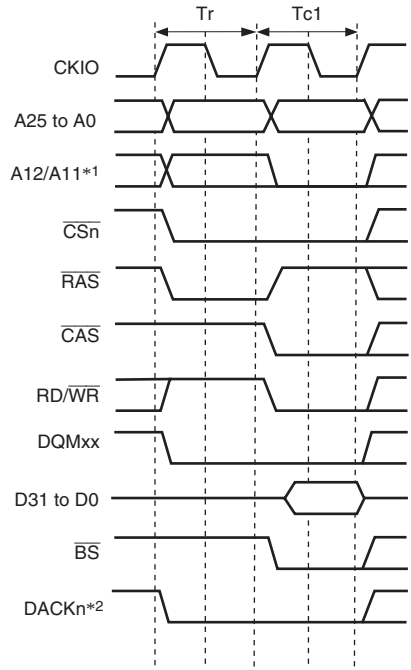
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.20 Burst Read Timing (Bank Active, Same Row Address)



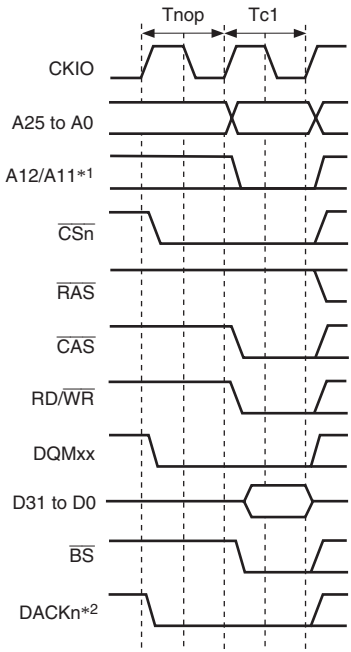
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.21 Burst Read Timing (Bank Active, Different Row Addresses)



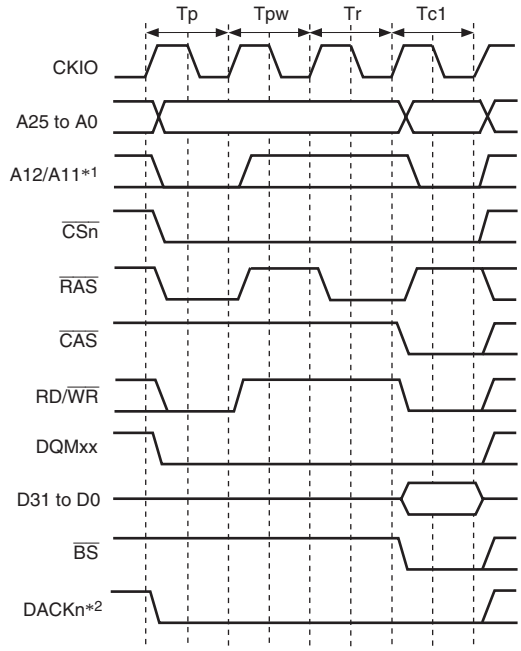
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.22 Single Write Timing (No Auto Precharge)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.23 Single Write Timing (Bank Active, Same Row Address)



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

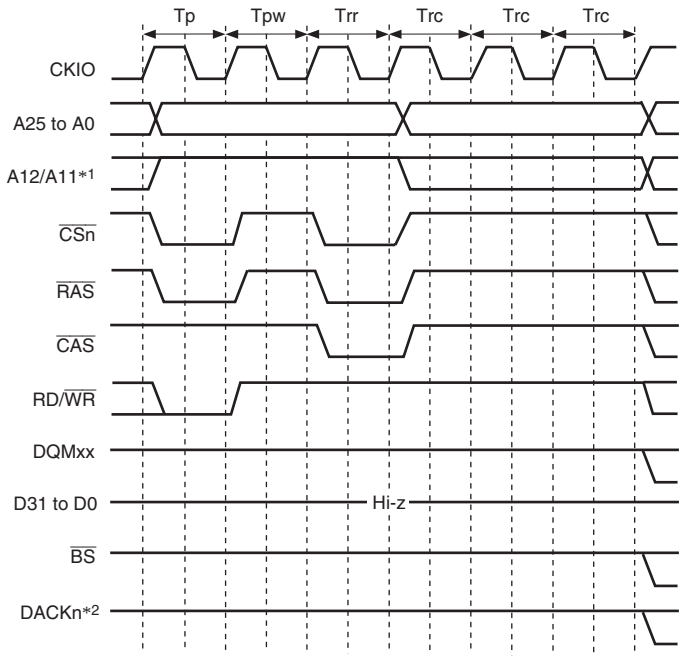
Figure 12.24 Single Write Timing (Bank Active, Different Row Addresses)

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC[2:0] bits in RTCSR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-refreshing

   Refreshing is performed at intervals determined by the input clock selected by bits CKS[2:0] in RTCSR, and the value set by in RTCOR. The value of bits CKS[2:0] in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS[2:0] and RRC[2:0] settings. When the clock is selected by bits CKS[2:0], RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC[2:0]. At the same time, RTCNT is cleared to 0 and the count-up is restarted. Figure 12.25 shows the auto-refresh cycle timing.

   After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to precharged state from active state when some bank is being precharged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the TRP[1:0]bits in CSnWCR. A new command is not issued for the duration of the number of cycles specified by the TRC[1:0] bits in CSnWCR after the Trr cycle. The TRC[1:0] bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). A NOP cycle is inserted between the Tp cycle and Trr cycle when the setting value of the TRP[1:0] bits in CSnWCR is longer than or equal to 1 cycle.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.25 Auto-Refresh Timing

2. Self-refreshing

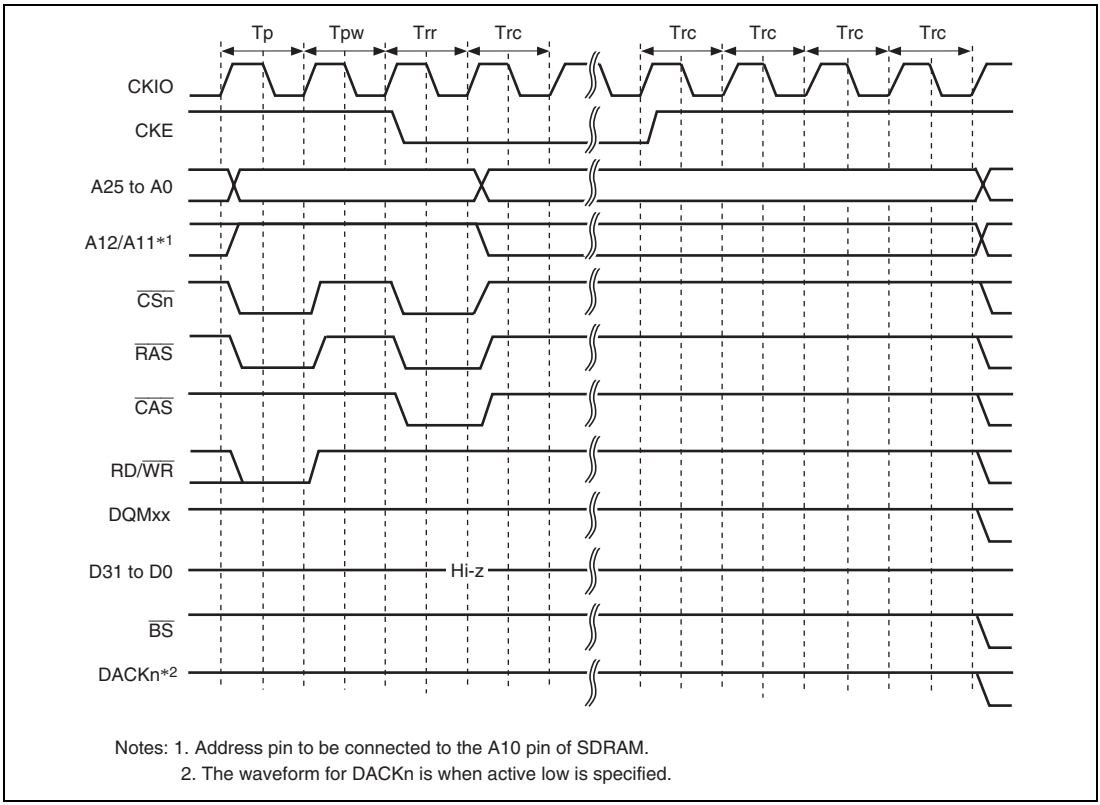
Self-refresh mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the TRP[1:0] bits in CSnWSR. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TRC[1:0] bits in CSnWCR.

Self-refresh timing is shown in figure 12.26. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode by an interrupt.

The self-refresh state is not cleared by a manual reset.

In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.26 Self-Refresh Timing

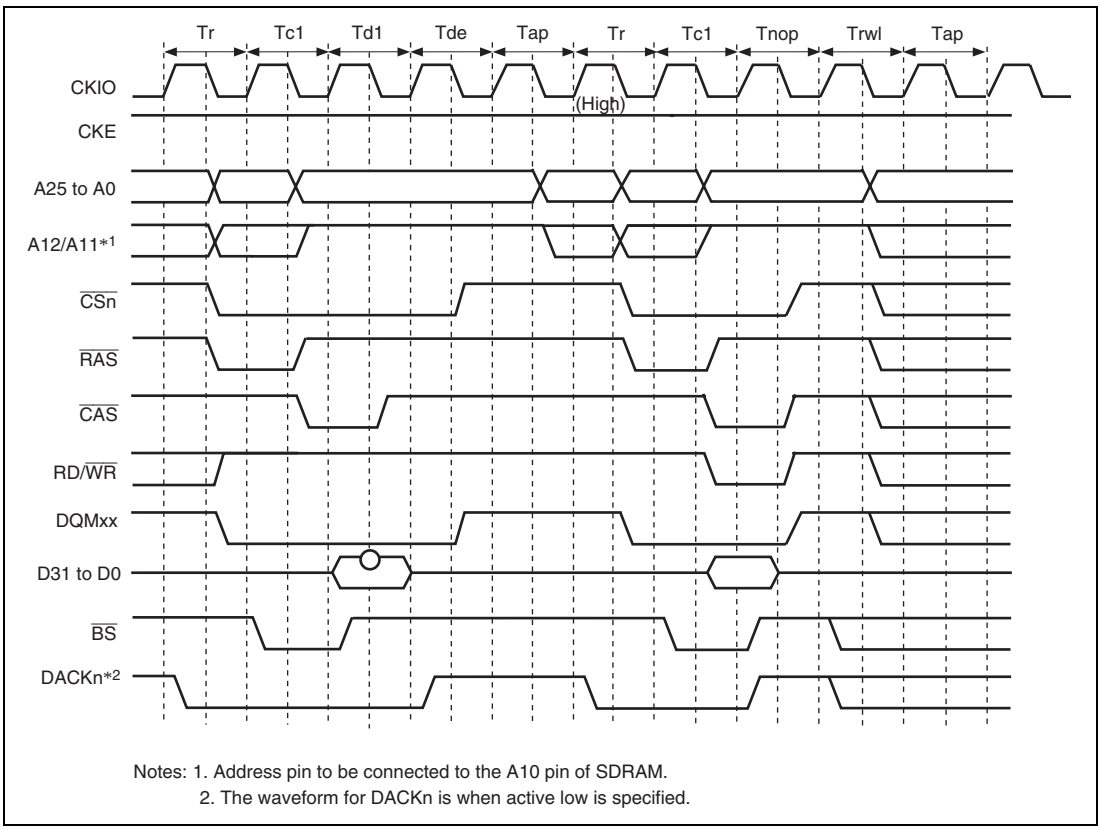
Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired. This LSI supports requests by the $\overline{\text{REFOUT}}$ pin for the bus mastership while waiting for the refresh request. The $\overline{\text{REFOUT}}$ pin is asserted low until the bus mastership is acquired.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring. If a bus mastership is requested during self-refresh, the bus will not be released until the self-refresh is completed.

Low-Frequency Mode: When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 12.27 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CKIO, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CKIO, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.



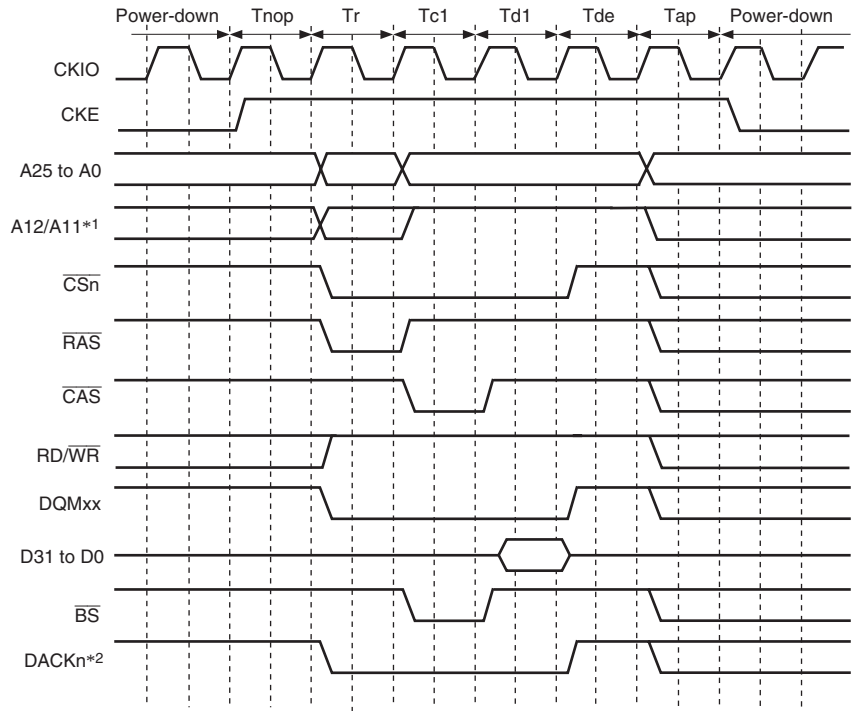
Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.27 Access Timing in Low-Frequency Mode

Power-Down Mode: If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in the power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note

that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle that asserts the CKE in order to cancel power-down mode is inserted.

Figure 12.28 shows the access timing in power-down mode.



Notes: 1.Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.28 Access Timing in Power-Down Mode

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{CSn}$, $\overline{RAS}$, $\overline{CAS}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'A4FD4000 + X for area 2 synchronous DRAM, and to address H'A4FD5000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 to 3,

wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 12.19. In this time 0 is output at the external address pins of A12 or later.

Table 12.19 Access Address in SDRAM Mode Register Write

- Setting for Area 2 (SDMR2)
  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4440	H'0000440
	3	H'A4FD4460	H'0000460
32 bits	2	H'A4FD4880	H'0000880
	3	H'A4FD48C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD4040	H'0000040
	3	H'A4FD4060	H'0000060
32 bits	2	H'A4FD4080	H'0000080
	3	H'A4FD40C0	H'00000C0

- Setting for Area 3 (SDMR3)
  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD5440	H'0000440
	3	H'A4FD5460	H'0000460
32 bits	2	H'A4FD5880	H'0000880
	3	H'A4FD58C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'A4FD5040	H'0000040
	3	H'A4FD5060	H'0000060
32 bits	2	H'A4FD5080	H'0000080
	3	H'A4FD50C0	H'00000C0

Mode register setting timing is shown in figure 12.29. A PALL command (all bank precharge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the TRP[1:0] bits in CSnWCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the TRC[1:0]bits in CSnWCR, are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

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

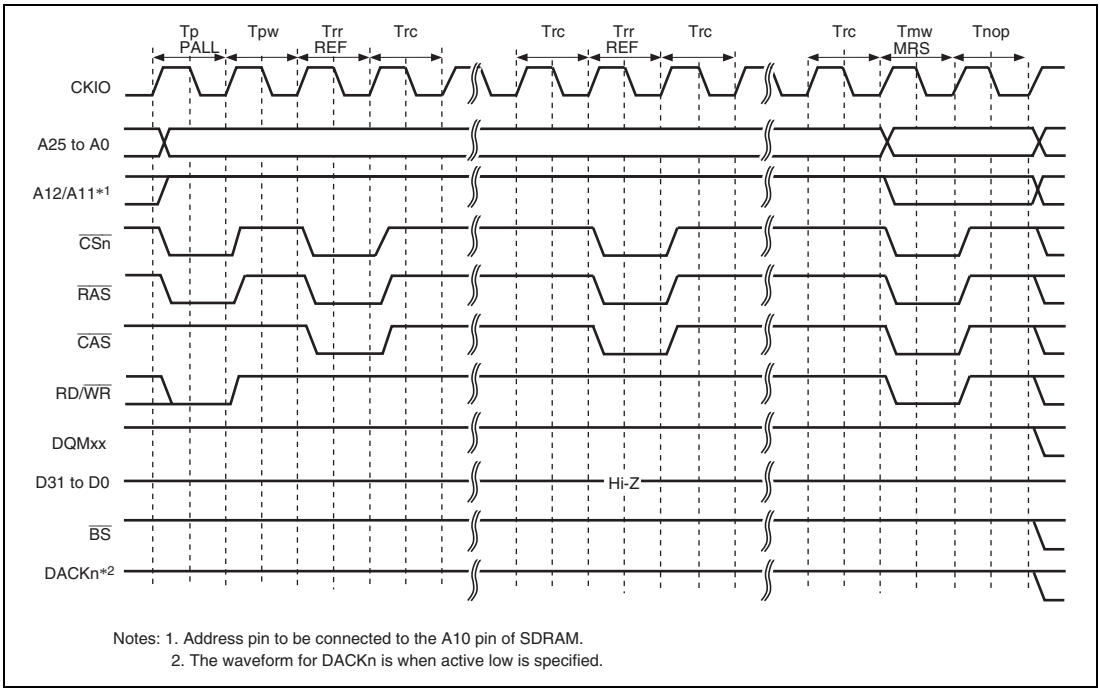


Figure 12.29 Write Timing for SDRAM Mode Register (Based on JEDEC)

Low-Power SDRAM: The low-power SDRAM can be accessed using the same protocol as the normal SDRAM. The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which data in a work area other than the specific area can be lost without severe repercussions. For details, refer to the data sheet for the low-power SDRAM to be used.

The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table 12.20. For example, if data H'0YYYYYYY is written to address H'A4FD5XXX in long-word, the commands are issued to the CS3 space in the following sequence: PALL -> REF x 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XXX and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'A4FD5XXX in long-word, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 12.20 Output Addresses when EMRS Command is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'A4FD4XXX	H'********	16 bits	H'0000XXX	—
CS3 MRS	H'A4FD5XXX	H'********	16 bits	H'0000XXX	—
CS2 MRS +EMRS (with refresh)	H'A4FD4XXX	H'0YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS3 MRS +EMRS (with refresh)	H'A4FD5XXX	H'0YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS2 MRS +EMRS (without refresh)	H'A4FD4XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY
CS3 MRS +EMRS (without refresh)	H'A4FD5XXX	H'1YYYYYYY	32 bits	H'0000XXX	H'YYYYYYY

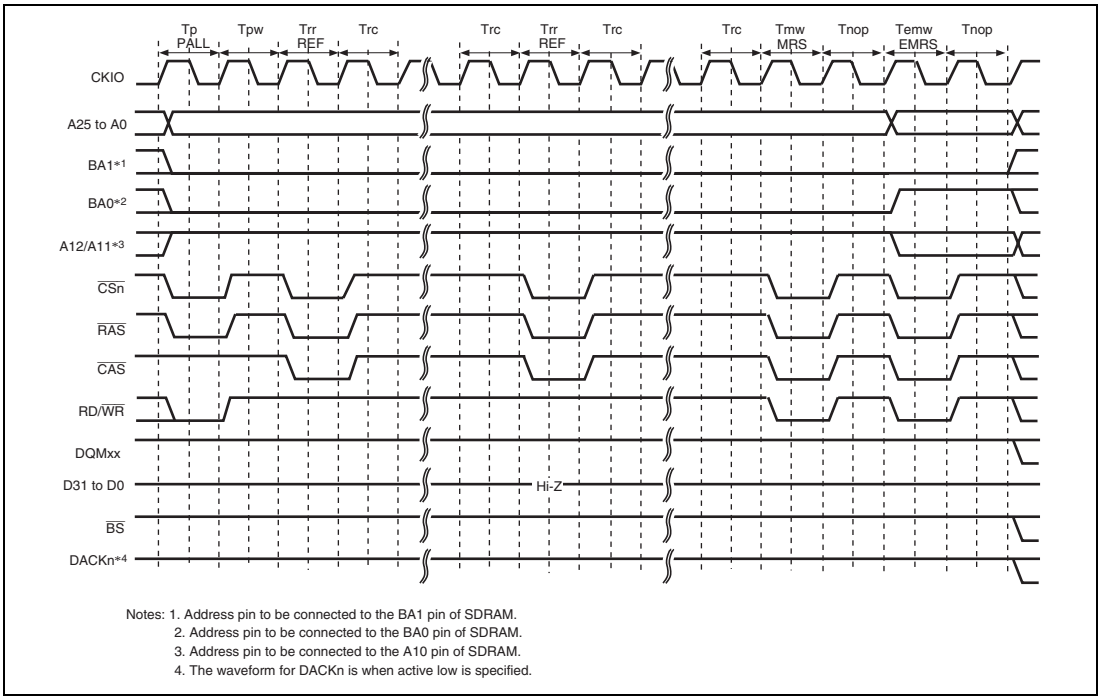
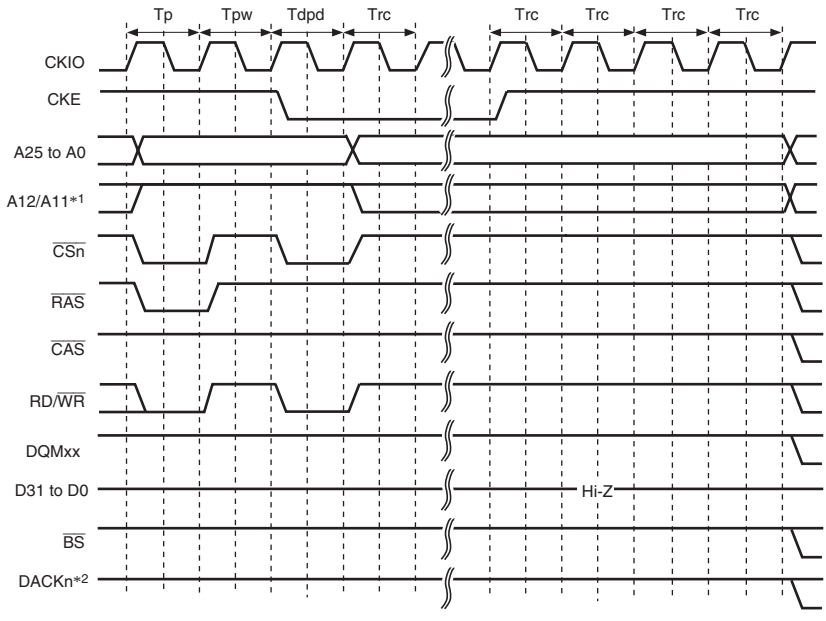


Figure 12.30 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports the deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In the deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit of the SDCR is set to 1 while the DEEP and RFSH bits of the SDCR are set to 1, the low-power SDRAM enters the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.



Notes: 1. Address pin to be connected to the A10 pin of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 12.31 Transition Timing in Deep Power-Down Mode

12.5.6 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent accesses are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent accesses, addresses are changed at the falling edge of the CKIO.

For the 1st access cycle, the number of wait cycles specified by the W[3:0] bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the BW[1:0] bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that do not perform the burst operation in the burst ROM (clock asynchronous) interface, access timing is same as a normal space.

Table 12.21 lists a relationship between bus width, access size, and the number of bursts. Figure 12.32 shows a timing chart.

Table 12.21 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	BEN Bit	Access Size	Number of Bursts	Number of Accesses
8 bits	Not affected	8 bits	1	1
	Not affected	16 bits	2	1
	Not affected	32 bits	4	1
	0	16 bytes	16	1
	1		4	4
16 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	2	1
	0	16 bytes	8	1
	1		2	4
32 bits	Not affected	8 bits	1	1
	Not affected	16 bits	1	1
	Not affected	32 bits	1	1
	Not affected	16 bytes	4	1

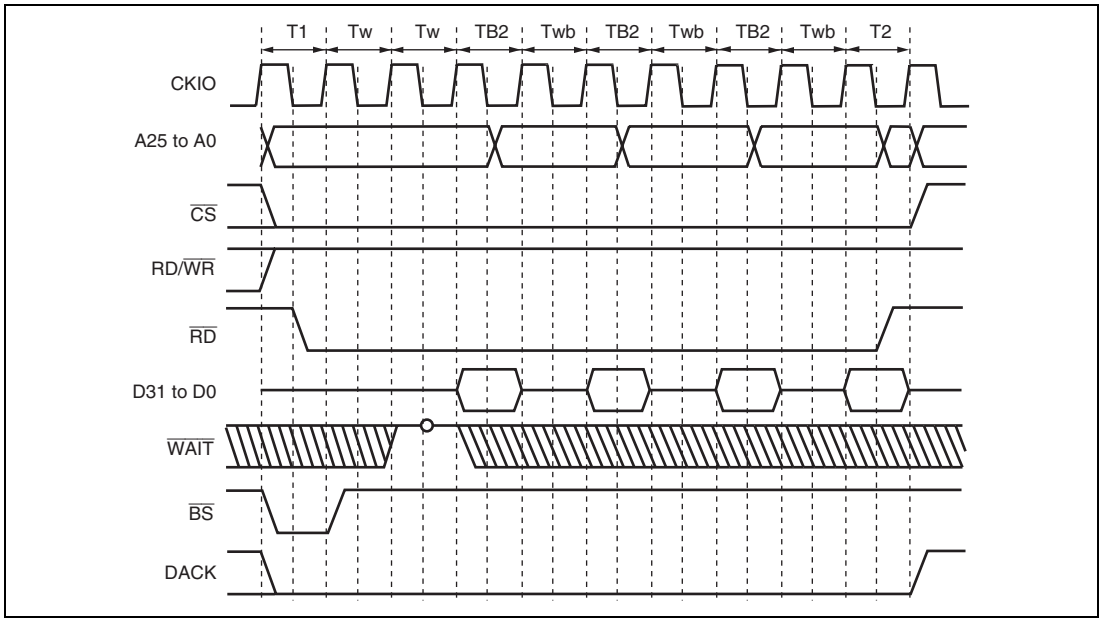


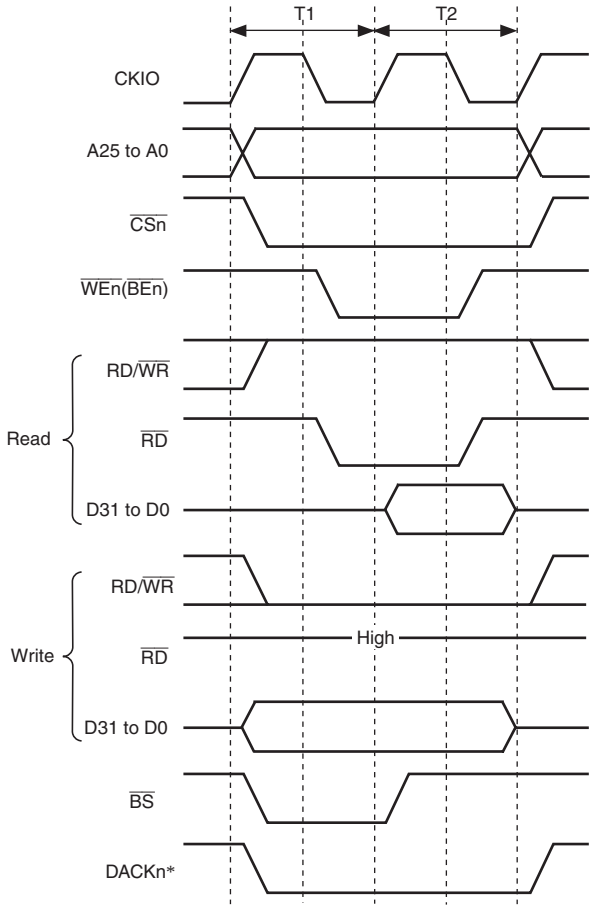
Figure 12.32 Burst ROM (Clock Asynchronous) Access (Bus Width = 32 Bits, 16-byte Transfer (Number of Bursts = 4), Access Wait for First Time = 2, Access Wait for 2nd Time and after = 1)

12.5.7 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

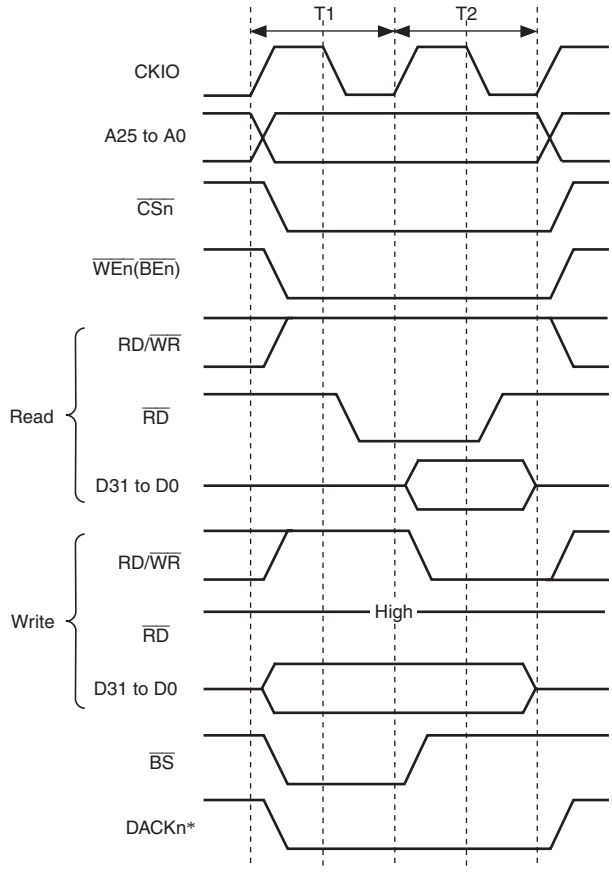
When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the byte-selection SRAM interface is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin, which is different from that for the normal space interface. The basic access timing is shown in figure 12.33. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). For details, refer to the data sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 12.34 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW[1:0] bits in CSnWCR. Figure 12.35 shows the access timing when a software wait is specified.



Note: The waveform for DACKn is when active low is specified.

Figure 12.33 Basic Access Timing for Byte-Selection SRAM (BAS = 0)



Note: The waveform for DACKn is when active low is specified.

Figure 12.34 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

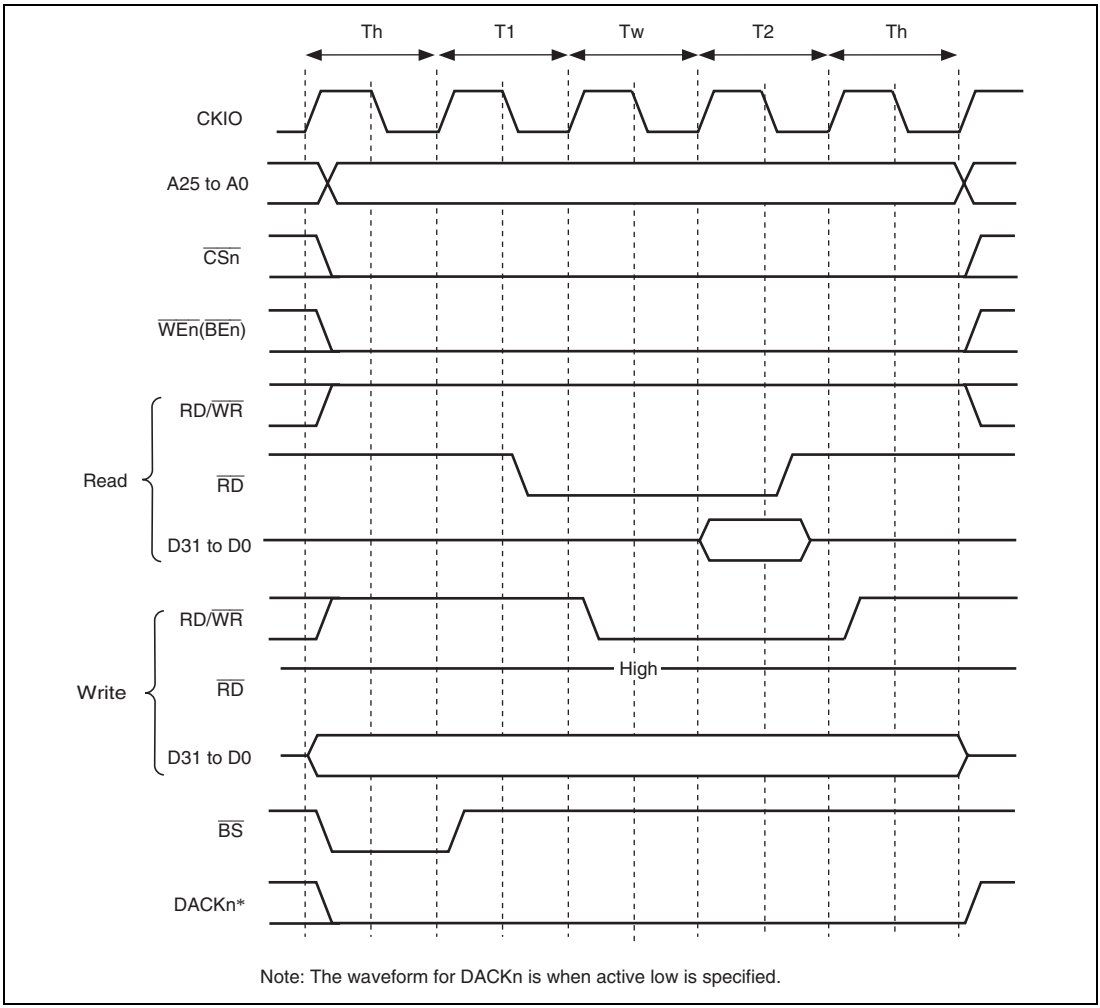


Figure 12.35 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only)

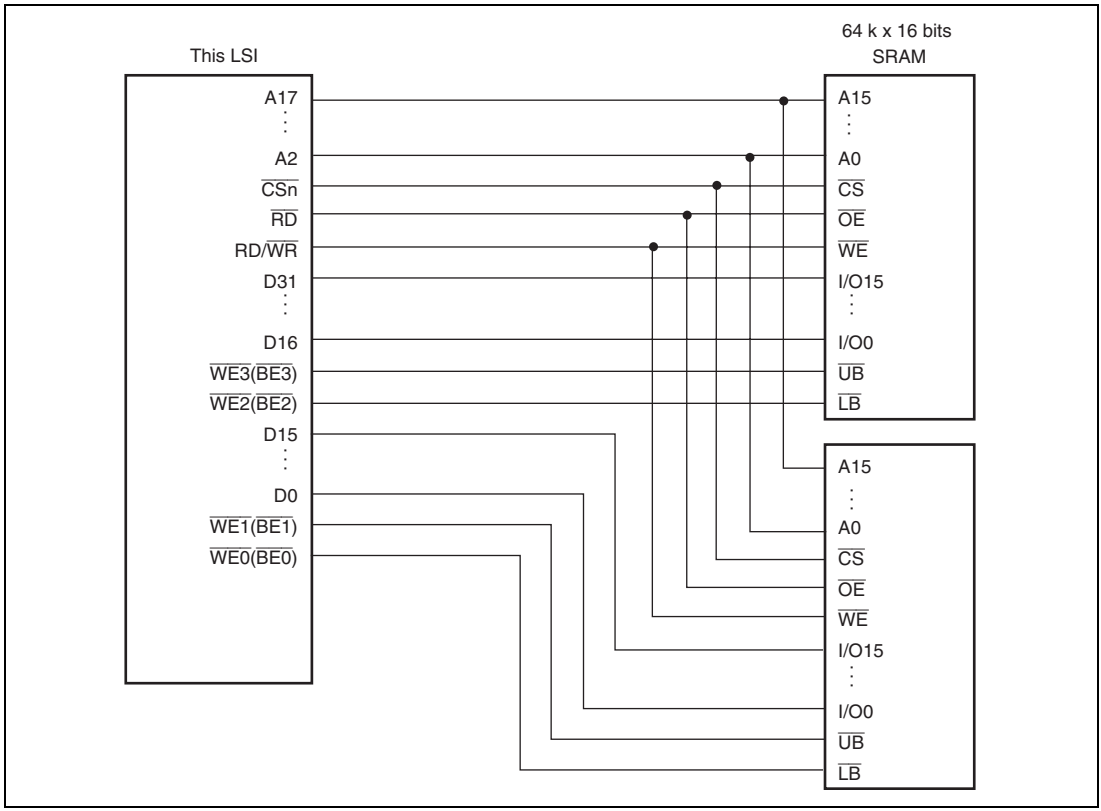


Figure 12.36 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM

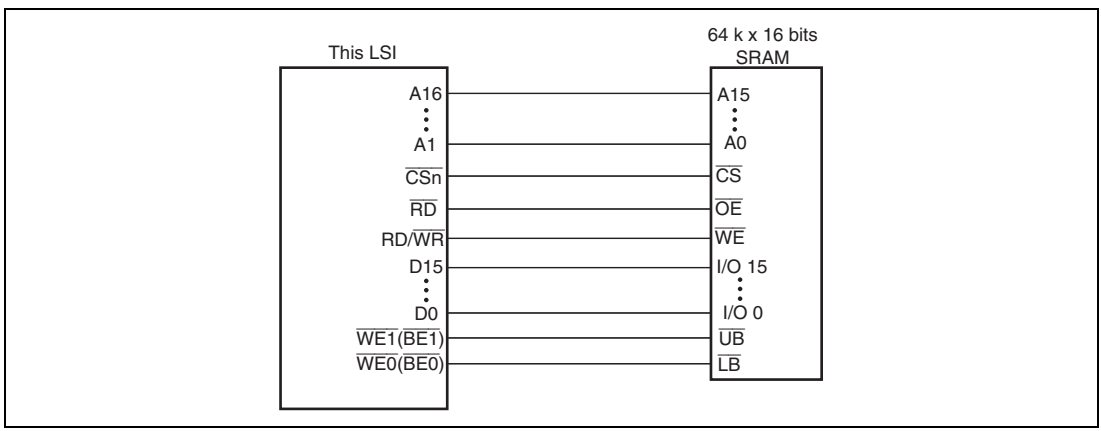


Figure 12.37 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

12.5.8 PCMCIA Interface

With this LSI, if address map (2) is selected using the MAP bit in CMNCR, the PCMCIA interface can be specified in areas 5 and 6. Areas 5 and 6 in the physical space can be used for the IC memory card and I/O card interface defined in the JEIDA specifications version 4.2 (PCMCIA2.1 Rev. 2.1) by specifying the TYPE[3:0] bits of CSnBCR (n = 5B, 6B) to B'0101. In addition, the SA[1:0] bits of CSnWCR (n = 5B, 6B) assign the upper or lower 32 Mbytes of each area to an IC memory card or I/O card interface. For example, if the SA1 and SA0 bits of the CS5BWCR are set to 1 and cleared to 0, respectively, the upper 32 Mbytes and the lower 32 Mbytes of area 5B are used as an IC memory card interface and I/O card interface, respectively.

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits using the BSZ[1:0] bits in CS5BBCR or CS6BBCR.

Figure 12.38 shows an example of a connection between this LSI and the PCMCIA card. To enable insertion and removal of the PCMCIA card during system power-on, a three-state buffer must be connected between the LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in the big endian mode is not clearly defined. Consequently, an original definition is provided for the PCMCIA interface in big endian mode in this LSI.

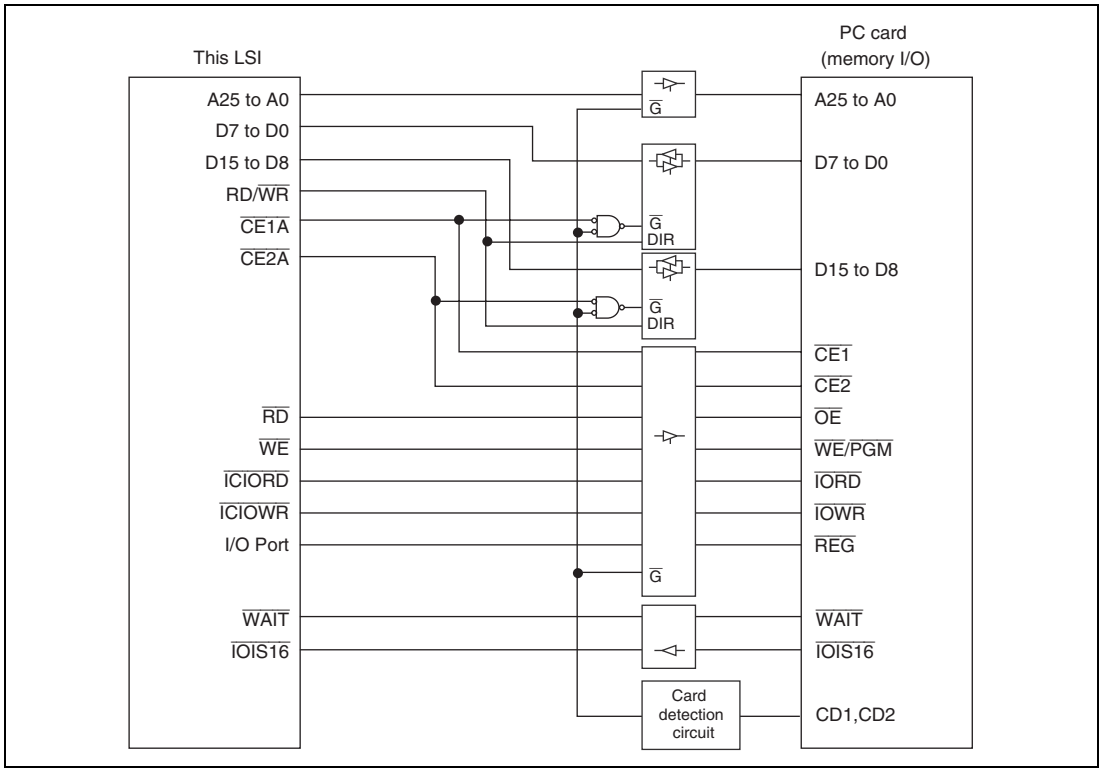


Figure 12.38 Example of PCMCIA Interface Connection

Basic Timing for Memory Card Interface: Figure 12.39 shows the basic timing of the PCMCIA IC memory card interface. If areas 5 and 6 in the physical space are specified as the PCMCIA interface, accessing the common memory areas in areas 5 and 6 automatically accesses the IC memory card interface. If the external bus frequency (CKIO) increases, the setup times and hold times for the address pins (A25 to A0) to RD and WE, card enable signals ($\overline{CE1A}$, $\overline{CE2A}$, $\overline{CE1B}$, $\overline{CE2B}$), and write data (D15 to D0) become insufficient. To prevent this error, the LSI can specify the setup times and hold times for areas 5 and 6 in the physical space independently, using CS5BWCR and CS6BWCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait can be inserted using the WAIT pin. Figure 12.40 shows the PCMCIA memory bus wait timing.

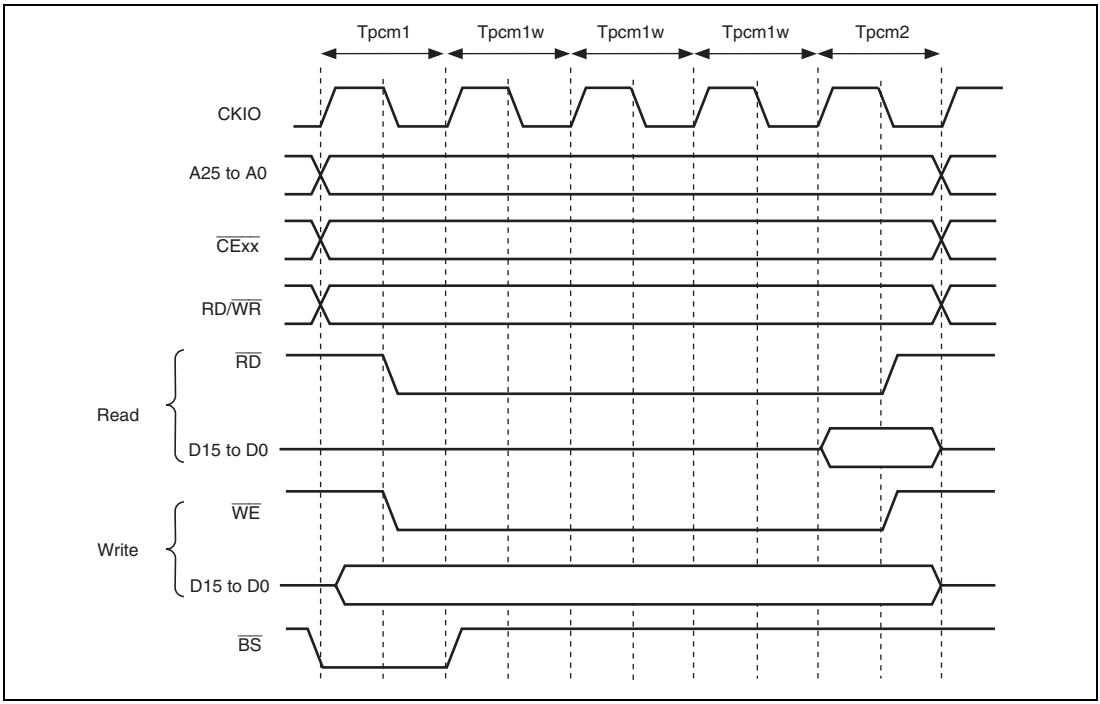


Figure 12.39 Basic Access Timing for PCMCIA Memory Card Interface

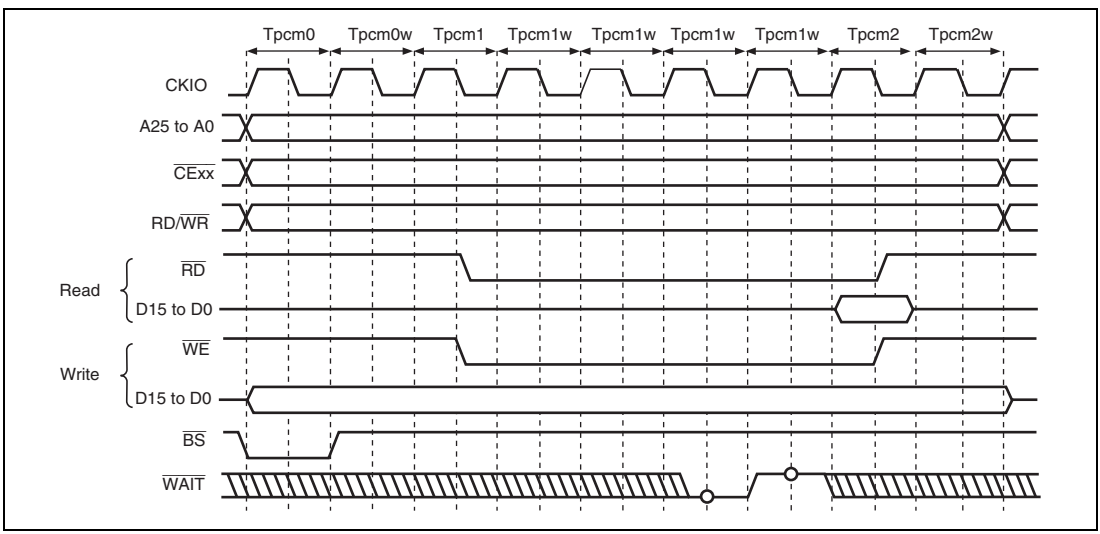


Figure 12.40 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait = 1)

If all 32 Mbytes of the memory space are used as an IC memory card interface, the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory can be generated by a port. If the memory space used for the IC memory card interface is 16 Mbytes or less, the A24 pin can be used as the $\overline{\text{REG}}$ signal by using the memory space as a 16-Mbyte common memory space and a 16-Mbyte attribute memory space.

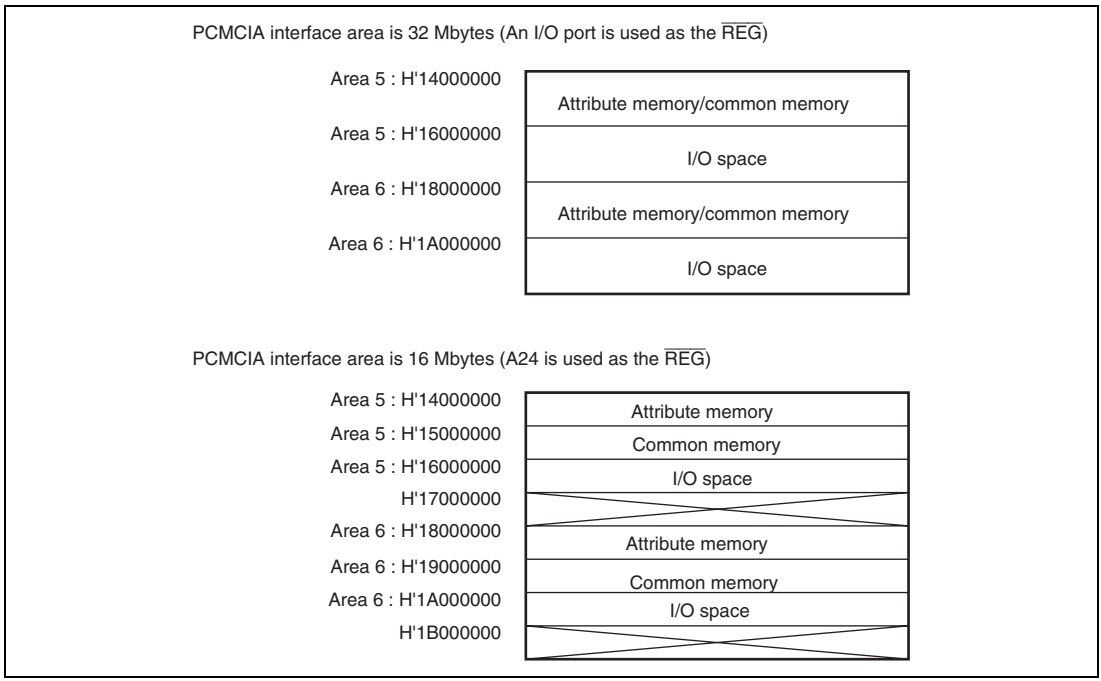


Figure 12.41 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B′10, CS6BWCR.SA[1:0] = B′10)

Basic Timing for I/O Card Interface: Figures 12.42 and 12.43 show the basic timings for the PCMCIA I/O card interface.

The I/O card and IC memory card interfaces can be switched using an address to be accessed. If area 5 of the physical space is specified as the PCMCIA, the I/O card interface can automatically be accessed by accessing the physical addresses from H’16000000 to H'17FFFFFF. If area 6 of the physical space is specified as the PCMCIA, the I/O card interface can automatically be accessed by accessing the physical addresses from H′1A000000 to H′1BFFFFFF.

Note that areas to be accessed as the PCMCIA I/O card must be non-cached if they are logical space (space P2 or P3) areas, or a non-cached area specified by the MMU.

If the PCMCIA card is accessed as an I/O card in little endian mode, dynamic bus sizing for the I/O bus can be achieved using the $\overline{\text{IOIS16}}$ signal. If the $\overline{\text{IOIS16}}$ signal is brought high in a word-size I/O bus cycle while the bus width of area 6 is specified as 16 bits, the bus width is recognized as 8 bits and data is accessed twice in 8-bit units in the I/O bus cycle to be executed.

The $\overline{\text{IOIS16}}$ signal is sampled at the falling edge of CKIO in the Tpci0, Tpci0w, and Tpci1 cycles when the TED[3:0] bits are specified as 1.5 cycles or more, and is reflected in the CE2 signal 1.5 cycles after the CKIO sampling point. The TED[3:0] bits must be specified appropriately to satisfy the setup time from $\overline{\text{ICIORD}}$ and $\overline{\text{ICIOWR}}$ of the PC card to CEn.

Figure 12.44 shows the dynamic bus sizing basic timing.

Note that the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode. In the big endian mode, the $\overline{\text{IOIS16}}$ signal must be fixed low.

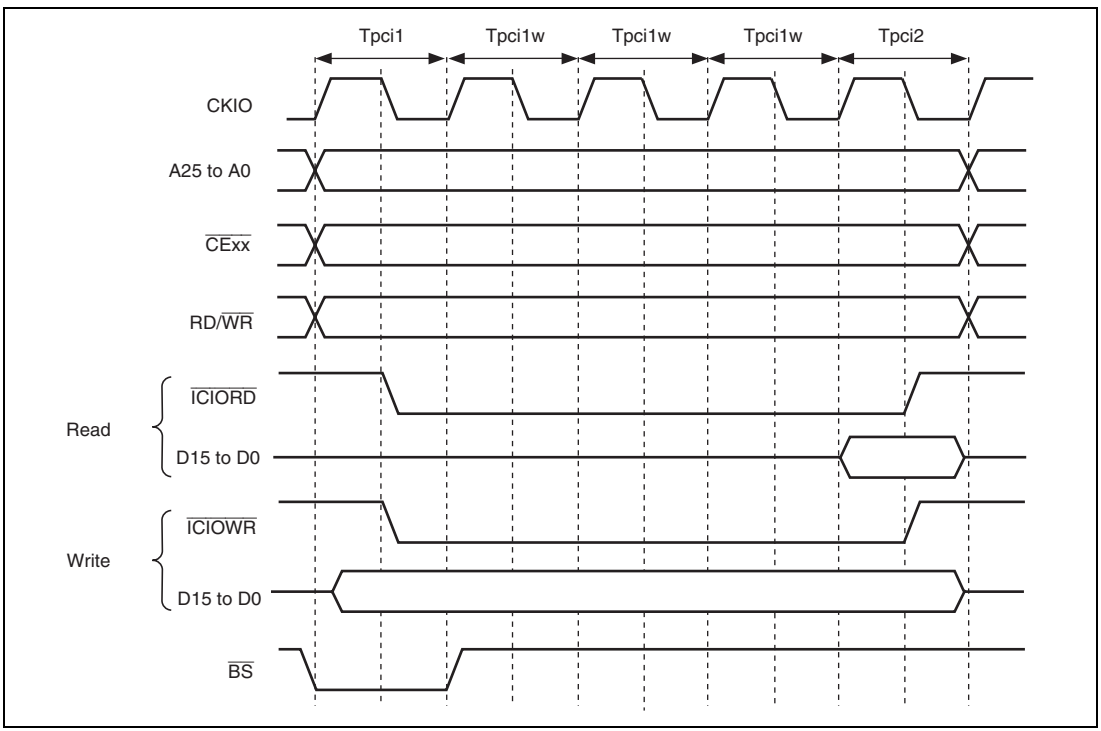


Figure 12.42 Basic Timing for PCMCIA I/O Card Interface

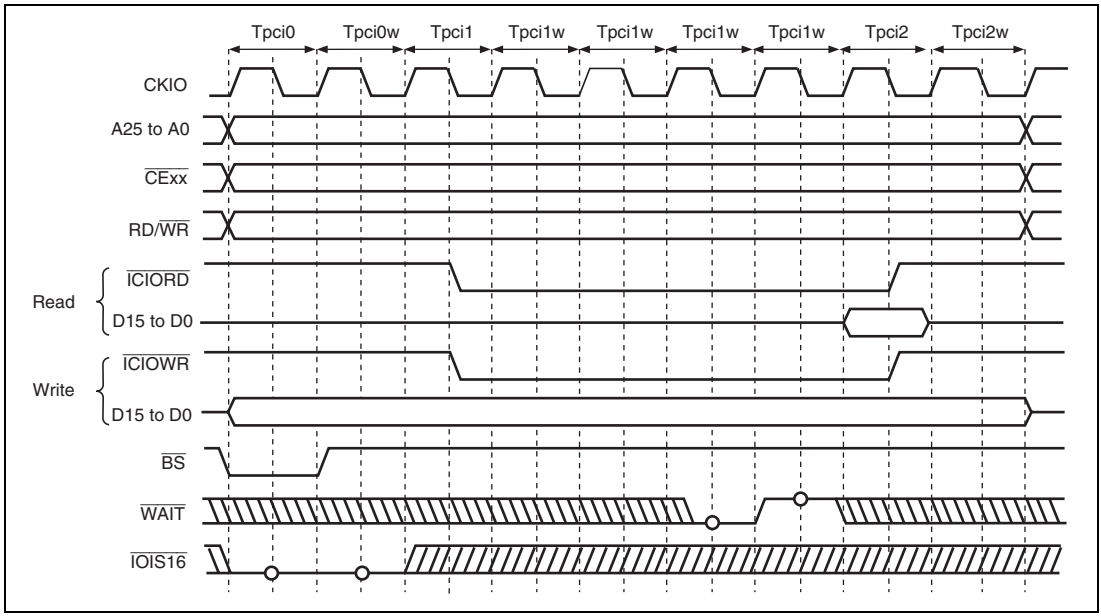


Figure 12.43 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Wait = 1, Hardware Wait = 1)

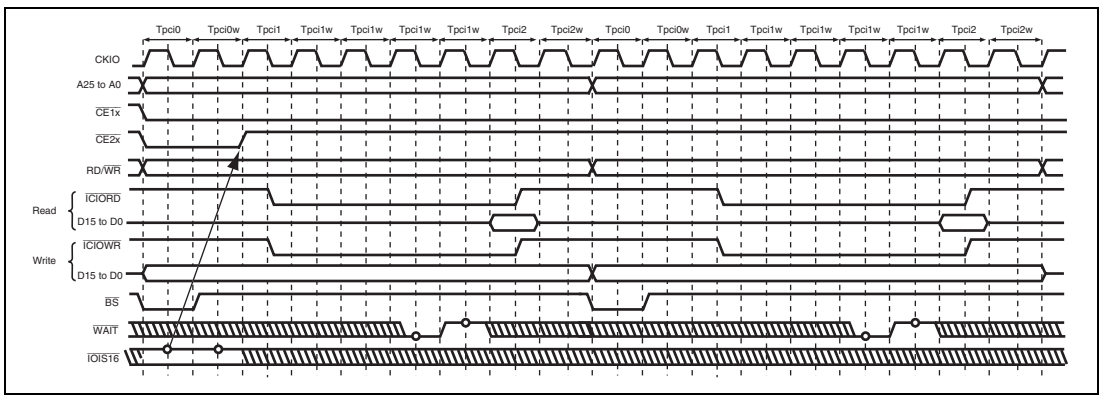


Figure 12.44 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B′0010, TEH[3:0] = B′0001, Software Waits = 3)

12.5.9 Burst ROM (Clock Synchronous) Interface

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W[3:0] bits of the CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW[1:0] bits of the CS0WCR.

While the burst ROM is accessed (clock synchronous), the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. If the bus width is 32 bits, the burst length must be specified as 4. The burst ROM interface does not support the 8-bit bus width for the burst ROM. The burst ROM interface performs burst operations for all read accesses. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times.

These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, a 16-byte read by cache fill or 16-byte read by the DMA should be used. The burst ROM interface performs write accesses in the same way as normal space access.

Note: The burst ROM (clock synchronous) must be accessed as cacheable space.

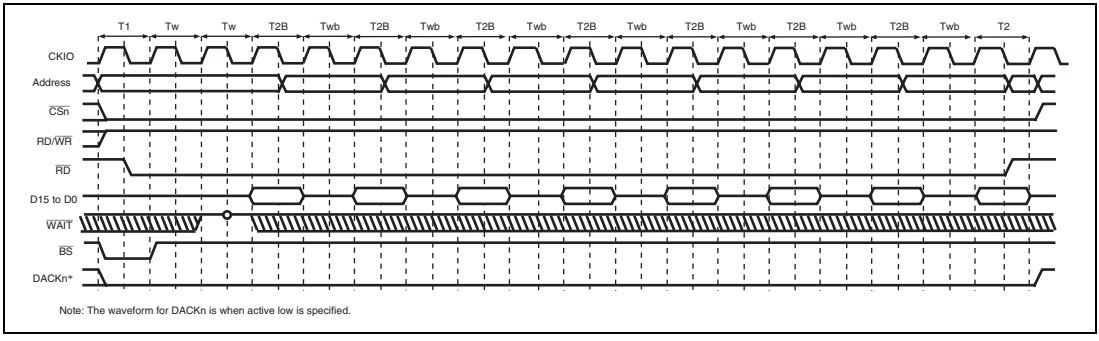


Figure 12.45 Burst ROM (Clock Synchronous) Access Timing (Burst Length = 8, Wait Cycles inserted in First Access = 2, Wait Cycles inserted in Second and Subsequent Accesses = 1)

12.5.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. This LSI has a function that avoids data collisions by inserting wait cycles between continuous access cycles.

The number of wait cycles between access cycles can be set by bits IWW[2:0], IWRWD[2:0], IWRWS[2:0], IWRRD[2:0], and IWRRS[2:0] in CSnBCR, and bits DMAIW[2:0] and DMAIWA in CMNCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Continuous accesses are write-read or write-write
2. Continuous accesses are read-write for different spaces
3. Continuous accesses are read-write for the same space
4. Continuous accesses are read-read for different spaces
5. Continuous accesses are read-read for the same space
6. Data output from an external device caused by DMA single transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single transfer is followed by any type of access (DMAIWA = 1)

12.5.11 Bus Arbitration

To prevent device malfunction while the bus mastership is transferred between master and slave, the LSI negates all of the bus control signals before bus release. When the bus mastership is received, all of the bus control signals are first negated and then driven appropriately. In this case, output buffer contention can be prevented because the master and slave drive the same signals with the same values. In addition, to prevent noise while the bus control signal is in the high impedance state, pull-up resistors must be connected to these control signals.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CSn}}$

signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. 16-byte transfer because of a cache miss
2. During copyback operation for the cache
3. Between the read and write cycles of a TAS instruction
4. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
5. 16-byte transfer by the DMAC or E-DMAC
6. Setting the BLOCK bit in CMNCR to 1

Bits DPRTY[1:0] in CMNCR can select whether or not the bus request is received during DMAC burst transfer.

This LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the slave has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

The SDRAM issues a all bank precharge command (PALL) when active banks exist and releases the bus after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CKIO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CS}}$n, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, DQMxx, $\overline{\text{WE}}$n ($\overline{\text{BE}}$n), $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-impedance state at subsequent rising edges of CKIO. Bus request signals are sampled at the falling edge of CKIO.

The sequence for reclaiming the bus mastership from a slave is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ is detected at the falling edge of CKIO, the bus control signals are driven high. The $\overline{\text{BACK}}$ is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CKIO where address and data signals are driven. Figure 12.46 shows the bus arbitration timing.

In an original slave device designed by the user, multiple bus accesses are generated continuously to reduce the overhead caused by bus arbitration. In this case, to execute SDRAM refresh correctly, the slave device must be designed to release the bus mastership within the refresh interval time. To achieve this, the LSI instructs the $\overline{\text{REFOUT}}$ pin to request the bus mastership

while the SDRAM waits for the refresh. The LSI asserts the $\overline{\text{REFOUT}}$ pin until the bus mastership is received. If the slave releases the bus, the LSI acquires the bus mastership to execute the SDRAM refresh.

The bus release by the $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ signal handshaking requires some overhead. If the slave has many tasks, multiple bus cycles should be executed in a bus mastership acquisition. Reducing the cycles required for master to slave bus mastership transitions streamlines the system design.

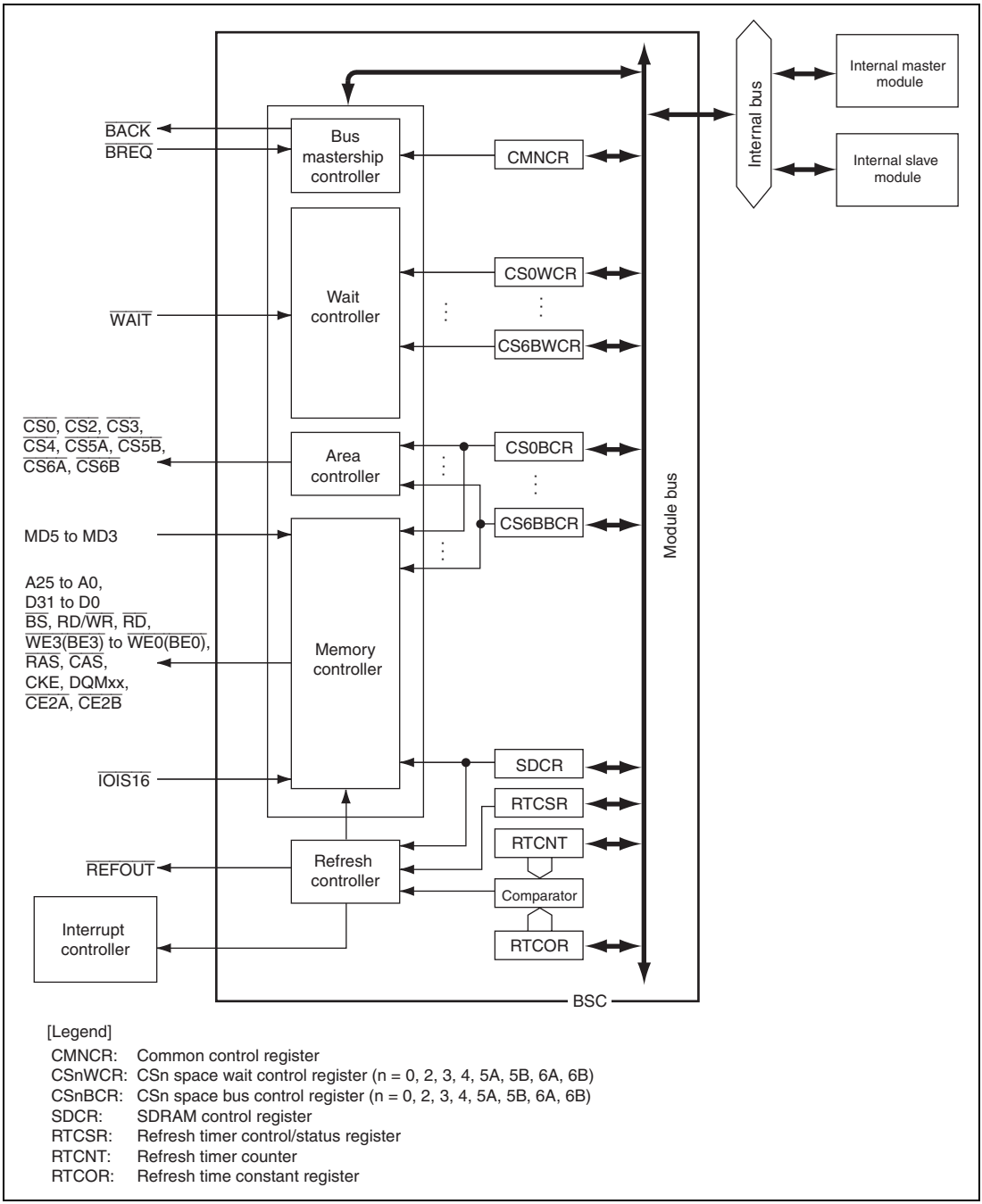


Figure 12.46 Bus Arbitration Timing

12.5.12 Others

Reset: The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are negated and output buffers are turned off regardless of the bus cycle state. All control registers are initialized. In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, the current bus cycle being executed is completed and then the access wait state is entered. If a 16-byte transfer is performed by a cache or if another LSI on-chip bus master module is executed when a manual reset occurs, the current access is cancelled in longword units because the access request is cancelled by the bus master at manual reset. If a manual reset is requested during cache fill operations, the contents of the cache cannot be guaranteed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle. Note, however, a bus arbitration request by the $\overline{\text{BREQ}}$ signal can't be accepted during manual reset signal assertion.

Some flash memories may specify a minimum time from reset release to the first access. To ensure this minimum time, the bus state controller supports a 5-bit reset wait counter (RWTCNT). At power-on reset, the RWTCNT is cleared to 0. After a power-on reset, RWTCNT is counted up synchronously together with CKIO and an external access will not be generated until RWTCNT is counted up to H′007F. At a manual reset, RWTCNT is not cleared. RWTCNT cannot be read from or written to.

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

Internal bus masters such as DMAC or E-DMAC other than the CPU can access on-chip memory other than the cache memory but cannot access the cache memory. If an on-chip bus master other than the CPU writes data to an external memory other than the cache, the contents of the external memory may differ from that of the cache memory. To prevent this problem, if the external memory whose contents is cached is written by an on-chip bus master other than the CPU, the corresponding cache memory should be purged by software.

If the CPU initiates read access for the cache, the cache is searched. If the cache stores data, the CPU latches the data and completes the read access. If the cache does not store data, the CPU performs four contiguous longword read cycles to perform cache fill operations via the internal bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word boundary

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

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

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

In write-back mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is then re-written to the cache. In the actual memory, data will not be re-written until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is first saved to the internal buffer, 16-byte data including the data corresponding to the address is then read, and data in the corresponding access of the cache is finally modified. Following these operations, a write-back cycle for the saved 16-byte data is executed.

In write-through mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is re-written to the cache simultaneously with the actual write via the internal bus. If data is not detected at the address corresponding to the cache, the cache is not modified but an actual write is performed via the internal bus.

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC or E-DMAC. Accordingly, to perform dual address DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next write cycle will not be initiated until the previous write cycle is completed.

On-Chip Peripheral Module Access: To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

Section 13 Direct Memory Access Controller (DMAC)

The direct memory access controller (DMAC) can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

13.1 Features

- Six channels (Two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, Word (two bytes), Longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests:
  An external request, on-chip peripheral module request, or auto request can be selected.
  The following modules can issue an on-chip peripheral module request.
  SCIF0, SCIF1, SIOF0, SIOF1
- Bus mode: Cycle steal mode or burst mode can be selected.
- 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 at the end of the specified counts of data transfer.
- External request detection: There are following four types of DREQ input detection.
  Low-level detection
  High-level detection
  Rising-edge detection
  Falling-edge detection
- Transfer request acknowledge and transfer end signals: Active levels for DACK and TEND can be set independently.

Figure 13.1 shows a block diagram of the DMAC.

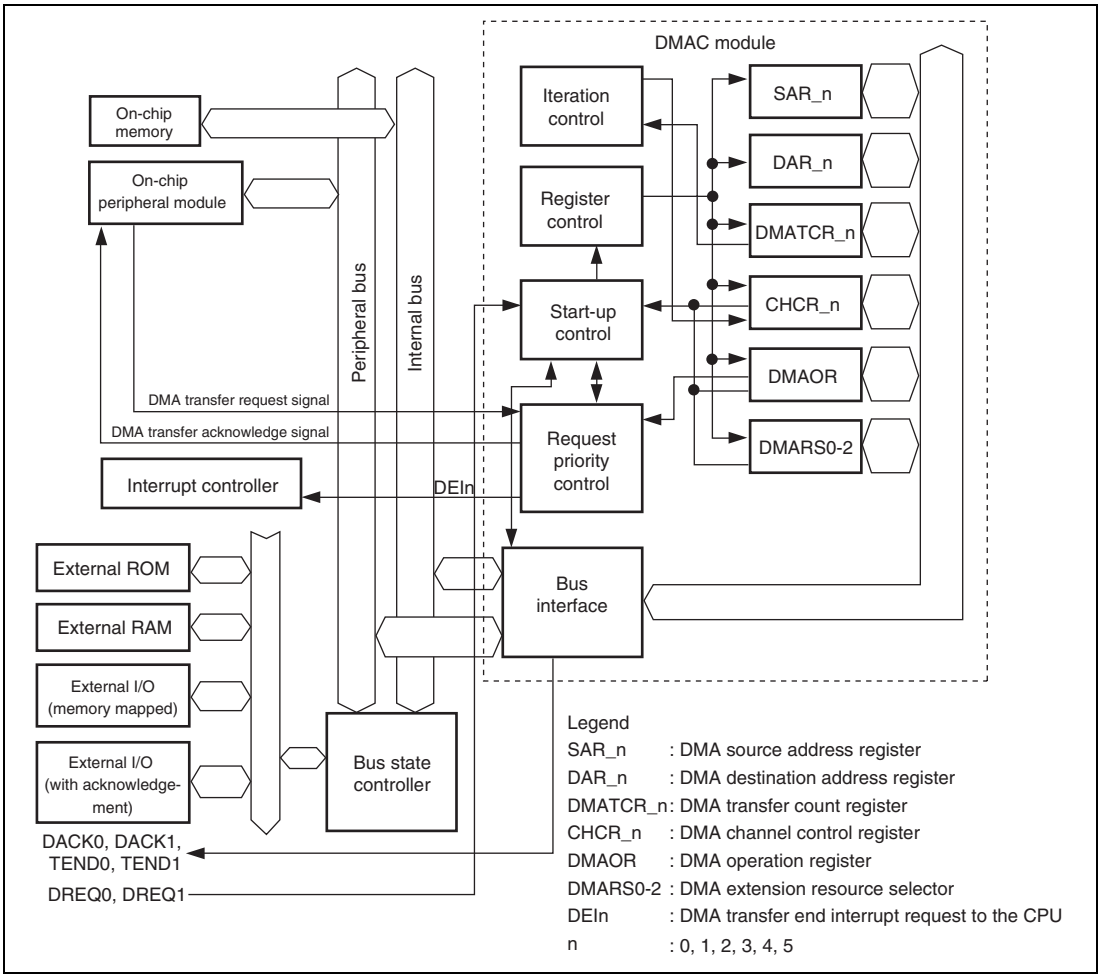


Figure 13.1 Block Diagram of DMAC

13.2 Input/Output Pins

The external pins for the DMAC are described below.

Table 13.1 lists the configuration of the pins that are connected to external bus. The DMAC has pins for 2 channels (channels 0 and 1) for external bus use.

Table 13.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknowledge output from channel 0 to external device
	DMA transfer end	TEND0	O	DMA transfer end output for channel 0
1	DMA transfer request	DREQ1	I	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request acknowledge output from channel 1 to external device
	DMA transfer end	TEND1	O	DMA transfer end output for channel 1

13.3 Register Descriptions

The DMAC has the following registers. Refer to section 24, List of Registers, for the addresses and access size of these registers.

Channel 0:

- DMA source address register_0 (SAR_0)
- DMA destination address register_0 (DAR_0)
- DMA transfer count register_0 (DMATCR_0)
- DMA channel control register_0 (CHCR_0)

Channel 1:

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

Channel 2:

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

Channel 3:

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

Channel 4:

- DMA source address register_4 (SAR_4)
- DMA destination address register_4 (DAR_4)
- DMA transfer count register_4 (DMATCR_4)
- DMA channel control register_4 (CHCR_4)

Channel 5:

- DMA source address register_5 (SAR_5)
- DMA destination address register_5 (DAR_5)
- DMA transfer count register_5 (DMATCR_5)
- DMA channel control register_5 (CHCR_5)

Common:

- DMA operation register (DMAOR)
- DMA extension resource selector 0 (DMARS0)
- DMA extension resource selector 1 (DMARS1)
- DMA extension resource selector 2 (DMARS2)

13.3.1 DMA Source Address Register (SAR)

SAR is a 32-bit readable/writable register that specifies the source address of a DMA transfer. During a DMA transfer, SAR indicates the next source address. When the data is transferred from an external device with the DACK in single address mode, SAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value.

SAR is undefined at reset and retains the current value in standby or module standby mode.

13.3.2 DMA Destination Address Register (DAR)

DAR is a 32-bit readable/writable register that specifies the destination address of a DMA transfer. During a DMA transfer, DAR indicates the next destination address. When the data is transferred to an external device with the DACK in single address mode, DAR is ignored.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value.

DAR is undefined at reset and retains the current value in standby or module standby mode.

13.3.3 DMA Transfer Count Register (DMATCR)

DMATCR is a 32-bit readable/writable registers that specifies 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, DMATCR indicates the remaining transfer count.

The upper eight bits of DMATCR will return 0 if read, and should only be written with 0.

To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

DMATCR is undefined at reset and retains the current value in standby or module standby mode.

13.3.4 DMA Channel Control Register (CHCR)

CHCR is a 32-bit readable/writable register that controls the DMA transfer mode.

CHCR is initialized to H'00000000 at reset and retains the current value in the standby or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether the DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR0 and CHCR1.This bit is reserved and always read as 0 in CHCR2 to CHCR5. The write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1

Bit	Bit Name	Initial Value	R/W	Description
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high active or low active. This bit is valid only in CHCR0 and CHCR1.This bit is reserved and always read as 0 in CHCR2 to CHCR5. The write value should always be 0. 0: Low-active output of TEND 1: High-active output of TEND
21 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	AM	0	R/W	Acknowledge Mode Specifies whether the DACK is output in data read cycle or in data write cycle in dual address mode. In single address mode, the DACK is always output regardless of the specification by this bit. This bit is valid only in CHCR0 and CHCR1.This bit is reserved and always read as 0 in CHCR2 to CHCR5. The write value should always be 0. 0: DACK output in read cycle (Dual address mode) 1: DACK output in write cycle (Dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK signal output is high active or low active. This bit is valid only in CHCR0 and CHCR1.This bit is reserved and always read as 0 in CHCR2 to CHCR5. The write value should always be 0. 0: Low-active output of DACK 1: High-active output of DACK

Bit	Bit Name	Initial Value	R/W	Description
15	DM1	0	R/W	Destination Address Mode
14	DM0	0	R/W	Specify whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, the DM1 and DM0 bits are ignored when data is transferred to an external device with the DACK.) 00: Fixed destination address (setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in byte transfer, +2 in word transfer, +4 in longword transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in byte transfer, –2 in word transfer, –4 in longword transfer; setting prohibited in 16-byte transfer) 11: Reserved (setting prohibited)
13	SM1	0	R/W	Source Address Mode
12	SM0	0	R/W	Specify whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, the SM1 and SM0 bits are ignored when data is transferred from an external device with the DACK.) 00: Fixed source address (setting prohibited in 16-byte transfer) 01: Source address is incremented (+1 in byte transfer, +2 in word transfer, +4 in longword transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in byte transfer, –2 in word transfer, –4 in longword transfer; setting prohibited in 16-byte transfer) 11: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
11	RS3	0	R/W	Resource Select
10	RS2	0	R/W	Specify which transfer requests will be sent to the DMAC. The change of transfer request source should be done in the state that the DMA enable bit (DE) is cleared to 0.
9	RS1	0	R/W	
8	RS0	0	R/W	0000: External request, dual address mode 0010: External request, single address mode External address space → external device with DACK 0011: External request, single address mode External device with DACK → external address space 0100: Auto request 1000: DMA extension resource selector Other than above: Reserved (setting prohibited) Note: An external request specification is valid only in CHCR0 and CHCR1. None of the external request specification can be selected in CHCR2 to CHCR5.
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	Specify the sampling method of the DREQ pin input and the sampling level. These bits are valid only in CHCR0 and CHCR1. These bits are reserved and always read as 0 in CHCR2 to CHCR5. The write value should always be 0. In channels 0 and 1, also, if the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by this bit is invalid. 00: DREQ detected in low level 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 the DMA transfers data. 0: Cycle steal mode 1: Burst mode

Bit	Bit Name	Initial Value	R/W	Description
4	TS1	0	R/W	Transfer Size
3	TS0	0	R/W	Specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte size 01: Word (two bytes) size 10: Longword (four bytes) size 11: 16-byte (four longwords) size
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when the TE bit is set to 1. 0: Interrupt request is disabled 1: Interrupt request is enabled
1	TE	0	R/(W)*	Transfer End Flag The TE bit is set to 1 when data transfer ends when DMATCR becomes to 0. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit and DME bit in the DMA operation register (DMAOR). 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 1: DMA transfer ends by the specified count (DMATCR = 0) [Clearing condition] Writing 0 after reading 1 from this bit

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

13.3.5 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register indicates the DMA transfer status.

DMAOR is initialized to H′0000 at a reset and retains the current value in standby or module standby mode.

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	PR1	0	R/W	Priority Mode
8	PR0	0	R/W	Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 01: Fixed mode 2: CH0 > CH2 > CH3 > CH1 > CH4 > CH5 10: Reserved (setting prohibited) 11: All channel round-robin mode

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	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred by the DMAC. When 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. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error [Clear conditions] Writing 0 after reading 1 from this bit
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. When 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. This bit can only be cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in progress can be done in one transfer unit. Even if the DMAC is not in operational, this bit is set to 1 when the NMI interrupt was input. 0: No NMI interrupt 1: NMI interrupt occurs [Clearing conditions] Writing 0 after reading 1 from this bit
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in CHCR are set to 1, DMA transfer is enabled. Note that transfer is enabled if the TE bit in CHCR and the NMIF and AE bits in DMAOR are all 0. If this bit is cleared, DMA transfers in all the channels can be terminated. 0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels

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

13.3.6 DMA Extension Resource Selector 0 to 2 (DMARS0 to DMARS2)

DMARS is a 16-bit readable/writable register that specifies 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 requests of the SCIF0, SCIF1, SIOF0, and SIOF1.

When MID and RID other than the values listed in table 13.2 are set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when bits RS3 to RS0 has been set to B'1000 in CHCR0 to CHCR5. Otherwise, even if DMARS has been set, a transfer request source is not accepted.

DMARS is initialized to H'0000 at reset and retains the current value in standby or module standby mode.

- DMARS0

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

- DMARS1

Bit	Bit Name	Initial Value	R/W	Description
15	C3MID5	0	R/W	Transfer request module ID for DMA channel 3 (MID)
14	C3MID4	0	R/W	See table 13.2.
13	C3MID3	0	R/W	
12	C3MID2	0	R/W	
11	C3MID1	0	R/W	
10	C3MID0	0	R/W	
9	C3RID1	0	R/W	Transfer request module ID for DMA channel 3 (RID)
8	C3RID0	0	R/W	See table 13.2.
7	C2MID5	0	R/W	Transfer request module ID for DMA channel 2 (MID)
6	C2MID4	0	R/W	See table 13.2.
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request module ID for DMA channel 2 (RID)
0	C2RID0	0	R/W	See table 13.2.

- DMARS2

Bit	Bit Name	Initial Value	R/W	Description
15	C5MID5	0	R/W	Transfer request module ID for DMA channel 5 (MID)
14	C5MID4	0	R/W	See table 13.2.
13	C5MID3	0	R/W	
12	C5MID2	0	R/W	
11	C5MID1	0	R/W	
10	C5MID0	0	R/W	
9	C5RID1	0	R/W	Transfer request module ID for DMA channel 5 (RID)
8	C5RID0	0	R/W	See table 13.2.
7	C4MID5	0	R/W	Transfer request module ID for DMA channel 4 (MID)
6	C4MID4	0	R/W	See table 13.2.
5	C4MID3	0	R/W	
4	C4MID2	0	R/W	
3	C4MID1	0	R/W	
2	C4MID0	0	R/W	
1	C4RID1	0	R/W	Transfer request module ID for DMA channel 4 (RID)
0	C4RID0	0	R/W	See table 13.2.

Transfer requests from the various modules are specified by the MID and RID as shown in table 13.2.

Table 13.2 DMARS Setting

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	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
SIOF0	H'51	B'010100	B'01	Transmit
	H'52		B'10	Receive
SIOF1	H'55	B'010101	B'01	Transmit
	H'56		B'10	Receive

13.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In the bus mode, the burst mode or the cycle steal mode can be selected.

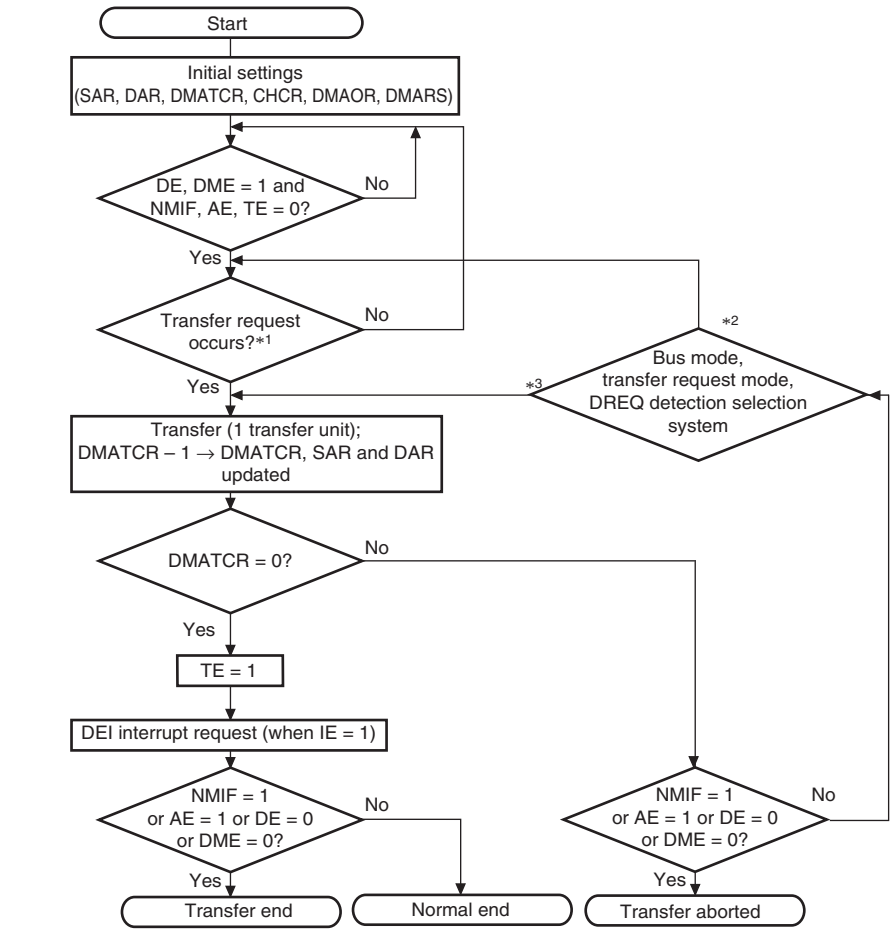
13.4.1 DMA Transfer Flow

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 extension resource selector (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 comes and transfer is enabled, the DMAC transfers 1 transfer unit of data (depending on the TS0 and TS1 settings). For an auto request, 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 of the CHCR is set to 1 at this time, a DEI 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 of the CHCR or the DME bit of the DMAOR are changed to 0.

Figure 13.2 is a flowchart of this procedure.



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

Figure 13.2 DMA Transfer Flowchart

13.4.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by devices and on-chip peripheral modules that are neither the source nor the destination.

Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected in the RS3 to RS0 bits in the DMA channel control registers 0 to 5 (CHCR_0 to CHCR_5), and the DMA extension resource selectors 0 to 2 (DMARS0 to DMARS2).

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 peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR_0 to CHCR_5 and the DME bit of the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR_0 to CHCR_5 AE bit of DMAOR, and the NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signals (DREQ0 or DREQ1) of an external device. Choose one of the modes shown in table 13.3 according to the application system. 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.

Table 13.3 Selecting External Request Modes with RS Bits

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

Whether the DREQ is detected by either the edge or level of the signal input is selected with the DREQ level (DL) bit and DREQ select (DS) bit in CHCR_0 and CHCR_1 as shown in table 13.4. The source of the transfer request does not have to be the data transfer source or destination.

Table 13.4 Selecting External Request Detection with DL, DS Bits

CHCR		
DL	**DS**	**Detection of External Request**
0	0	Low level detection
	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 (non-sensitive period). 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 13.5 Selecting External Request Detection with DO Bit

CHCR	
DO	**External Request**
0	Overrun 0
1	Overrun 1

On-Chip Peripheral Module Request Mode: In this mode, the transfer is performed in response to the transfer request signal of an on-chip peripheral module.

The DMA transfer request signals comprise the transmit data empty transfer request and receive data full transfer request from the SCIF0, SCIF1, SIOF0, and SIOF1 set by DMARS0 to DMARS2.

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 transfer request is set to TXI of the SCIF0, the transfer destination must be the SCIF0's transmit data

register. Likewise, when a transfer request is set to RXI of the SCIF0, the transfer source must be the SCIF0's receive data register. These conditions also apply to the SCIF1, SIOF0, and SIOF1.

Depending on the on-chip peripheral module, the number of receive FIFO triggers can be set as a transfer request. If the receive FIFO trigger condition is not satisfied, data may be remained in the receive FIFO. Therefore, data needs to be read upon completion of the DMA transfer.

Table 13.6 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	001000	01	SCIF0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR_0	Cycle steal
		10	SCIF0 receiver	RXI (receive FIFO data full interrupt)	SCFRDR_0	Any	Cycle steal
	001010	01	SCIF1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR_1	Cycle steal
		10	SCIF1 receiver	RXI (receive FIFO data full interrupt)	SCFRDR_1	Any	Cycle steal
	010100	01	SIOF0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR_0	Cycle steal
		10	SIOF0 receiver	RXI (receive FIFO data full interrupt)	SIOF0/ SIRDR_0	Any	Cycle steal
	010101	01	SIOF1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR_1	Cycle steal
		10	SIOF1 receiver	RXI (receive FIFO data full interrupt)	SIOF1/ SIRDR_1	Any	Cycle steal

13.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. The two modes (fixed mode and round-robin mode) can be selected using bits PR0 and PR1 in DMAOR.

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

Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5

Fixed mode 2: CH0 > CH2 > CH3 > CH1 > CH4 > CH5

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

Round-Robin Mode: Each time one word, byte, longword or 16-byte unit is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority order. The round-robin mode operation is shown in figure 13.3. The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 immediately after a reset. When round-robin mode is specified, the bus mode setting of multiple channels does not allow a mixture of cycle steal mode and burst mode.

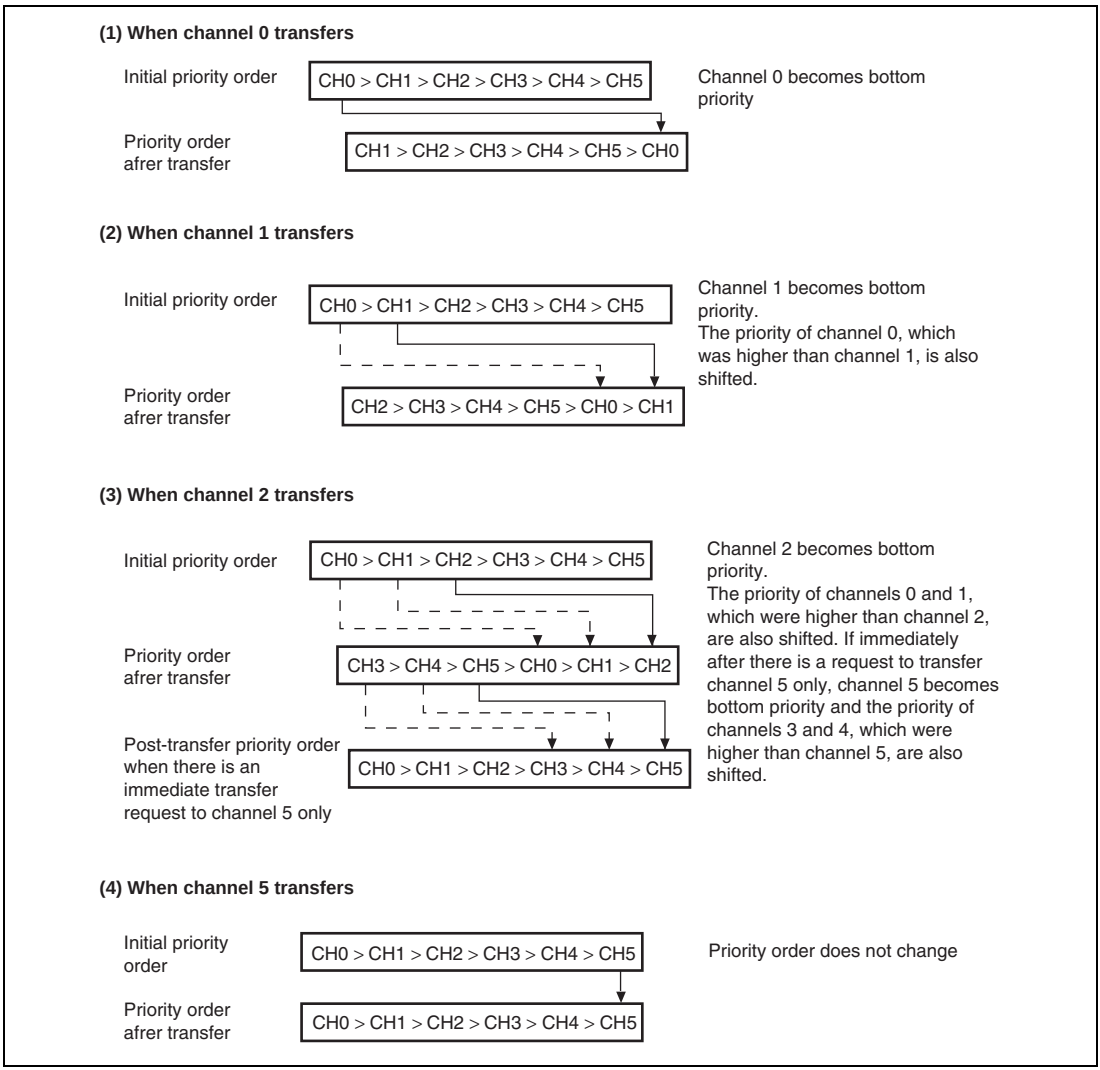


Figure 13.3 Round-Robin Mode

Figure 13.4 shows how the priority order 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.

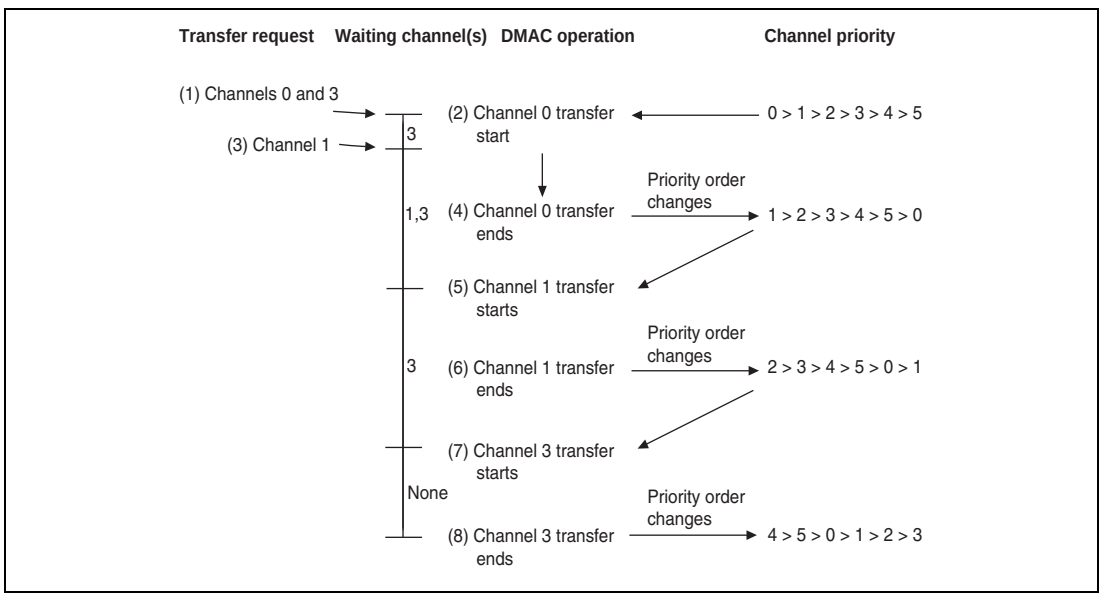


Figure 13.4 Changes in Channel Priority in Round-Robin Mode

13.4.4 DMA Transfer Types

DMA transfer has two types; single address mode transfer and dual address mode transfer, They depend on the number of bus cycles of access to source and destination. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode. The DMAC supports the transfers shown in table 13.7.

Table 13.7 Supported DMA Transfers

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**X/Y Memory**
External device with DACK	Not available	Single	Single	Not available	Not available
External memory	Single	Dual	Dual	Dual	Dual
Memory-mapped external device	Single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
X/Y memory	Not available	Dual	Dual	Dual	Dual

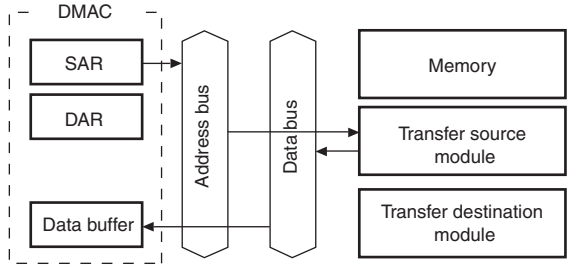
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is not available for on-chip peripheral modules.

Address Modes:

1. Dual Address Mode

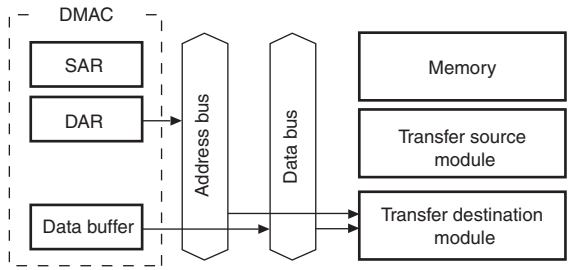
   In the dual address mode, both the transfer source and destination are accessed (selected) by an address. The source and destination can be located externally or internally.

   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 13.5, 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 13.5 Data Flow in Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. The AM bit of the channel control register (CHCR) can specify whether the DACK is output in read cycle or write cycle.

Figure 13.6 shows an example of DMA transfer timing in dual address mode.

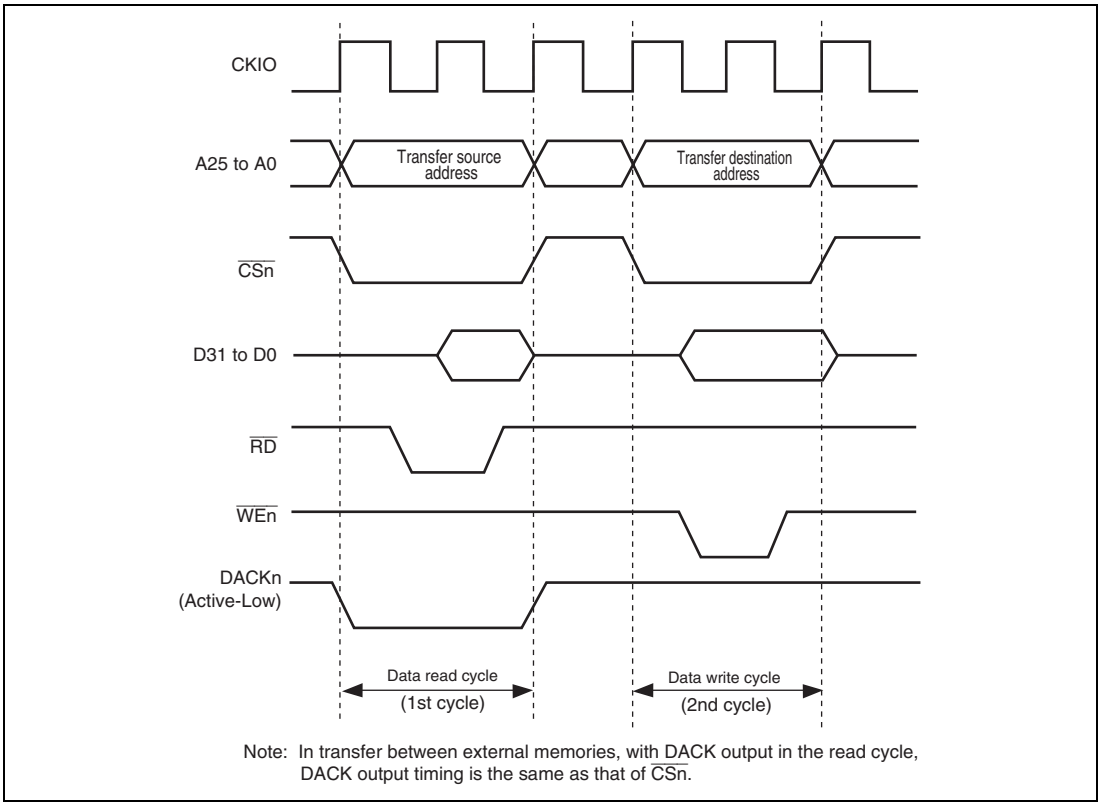


Figure 13.6 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary memory, Destination: Ordinary memory)

2. Single Address Mode

   In single address mode, either the transfer source or transfer destination external device is accessed (selected) by means of the DACK signal, and the other device is accessed by address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 13.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

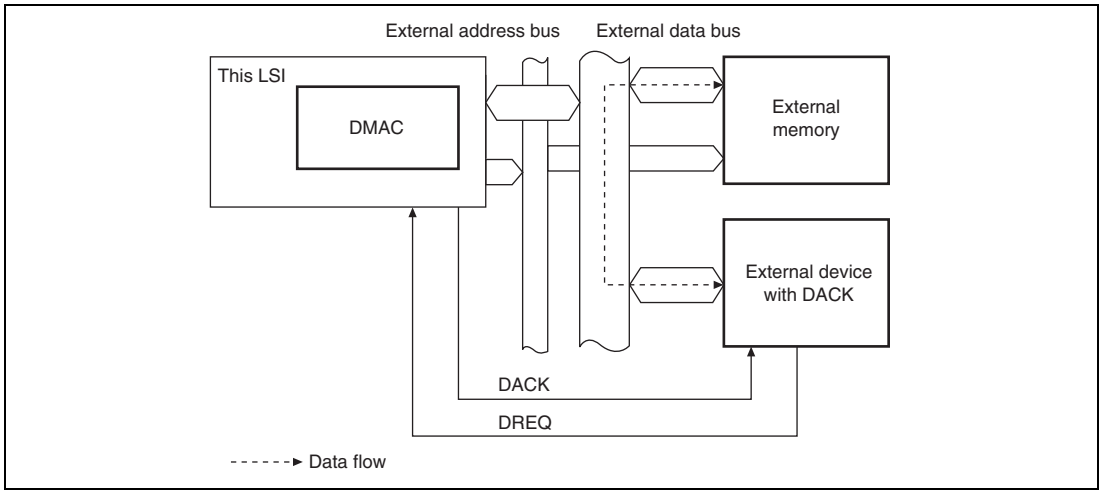


Figure 13.7 Data Flow in Single Address Mode

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

Figures 13.8 shows example of DMA transfer timing in single address mode.

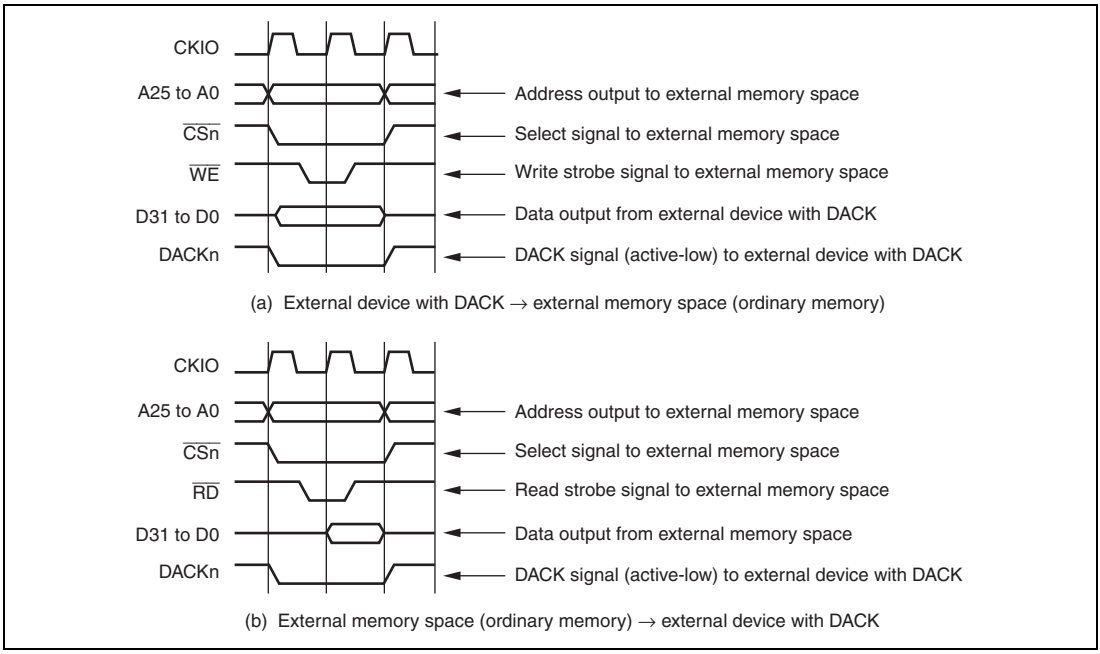


Figure 13.8 Example of DMA Transfer Timing in Single Address Mode

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

- Cycle-Steal Mode

  In the cycle-steal mode, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, long-word, or 16 bytes unit) DMA transfer. When another transfer request occurs, the bus masterships are 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 the cycle-steal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination.

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

1. Dual address mode
2. DREQ low level detection

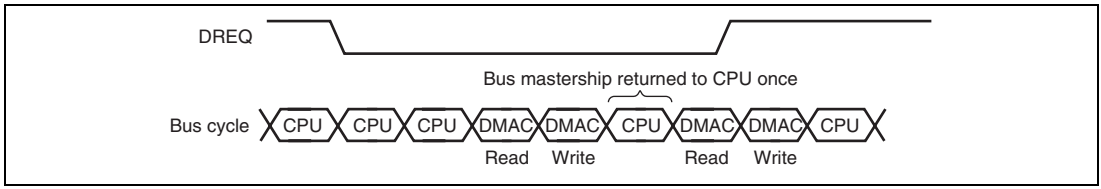


Figure 13.9 DMA Transfer Example in Cycle-Steal Mode (Dual Address, DREQ Low Level Detection)

- Burst Mode

  In the burst mode, once the bus mastership is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In the external request mode with low level detection of the DREQ pin, however, when the DREQ pin is driven high, the bus 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.

  The burst mode cannot be used when the on-chip peripheral module is the transfer request source. Figure 13.10 shows DMA transfer timing in burst mode.

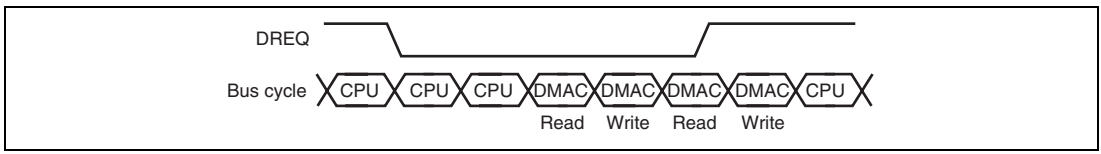


Figure 13.10 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

Relationship between Request Modes and Bus Modes by DMA Transfer Category: Table 13.8 shows the relationship between request modes and bus modes by DMA transfer category.

Table 13.8 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0, 1
	External memory and external memory	External, auto	B/C	8/16/32/128	0 to 5*[2]
	External memory and memory-mapped external device	External, auto	B/C	8/16/32/128	0 to 5*[2]
	Memory-mapped external device and memory-mapped external device	External, auto	B/C	8/16/32/128	0 to 5*[2]
	External memory and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0 to 5*[2]
	Memory-mapped external device and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0 to 5*[2]
	On-chip peripheral module and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0 to 5*[2]
	X/Y memory and X/Y memory	External, auto	B/C	8/16/32/128	0 to 5*[2]
	X/Y memory and memory-mapped external device	External, auto	B/C	8/16/32/128	0 to 5*[2]
	X/Y memory and on-chip peripheral module	All*[1]	C	8/16/32*[3]	0 to 5*[2]
	X/Y memory and external memory	External, auto	B/C	8/16/32/128	0 to 5*[2]
Single	External device with DACK and external memory	External	B/C	8/16/32	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1

[Legend] B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However, the request-source register must be designated as the transfer source or the transfer destination.

2. If the transfer request is an external request, channels 0 and 1 are only available.

3. Access size permitted for each module must be used when accessing the on-chip peripheral module.

Bus Mode and Channel Priority: Even if channel 1 is performing burst-mode transfer in priority fixed mode (CH0 > CH1), channel 0 starts transfer immediately when a request is made for transfer on channel 0 with higher priority.

If channel 0 is also in burst mode at this time, channel 1 resumes transfer after transfer on channel 0 with higher priority is completed.

If channel 0 is in cycle-steal mode, channel 0 with higher priority transfers one transfer unit then allows channel 1 to perform transfers without releasing bus mastership. Next, transfers are performed alternately by channel 0, channel 1, channel 0, channel 1, and so on. This means that a bus state is set for the CPU cycle after completion of the cycle-steal mode transfer is replaced with the burst-mode transfer. (This operation is hereinafter referred to as burst-mode priority execution.) Figure 13.11 shows an example.

If multiple channels are conflicting in burst mode, the channel with the highest priority is selected for execution.

If multiple channels perform DMA transfers, bus mastership is not released to the bus master until all conflicting burst transfers are completed.

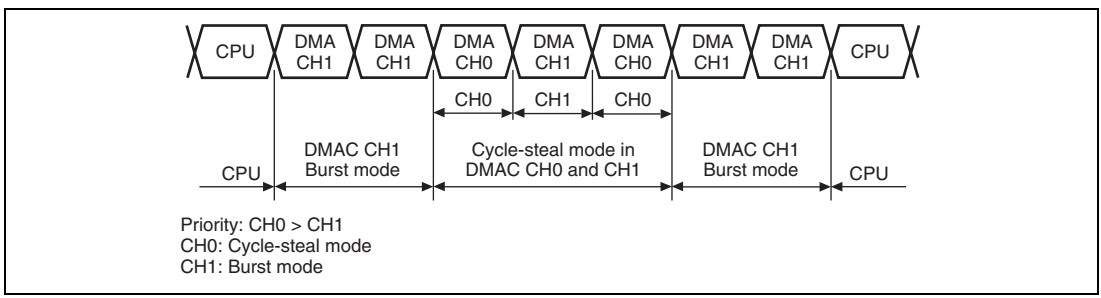


Figure 13.11 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes according to the specification shown in figure 13.11. However, no mixture of channels in cycle-steal mode and channels in burst mode is allowed.

13.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 12, Bus State Controller (BSC).

DREQ Pin Sampling Timing:

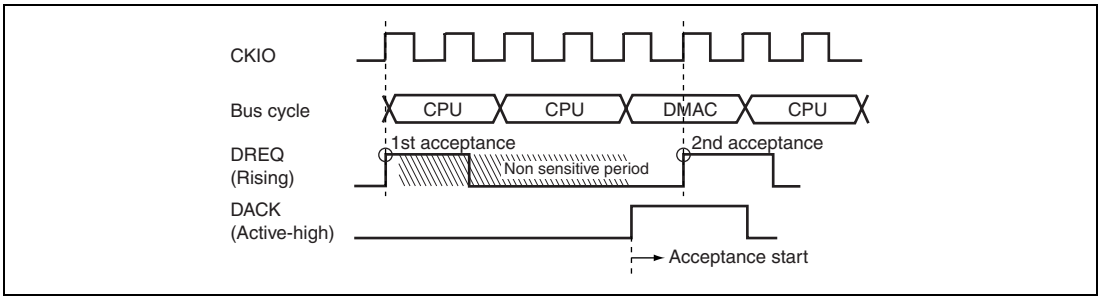


Figure 13.12 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

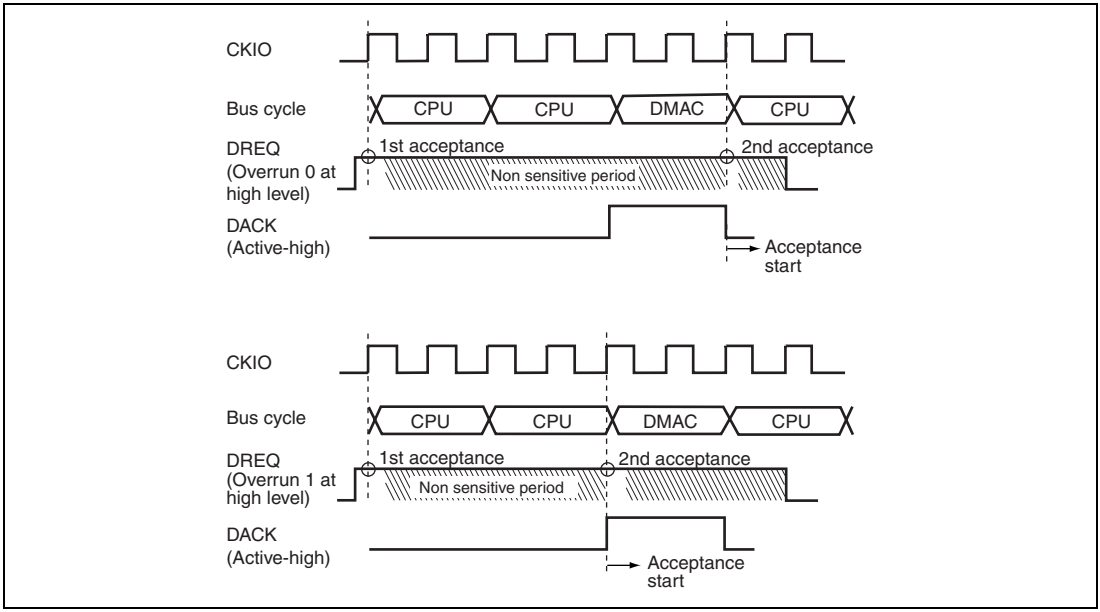


Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

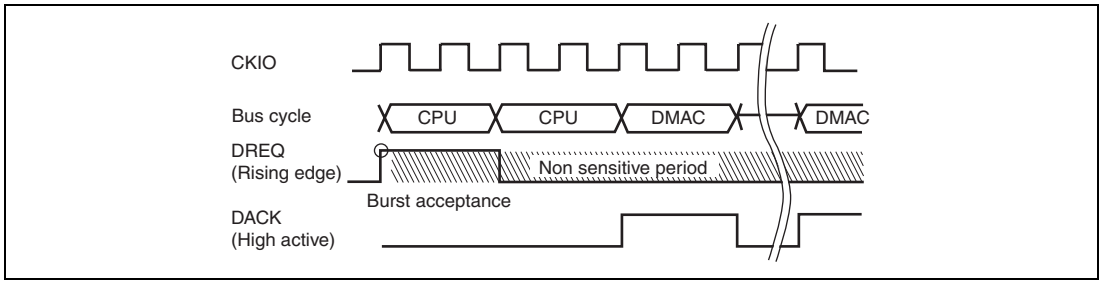


Figure 13.14 Example of DREQ Input Detection in Burst Mode Edge Detection

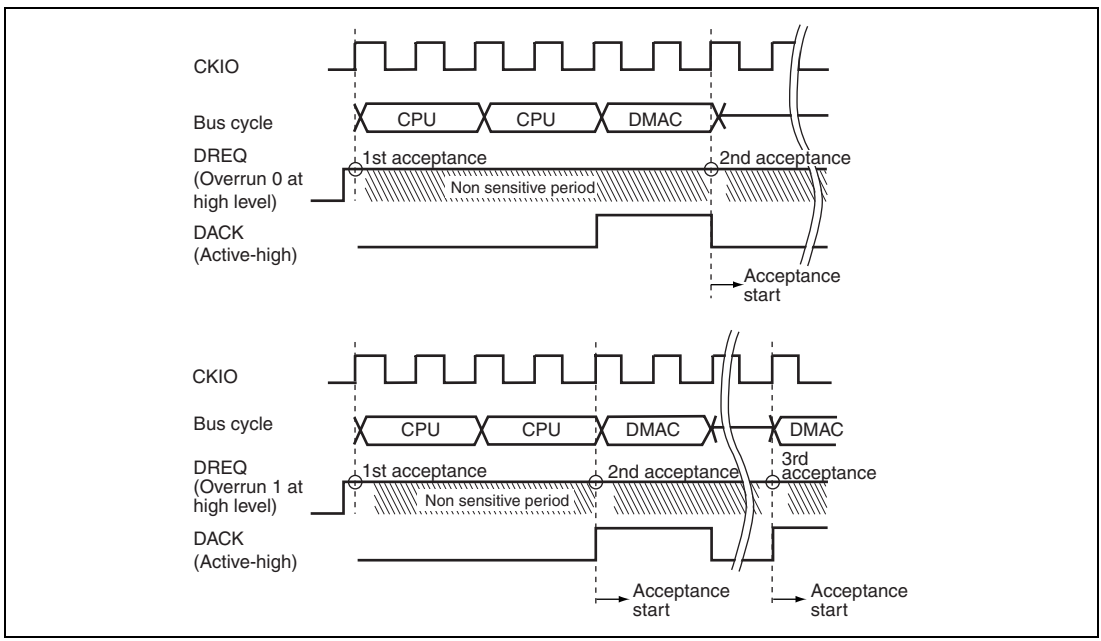


Figure 13.15 Example of DREQ Input Detection in Burst Mode Level Detection

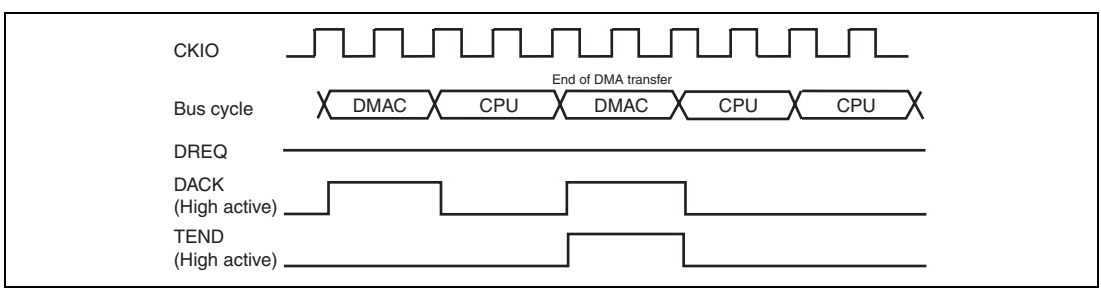
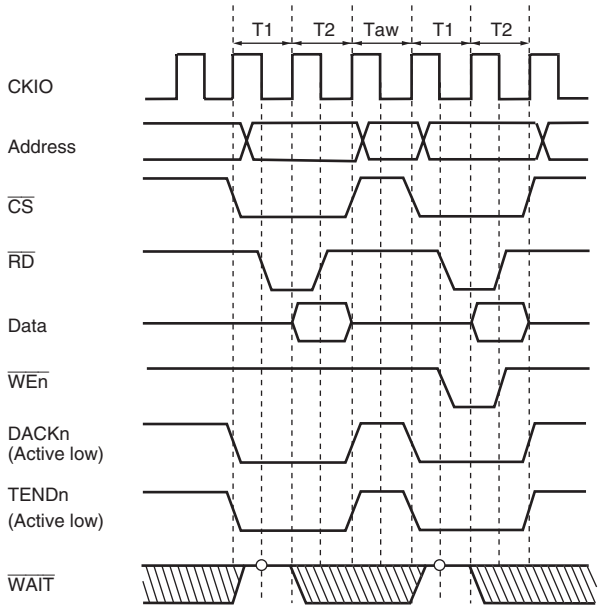


Figure 13.16 Example of DMA Transfer End Timing (Cycle Steal Level Detection)



Note: TEND is asserted for the last transfer unit of DMA transfers.
If a transfer unit is divided into multiple bus cycles and
if CS is negated during the bus cycle, TEND is also divided.

Figure 13.17 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle = 1, Longword Access to 16-bit Device)

13.5 Usage Note

When using the DMAC, note the following:

Note on Using TEND Pin:

If a DMA transfer is performed under one of the conditions described below and, after completion of the transfer, retransfer is performed on the same channel, the TEND pin is asserted once in the first DMA transfer in retransfer when the retransfer condition satisfies (1) DACK is output in a dual address mode read cycle (with the AM bit in CHCR cleared to 0) and the DMA transfer source address (SAR) is in external memory space or (2) in single address mode.

Conditions:

- DACK is output in a dual address mode read cycle (with the AM bit in CHCR cleared to 0) and the DMA transfer source address (SAR) is in external memory space.
- DACK is output in a dual address mode write cycle (with the AM bit in CHCR set to 1) and the DMA transfer destination address (DAR) is in external memory space.
- Single address mode

Method of Avoidance:

Perform a dummy DMA transfer under one of the settings below. After start of a dummy DMA transfer, clear all bits in the DMA channel control register (CHCR) of the corresponding channel to suspend the dummy DMA transfer forcibly.

- DACK is output in a dual address mode read cycle (with the AM bit in CHCR cleared to 0) and the DMA transfer source address (SAR) is in external memory space.
- DACK is output in a dual address mode write cycle (with the AM bit in CHCR set to 1) and the DMA transfer destination address (DAR) is in external memory space.

Section 14 Timer Unit (TMU)

This LSI includes a three-channel (channel 0 to 2) 32-bit timer unit (TMU).

14.1 Features

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter
- All channels are provided with 32-bit constant registers and 32-bit down counters for an auto-reload function that can be read or written to at any time
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection among 4 counter input clocks: Pϕ/4, Pϕ/16, Pϕ/64, and Pϕ/256

Note: Pϕ is the internal clock for peripheral modules. See section 11, On-Chip Oscillation Circuits, for more information on the clock pulse generator.

14.1.1 Block Diagram

Figure 14.1 shows a block diagram of the TMU.

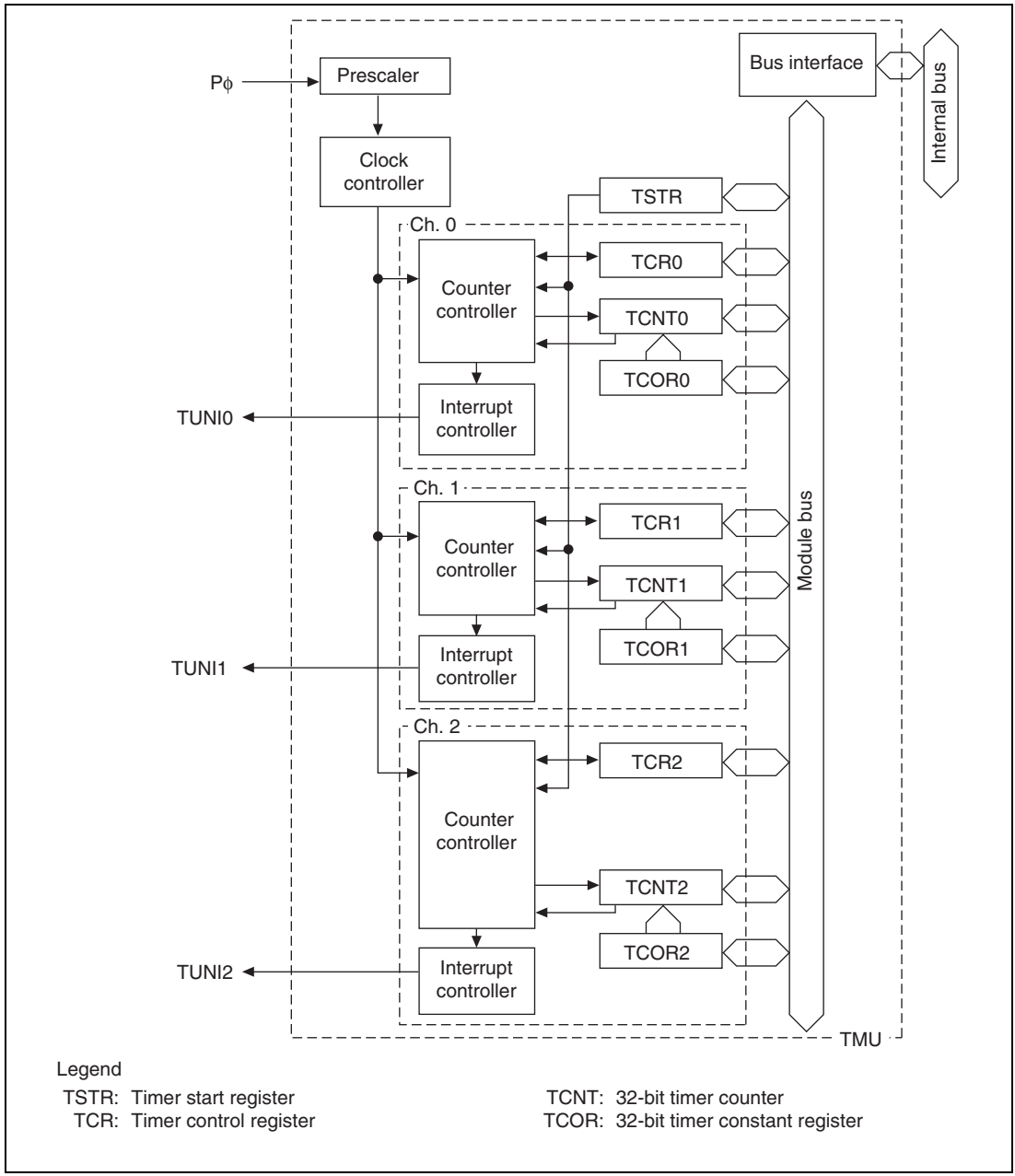


Figure 14.1 TMU Block Diagram

14.2 Register Descriptions

The TMU has the following registers. Refer the section 24, List of Registers, for the addresses and access size for these registers.

- Timer start register (TSTR)
- Timer constant register 0 (TCOR0)
- Timer counter 0 (TCNT0)
- Timer control register 0 (TCR0)
- Timer constant register 1 (TCOR1)
- Timer counter 1 (TCNT1)
- Timer control register 1 (TCR1)
- Timer constant register 2 (TCOR2)
- Timer counter 2 (TCNT2)
- Timer control register 2 (TCR2)

14.2.1 Timer Start Register (TSTR)

The timer start register (TSTR) selects whether to run or halt the timer counters (TCNT) for channels 0 to 2.

TSTR is an 8-bit readable/writable register. It is initialized to H'00 by a power-on reset or manual reset. It is initialized in standby mode when the multiplication ratio of PLL circuit 1 (PLL1) is changed or when the MSTP2 bit in STBCR is set to 1.

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 Selects whether to run or halt TCNT2. 0: TCNT2 count halted 1: TCNT2 counts
1	STR1	0	R/W	Counter Start 1 Selects whether to run or halt TCNT1. 0: TCNT1 count halted 1: TCNT1 counts

Bit	Bit Name	Initial Value	R/W	Description
0	STR0	0	R/W	Counter Start 0 Selects whether to run or halt TCNT0. 0: TCNT0 count halted 1: TCNT0 counts

14.2.2 Timer Control Registers (TCR)

The timer control registers (TCR) control the timer counters (TCNT) and interrupts. The TMU has three TCR registers, one for each channel.

The TCR registers control the issuance of interrupts when the flag indicating timer counters (TCNT) underflow has been set to 1, and also carry out counter clock selection.

The TCR registers are 16-bit readable/writable registers. They are initialized to H'0000 by a power-on reset and manual reset, but are not initialized, and retain their contents, in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15 or 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a TCNT underflow. 0: TCNT has not underflowed [Clearing condition] 0 is written to UNF 1: TCNT has underflowed [Setting condition] TCNT underflows* Note: * Contents do not change when 1 is written to UNF.
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	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation when the status flag (UNF) indicating TCNT underflow has been set to 1. 0: Interrupt due to UNF (TUNI) is disabled 1: Interrupt due to UNF (TUNI) is enabled
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	Select the TCNT count clock.
0	TPSC0	0	R/W	000: Count on Pϕ/4 001: Count on Pϕ/16 010: Count on Pϕ/64 011: Count on Pϕ/256 100: Reserved (Setting prohibited) 101: Reserved (Setting prohibited) 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

14.2.3 Timer Constant Registers (TCOR)

The TMU has three timer constant registers (TCOR), one for each channel. The TCOR registers set the value to be set in TCNT when TCNT underflows.

The TCOR registers are 32-bit readable/writable registers. They are initialized to H'FFFFFFFF by a power-on reset or manual reset, but are not initialized, and retain their contents, in standby mode.

14.2.4 Timer Counters (TCNT)

The TMU has three timer counters (TCNT), one for each channel. The timer counters (TCNT) counts down upon input of a clock. The input clock is selected using the TPSC2 to TPSC0 bits in the timer control registers (TCR).

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in TCR of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the count-down continues from that value.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset, but is not initialized, and retains its contents, in standby mode.

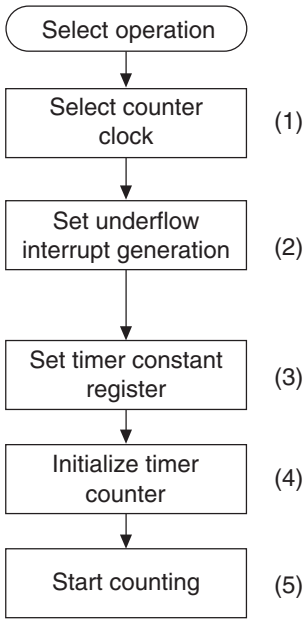
14.3 TMU Operation

Each of the three channels has a 32-bit TCNT and a 32-bit TCOR. TCNT counts down. The auto-reload function enables synchronized counting and counting by external events.

14.3.1 Counter Operation

When the STR0 to STR2 bits in TSTR are set to 1, the corresponding TCNT starts counting. When TCNT underflows, the underflow flag (UNF) of the corresponding TCR is set. At this time, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the down-count operation is continued.

Count Operation Setting Procedure: An example of the procedure for setting the count operation is shown in figure 14.2.



(1) Select the counter clock with the TPSC2 to TPSC0 bits in TCR.
(2) Use the UNIE bit in TCR to set whether to generate an interrupt when TCNT underflows.
(3) Set a value in TCOR (the cycle is the set value plus 1).
(4) Set the initial value in TCNT.
(5) Set the STR bit in TSTR to 1 to start operation.

Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 14.2 Setting Count Operation

Auto-Reload Count Operation: Figure 14.3 shows the TCNT auto-reload operation.

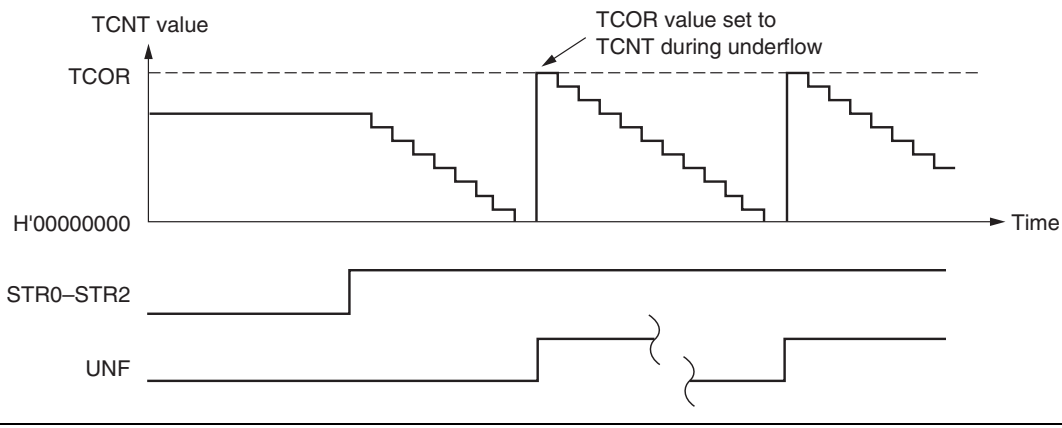


Figure 14.3 Auto-Reload Count Operation

TCNT Count Timing: Set the TPSC2 to TPSC0 bits in TCR to select whether peripheral module clock Pϕ or one of the four internal clocks created by dividing it is used (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256). Figure 14.4 shows the timing.

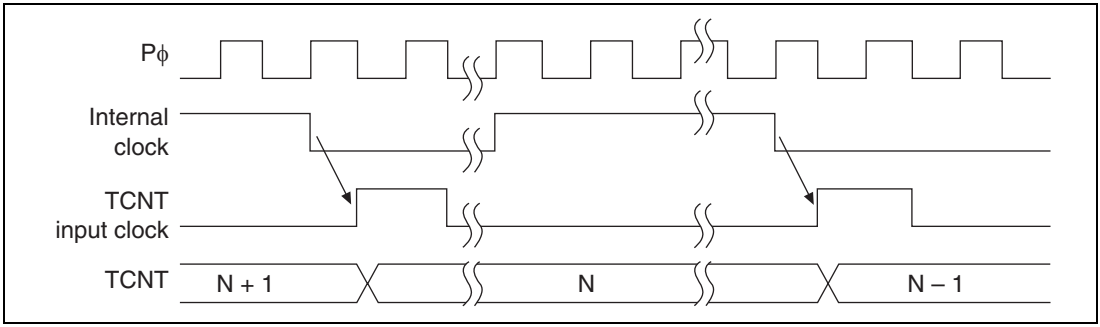


Figure 14.4 Count Timing when Internal Clock Is Operating

14.4 Interrupts

The interrupt source of TMU is underflow interrupt (TUNI).

14.4.1 Status Flag Set Timing

The UNF bit is set to 1 when the TCNT underflows. Figure 14.5 shows the timing.

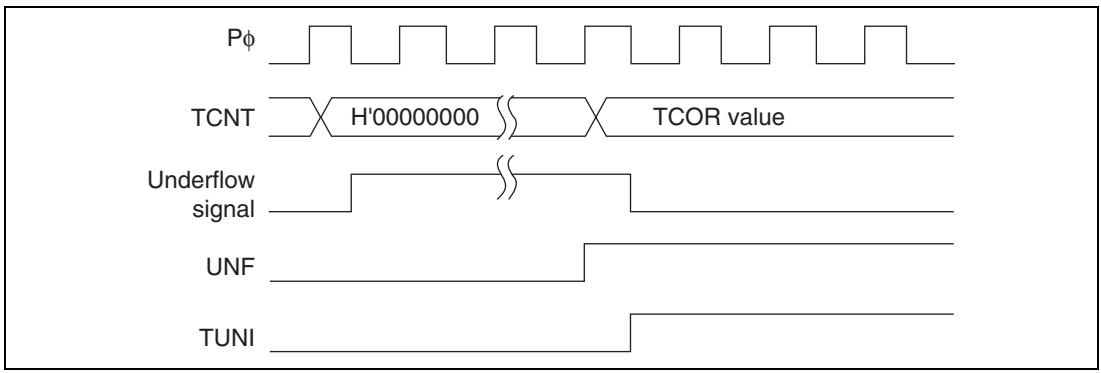


Figure 14.5 UNF Set Timing

14.4.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 14.6 shows the timing.

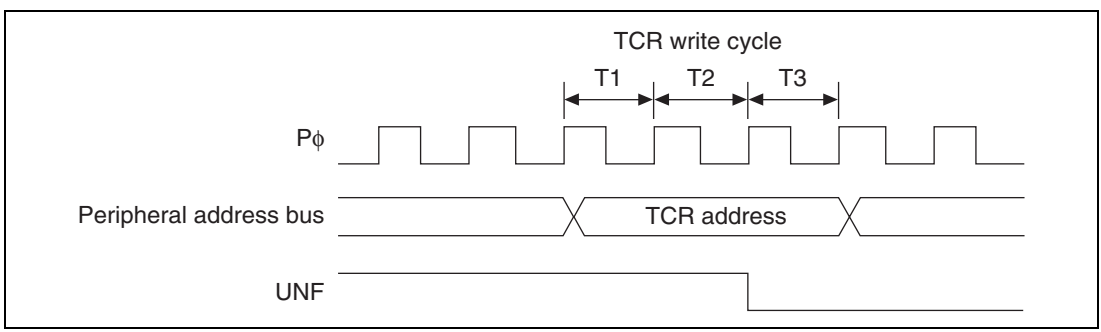


Figure 14.6 Status Flag Clear Timing

14.4.3 Interrupt Sources and Priorities

The TMU generates underflow interrupts for each channel. When the interrupt request flag and interrupt enable bit are both set to 1, the interrupt is requested. Codes are set in the interrupt event register (INTEVT2) for these interrupts and interrupt processing occurs according to the codes.

The relative priorities of channels can be changed using the interrupt controller (see section 4, Exception Handling, and section 8, Interrupt Controller (INTC)). Table 14.1 lists TMU interrupt sources.

Table 14.1 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	Low

14.5 Usage Notes

14.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits for the channel (STR2 to STR0) in TSTR to halt timer counting.

14.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When timer counting and register read processing are performed simultaneously, the register value before TCNT counting down (with synchronization processing) is read.

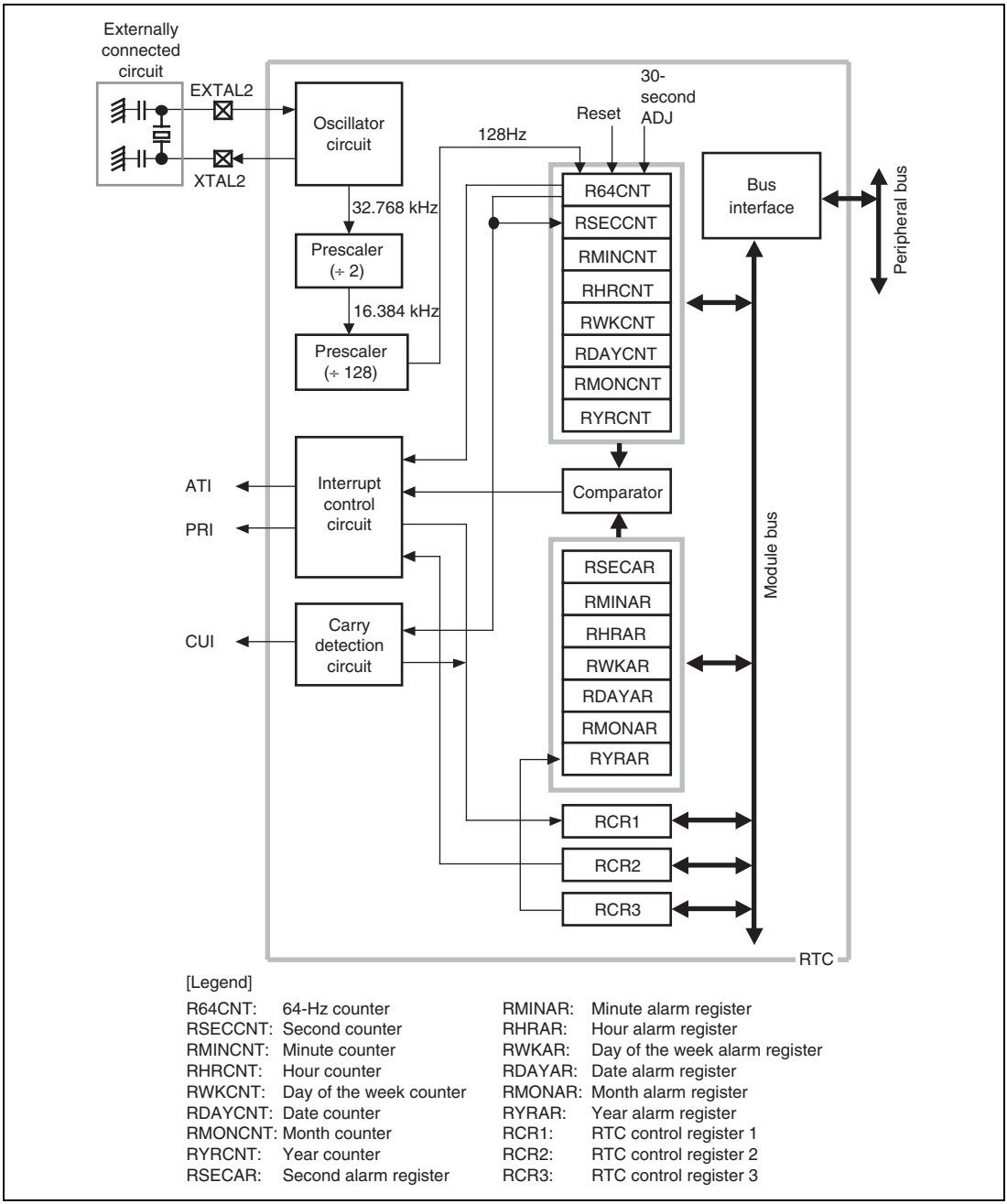
Section 15 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator. A block diagram of the RTC is shown in figure 15.1.

15.1 Feature

The RTC has following features:

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary format)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment



[Legend]

R64CNT:	64-Hz counter	RMINAR:	Minute alarm register
RSECCNT:	Second counter	RHRAR:	Hour alarm register
RMINCNT:	Minute counter	RWKAR:	Day of the week alarm register
RHRCNT:	Hour counter	RDAYAR:	Date alarm register
RWKCNT:	Day of the week counter	RMONAR:	Month alarm register
RDAYCNT:	Date counter	RYRAR:	Year alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2
RSECAR:	Second alarm register	RCR3:	RTC control register 3

Figure 15.1 RTC Block Diagram

15.2 Input/Output Pins

Table 15.1 shows the RTC pin configuration.

Table 15.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
RTC external clock	EXTAL2	I	Connects crystal to RTC oscillator*[1]
RTC crystal	XTAL2	O	Connects crystal to RTC oscillator*[1]
RTC oscillator power supply	VccQ-RTC	—	Dedicated power-supply pin for RTC*[2]
RTC oscillator ground	VssQ-RTC	—	Dedicated GND pin for RTC

Note: 1. Pull up (VccQ-RTC) the EXTAL2 pin, and open (NC) the XTAL2 pin when the realtime clock (RTC) is not used.

2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal oscillator circuit for RTC operates with VccQ-RTC. The control circuit and the RTC counter operate with Vcc (common to the internal circuit). Therefore, all power supplies other than VccQ-RTC should always be turned on even if only RTC operates.

15.3 Register Descriptions

The RTC has the following registers. Refer to section 24, List of Registers, for more details on the address and access size.

- 64-Hz counter (R64CNT)
- Second counter (RSECCNT)
- Minute counter (RMINCNT)
- Hour counter (RHRCNT)
- Day of week counter (RWKCNT)
- Date counter (RDAYCNT)
- Month counter (RMONCNT)
- Year counter (RYRCNT)
- Second alarm register (RSECAR)
- Minute alarm register (RMINAR)
- Hour alarm register (RHRAR)
- Day of week alarm register (RWKAR)
- Date alarm register (RDAYAR)
- Month alarm register (RMONAR)
- Year alarm register (RYRAR)

- RTC control register 1 (RCR1)
- RTC control register 2 (RCR2)
- RTC control register 3 (RCR3)

15.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit (RTC prescaler and R64CNT) between 64 Hz and 1 Hz.

R64CNT is reset to H'00 by setting the RESET bit in RCR2 or the ADJ bit in RCR2 to 1.

R64CNT is an 8-bit read-only register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 0	—	—	R	64-Hz Counter Each bit (bits 6 to 0) indicates the state of the RTC divider circuit between 64 and 1Hz. Bit Frequency 6: 1 Hz 5: 2 Hz 4: 4 Hz 3: 8 Hz 2: 16 Hz 1: 32 Hz 0: 64 Hz

15.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count operation is performed by a carry for each second of the 64-Hz counter.

The range of second can be set is 0 to 59 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RSECCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	—	—	R/W	Counter for 10-unit of second in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of second in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count operation is performed by a carry for each minute of the second counter.

The range of minute can be set is 0 to 59 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RMINCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.The write value should always be 0.
6 to 4	—	—	R/W	Counter for 10-unit of minute in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of minute in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The range of hour can be set is 0 to 23 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RHRCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.The write value should always be 0.
5, 4	—	—	R/W	Counter for 10-unit of hour in the BCD-code. The range can be set from 0 to 2 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of hour in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The range for day of the week can be set is 0 to 6 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RWKCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

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 to 0	—	—	R/W	Counter for the day of week in the BCD-code. The range can be set from 0 to 6 (decimal). Code Day of Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

15.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. The count operation is performed by a carry for each day of the hour counter.

Though the range of date which can be set is 1 to 31 (decimal), it changes with each month and in leap years. Please confirm the correct setting. Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RDAYCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	—	—	R/W	Counter for 10-unit of date in the BCD-code. The range can be set from 0 to 3 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of date in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count operation is performed by a carry for each month of the date counter.

The range of month can be set is 1 to 12 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2.

RMONCNT is an 8-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	—	R/W	Counter for 10-unit of month in the BCD-code. The range can be set from 0 to 1 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of month in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The 4 digits of the year are displayed. The count operation is performed by a carry for each year of the month counter.

The range for year which can be set is 0000 to 9999 (decimal). Errant operation will result if any other value is set. Carry out write processing after stopping the count operation with the START bit in RCR2 or using a carry flag.

RYRCNT is a 16-bit readable/writable register and not initialized by a power-on reset or manual reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fractional result of 0. The year counter value of 0000 is included in the leap year.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	—	R/W	Counter for 1000-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal)
11 to 8	—	—	R/W	Counter for 100-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal).
7 to 4	—	—	R/W	Counter for 10-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal).
3 to 0	—	—	R/W	Counter for 1-unit of year in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of second alarm which can be set is 0 to 59 (decimal). Errant operation will result if any other value is set.

RSECAR is an 8-bit readable/writable register. The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSECAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Second Alarm Enable Specifies whether comparison of RSECCNT and RSECAR is performed as an alarm condition. 0: Not compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of second alarm in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of second alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of minute alarm which can be set is 0 to 59 (decimal). Errant operation will result if any other value is set.

RMINAR is an 8-bit readable/writable register. The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Minute Alarm Enable Specifies whether comparison of RMINCNT and RMINAR is performed as an alarm condition. 0: Not compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of minute alarm in the BCD-code. The range can be set from 0 to 5 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of minute alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the hour counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of hour alarm which can be set is 0 to 23 (decimal). Errant operation will result if any other value is set.

RHRAR is an 8-bit readable/writable register. The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Hour Alarm Enable Specifies whether comparison of RHRCNT and RHRAR is performed as an alarm condition. 0: Not compared 1: Compared
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	—	R/W	Setting value for 10-unit of hour alarm in the BCD-code. The range can be set from 0 to 2 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of hour alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of day of the week alarm which can be set is 0 to 6 (decimal). Errant operation will result if any other value is set.

RWKAR is an 8-bit readable/writable register. The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Day of Week Alarm Enable Specifies whether comparison of RWKCNT and RWKAR is performed as an alarm condition. 0: Not compared 1: Compared
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	—	—	R/W	Day of Week Alarm Code The range can be set from 0 to 6 (decimal). Code Day of the Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

15.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the date counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of date alarm which can be set is 1 to 31 (decimal). Errant operation will result if any other value is set. The RDAYCNT range that can be set changes with some months and in leap years. Please confirm the correct setting.

RDAYAR is an 8-bit readable/writable register. The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Date Alarm Enable Specifies whether comparison of RDAYCNT and RDAYAR is performed as an alarm condition. 0: Not compared 1: Compared
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	—	R/W	Setting value for 10-unit of date alarm in the BCD-code. The range can be set from 0 to 3 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of date alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of month alarm which can be set is 1 to 12 (decimal). Errant operation will result if any other value is set.

RMONAR is an 8-bit readable/writable register. The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Month Alarm Enable Specifies whether comparison of RMONCNT and RMONAR is performed as an alarm condition. 0: Not compared 1: Compared
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	—	R/W	Setting value for 10-unit of month alarm in the BCD-code. The range can be set from 0 to 1 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of month alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. When the YAEN bit in RCR3 is set to 1, a comparison with the RYRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/ and RMONAR, the counter and alarm register comparison is performed only on those with ENB bits and the YAEN bit in RCR3 set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range of year alarm which can be set is 0000 to 9999 (decimal). Errant operation will result if any other value is set.

RYRAR is a 16-bit readable/writable register. The contents are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	—	R/W	Setting value for 1000-unit of year alarm in the BCD-code. The range can be set from 0 to 9 (decimal).
11 to 8	—	—	R/W	Setting value for 100-unit of year alarm in the BCD-code. The range can be set from 0 to 9 (decimal).
7 to 4	—	—	R/W	Setting value for 10-unit of year alarm in the BCD-code. The range can be set from 0 to 9 (decimal).
3 to 0	—	—	R/W	Setting value for 1-unit of year alarm in the BCD-code. The range can be set from 0 to 9 (decimal).

15.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag. Because flags are sometimes set after an operand read, do not use this register in read-modify-write processing.

RCR1 is an 8-bit readable/writable register. RCR1 is initialized to H'00 by a power-on reset or a manual reset, all bits are initialized to 0 except for the CF flag, which is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to R64CNT or RSECCNT occurs. A count register value read at this time cannot be guaranteed; another read is required. 0: No count up of R64CNT or RSECCNT. Clearing condition: When 0 is written to CF 1: Count up of R64CNT or RSECCNT. Setting condition: When 1 is written to CF or if the carry of R64CNT or RSECCNT occurs when R64CNT or RSECCNT is read.

Bit	Bit Name	Initial Value	R/W	Description
6	—	0	R	Reserved
5	—	0	R	These bits are always read as 0. The write value should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1
2	—	0	R	Reserved
1	—	0	R	These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set to 1 when the alarm time set in an alarm register (only registers with the ENB bit of the corresponding alarm registers and YAEN bit in RCR3 set to 1) matches the clock and calendar time. This flag is cleared to 0 when 0 is written, but holds the previous value when 1 is to be written. 0: Clock/calendar and alarm register have not matched. Clearing condition: When 0 is written to AF 1: Clock/calendar and alarm register have matched. Setting condition: Clock/calendar and alarm register have matched (only registers with the ENB bit and YAEN bit in RCR3 set to 1)

15.3.17 RTC Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop control.

RCR2 is an 8-bit readable/writable register. It is initialized to H'09 by a power-on reset. It is initialized except for RTCEN and START by a manual reset. It is not initialized in standby mode, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the PES bits. Clearing condition: When 0 is written to PEF 1: Interrupts generated with the period designated by the PES bits. Setting condition: When an interrupt is generated with the period designated by the PES0 to PES2 bits or when 1 is written to the PEF flag
6	PES2	0	R/W	Periodic Interrupt Flags
5	PES1	0	R/W	These bits specify the periodic interrupt.
4	PES0	0	R/W	000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds

Bit	Bit Name	Initial Value	R/W	Description
3	RTCEN	1	R/W	Controls the operation of the crystal oscillator for the RTC. 0: Halts the crystal oscillator for the RTC. 1: Runs the crystal oscillator for the RTC.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (RTC prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset When 1 is written, initializes the divider circuit (RTC prescaler and R64CNT). This bit always reads 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally. Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

15.3.18 RTC Control Register 3 (RCR3)

RCR3 is a register that controls comparison of the BCD-coded year counter RYRCNT and the year alarm register RYRAR of the RTC.

RCR3 is an 8-bit readable/writable register.

Bit	Bit Name	Initial Value	R/W	Description
7	YAEN	0	R/W	Year Alarm Enable When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed. From among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

15.4 Operation

15.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

15.4.2 Setting Time

Figure 15.2 shows how to set the time when the clock is stopped.

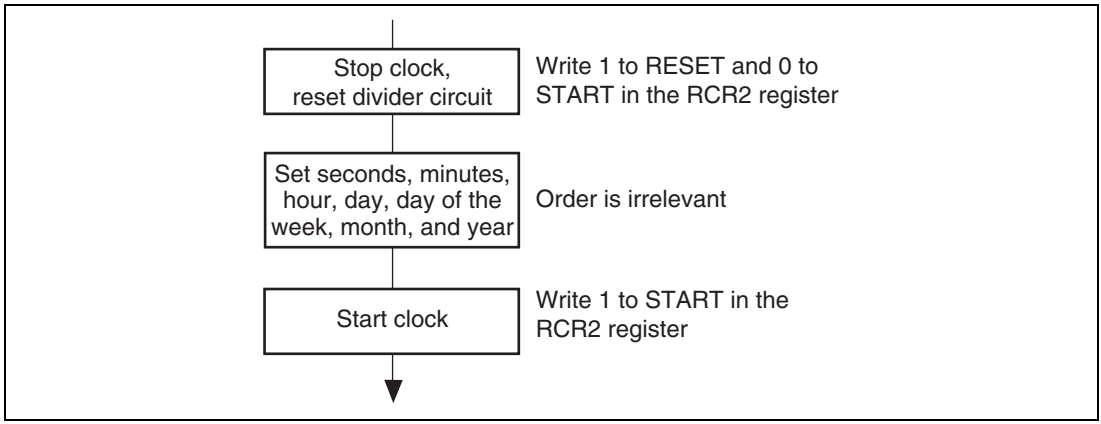


Figure 15.2 Setting Time

15.4.3 Reading Time

Figure 15.3 shows how to read the time.

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 15.3 shows the method of reading the time without using interrupts; part (b) in figure 15.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

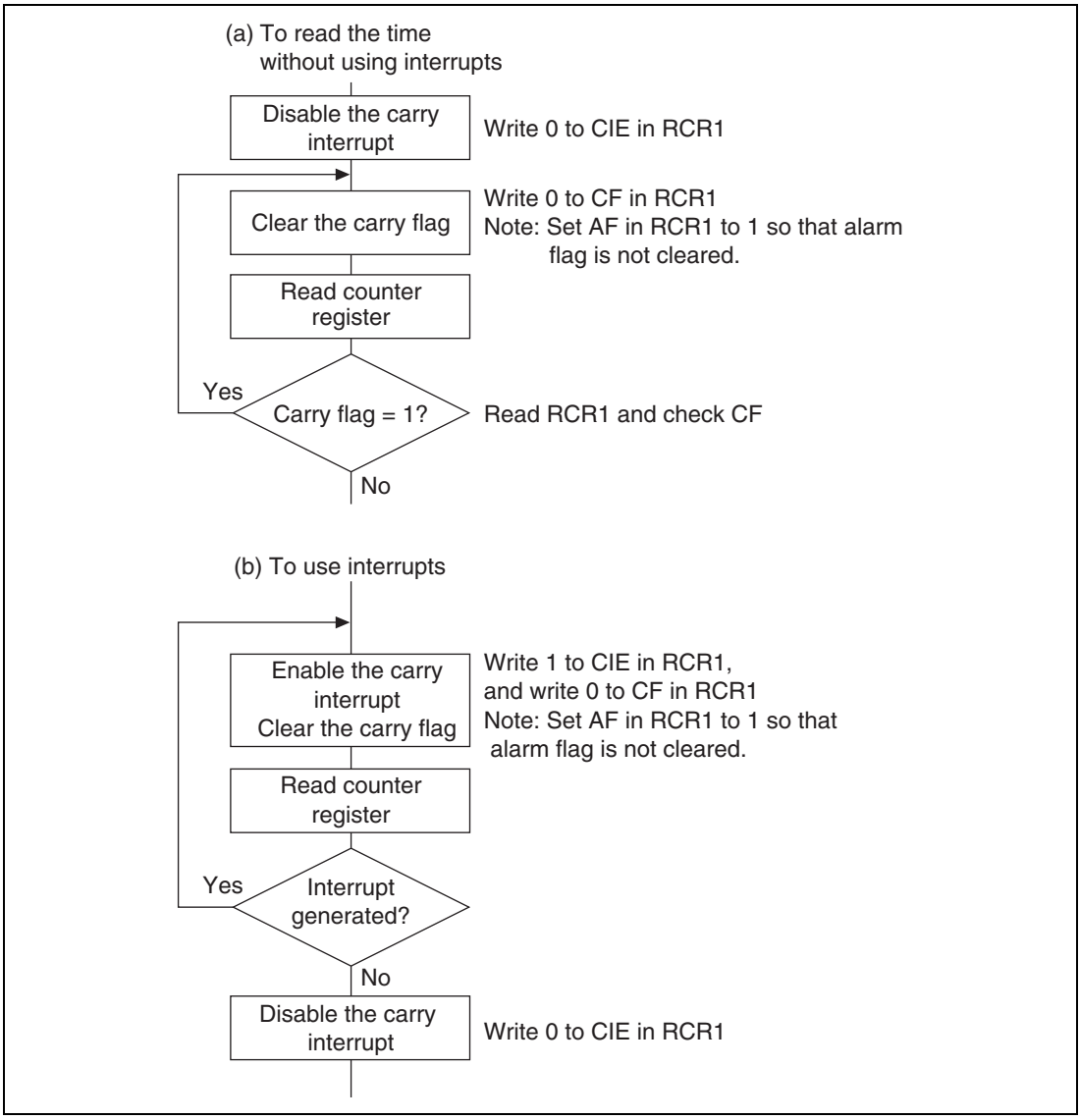


Figure 15.3 Reading Time

15.4.4 Alarm Function

Figure 15.4 shows how to use the alarm function.

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit or YAEN bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

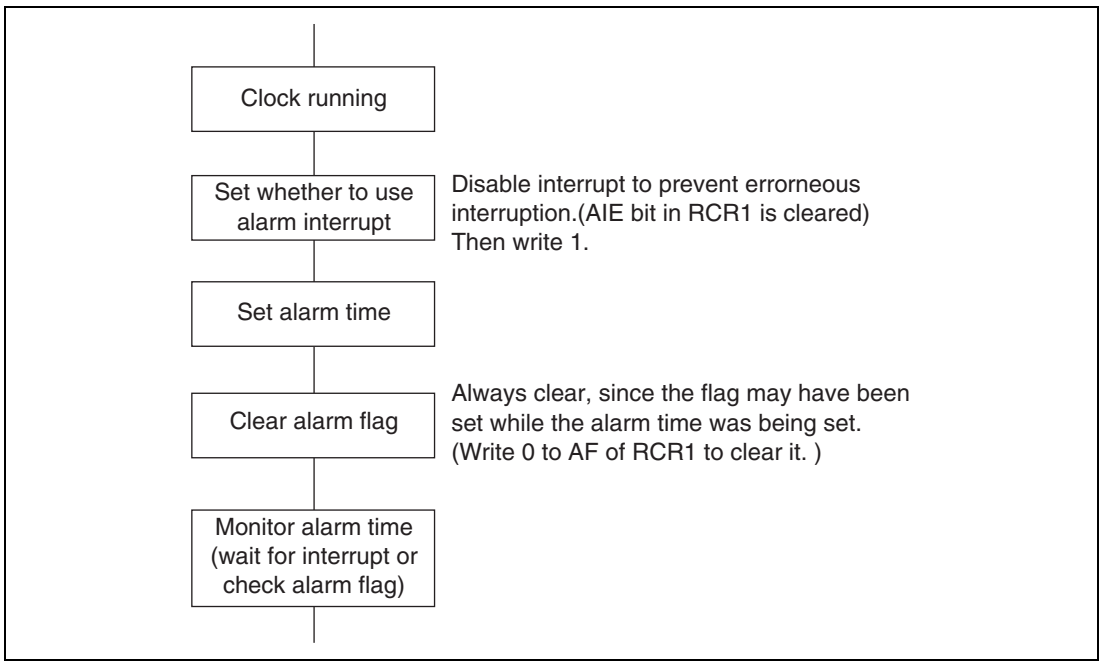


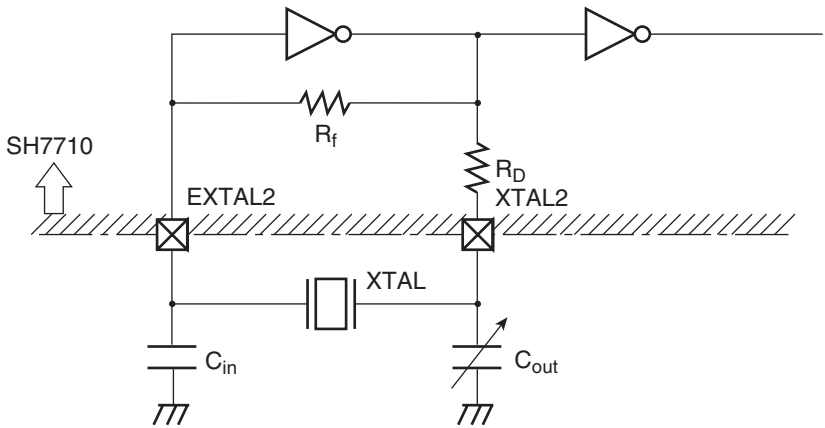
Figure 15.4 Using Alarm Function

15.4.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 15.2, and the RTC crystal oscillator circuit in figure 15.5.

Table 15.2 Recommended Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes:
1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include stray capacitance due to the wiring. Take care when using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit constants, stray 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, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 15.5 Example of Crystal Oscillator Circuit Connection

15.5 Usage Notes

15.5.1 Register Writing during RTC Count

The following RTC registers cannot be written to during an RTC count (while the START bit = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be stopped before writing to any of the above registers.

15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 15.6.

A periodic interrupt can be generated periodically at the interval set by the periodic interrupt flag (PES0–PES2) in RCR2. When the time set by the PES0–PES2 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when the periodic interrupt flag (PES0 to PES2) is set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

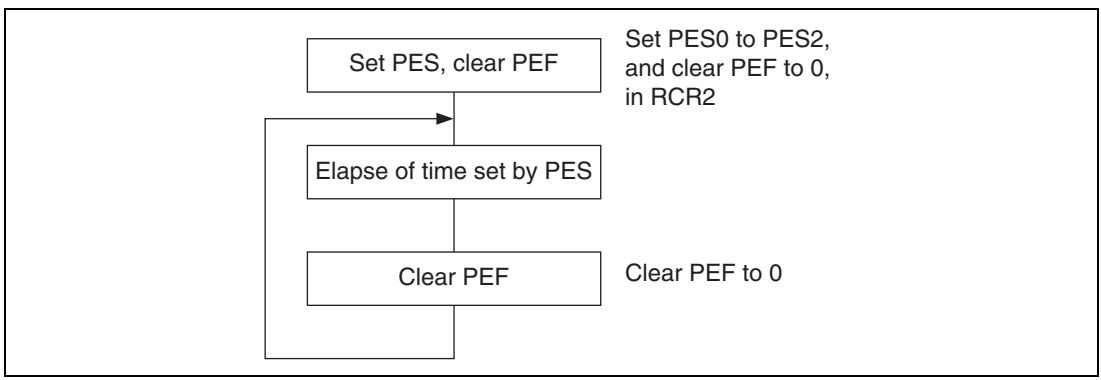


Figure 15.6 Using Periodic Interrupt Function

15.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after waiting for two RTC clocks or more.

15.5.4 Usage Note about RTC Power Supply

RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal oscillator circuit for RTC operates with VccQ-RTC. The control circuit and the RTC counter operate with Vcc (common to the internal circuit). Therefore, all power supplies other than VccQ-RTC should always be turned on even if only RTC operates.

Section 16 Serial Communication Interface with FIFO (SCIF)

This LSI has a two-channel serial communication interface with on-chip FIFO buffers (Serial Communication Interface with FIFO: SCIF). The SCIF can perform asynchronous and clock synchronous serial communication.

The SCIF provides a 16-stage FIFO register for both transmission and reception, enabling fast, efficient, and continuous communication.

16.1 Features

The SCIF features are listed below.

- Asynchronous 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 communication 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: If a framing error is following by at least one frame at the space “0” (low) level, a break is detected.
- Clock synchronous mode

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other chips that have a synchronous communication function.

  Data length: 8 bits

  Receive error detection: Overrun error
- 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 16-stage FIFO buffer structure, enabling fast and continuous serial data transmission and reception.
- On-chip baud rate generator allows any bit rate to be selected.

- Choice of serial clock source: Internal clock from the baud rate generator or external clock from the SCIF0CK and SCIF1CK pins.
- There are four interrupt sources—transmit-FIFO-data-empty, receive-FIFO-data-full, receive-error, and break. Each source can be requested independently.
- The DMA controller (DMAC) can be activated to execute a data transfer in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full.
- On-chip modem control functions ($\overline{\text{CTS0}}$/$\overline{\text{CTS1}}$ and $\overline{\text{RTS0}}$/$\overline{\text{RTS1}}$)
- When not in use, the SCIF can be stopped by halting its clock supply to reduce power consumption.
- 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.
- On reception a time out error (DR) can be detected
- The contents of the transmit FIFO data register (SCFTDR) and receive FIFO data register (SCFRDR) are undefined after a power-on or manual reset. Other registers are initialized by a power-on or manual reset, and retain their values in standby mode and in the module standby state.

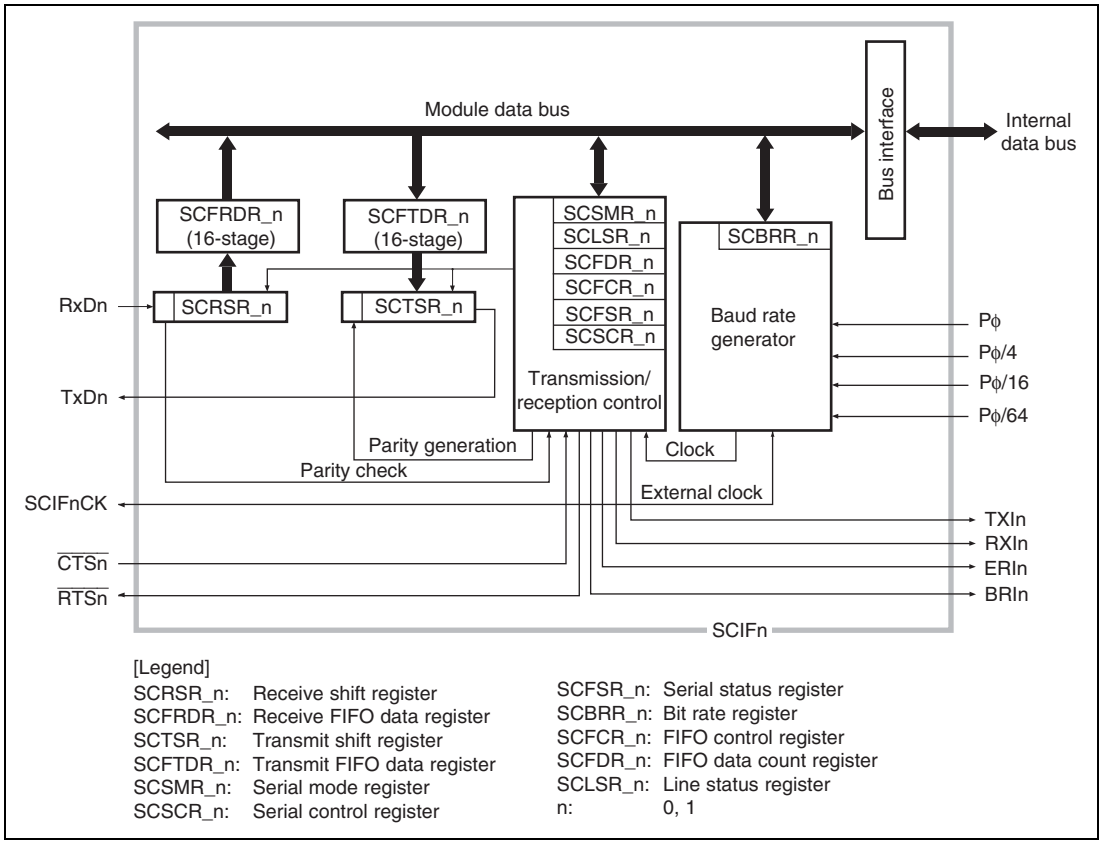


Figure 16.1 Block Diagram of SCIF

16.2 Input/Output Pins

Table 16.1 shows the SCIF pin configuration.

Table 16.1 Pin Configuration

Channel	Pin Name	Abbreviation	I/O	Function
0	Serial clock pin	SCIF0CK	Input/output	Clock input/output
	Receive data pin	RxD0	Input	Receive data input
	Transmit data pin	TxD0	Output	Transmit data output
	Modem control pin	$\overline{\text{CTS0}}$	Input	Transmission clear
	Modem control pin	$\overline{\text{RTS0}}$	Output	Transmit request
1	Serial clock pin	SCIF1CK	Input/output	Clock input/output
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data output
	Modem control pin	$\overline{\text{CTS1}}$	Input	Transmission clear
	Modem control pin	$\overline{\text{RTS1}}$	Output	Transmit request

Note: These pins function as serial pins by making the SCIF operation settings with the C/$\overline{\text{A}}$ bit in SCSMR, TE, RE, CKE1, and CKE0 bits in SCSCR, and MCE bit in SCFCR.

16.3 Register Descriptions

The SCIF has the following registers. For details on addresses and access sizes of these registers, see section 24, List of Registers.

Channel 0:

- Serial mode register_0 (SCSMR_0)
- Bit rate register_0 (SCBRR_0)
- Serial control register_0 (SCSCR_0)
- Transmit FIFO data register_0 (SCFTDR_0)
- Serial status register_0 (SCFSR_0)
- Receive FIFO data register_0 (SCFRDR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Line status register_0 (SCLSR_0)
- Receive shift register_0 (SCRSR_0)
- Transmit shift register_0 (SCTSR_0)

Channel 1:

- Serial mode register_1 (SCSMR_1)
- Bit rate register_1 (SCBRR_1)
- Serial control register_1 (SCSCR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial status register_1 (SCFSR_1)
- Receive FIFO data register_1 (SCFRDR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Line status register_1 (SCLSR_1)
- Receive shift register_1 (SCRSR_1)
- Transmit shift register_1 (SCTSR_1)

16.3.1 Receive Shift Register (SCRSR)

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the 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 the receive FIFO data register SCFRDR, automatically.

SCRSR cannot be directly read or written to by the CPU.

16.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-stage FIFO register 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 the receive FIFO data register is full (16 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 the receive FIFO data register, an undefined value will be returned. When the receive FIFO data register is full of receive data, subsequent serial data is lost.

The contents of SCFRDR are undefined after a power-on reset or manual reset.

16.3.3 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 sequentially to the 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 or written to by the CPU.

16.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit 16-stage FIFO data register 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 16 bytes of transmit data. Data written in this case is ignored.

The contents of SCFTDR are undefined after a power-on reset or manual reset.

16.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial communication format and select the clock source of the baud rate generator.

SCSMR can be read or written to by the CPU at all times.

SCSMR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

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 clock synchronous mode as the SCIF operating mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode data length. In clock synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register (SCFTDR) is not transmitted.
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 in reception. In clock synchronous mode, parity bit addition and checking is not performed, 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.

Bit	Bit Name	Initial Value	R/W	Description
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clock synchronous mode, and when parity addition and checking is disabled in asynchronous mode. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. 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. 2. 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 Selects 1 or 2 bits as the stop bit length. The STOP bit setting is only valid in asynchronous mode. When clock synchronous mode is set, the STOP bit setting is invalid since stop bits are not added. 0: 1 stop bit*[1] 1: 2 stop bits*[2] Notes: 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. 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.
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	Select the clock source for the on-chip baud rate generator. 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64

16.3.6 Serial Control Register (SCSCR)

SCSCR performs enabling or disabling of the SCIF transfer operations and interrupt requests, and selection of the serial clock source.

SCSCR can be read or written to by the CPU at all times.

SCSCR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

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 generation of a transmit-FIFO-data-empty interrupt (TXI) request when the TDFE flag in SCFSR is set to 1 after the serial transmit data is transferred from SCFTDR to SCTSR and the number of data bytes in the transmit FIFO register is equal to or below the trigger set number. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled* 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled Note: * TXI interrupt requests can be cleared by writing transmit data exceeding the transmit trigger set number to SCFTDR, reading 1 from the TDFE flag, then clearing it to 0, or by clearing the TIE bit to 0.

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, receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, or 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 request can be cleared by reading 1 from the RDF flag or DR flag, then clearing the flag to 0, or by clearing the RIE bit to 0. The ERI and BRI requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or 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. 0: Transmission disabled 1: Transmission enabled* Note: * SCSMR and SCFCR settings must be made, the transmit 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. 0: Reception disabled*[1] 1: Reception enabled*[2] Notes: 1. Clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER, and ORER flags, which retain their state. 2. SCSMR and SCFCR settings must be made, the receive 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) request and break interrupt (BRI) request. The REIE bit setting is available when the RIE bit is cleared to 0. 0: Receive-error interrupt (ERI) request and break interrupt (BRI) request disabled* 1: Receive-error interrupt (ERI) request and break interrupt (BRI) request enabled Note: * A receive-error interrupt (ERI) request and break interrupt (BRI) request can be cleared by reading 1 from the ER, BRK, and ORER flags, then clearing the flags to 0, or by clearing the RIE and REIE bits to 0. Even if the RIE bit is cleared to 0, setting the REIE bit to 1 enables generation of the ERI and BRI requests. This setting is achieved to notify the ERI and BRI requests to the interrupt controller at the DMAC transfer.
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 and 0
0	CKE0	0	R/W	Select the SCIF clock source and enable or disable clock output from the SCIFnCK pin. The combination of the CKE1 bit and CKE0 bit determines whether the SCIFnCK pin is set as a serial clock output pin or a serial clock input pin. The setting of the CKE0 bit is available in internal clock operation (CKE1 = 0). In the case of external clock operation (CKE1 = 1), the setting of the CKE0 bit is not available. The CKE1 and CKE0 bits must be set before the SCIF operating mode is selected by SCSMR. • Asynchronous mode 00: Internal clock/SCIFnCK pin functions as input pin (input signal ignored) 01: Internal clock/SCIFnCK pin functions as clock output*[2] 1-*[1]: External clock/SCIFnCK pin functions as clock input*[3] • Clock synchronous mode 00: Internal clock/SCIFnCK pin functions as synchronous clock output 01: Internal clock/SCIFnCK pin functions as synchronous clock output 1-*[1]: External clock/SCIFnCK pin functions as synchronous clock input Notes: 1. When CKE1 = 1, the value of CKE0 is don't care. 2. The output clock frequency is 16 times the bit rate. 3. The input clock frequency is 16 times the bit rate.

16.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The lower 8 bits specify the status flags that indicate the SCIF operating status. The upper 8 bits indicate the receive error number of data in the receive-FIFO register.

SCFSR can be read or written to by the CPU at all times. However, 1 cannot be written to the ER, TEND, TDFE, BRK, RDF, and DR flags. Also note that in order to clear these flags to 0, they must be read as 1 beforehand.

The FER and PER flags are read-only flags and cannot be modified.

SCFSR is initialized to H'0060 by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
15	PER3	0	R	Parity Error Number 3 to 0
14	PER2	0	R	Indicate the number of data bytes, in which parity errors are generated, in receive data stored in SCFRDR.
13	PER1	0	R	
12	PER0	0	R	After setting the ER bit in SCFSR, the values of bits 15 to 12 indicate the number of parity error generated data. When all 16 bytes of receive data in SCFRDR has parity errors, the PER3 to PER0 bits indicate 0.
11	FER3	0	R	Framing Error Number 3 to 0
10	FER2	0	R	Indicate the number of data bytes, in which framing errors are generated, in receive data stored in SCFRDR.
9	FER1	0	R	
8	FER0	0	R	After setting the ER bit in SCFSR, the values of bits 11 to 8 indicate the number of framing error generated data. When all 16 bytes of receive data in SCFRDR has framing errors, the FER3 to FER0 bits indicate 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.*[1] 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*[2] • 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 Notes: 1. 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 data read from SCFRDR. 2. When the stop length is 2 bits, only the first stop bit is checked for a value of 1; the second stop bit is not checked.

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	Transmit End Indicates that there is no valid data in SCFTDR when the last bit of the transmit character is sent, and transmission has been ended. 0: Transmission is in progress [Clearing conditions] • When the TEND flag is cleared to 0 after the transmit data is written to SCFTDR and TEND = 1 is read • When data is written to SCFTDR by the DMAC 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • The TE bit in SCSCR is cleared to 0 • When there is no transmit data in SCFTDR on transmission of the last bit of 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 been equal 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, and 0 is written to the TDFE bit after reading TDFE = 1 • 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 [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes is equal to or below the transmit trigger set number as the result of a transmit operation* Note: * As SCFTDR is a 16-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 16 – (transmit trigger set number). Data written in excess of this will be ignored. The number of data bytes in SCFTDR is indicated by the upper bits in SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	Break Detect In asynchronous mode, indicates that a receive data break signal has been detected or not. 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 that there is a framing error or not in the data read from SCFRDR. 0: There is no framing error in the receive data read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in SCFRDR read data 1: There is a framing error in the receive data read from SCFRDR [Setting condition] When there is a framing error in SCFRDR read data

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error In asynchronous mode, indicates that there is a parity error or not in the data read from SCFRDR. 0: There is no parity error in the receive data read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in SCFRDR read data 1: There is a parity error in the receive data read from SCFRDR [Setting condition] When there is a parity error in SCFRDR read data

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 is less than the receive trigger set number, and 0 is written to RDF after reading RDF = 1 • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR is less than 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 16-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If data is read when SCFRDR is empty, an undefined value will be returned. The number of receive data bytes in SCFRDR is indicated by the lower bits in SCFDR.

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. 0: Reception is in progress or has ended successfully 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, and 0 is written to DR after reading DR = 1 • When all the receive data in SCFRDR is 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* Note: * Corresponds to 1.5 frame time when the format of 8-bit length and 1 stop bit is used. etu: Elementary time unit (time for transfer of 1 bit)

Note: * Only 0 can be written for clearing the flags.

16.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

SCBRR can be read or written to by the CPU at all times.

SCBRR is initialized to H'FF by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See table 16.2 for the relation between n and the clock.)

Table 16.2 Relationship between n and Clock

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

16.3.9 FIFO Control Register (SCFCR)

SCFCR performs data count resetting and trigger data number setting for the transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can be read or written to by the CPU at all times.

SCFCR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

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	$\overline{\text{RTS}}$ Output Active Trigger 2 to 0
9	RSTRG1	0	R/W	The $\overline{\text{RTS}}$ signal goes high when the number of receive data bytes in SCFRDR is equal to or greater than the trigger set number shown in below.
8	RSTRG0	0	R/W	$\overline{\text{RTS}}$ active trigger: 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14

Bit	Bit Name	Initial Value	R/W	Description
7	RTRG1	0	R/W	Receive FIFO Data Number Trigger 1 and 0
6	RTRG0	0	R/W	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 in below. Asynchronous mode — Clock synchronous mode 00: 1 — 00: 1 01: 4 — 01: 2 10: 8 — 10: 8 11: 14 — 11: 14
5	TTRG1	0	R/W	Transmit FIFO Data Number Trigger 1 and 0
4	TTRG0	0	R/W	Set the number of remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when, as the result of a transmit operation, the number of transmit data bytes in SCFTDR is equal to or below the trigger set number shown in below. 00: 8 (8) 01: 4 (12) 10: 2 (14) 11: 0 (16) Note: The values in parentheses are the number of empty bytes in SCFTDR when the flag is set.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. This bit is valid only in asynchronous mode. 0: Modem signal disabled* 1: Modem signal enabled Note: * $\overline{\text{CTS}}$ is fixed at active 0 regardless of the input value, and $\overline{\text{RTS}}$ is also fixed at 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the transmit data in the transmit FIFO data register and resets it to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * A reset operation is performed in the event of a power-on reset or manual reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in the receive FIFO data register and resets it to the empty state. 0: Reset operation disabled* 1: Reset 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 (TxD) and receive input pin (RxD), and $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin, enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

16.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register that indicates the number of data bytes stored in SCFTDR and SCFRDR.

Bits 12 to 8 show the number of transmit data bytes in SCFTDR, and bits 4 to 0 show the number of receive data bytes in SCFRDR.

SCFDR can be read by the CPU at all times.

SCFDR is initialized to H'0000 by a power-on reset or manual reset. It is not initialized in standby mode or in the module standby state, and retains its contents.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	T4	0	R	Bits 12 to 8 in SCFDR show the number of untransmitted data bytes in SCFTDR. A value of H'00 means that there is no transmit data, and a value of H'10 means that SCFTDR is full of transmit data.
11	T3	0	R	
10	T2	0	R	
9	T1	0	R	
8	T0	0	R	
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	R4	0	R	Bits 4 to 0 in SCFDR show the number of receive data bytes in SCFRDR. A value of H'00 means that there is no receive data, and a value of H'10 means that SCFRDR is full of receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

16.3.11 Line Status Register (SCLSR)

SCLSR is a 16-bit register that indicates whether an overrun error occurs or not during reception.

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 and reception is ended abnormally. 0: Reception is in progress, or reception has ended successfully*[1] [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1 1: An overrun error occurred during reception*[2] [Setting condition] When serial reception is completed while the receive FIFO is full Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. 2. 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.

16.4 Operation

16.4.1 Overview

The SCIF can carry out serial communication in asynchronous mode, in which synchronization is achieved character by character, and in clock synchronous mode, in which synchronization is achieved with clock pulses.

16-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead and enabling fast, continuous communication to be performed. Also, the $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals are included as modem control signals. Transfer format is selected by SCSMR. This is shown in table 16.3. 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. This is shown in table 16.4.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- 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, overrun errors, receive-FIFO-data-full state, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- The SCIF clock source: Choice of internal or external clock

  When internal clock is selected: The SCIF operates on the baud rate generator clock.

  When external clock is selected: Clock with frequency16 times the bit rate must be input. (The on-chip baud rate generator is not used.)

Clock synchronous Mode

- Transfer format: Fixed to 8-bit data
- Detection of overrun errors during reception
- The SCIF clock source: Choice of internal or external clock

  When internal clock is selected: The SCIF operates on the baud rate generator clock and outputs the synchronous clock

  When external clock is selected: The SCIF operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 16.3 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					The 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	*	*	*	Clock synchronous mode	8-bit data	No	No

Table 16.4 SCSMR and SCSCR Settings for the SCIF Clock Source Selection

SCSMR	SCSCR				
Bit 7: C/$\overline{A}$	Bit 1: CKE1	Bit 0: CKE0	Mode	Clock Source	SCIFnCK Pin Function
0	0	0	Asynchronous mode	Internal	SCIF does not use the SCIFnCK pin
		1			Outputs a clock with frequency 16 times the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1			
1	0	0	Clock synchronous mode	Internal	Outputs the synchronous clock
		1			
	1	0		External	Inputs the synchronous clock
		1			

16.4.2 Serial Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time. Figure 16.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial communication character consists of a start bit (low), data (LSB first), parity bit (high/low), and stop bit (high), in this order. In asynchronous mode, the SCIF synchronizes at the falling edge of the start bit during reception. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Therefore, communication data is latched at the center of each bit.

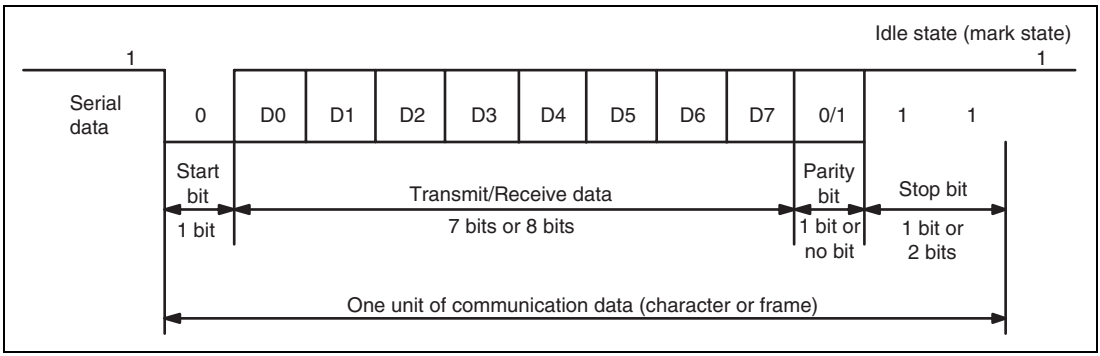


Figure 16.2 Data Format in Asynchronous Communication (Example of 8-Bit Data with Parity and 2 Stop Bits)

Data Transfer Format: Table 16.5 shows the transfer formats that can be used in asynchronous mode. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 16.5 Serial Transfer Formats

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		
		1	S	8-bit data								STOP	STOP	
	1	0	S	8-bit data								P	STOP	
		1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
		1	S	7-bit data							STOP	STOP		
	1	0	S	7-bit data							P	STOP		
		1	S	7-bit data							P	STOP	STOP	

S: Start bit
STOP: Stop bit
P: Parity bit

Clock: The SCIF transfer clock is set by the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details, see table 16.4.

When an external clock is input to the SCIFnCK pin, the clock with frequency 16 times the bit rate must be input.

When the SCIF operates on an internal clock, it can output a clock from the SCIFnCK pin. At this time output clock frequency is 16 times the bit rate.

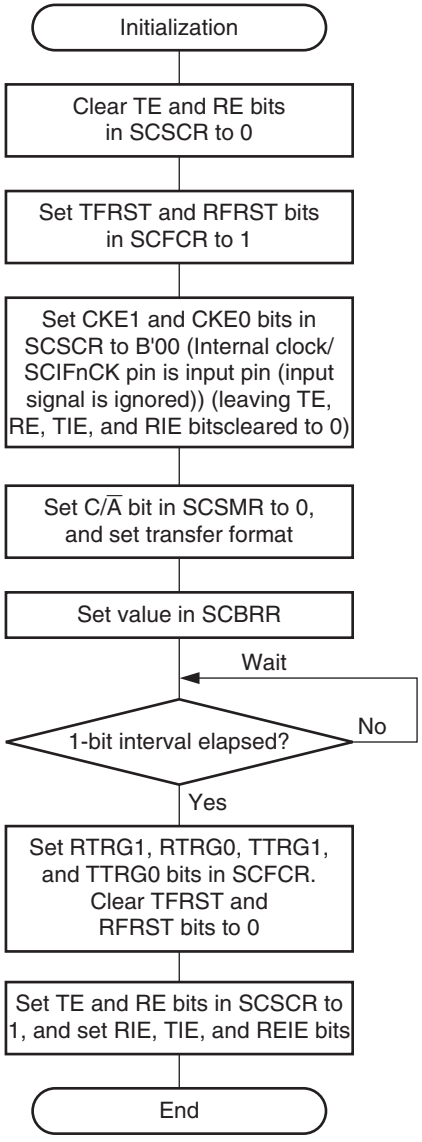
Data Transfer Operations:

The SCIF Initialization: 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 transfer format, etc., is changed, 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 TE and RE bits to 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND bit in SCFSR has been set to 1. Clearing to 0 can also be performed during transmission, but the data being transmitted will go to the high-impedance state after the clearance. Before setting TE to 1 again to start transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCFTDR.

When an external clock is used, the clock should not be stopped during operation including initialization because its operation becomes unreliable.

Figure 16.3 shows a sample the SCIF initialization flowchart.

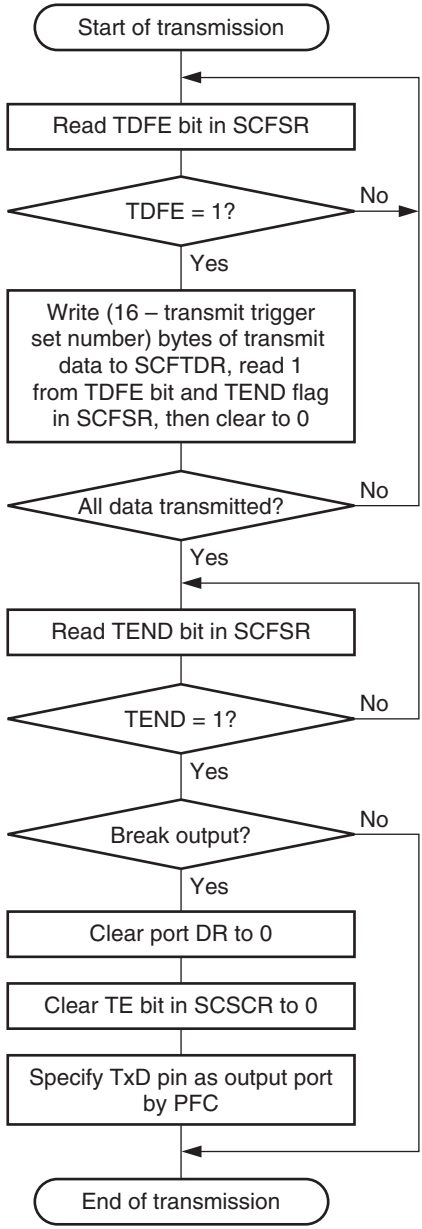


1. Set the clock selection in SCSCR. Be sure to clear bits RIE, TIE, TE, and RE to 0.
2. Set the transfer format in SCSMR.
3. Write a value corresponding to the bit rate into SCBRR. (Not necessary if an external clock is used.)
4. Wait at least one bit interval, then set the TE bit or RE bit in SCSCR to 1. Also set the RIE, TIE, and REIE bits.
   Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 16.3 Sample the SCIF Initialization Flowchart

Serial Data Transmission: Figure 16.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



1. SCIF status check and transmit data write:
   Read the serial status register (SCFSR) and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, read 1 from the TDFE and TEND flags, then clear these flags to 0.
   The number of data bytes that can be written is 16 – (transmit trigger set number).
2. Serial transmission continuation procedure:
   To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE bit to 0.
3. Break output at the end of serial transmission:
   To output a break in serial transmission, clear the port data register (DR) to 0 before clearing the TE bit in SCSCR to 0, and then specify the TxD pin as an output port by the PFC.
   In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 16.4 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 16 – (transmit trigger set number).
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 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, 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.

Figure 16.5 shows an example of the operation for transmission in asynchronous mode.

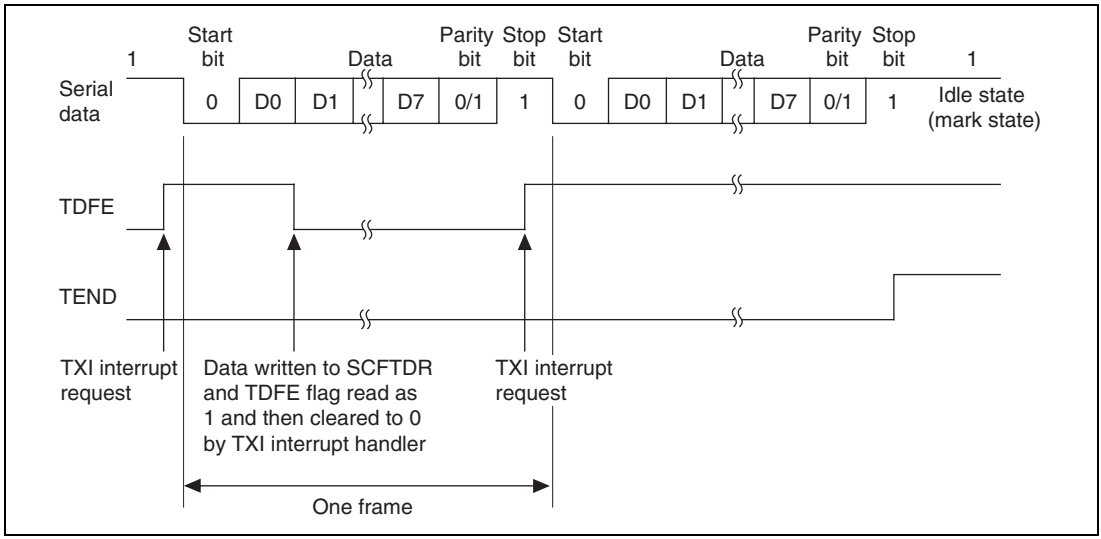
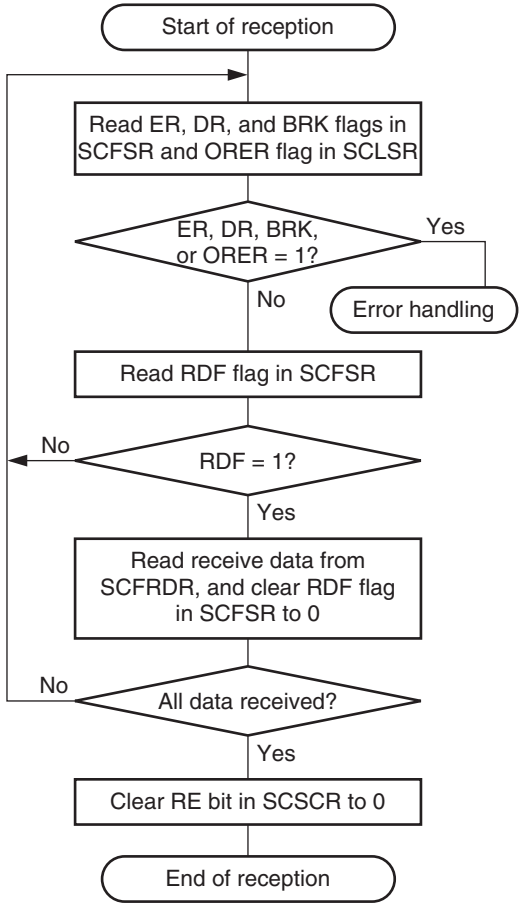


Figure 16.5 Example of Transmit Operation (Example of 8-Bit Data with Parity and 1 Stop Bit)

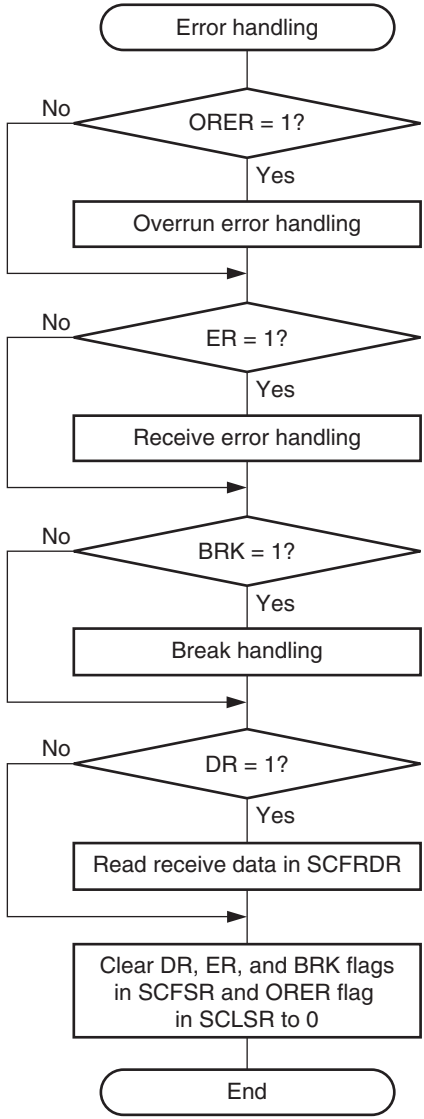
Serial Data Reception: Figures 16.6 and 16.7 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.



1. Receive error handling and break detection: Read the DR, ER, and BRK flags in SCFSR and ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.
2. SCIF status check and receive data read: Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.
3. Serial reception continuation procedure: To continue serial reception, read at least the receive trigger set number of data bytes from SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading the lower bits of SCFDR.

Figure 16.6 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data 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 16.7 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the communication 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 cleared to 0 and an overrun error does not occur.

   D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

Note: Reception continues when a receive error (a framing error or parity error) occurs.

4. If the RIE bit in SCSCR is set to 1 when the RDF flag 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 flag or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 16.8 shows an example of the operation for reception in asynchronous mode.

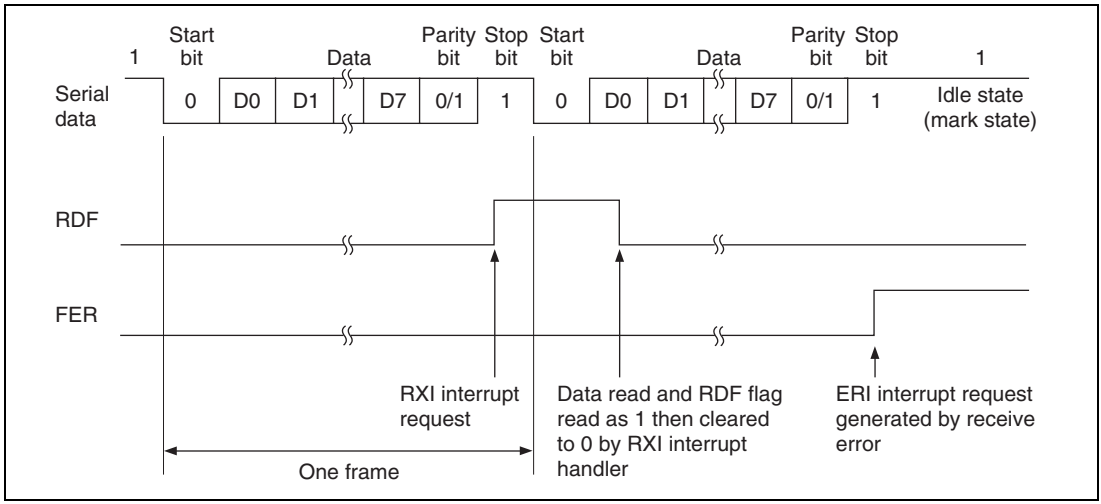


Figure 16.8 Example of SCIF Receive Operation (Example of 8-Bit Data with Parity and 1 Stop Bit)

Modem Function: When using a modem function, transmission can be stopped and started again according to the $\overline{\text{CTS}}$ input value. When the $\overline{\text{CTS}}$ is set to 1 during transmission, the data enters a mark state after transmitting one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting with a start bit.

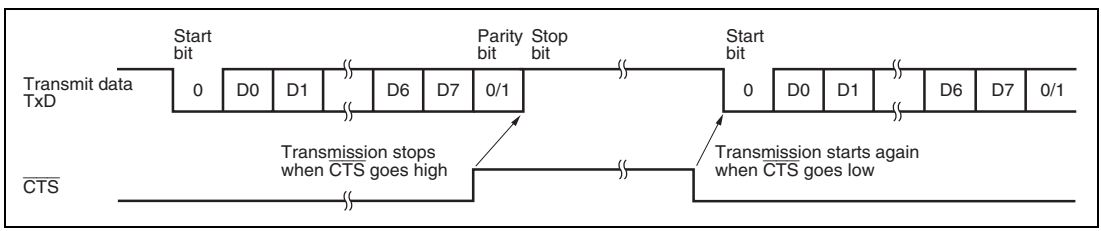


Figure 16.9 $\overline{\text{CTS}}$ Control Operation

When using a modem function and the receive FIFO (SCFRDR) is at least the number of the $\overline{\text{RTS}}$ output trigger, the $\overline{\text{RTS}}$ signal goes high.

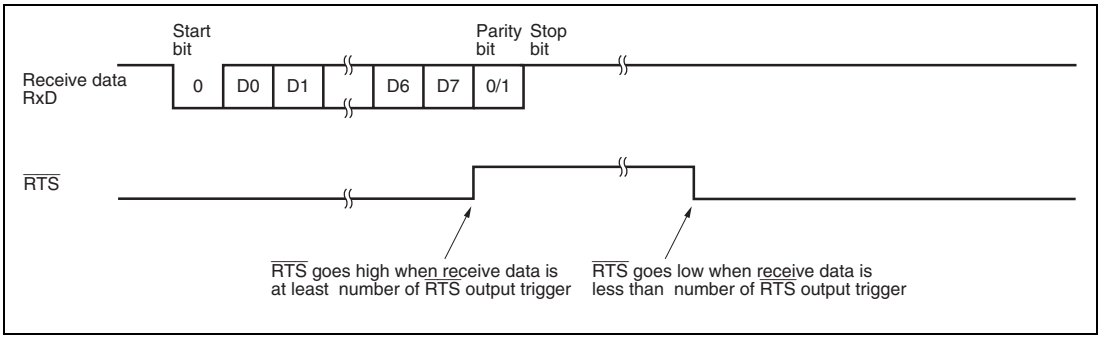


Figure 16.10 $\overline{\text{RTS}}$ Control Operation

16.4.3 Serial Operation in Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data synchronizing with clock pulses. This mode is suitable for high-speed serial communication. In the SCIF, the transmitter and receiver are independent. Therefore, by sharing the same clock, full-duplex communication can be performed. Figure 16.11 shows the general format of clock synchronous serial communication.

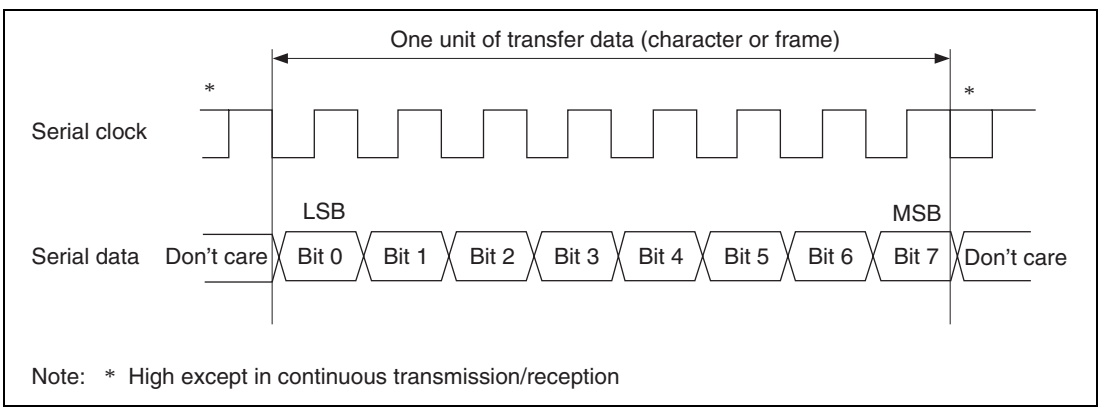


Figure 16.11 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, data on the communication line is output from one fall of the serial clock to the next. Data is guaranteed valid at the rise of the serial 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 MSB.

In clock synchronous mode, the SCIF receives data in synchronization with the rise of the serial clock.

Data Transfer Format: A fixed 8-bit data format is used. No parity bit is added.

Clock: An internal clock generated by the on-chip baud rate generator or an external synchronous clock input from the SCIFnCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SCSMR and bits CKE1 and CKE0 in SCSCR. For details, see table 16.4.

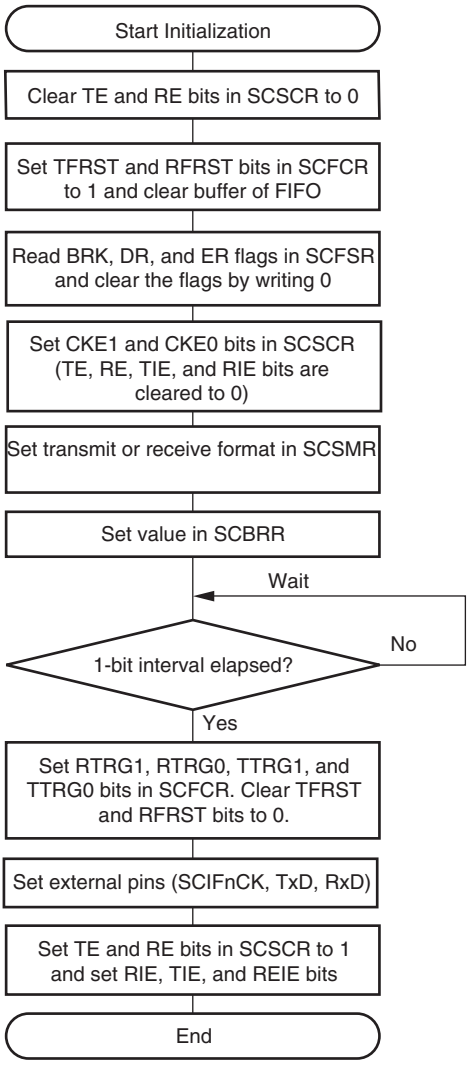
When the SCIF is operated on an internal clock, synchronous clock is output from the SCIFnCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transmission/reception is performed the clock is fixed high. If an internal clock is selected when only reception is performed, clock pulse is output continuously until the number of data bytes in the receive FIFO reaches the receive trigger set number while the RE bit in SCSCR is 1.

Data Transfer Operations:

The SCIF Initialization: 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 mode, communication format etc., is changed, 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 the RDF, PER, FER, and ORER flags and contents of SCFRDR are retained even if the RE bit is cleared to 0.

Figure 16.12 shows a sample the SCIF initialization flowchart.



1. Keep the TE and RE bits cleared to 0 until initialization has been completed.
2. Set the CKE1 and CKE0 bits.
3. Set the transfer or receive format in SCSMR.
4. Write a value corresponding to the bit rate in SCBRR. (Not necessary if an external clock is used.) After this setting wait for at least 1-bit interval.
5. Set the external pins. Specifies the pins as RxD input in reception and TxD output in transmission. Set the SCIFnCK input/output according to the CKE1 and CKE0 settings.
6. Set the TE bit or RE bit in SCSCR to 1. Also, set the TIE, RIE, and REIE bits. At this time, the TxD, RxD, and SCIFnCK pins can be used. In transmission, the TxD pin is in the mark state. When reception in clock synchronous mode and synchronous clock output (clock master) are selected, a clock is output from the SCIFnCK pin.

Figure 16.12 Sample the SCIF Initialization Flowchart

Serial Data Transmission: Figure 16.13 shows a sample flowchart for serial transmission. Use the following procedure for serial data transmission after enabling the SCIF for transmission.

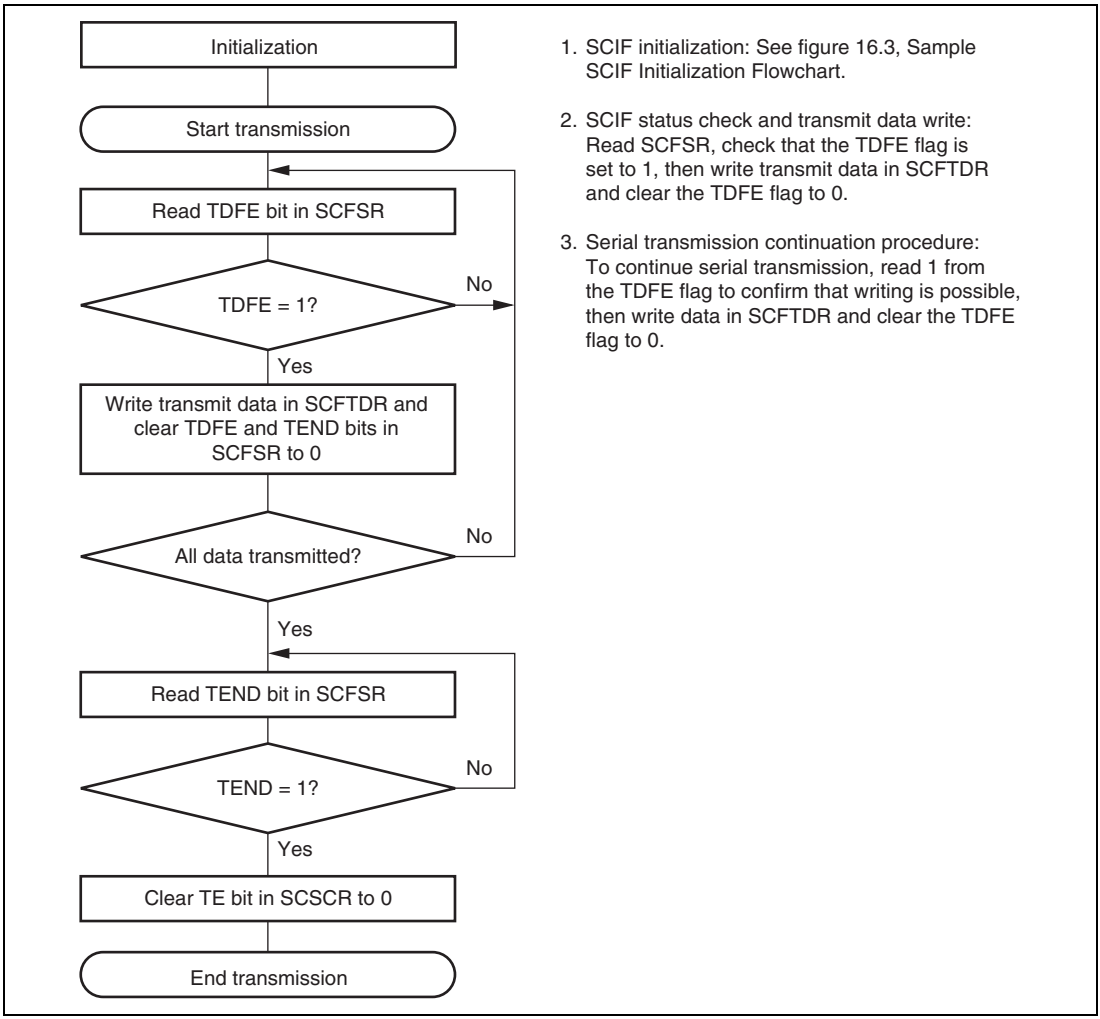


Figure 16.13 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCFTSR 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 16 – (transmit trigger set number).

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 and external clock is specified, the SCIF outputs data in synchronization with the input clock. Serial transmit data is output in order from LSB to MSB from the TxD pin.
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, the stop bit is sent, 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, the stop bit is sent, and then the state of the TxD pin is held.
4. After serial transmission has completed, the SCIFnCK pin is fixed to high.

Figure 16.14 shows an example of the SCIF transmit operation.

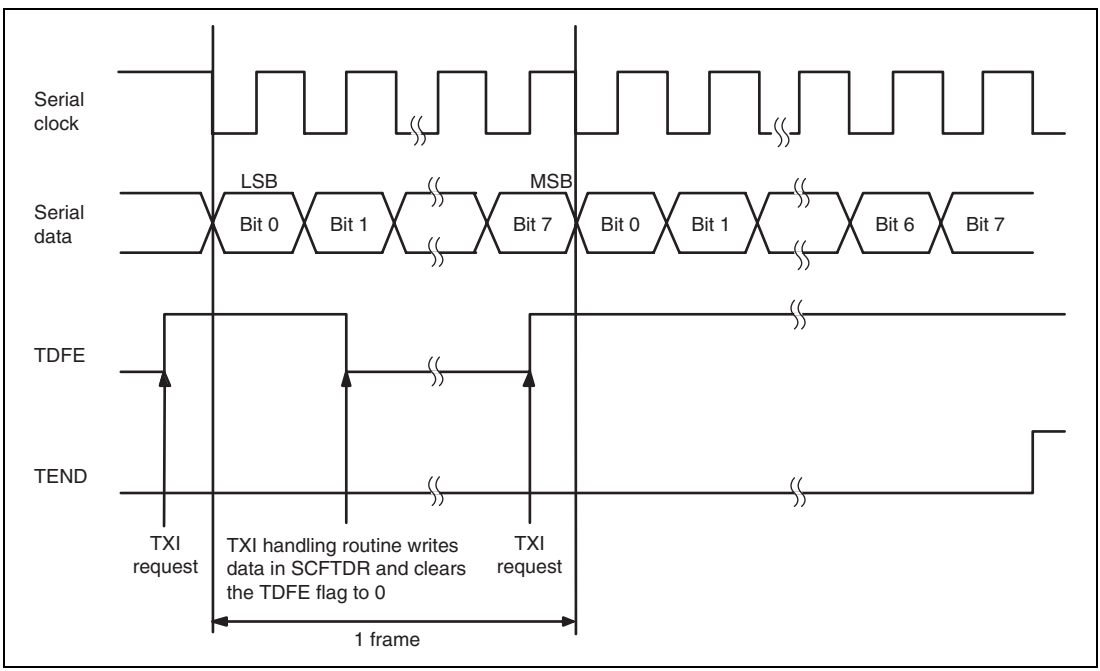
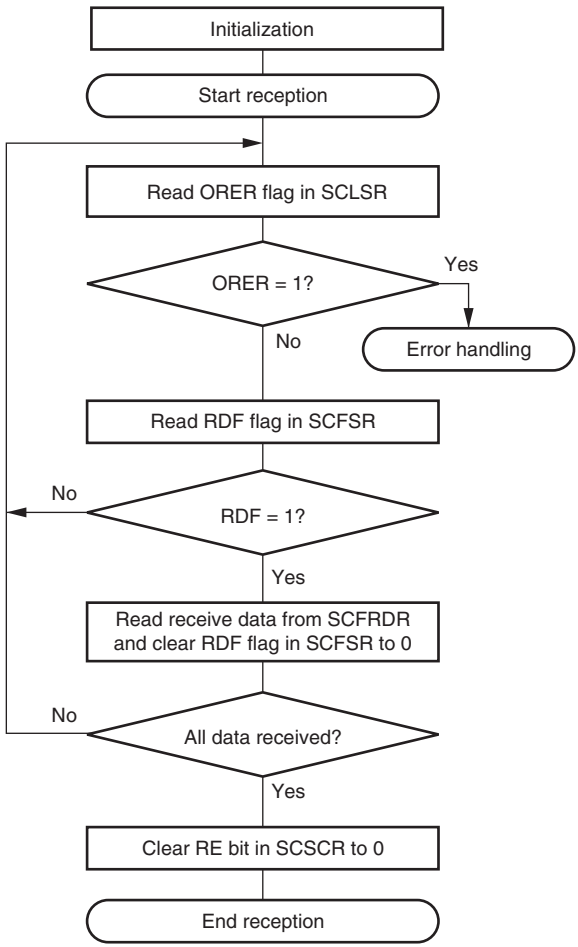


Figure 16.14 Example of the SCIF Transmit Operation

Serial Data Reception: Figures 16.15 and 16.16 show sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

To change the operating mode from asynchronous mode to clock synchronous mode without initialization, be sure to confirm that the flags ORER, PER3 to PER0, and FER3 to FER0 are cleared to 0.



1. SCIF initialization: See figure 16.3, Sample SCIF Initialization Flowchart.
2. Receive error handling: If a receive error occurs, read the ORER flag in SCLSR, then after executing the necessary error handling, clear the ORER flag to 0. Serial reception cannot be continued while the ORER flag is set to 1.
3. SCIF status check and receive data read: Read SCFSR, check that the RDF flag is set to 1, then read receive data in SCFRDR and clear the RDF flag to 0. Notification that the RDF flag has changed from 0 to 1 can also be given by the RXI.
4. Serial reception continuation procedure: To continue serial reception, read at least the receive trigger set number of data bytes from SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The number of receive data in SCFRDR can be ascertained by reading the lower bits of SCFDR. However, if the DMAC is activated by the RXI and the value is read from SCFRDR, the RDF flag is cleared automatically.

Figure 16.15 Sample Serial Reception Flowchart

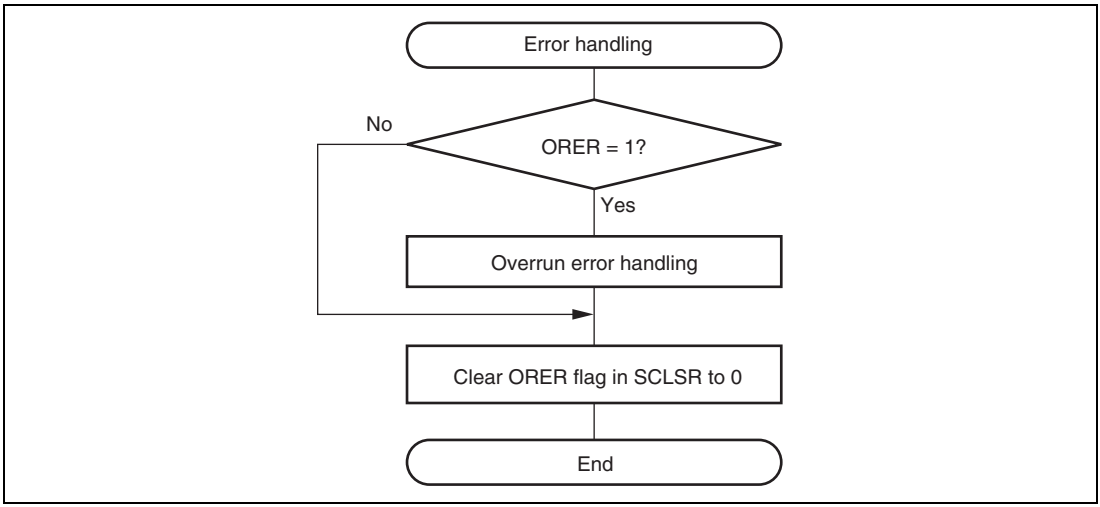


Figure 16.16 Sample Serial Reception Flowchart

In serial reception, the SCIF operates as described below.

1. The SCIF internally initializes in synchronization with the synchronous clock input or output.
2. The SCIF stores receive data in SCRSR in order from LSB to MSB. After reception, the SCIF checks whether receive data can be transmitted from SCRSR to SCFRDR. If this check is passed, the SCIF stores the receive data in SCFRDR. If an overrun error is detected by an error check, following reception can not be performed.
3. If the RDF flag is set to 1 and the RIE bit in SCSCR is set to 1, the SCIF requests a receive-FIFO-data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit or REIE bit in SCSCR is set to 1, the SCIF requests a break interrupt (BRI).

Figure 16.17 shows an example of the SCIF receive operation.

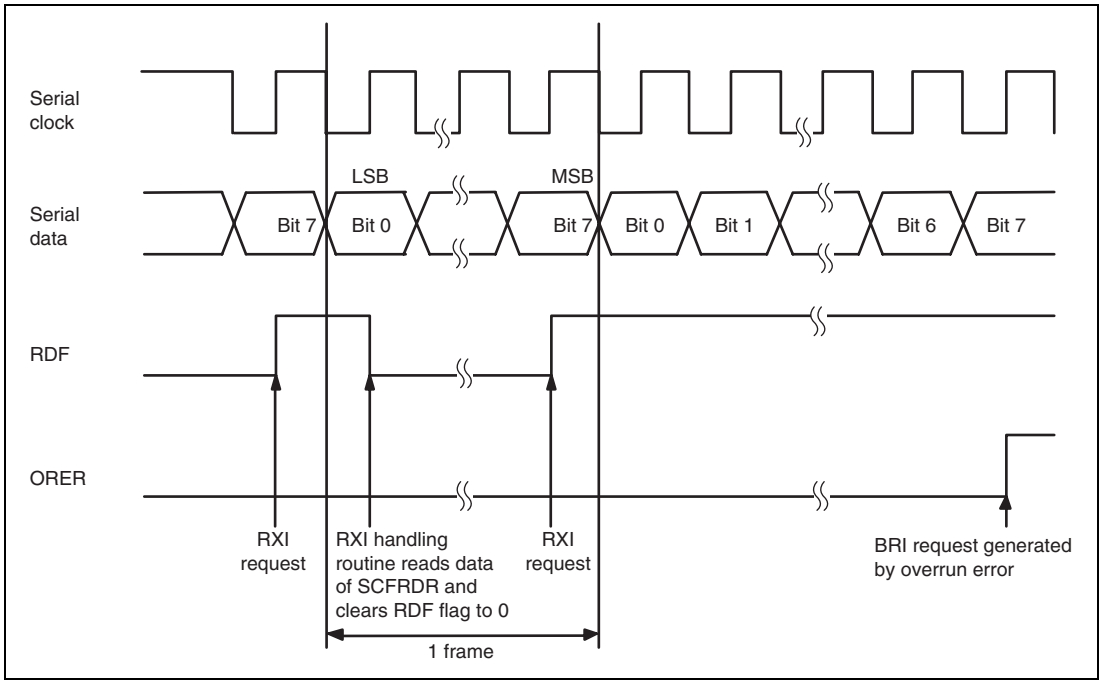
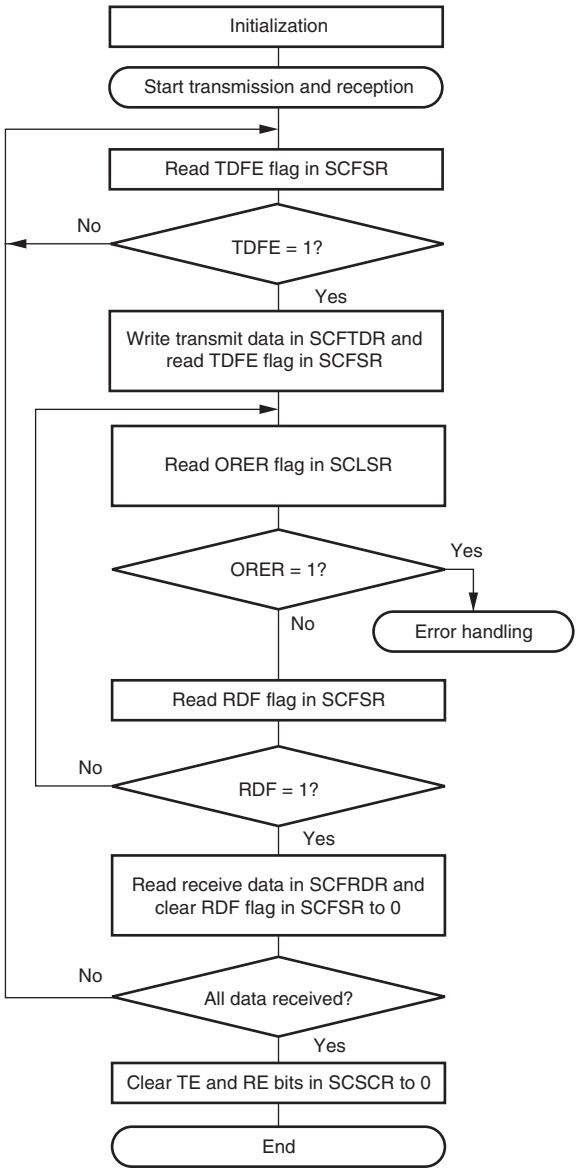


Figure 16.17 Example of the SCIF Receive Operation

Simultaneous Serial Data Transmission and Reception: Figure 16.18 shows a sample flowchart for simultaneous serial transmission and reception.

Before performing simultaneous serial data transmission and reception according to the processes described below, specify the SCIF as the transmit/receive enable state.



1. SCIF initialization: See figure 16.3, Sample SCIF Initialization Flowchart.

2. SCIF status check and receive data write: Read SCFSR, check that the TDFE flag is set to 1, then write transmit data in SCFTDR and clear the TDFE flag to 0. Notification that the TDFE flag has changed from 0 to 1 can also be given by the TXI.

3. Receive error handling: If a receive error occurs, read the ORER flag in SCLSR, then after executing the necessary error handling, clear the ORER flag to 0. Serial reception cannot be continued while the ORER flag is set to 1.

4. SCIF status check and receive data read: Read SCFSR, check that the RDF flag is set to 1, then read receive data from SCFRDR and clear the RDF flag to 0. Notification that the RDF flag has changed from 0 to 1 can also be given by the RXI.

5. Serial transmission/reception continuation procedure: To continue serial reception, before the MSB of the current frame is received, read the RDF flag and SCFRDR and clear the RDF flag to 0. Also, check that the TDFE flag is set to 1 and data can be written before transmitting the MSB of the current frame. Futhermore, write data in SCFTDR and clear the TDFE flag to 0.

Note: When switching from transmission or reception to simultaneous transmission and reception, clear the TE and RE bits to 0, then set the both bits to 1 simultaneously.

Figure 16.18 Sample Serial Data Transmission/Reception Flowchart

16.5 SCIF Interrupt Sources and DMAC

The SCIF supports four interrupt sources—transmit-FIFO-data-empty interrupt (TXI), receive-error interrupt (ERI), receive-FIFO-data-full interrupt (RXI), and break interrupt (BRI). Table 16.6 shows the interrupt sources and their order of priority. For priorities and the relationship with non-the SCIF interrupts, see section 4, Exception handling. The interrupt sources can be 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.

When the TXI is enabled by the TIE bit, if the TDFE flag in SCFSR is set to 1, a TXI request and a transmit-FIFO-data-empty DMA transfer request are generated. When the TXI is disabled by the TIE bit, if the TDFE flag is set to 1, only the transmit-FIFO-data-empty DMA transfer request is generated. The DMAC can be activated and data transfer performed on generation of the transmit-FIFO-data-empty DMA transfer request.

When the RXI is enabled by the RIE bit, if the RDF flag or DR flag in SCFSR is set to 1, an RXI request and a receive-FIFO-data-full DMA transfer request are generated. When the RXI is disabled by the RIE bit, if the RDF flag or DR flag is set to 1, only the receive-FIFO-data-full DMA transfer request is generated. The DMAC can be activated and data transfer performed on generation of the receive-FIFO-data-full DMA transfer request. The generation of the RXI and the receive-FIFO-data-full DMA transfer requests 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 request is generated. When using the DMAC for transmission/reception, set and enable the DMAC before making the SCIF settings. See section 13, Direct Memory Access Controller (DMAC), for details on the DMAC setting procedure. Also set the RXI and TXI requests not to be output to the interrupt controller. If the interrupt requests are set to be generated, the interrupt requests to the interrupt controller are cleared by the DMAC regardless of the interrupt handling program.

When the RIE bit is cleared to 0 and the REIE bit is set to 1 in SCSCR, the ERI or BRI request can be generated without generating the RXI request. Note that the TXI indicates that writing the transmit data is enabled, while the RXI indicates that the receive data is in SCFRDR.

Table 16.6 The SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)*	Possible	
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

Note: * The RXI by the DR is enabled only in asynchronous mode.
See section 4, Exception Handling, for priorities and the relationship with non-the SCIF interrupts.

16.6 Usage Notes

Note the following when using the SCIF.

SCFTDR Writing and 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 the TDFE flag 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 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 bits 12 to 8 in SCFDR.

SCFRDR Reading and 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 the RDF flag 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 in SCFRDR is still equal to or greater than the trigger number after a read, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after all receive data has been read.

The number of receive data bytes in SCFRDR can be found from bits 4 to 0 in SCFDR.

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the 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.

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 transfer 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 16.19.

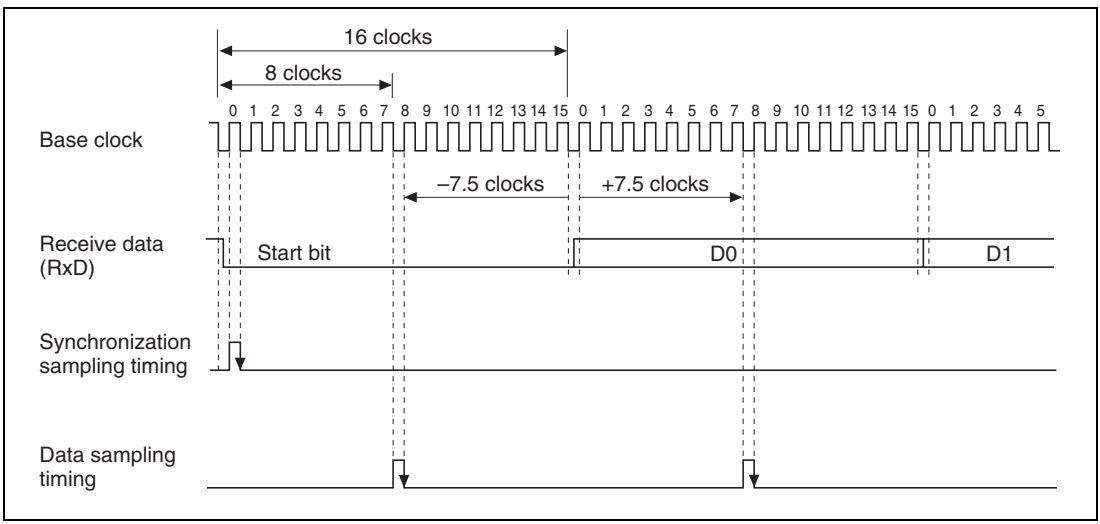


Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

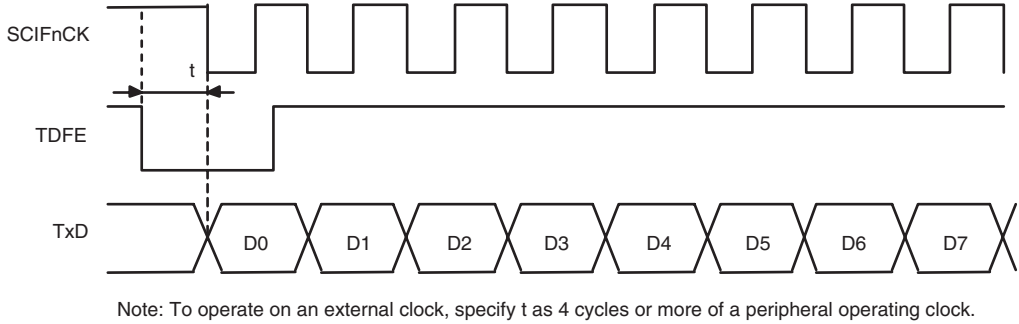
From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \quad \text{..... (2)}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Notes on DMAC Usage: To use an external clock source for a synchronous clock, the external clock should be input after SCFTDR has been updated by the DMAC and then five cycles or more of a peripheral operating clock has passed. If a transmit clock is input within four cycles after SCFTDR has updated, erroneous operation may occur (figure 16.20).



Note: To operate on an external clock, specify t as 4 cycles or more of a peripheral operating clock.

Figure 16.20 Sample Transfer of Synchronous Clock by DMAC

Section 17 Serial I/O with FIFO (SIOF)

This LSI includes a two-channel clocked synchronous serial I/O module with FIFO (SIOF) which can be directly connected to the audio CODEC. The functions of the SIOF0 and SIOF1 are common.

17.1 Features

The features of the SIOF are described below.

- Serial transfer
  Sixteen-stage 32-bit FIFOs (transmission/reception independently)
  Supports 8-bit data/16-bit data/16-bit stereo audio input/output
  MSB or LSB first for data transmission/reception
  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 every A-Law or μ-Law CODEC linear audio chip manufactured by any company
  Supports both master and slave modes
- Serial clock
  An external pin input or internal clock (P_CLK) can be selected as the clock source.
- Interrupts
  Following four interrupts can be requested independently.
  Transmission interrupt
  Reception interrupt
  Error interrupt
  Control interrupt
- DMA transfer
  Supports DMA transfer by a transfer request for transmission/reception

17.1.1 Block Diagram

Figure 17.1 shows a block diagram of the SIOF.

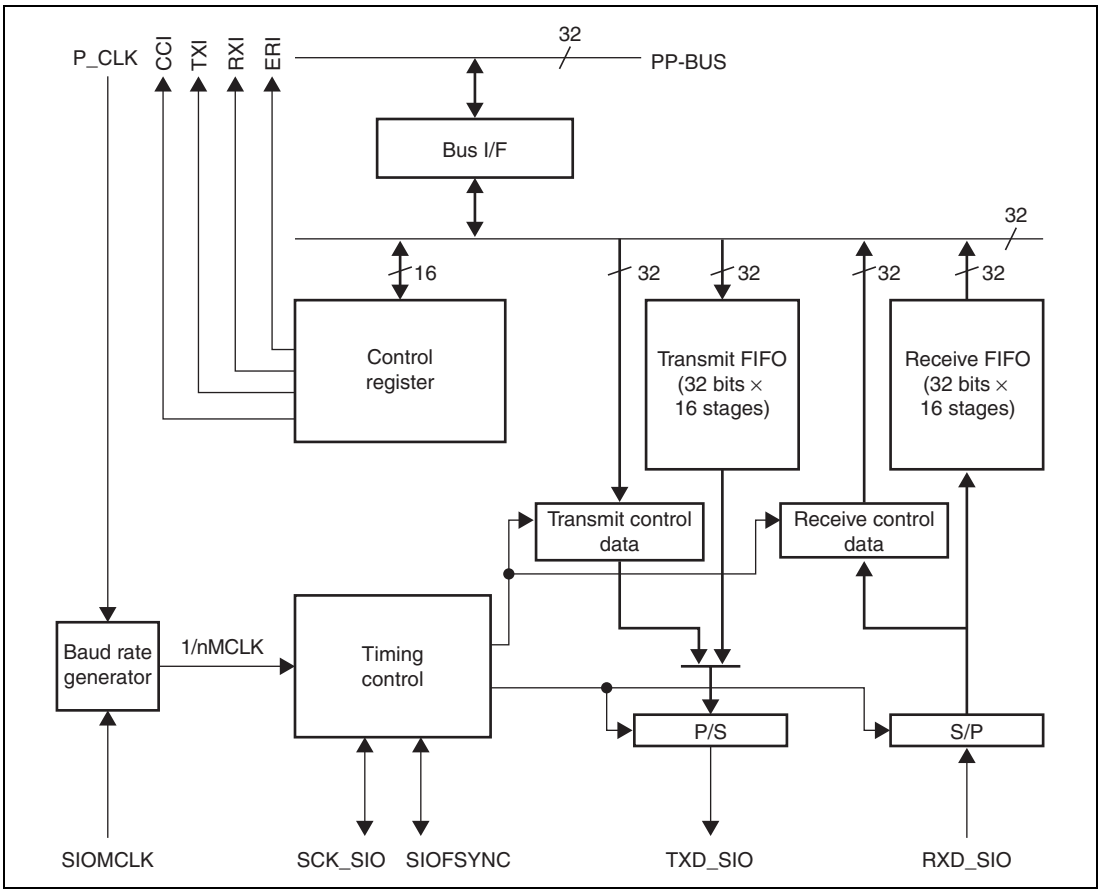


Figure 17.1 Block Diagram of SIOF

17.2 Input/Output Pins

The pin configuration of the SIOF0/1 is shown in table 17.1.

Table 17.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	Clock input pin	SIOMCLK0	Input	Master clock input
	Communication clock pin	SCK_SIO0	Input/ output	Serial clock (common to transmission/reception)
	Frame synchronous pin	SIOFSYNC0	Input/ output	Frame synchronous signal (common to transmission/reception)
	Transmit data pin	TXD_SIO0	Output	Transmit data
	Receive data pin	RXD_ SIO0	Input	Receive data
1	Clock input pin	SIOMCLK1	Input	Master clock input
	Communication clock pin	SCK_SIO1	Input/ output	Serial clock (common to transmission/reception)
	Frame synchronous pin	SIOFSYNC1	Input/ output	Frame synchronous signal (common to transmission/reception)
	Transmit data pin	TXD_SIO1	Output	Transmit data
	Receive data pin	RXD_SIO1	Input	Receive data

17.3 Register Descriptions

The SIOF has the following registers. For the addresses and access size of these registers, refer to section 24, List of Registers.

1. Channel 0

- SIOF mode register_0 (SIMDR_0)
- Serial clock select register_0 (SISCR_0)
- Serial transmit data assign register_0 (SITDAR_0)
- Serial receive data assign register_0 (SIRDAR_0)
- Serial control data assign register_0 (SICDAR_0)
- SIOF control register_0 (SICTR_0)
- SIOF FIFO control register_0 (SIFCTR_0)
- SIOF status register_0 (SISTR_0)
- SIOF interrupt enable register_0 (SIIER_0)
- Serial transmit data register_0 (SITDR_0)
- Serial receive data register_0 (SIRDR_0)
- Serial transmit control data register_0 (SITCR_0)
- Serial receive control data register_0 (SIRCR_0)

2. Channel 1

- SIOF mode register_1 (SIMDR_1)
- Serial clock select register_1 (SISCR_1)
- Serial transmit data assign register_1 (SITDAR_1)
- Serial receive data assign register_1 (SIRDAR_1)
- Serial control data assign register_1 (SICDAR_1)
- SIOF control register_1 (SICTR_1)
- SIOF FIFO control register_1 (SIFCTR_1)
- SIOF status register_1 (SISTR_1)
- SIOF interrupt enable register_1 (SIIER_1)
- Serial transmit data register_1 (SITDR_1)
- Serial receive data register_1 (SIRDR_1)
- Serial transmit control data register_1 (SITCR_1)
- Serial receive control data register_1 (SIRCR_1)

17.3.1 SIOF Mode Register (SIMDR)

SIMDR is a register that sets the SIOF0/1 operating mode. SIMDR is initialized by a power-on reset or manual reset.

Bit	Bit Name	Initial Value	R/W	Description
15	TRMD1	0	R/W	Transfer Mode
14	TRMD0	0	R/W	Selects transfer mode. 00: Slave mode 1 01: Slave mode 2 10: Master mode 1 11: Master mode 2 Note: For the operation in each mode, see section 17.4.3, Transfer Data Format.
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	REDG	0	R/W	Receive Data Sampling Edge The TXD_SIO signal is transmitted at the opposite edge where the RXD_SIO signal is sampled (see figure 17.4). 0: RXD_SIO is sampled at the falling edge of SCK_SIO 1: RXD_SIO is sampled at the rising edge of SCK_SIO Note: This bit is valid in master mode 1 and master mode 2.

Bit	Bit Name	Initial Value	R/W	Description
11	FL3	0	R/W	Frame Length
10	FL2	0	R/W	00xx: Slot length is 8 bits and frame length is 8 bits
9	FL1	0	R/W	0100: Slot length is 8 bits and frame length is 16 bits
8	FL0	0	R/W	0101: Slot length is 8 bits and frame length is 32 bits 0110: Slot length is 8 bits and frame length is 64 bits 0111: Slot length is 8 bits and frame length is 128 bits 10xx: Slot length is 16 bits and frame length is 16 bits 1100: Slot length is 16 bits and frame length is 32 bits 1101: Slot length is 16 bits and frame length is 64 bits 1110: Slot length is 16 bits and frame length is 128 bits 1111: Slot length is 16 bits and frame length is 256 bits Notes: 1. When slot length is specified as 8 bits, control data cannot be transmitted or received. 2. When LSB is first transmitted or received, control data cannot be transmitted or received. x: Don't care
7	TXDIZ	0	R/W	High-Impedance Output when Transmission is Invalid Specifies high-impedance output when transmission is invalid. 0: High output (1 output) when invalid 1: High-impedance output 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	LSBF	0	R/W	LSB-First Transmission/Reception Selects the bit order of a transmit/receive frame. 0: MSB-first 1: LSB-first
5	RCIM	0	R/W	Receive Control Data Interrupt Mode Selects the set timing of the RCRDY bit in SISTR. 0: Sets the RCRDY bit in SISTR when the contents of SIRCR change. 1: Sets the RCRDY bit in SISTR each time when SIRCR receives control data.

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

17.3.2 Serial Clock Select Register (SISCR)

SISCR is used to set the baud rate generator operation. SISCR can be specified when the TRMD1 and TRMD0 bits in SIMDR are specified as B′10 or B′11. SISCR is initialized by a power-on reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection The master clock is the clock input to the baud rate generator. 0: Uses the SIOMCLK pin input signal as the master clock 1: Uses PCLK as the master clock
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the baud rate generator output clock as the clock source 1: Uses the master clock itself as the clock source
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	BRPS4	0	R/W	Baud Rate Generator's Prescalar Setting (BRPS)
11	BRPS3	0	R/W	Set the master clock division ratio BRPS.
10	BRPS2	0	R/W	00000: (× 1/32)
9	BRPS1	0	R/W	00001: (× 1/1)
8	BRPS0	0	R/W	00010: (× 1/2) 11111: (× 1/31)
7 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	BRDV2	0	R/W	Baud Rate Generator's Division Ratio Setting (BRDV)
1	BRDV1	0	R/W	Set the frequency division ratio BRDV for the output stage of the baud rate generator. The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (maximum 1/1024).
0	BRDV0	0	R/W	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 Note: Other than above is reserved (setting prohibited).

17.3.3 Serial Transmit Data Assign Register (SITDAR)

SITDAR is used to specify the position of the transmit data in a frame (slot number). SITDAR is initialized by a power-on reset and software reset.

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	TDLA3	0	R/W	Transmit Left Channel Data Assigns
10	TDLA2	0	R/W	Specify the position of left-channel data in transmit frame as B′0000 to B′1110. Transmit data for the left channel is specified in bits SITDL15 to SITDL0 in SITDR.
9	TDLA1	0	R/W	
8	TDLA0	0	R/W	Note: If the TDLA3 to TDLA0 bits are set to B′1111, operation is not guaranteed.
7	TDRE	0	R/W	Transmit Right Channel Data Enable 0: Disables right channel data transmission 1: Enables right channel data transmission

Bit	Bit Name	Initial Value	R/W	Description
6	TLREP	0	R/W	Transmit Left Channel Repeat This bit setting is valid when the TDRE bit is set to 1. When this bit is set to 1, settings of bits SITDR15 to SITDR0 in SITDR are ignored. 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
5	—	0	R	Reserved
4	—	0	R	These bits are always read as 0. The write value should always be 0.
3	TDRA3	0	R/W	Transmit Right Channel Data Assigns
2	TDRA2	0	R/W	Specify the position of right-channel data in transmit frame as B′0000 to B′1110. Transmit data for the right channel is specified in bits SITDR15 to SITDR0 in SITDR. Note: If the TDRA3 to TDRA0 bits are set to B′1111, operation is not guaranteed.
1	TDRA1	0	R/W	
0	TDRA0	0	R/W	

17.3.4 Serial Receive Data Assign Register (SIRDAR)

SIRDAR is used to specify the position of the receive data in a frame. SIRDAR is initialized by a power-on reset or software reset.

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.

Bit	Bit Name	Initial Value	R/W	Description
11	RDLA3	0	R/W	Receive Left Channel Data Assigns 3 to 0
10	RDLA2	0	R/W	Specify the position of left-channel data in a receive frame as B′0000 to B′1110. Receive data for the left channel is stored in bits SIRDL15 to SIRDL0 in SIRDR.
9	RDLA1	0	R/W	
8	RDLA0	0	R/W	Note: If the RDLA3 to RDLA0 bits are set to B′1111, operation is not guaranteed.
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	RDRA3	0	R/W	Receive Right Channel Data Assigns 3 to 0
2	RDRA2	0	R/W	Specify the position of right-channel data in a receive frame as B′0000 to B′1110. Receive data for the right channel is stored in bits SIRDR15 to SIRDR0 in SIRDR.
1	RDRA1	0	R/W	
0	RDRA0	0	R/W	Note: If the RDRA3 to RDRA0 bits are set to B′1111, operation is not guaranteed.

17.3.5 Serial Control Data Assign Register (SICDAR)

SICDAR is used to specify the position of the control data in a frame. SICDAR can be specified only when the FL3 to FL0 bits in SIMDR are specified as 1xxx. SICDAR is initialized by a power-on reset or software reset.

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	CD0A3	0	R/W	Control Channel 0 Data Assigns 3 to 0
10	CD0A2	0	R/W	Specify the position of control channel 0 data in a receive or transmit frame as B′0000 to B′1110. Transmit data for the control channel 0 data is specified in bits SITC015 to SITC00 in SITCR. Receive data for the control channel 0 data is stored in bits SIRC015 to SIRC00 in SIRCR. Note: If the CD0A3 to CD0A0 bits are set to B′1111, operation is not guaranteed.
9	CD0A1	0	R/W	
8	CD0A0	0	R/W	
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	CD1A3	0	R/W	Control Channel 1 Data Assigns 3 to 0
2	CD1A2	0	R/W	Specify the position of control channel 1 data in a receive or transmit frame as B′0000 to B′1110. Transmit data for the control channel 1 data is specified in bits SITC115 to SITC10 in SITCR. Receive data for the control channel 1 data is stored in bits SIRC115 to SIRC10 in SIRCR. Note: If the CD1A3 to CD1A0 bits are set to B′1111, operation is not guaranteed.
1	CD1A1	0	R/W	
0	CD1A0	0	R/W	

17.3.6 SIOF Control Register (SICTR)

SICTR is used to set the SIOF operating state. SICTR is initialized by a power-on reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 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 in the baud rate generator to the SCK_SIO pin. 0: Disables the SCK_SIO output (outputs 0) 1: Enables the SCK_SIO output
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. If this bit is set to 1, the SIOF initializes the frame counter and initiates the operation. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	TXE	0	R/W	Transmission Enable This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal) and when valid data is stored in the transmit FIFO. When the 1 setting for this bit becomes valid, the SIOF issues a transmission 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 TXD_SIO pin begins. This bit is initialized by a transmit reset. 0: Disables data transmission from TXD_SIO (outputs 1) 1: Enables data transmission from TXD_SIO

Bit	Bit Name	Initial Value	R/W	Description
8	RXE	0	R/W	Reception Enable This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOFSYNC signal). When the 1 setting for this bit becomes valid, the SIOF begins the reception of data from the RXD_SIO pin. When receive data is stored in receive FIFO, the SIOF issues a reception transfer request according to the setting of the RFWM bit in SIFCTR. This bit is initialized by a receive reset. 0: Disables data reception from RXD_SIO 1: Enables data reception from RXD_SIO
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TXRST	0	R/W	Transmission Reset This bit setting becomes valid immediately. When the 1 setting for this bit becomes valid, the SIOF immediately sets transmit data from the TXD_SIO pin to 1, and initializes the transmission data register and transmission-related status register. The following are initialized. • SITDR • Transmit FIFO write/read pointer • TCRDY, TFEMP, and TDREQ bits in SISTR • TXE bit As the SIOF is cleared automatically at the completion of reset operation, this bit is always read as 0. 0: Transmission operation is not reset 1: Resets transmission operation

Bit	Bit Name	Initial Value	R/W	Description
0	RXRST	0	R/W	Reception Reset This bit setting becomes valid immediately. When the 1 setting for this bit becomes valid, the SIOF immediately disables reception from the RXD_SIO pin, and initializes the reception data register and reception-related status register. The following are initialized. • SIRDR • Receive FIFO write/read pointer • RCRDY, RFFUL, and RDREQ bits in SISTR • RXE bit As the SIOF is cleared automatically at the completion of reset operation, this bit is always read as 0. 0: Reception operation is not reset 1: Resets reception operation

17.3.7 SIOF FIFO Control Register (SIFCTR)

SIFCTR is used to indicate the area available for the transmit/receive FIFO transfer. SIFCTR is initialized by a power-on reset or software reset.

Bit	Bit Name	Initial Value	R/W	Description
15	TFWM2	0	R/W	Transmit FIFO Watermark
14	TFWM1	0	R/W	A transfer request to the transmit FIFO is issued by the TDREQ bit in SISTR. The transmit FIFO is always used as 16 stages of FIFO regardless of these bit settings.
13	TFWM0	0	R/W	
				000: Issue a transfer request when 16 stages of transmit FIFO are empty.
				001: Reserved (setting prohibited)
				010: Reserved (setting prohibited)
				011: Reserved (setting prohibited)
				100: Issue a transfer request when 12 or more stages of transmit FIFO are empty.
				101: Issue a transfer request when 8 or more stages of transmit FIFO are empty.
				110: Issue a transfer request when 4 or more stages of transmit FIFO are empty.
				111: Issue a transfer request when 1 or more stages of transmit FIFO are empty.
12	TFUA4	1	R	Transmit FIFO Usable Area
11	TFUA3	0	R	Indicate the number of words that can be transferred by the CPU or DMAC as B′00000 to B′10000.
10	TFUA2	0	R	
9	TFUA1	0	R	
8	TFUA0	0	R	

Bit	Bit Name	Initial Value	R/W	Description
7	RFWM2	0	R/W	Receive FIFO Watermark
6	RFWM1	0	R/W	A transfer request to the receive FIFO is issued by the RDREQ bit in SISTR. The receive FIFO is always used as 16 stages of FIFO regardless of these bit settings. 000: Issue a transfer request when 1 stage or more of receive FIFO are valid. 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: Reserved (setting prohibited) 100: Issue a transfer request when 4 or more stages of receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of receive FIFO are valid. 111: Issue a transfer request when 16 stages of receive FIFO are valid.
5	RFWM0	0	R/W	
4	RFUA4	0	R	Receive FIFO Usable Area
3	RFUA3	0	R	Indicate the number of words that can be transferred by the CPU or DMAC as B′00000 to B′10000.
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

17.3.8 SIOF Status Register (SISTR)

SISTR shows the SIOF state. Each of the bits of this register becomes an SIOF interrupt source when the corresponding bit in SIIER is set to 1. SISTR is initialized by a power-on reset or software reset.

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 This bit indicates a state of the SIOF. If SITCR is written, the SIOF clears this bit. This bit is valid when the TXE bit in SICTR is set to 1. If the issue of interrupts by this bit is enabled, the SIOF issues a control interrupt. 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 TXD_SIO pin. 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled Note: When using this bit, see 2 in section 17.5, Usage Notes.
13	TFEMP	0	R	Transmit FIFO Empty This bit indicates a state; if SITDR is written, the SIOF clears this bit. This bit is valid when the TXE bit in SICTR is 1. If the issue of interrupts by this bit is enabled, the SIOF issues a control interrupt. 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty
12	TDREQ	0	R	Transmit Data Transfer Request 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. This bit is valid when the TXE bit in SICTR is 1. This bit indicates a state of the SIOF. 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, the SIOF issues a transmit interrupt. 0: No transfer request 1: Transfer request

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	RCRDY	0	R	Receive Control Data Ready This bit indicates a state of the SIOF. If SIRCR is read, the SIOF clears this bit. This bit is valid when the RXE bit in SICTR is set to 1. If the issue of interrupts by this bit is enabled, the SIOF issues a control interrupt. If SIRCR is written when this bit is set to 1, SIRCR is modified by the latest data. 0: Indicates that SIRCR stores no valid data 1: Indicates that SIRCR stores valid data
9	RFFUL	0	R	Receive FIFO Full This bit indicates a state. If SIRDR is read, the SIOF clears this bit. This bit is valid when the RXE bit in SICTR is 1. If the issue of interrupts by this bit is enabled, the SIOF issues a control interrupt. 0: Receive FIFO not full 1: Receive FIFO full
8	RDREQ	0	R	Receive Data Transfer Request A receive data transfer request is issued when the valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. This bit is valid when the RXE bit in SICTR is 1. This bit indicates a state the SIOF. If the size of valid 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, the SIOF issues a receive interrupt. 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.
7 to 5	—	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
4	FSERR	0	R/W	Frame Synchronization Error 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. If the issue of interrupts by this bit is enabled, the SIOF issues an error interrupt. 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs
3	TFOVR	0	R/W	Transmit FIFO Overrun Transmit FIFO overrun means that there has been an attempt to write to SITDR when the transmit FIFO is full. When a transmit overrun occurs, written data is ignored. This bit is valid when the TXE 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, the SIOF issues an error interrupt. 0: No transmit FIFO overrun 1: Transmit FIFO overrun
2	TFUDR	0	R/W	Transmit FIFO Underrun Transmit FIFO underrun means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit underrun 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. If the issue of interrupts by this bit is enabled, the SIOF issues an error interrupt. 0: No transmit FIFO underrun 1: Transmit FIFO underrun

Bit	Bit Name	Initial Value	R/W	Description
1	RFUDR	0	R/W	Receive FIFO Underrun Receive FIFO underrun means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive underrun 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. If the issue of interrupts by this bit is enabled, the SIOF issues an error interrupt. 0: No receive FIFO underrun 1: Receive FIFO underrun
0	RFOVR	0	R/W	Receive FIFO Overrun Receive FIFO overrun means that writing has occurred when the receive FIFO is full. When a receive overrun occurs, the SIOF indicates the overrun, 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. If the issue of interrupts by this bit is enabled, the SIOF issues an error interrupt. 0: No receive FIFO overrun 1: Receive FIFO overrun

17.3.9 SIOF Interrupt Enable Register (SIIER)

SIIER is a used to enable the issue of SIOF interrupts. When each of the bits of this register is set to 1, and the corresponding bit of the SISTR is set to 1, the SIOF issues an interrupt. SIIER is initialized by a power-on reset or software reset.

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	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 (control interrupt)
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 (control interrupt)
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 (transmit interrupt)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
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 (control interrupt)
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 (control interrupt)

Bit	Bit Name	Initial Value	R/W	Description
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 (receive interrupt)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 (error interrupt)
3	TFOVRE	0	R/W	Transmit FIFO Overrun Enable 0: Disables interrupts due to transmit FIFO overrun 1: Enables interrupts due to transmit FIFO overrun (error interrupt)
2	TFUDRE	0	R/W	Transmit FIFO Underrun Enable 0: Disables interrupts due to transmit FIFO underrun 1: Enables interrupts due to transmit FIFO underrun (error interrupt)
1	RFUDRE	0	R/W	Receive FIFO Underrun Enable 0: Disables interrupts due to receive FIFO underrun 1: Enables interrupts due to receive FIFO underrun (error interrupt)
0	RFOVRE	0	R/W	Receive FIFO Overrun Enable 0: Disables interrupts due to receive FIFO overrun 1: Enables interrupts due to receive FIFO overrun (error interrupt)

17.3.10 Serial Transmit Data Register (SITDR)

SITDR is used to specify the SIOF transmit data. The setting data for this register is stored in the transmit FIFO. SITDR is initialized by a power-on reset, software reset, or transmit reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL15 to SITDL0	All 0	W	Left Channel Transmit Data Specify data to be output from the TXD_SIO pin as left channel data. The position of the left channel data in the transmission 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	SITDR15 to SITDR0	All 0	W	Right Channel Transmit Data Specify data to be output from the TXD_SIO pin as right channel data. The position of the right channel data in the transmission frame is specified by the TDRA bit in SITDAR. These bits are valid only when the TDLE bit and TLREP bit in SITDAR are set to 1 and cleared to 0, respectively.

17.3.11 Serial Receive Data Register (SIRDR)

SIRDR is used to read receive data of the SIOF. SIRDR stores data in the receive FIFO. SIRDR is initialized by a power-on reset, software reset, or receive reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL15 to SIRDL0	All 0	R	Left Channel Receive Data Store data received from the RXD_SIO pin as left channel data. The position of the left channel data in a 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	SIRDR15 to SIRDR0	All 0	R	Right Channel Receive Data Store data received from the RXD_SIO pin as right channel data. The position of the right channel data in the reception frame is specified by the RDRA bit in SIRDAR. These bits are valid only when the RDRE bit in SIRDAR is set to 1.

17.3.12 Serial Transmit Control Data Register (SITCR)

SITCR is used to specify the SIOF transmit control data. SITCR can be specified only when the FL3 to EL0 bits in SIMDR are specified as 1xxx. SITCR is initialized by a power-on reset, software reset, or transmit reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC015 to SITC00	All 0	W	Control Channel 0 Transmit Data Specify data to be output from the TXD_SIO pin as control channel 0 transmit data. The position of the control channel 0 data in the transmission or reception 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	SITC115 to SITC10	All 0	W	Control Channel 1 Transmit Data Specify data to be output from the TXD_SIO pin as control channel 1 transmit data. The position of the control channel 1 data in the transmission or reception frame is specified by the CD1A bit in SICDAR. These bits are valid only when the CD1E bit in SICDAR is set to 1.

17.3.13 Serial Receive Control Data Register (SIRCR)

SIRCR is used to store the SIOF receive control data. SIRCR can be specified only when the FL3 to FL0 bits in SIMDR are specified as 1xxx. SIRCR is initialized by a power-on reset, software reset, or receive reset.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRC015 to SIRC00	All 0	R	Control Channel 0 Receive Data Store data received from the RXD_SIO pin as control channel 0 receive data. The position of the control channel 0 data in the transmission or reception 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	SIRC115 to SIRC10	All 0	R	Control Channel 1 Receive Data Store data received from the RXD_SIO pin as control channel 1 receive data. The position of the control channel 1 data in the transmission or reception frame is specified by the CD1A bit in SICDAR. These bits are valid only when the CD1E bit in SICDAR is set to 1.

17.4 Operation

17.4.1 Serial Clocks

Master/Slave Modes: The following modes are available as the SIOF clock mode.

- Slave mode: SCK_SIO*, SIOFSYNC input
- Master mode: SCK_SIO*, SIOFSYNC output

Note: * In master mode, the SCK_SIO signal is kept output even if there is data or not.

Baud Rate Generator (BRG): In SIOF master mode, the baud rate generator (BRG) is used to generate the serial clock. The division ratio is from 1/2 to 1/1024.

Figure 17.2 shows connections for supply of the serial clock.

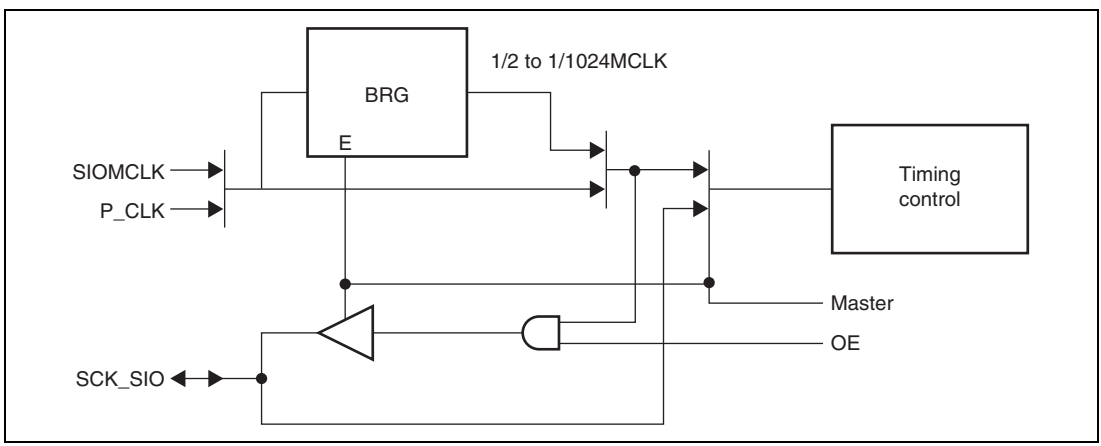


Figure 17.2 Serial Clock Supply

Table 17.2 shows an example of serial clock frequency.

Table 17.2 SIOF Serial Clock Frequency

	Sampling Rate			
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**	**96 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz	3.072 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz	—
128 bits	1.024 MHz	5.6648 MHz	6.144 MHz	—
256 bits	2.048 MHz	11.2896 MHz	12.288 MHz	—

17.4.2 Serial Timing

SIOFSYNC: The SIOFSYNC 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 17.3 shows the SIOFSYNC synchronization timing. The timing of master mode 1, slave mode 1, and slave mode 2 is shown in (a) in figure 17.3. The timing of master mode 2 is shown in (b) in figure 17.3.

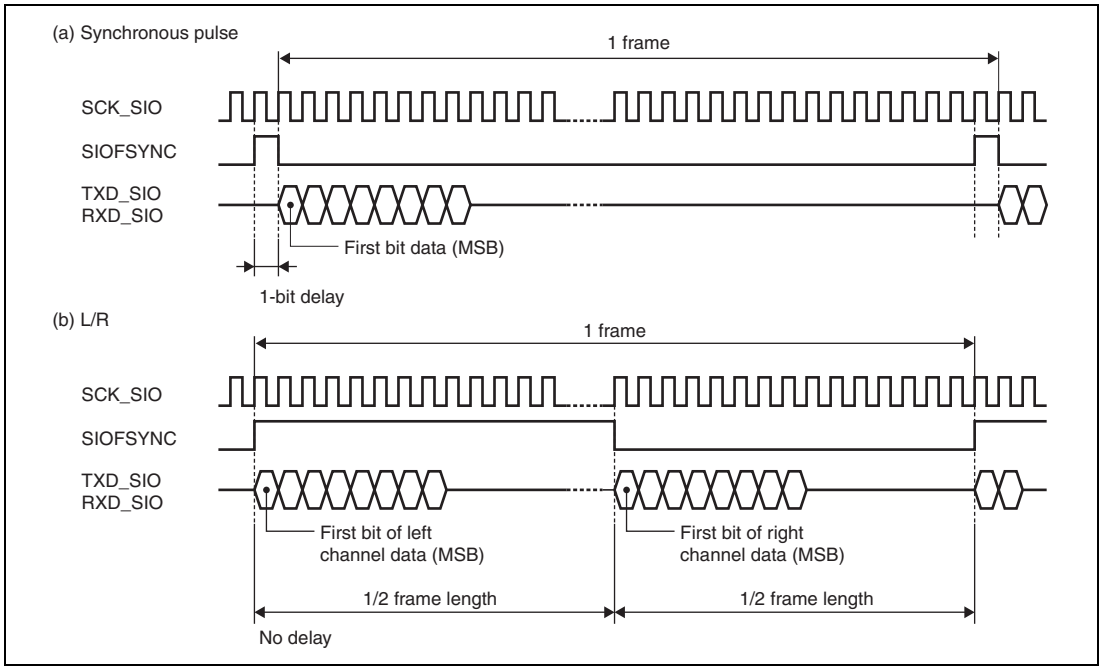


Figure 17.3 Serial Data Synchronization Timing

Transmit/Receive Timing: The TXD_SIO transmission timing and RXD_SIO reception timing relative to the SCK_SIO signal can be set as the sampling timing in the following two ways. The transmit/receive timing is set using the REDG bit in SIMDR. In slave mode 1 and slave mode 2, only falling-edge sampling is available.

- Falling-edge sampling
- Rising-edge sampling

Figure 17.4 shows the transmit/receive timing.

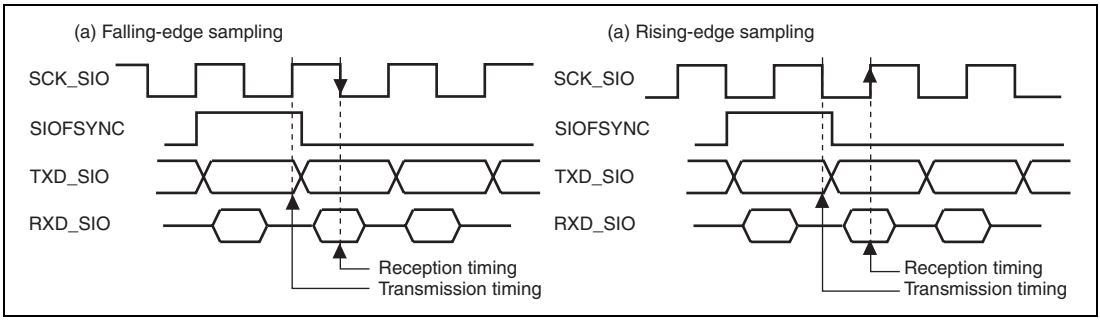


Figure 17.4 SIOF Transmit/Receive Timing

17.4.3 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 17.3. The transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 17.3 Serial Transfer Modes

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

Frame Length: The length of the frame to be transferred by the SIOF is specified by the FL3 to FL0 bits in SIMDR. Table 17.4 shows the relationship between the settings of the FL3 to FL0 bits and frame length.

Table 17.4 Frame Length

FL3 to FL0	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

[Legend]
x: Don't care.

Slot Position: The SIOF can specify the position of transmit data, receive data, and control data (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 slot leugth is valid for control register. In addition, control data is always assigned to the same slot number both in transmission and reception.

17.4.4 Register Allocation of Transfer Data

Transmit/Receive Data: Writing and reading of transmit or receive data are performed for the following registers.

- Transmit data writing: SITDR (32-bit access)
- Receive data reading: SIRDR (32-bit access)

Figure 17.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

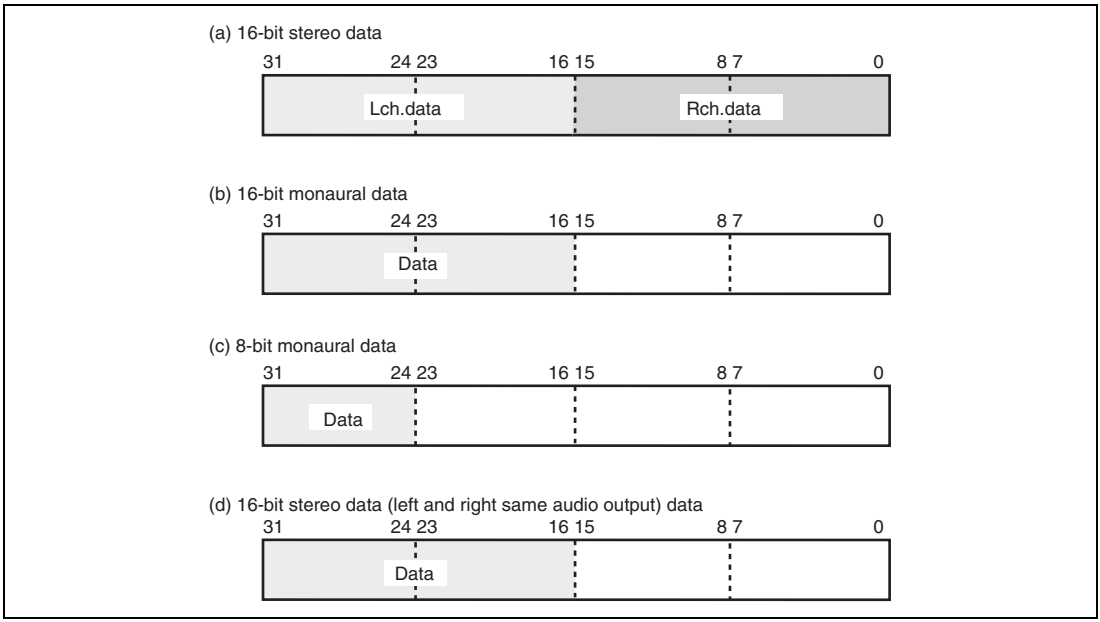


Figure 17.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid 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 the transmit data, specify the TLREP bit in SITDAR. Tables 17.5 and 17.6 show the audio mode specification for transmit data and that for receive data, respectively. To execute 8-bit monaural transmission or reception, use the left channel.

Table 17.5 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 17.6 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.

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 17.6 shows the control data and bit alignment in SITCR and SIRCR.

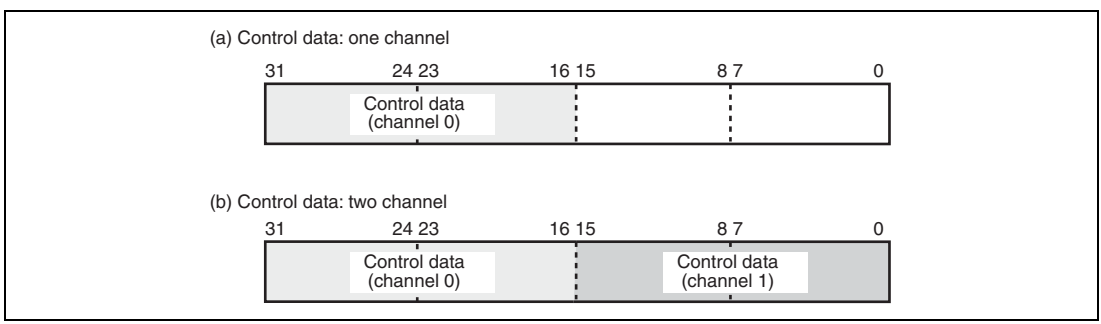


Figure 17.6 Control Data Bit Alignment

The number of channels in control data is specified by CD0E and CD1E bits in SICDAR. Table 17.7 shows the relationship between the number of channels in control data and bit settings. To use only one channel in control data, use channel 0.

Table 17.7 Setting for Number of Control Data Channels

	Bit	
Number of Channels	**CD0E**	**CD1E**
1	1	0
2	1	1

17.4.5 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 slot length is specified as 16 bits and MSB-first transmission/reception is selected.

Control by Slot Position (Master 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 17.7 shows an example of control data interface timing by slot position control.

Note: When using this control method, use PCLK as the master clock (master clock selection (MSSEL) = 1).

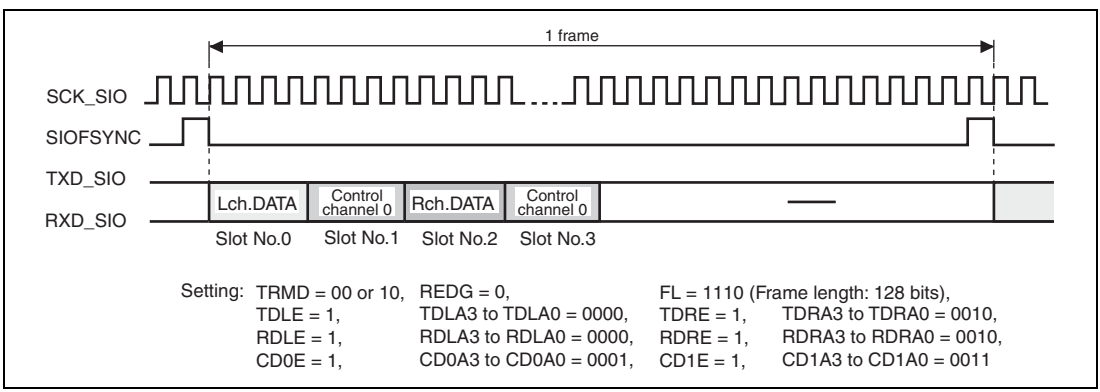


Figure 17.7 Control Data Interface (Slot Position)

Control by Secondary FS (Slave Mode 2): The CODEC normally outputs SIOFSYNC 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 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 control data (data specified by SITCR) or receives control data (stores in SIRCR) synchronously with the secondary FT.

Figure 17.8 shows an example of control data interface timing by secondary FS.

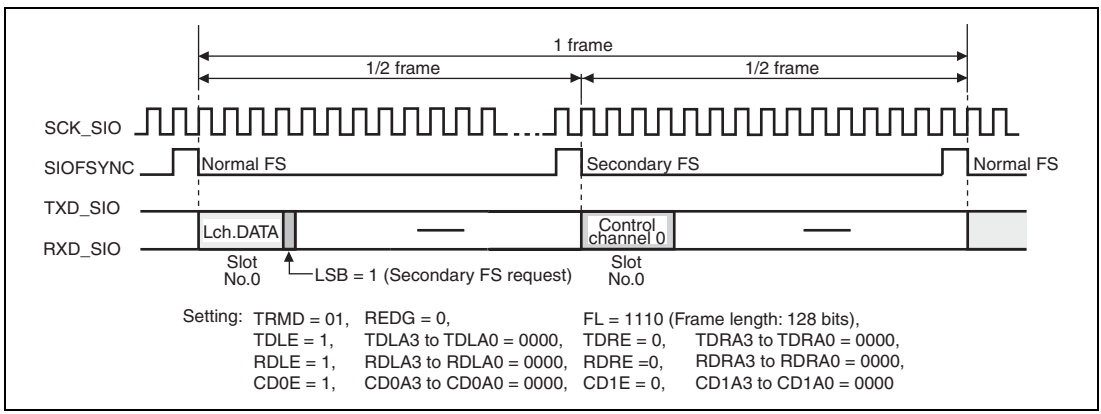


Figure 17.8 Control Data Interface (Secondary FS)

17.4.6 FIFO

Overview: The transmit and receive FIFOs of the SIOF have the following features.

- Sixteen-stage 32-bit FIFOs for transmission and reception
- The FIFO pointer can be modified 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.)
- Regardless of access size, the number of access cycles is always two cycles of the P-bus cycle.

Transfer Request: The SIOF indicates a transfer request of the FIFO in the following two bits of SISTR.

- 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 TFWM2 to TFWM0 bits and RFWM2 to RFWM0 bits in SIFCTR, respectively. Tables 17.8 and 17.9 summarize the conditions specified by SIFCTR.

Table 17.8 Conditions to Issue Transmit Request

TFWM2 to TFWM0	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 17.9 Conditions to Issue Receive Request

RFWM2 to RFWM0	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 above stage number. Accordingly, an overrun error or underrun error occurs if data area or empty area exceeds sixteen FIFO stages. FIFO transmission or reception request is cancelled 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 are indicated by the TFUA4 to TFUA0 bits in SIFCTR.

- Receive FIFO: The number of valid data stages that can be transferred by the CPU or DMAC are indicated by the RFUA4 to RFUA0 bits in SIFCTR.

The above contents show the number of data which CPU or DMAC can transfer.

17.4.7 Transmission and Reception Procedures

Transmission in Master Mode: Figure 17.9 shows an example of settings and operation for master mode transmission.

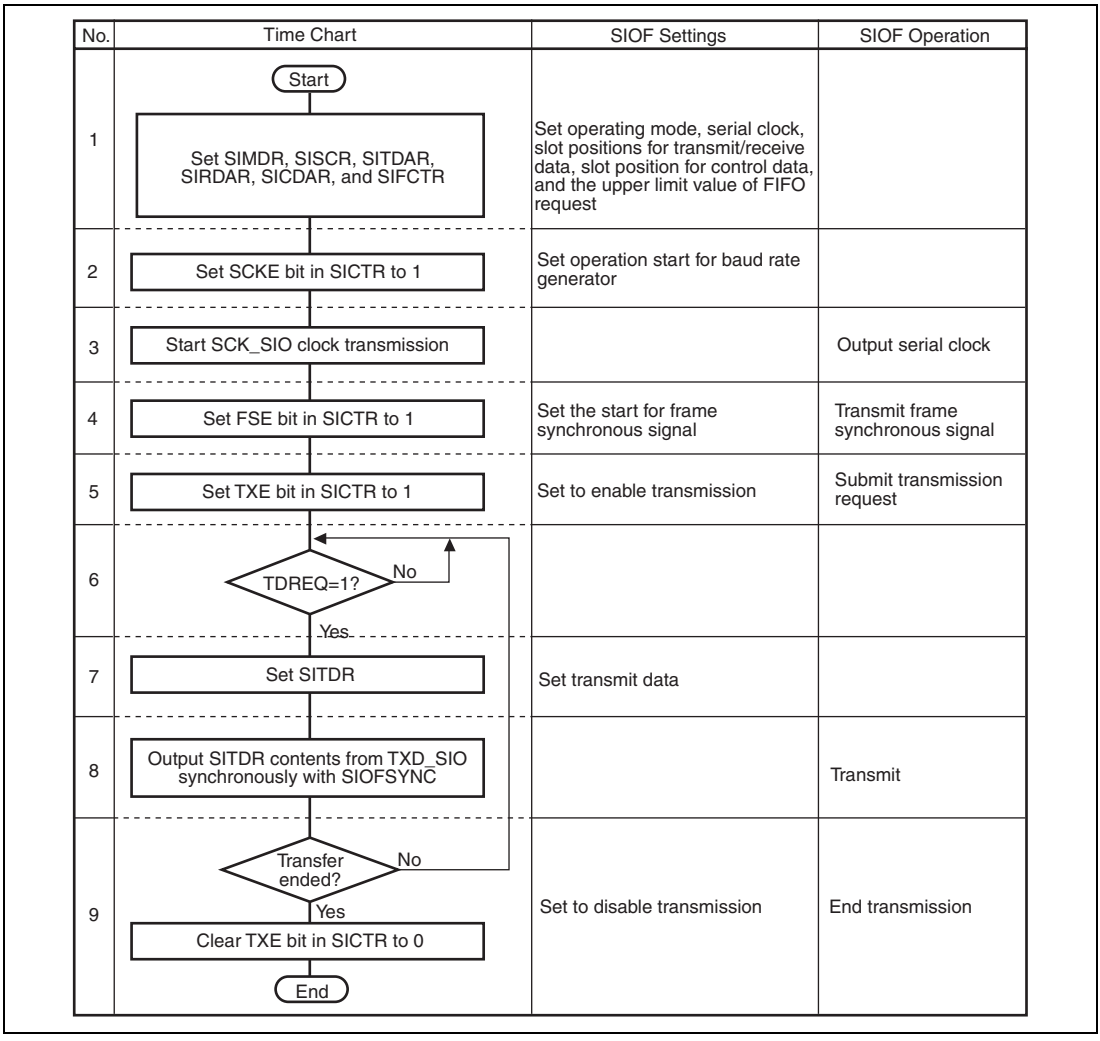


Figure 17.9 Example of Transmission Operation in Master Mode

Reception in Master Mode: Figure 17.10 shows an example of settings and operation for master mode reception.

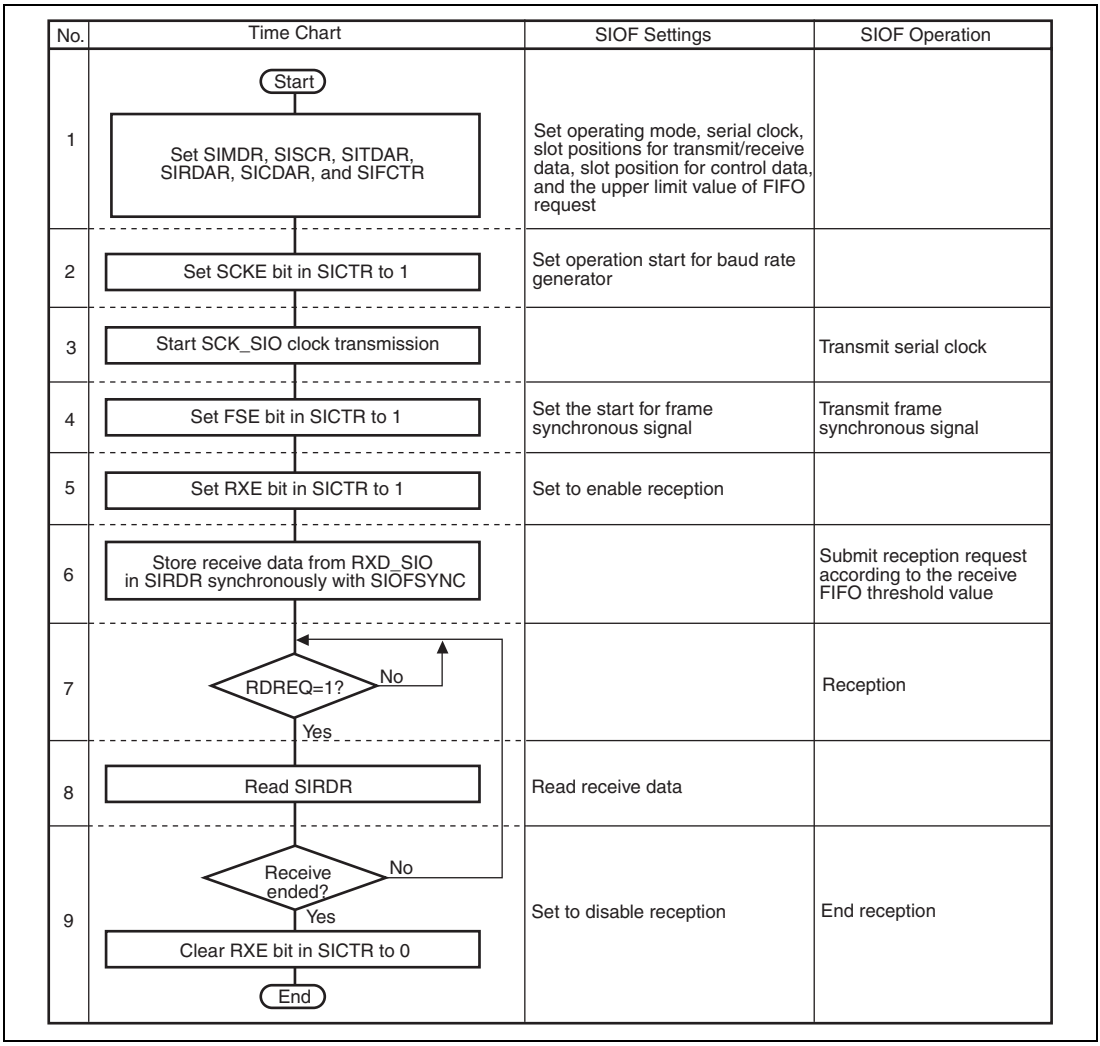


Figure 17.10 Example of Reception Operation in Master Mode

Transmission in Slave Mode: Figure 17.11 shows an example of settings and operation for slave mode transmission.

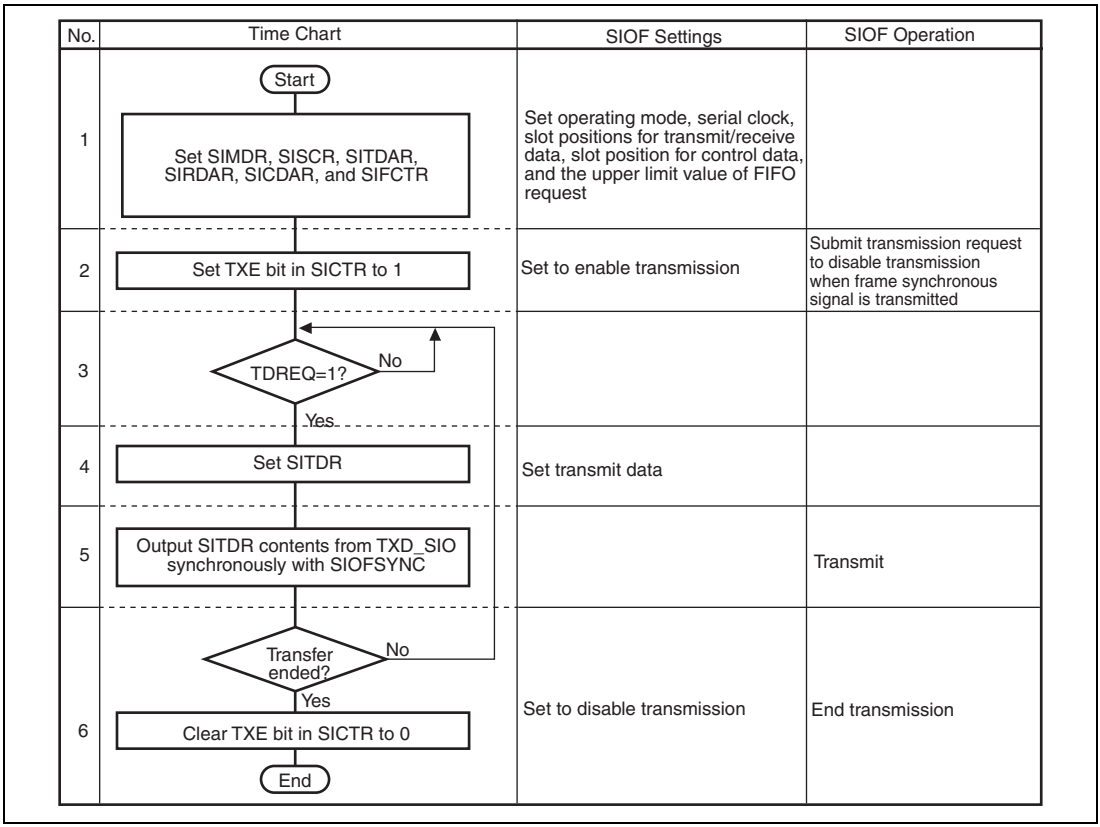


Figure 17.11 Example of Transmission Operation in Slave Mode

Reception in Slave Mode: Figure 17.12 shows an example of settings and operation for slave mode reception.

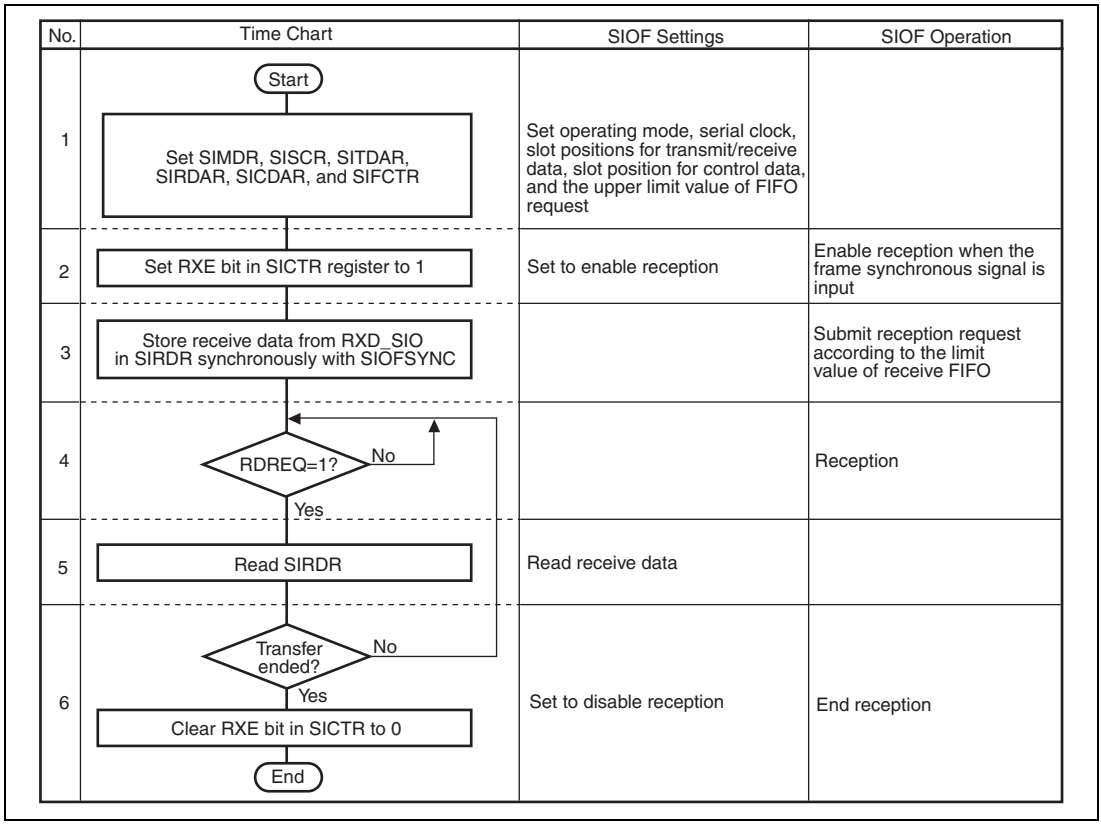


Figure 17.12 Example of Reception Operation in Slave Mode

Transmission/Reception Reset: The SIOF can separately reset the transmission and reception units by setting the following bits to 1.

- Transmission reset: TXRST bit in SICTR
- Reception reset: RXRST bit in SICTR

Table 17.10 shows the details of initialization upon transmission or reception reset.

Table 17.10 Transmission and Reception Reset

Type	Objects Initialized
Transmission reset	SITDR
	Transmit FIFO write pointer, transmit FIFO read pointer
	TCRDY bit, TFEMP bit, TDREQ bit in SISTR
	TXE bit in SICTR
Reception reset	SIRDR
	Receive FIFO write pointer, receive FIFO read pointer
	RCRDY bit, RFFUL bit, RDREQ bit in SISTR
	RXE bit in SICTR

Module Stop: In the module stop state, the SIOF stops transmit/receive operation with contents of all registers retained. If transmit/receive operation is not performed immediately after the module stop state is cleared, issue a transmit/receive reset.

17.4.8 Interrupts

The SIOF has four types of interrupts listed below. This classification is reflected to the IRR7 (SIOF0) and IRR8 (SIOF1) of the interrupt controller (INTC).

- Transmit interrupt (TXI)
- Receive interrupt (RXI)
- Control interrupt (CCI)
- Error interrupt (ERI)

Interrupt Sources: Interrupts can each be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 17.11 lists the SIOF interrupt sources.

Table 17.11 SIOF Interrupt Sources

No.	Classification	Bit Name	Function name	Description
1	Transmission (TXI)	TDREQ	Transmit data transfer request	The number of transmit FIFO data is equal to or less than the specified value by transmit operation.
2	Reception (RXI)	RDREQ	Receive data transfer request	The receive FIFO stores data of specified value or more.
3	Control (CCI)	TCRDY	Transmit control data ready	The transmit control data register is ready to be written.
4		RCRDY	Receive control data ready	The receive control data register stores valid data.
5		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
6		RFFUL	Receive FIFO full	The receive FIFO is full.
7	Error (ERI)	TFUDR	Transmit FIFO underrun	Serial data transmission timing has arrived while the transmit FIFO is empty.
8		TFOVR	Transmit FIFO overrun	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVR	Receive FIFO overrun	Serial data is received while the receive FIFO is full.
10		RFUDR	Receive FIFO underrun	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	Frame synchronization error	A synchronous signal is input before the specified bit time has been passed (in slave mode).

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, the SIOF issues each interrupt.

Transmit/Receive Interrupt Flag: Transmit and receive interrupts are sent to the INTC or DMAC by this interrupt flag based on the values of bits TDREQ and RDREQ in SISTR. Table17.12 shows the setting condition of the transmit/receive interrupt flag.

Table 17.12 Setting Condition of Transmit/Receive Interrupt Flag

	Setting Condition	Reset Condition
Transmit interrupt flag	TDREQ bit in SISTR is set to 1	TDREQ bit in SISTR is cleared to 0 Acknowledge from DMAC
Receive interrupt flag	RDREQ bit in SISTR is set to 1	RDREQ bit in SISTR is cleared to 0 Acknowledge from DMAC

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 underrun (TFUDR)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overrun (TFOVR)
  The contents of the transmit FIFO are protected, and the write operation causing the overrun is ignored.
- Receive FIFO overrun (RFOVR)
  Data causing the overrun is discarded and lost.
- Receive FIFO underrun (RFUDR)
  The latest read data is output on the bus (undefined value as specification).
- Frame synchronization error (FSERR)
  The internal counter is reset according to the FSYN signal in which an error occurs.

17.4.9 Transmission and Reception Timing

Examples of the SIOF serial transmission and reception are shown in figure 17.13 through figure 17.19.

8-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, frame length = 8 bits

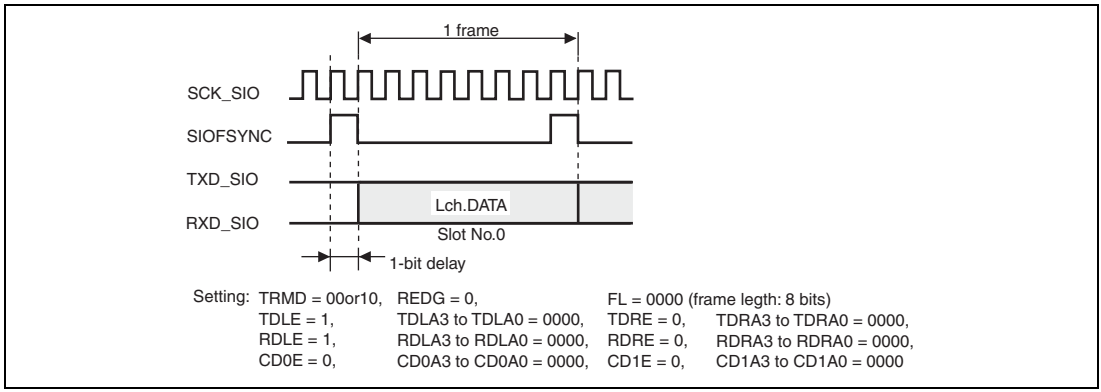


Figure 17.13 Transmission and Reception Timings (8-Bit Monaural Data (1))

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

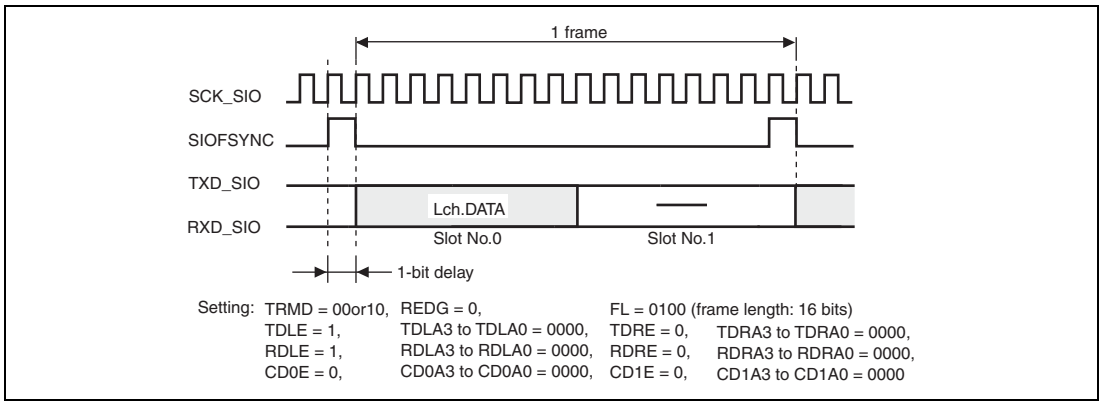


Figure 17.14 Transmission and Reception Timings (8-Bit Monaural Data (2))

16-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, frame length = 64 bits

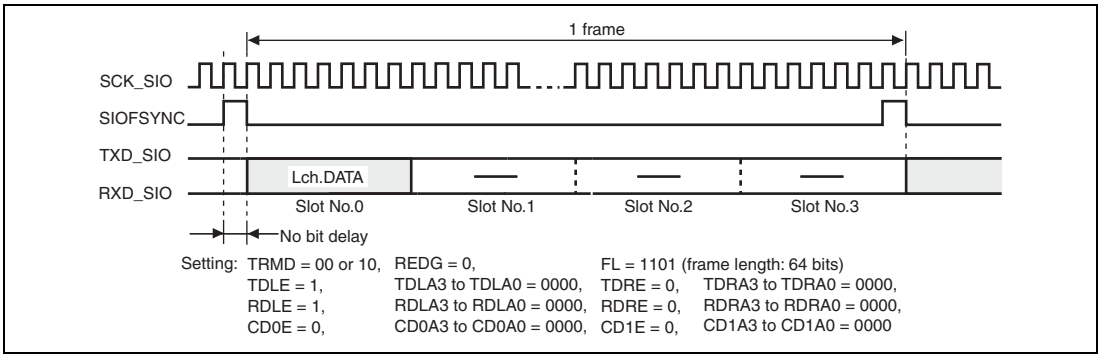


Figure 17.15 Transmission and Reception Timings (16-Bit Monaural Data (1))

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, frame length = 32 bits

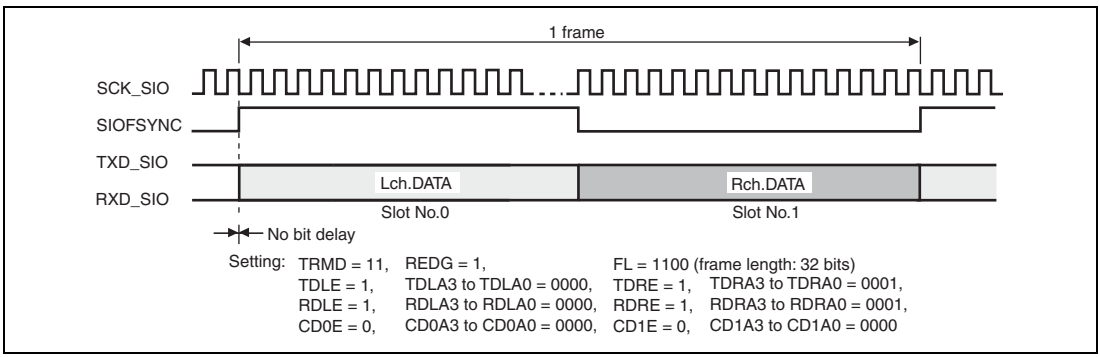


Figure 17.16 Transmission and Reception Timings (16-Bit Stereo Data (1))

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, frame length = 64 bits

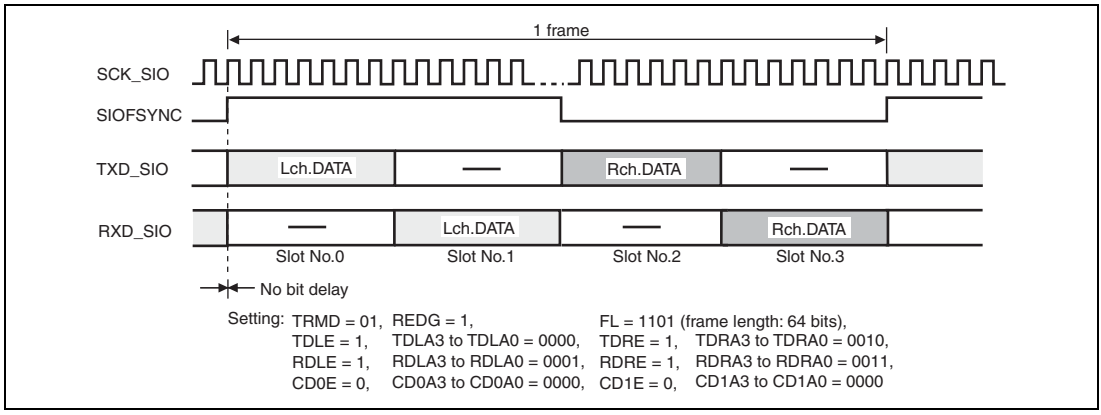


Figure 17.17 Transmission and Reception Timings (16-Bit Stereo Data (2))

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

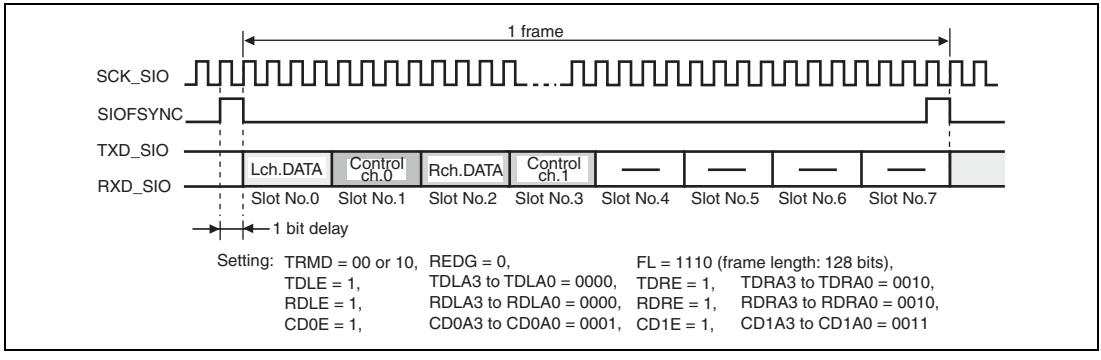


Figure 17.18 Transmission and Reception Timings (16-Bit Stereo Data (3))

16-bit Monaural Data (2): Synchronous pulse method, falling edge sampling, request for secondary FS, slot No.0 used for left channel data, slot No.0 used for control channel 0 data, frame length = 128 bits

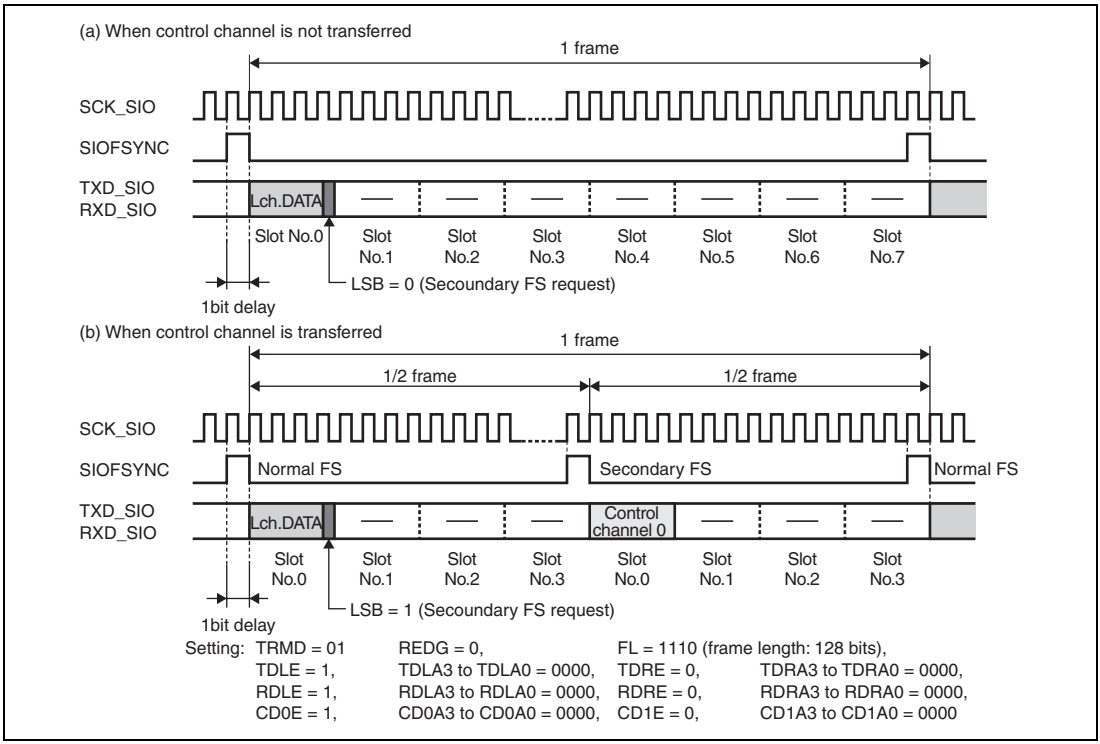


Figure 17.19 Transmission and Reception Timings (16-Bit Monaural Data (2))

17.5 Usage Notes

Note the following when using the SIOF.

1. Using the transmit function in slave mode

   If transmission is enabled when data has already been written to the transmit FIFO, one or two of the first data bytes may be lost.

   Therefore, data should not be written to the transmit FIFO before enabling transmission.
2. Using control data transmission/reception consecutively on control data interface (secondary FS position)

   The TCRDY value may become 1 before transmit control data is sent, and if the next control data is written to the control data register at this point, the control data waiting to be sent will be overwritten and erased.

   At this time, also, the control sequence is disrupted and the SIOF switches around the primary FS and secondary FS, with the result that transmission/reception of data and control data can no longer be performed normally.

   The control data register should therefore be written to after transmit control data has been sent.

   Example:

   Check RCRDY, and write to the control data register when RCRDY is 1.

   After transmit control data has been written to, it is essential to read the receive control register (SIRCR) and clear RCRDY.
3. DMA transfer

   Do not use 16-byte DMA transfer. (See section 13.4.4, DMA Transfer Types.)
4. Access from the CPU

   When performing access from the CPU, do not access the SIOF's transmit/receive FIFO consecutively, but instead insert an access to somewhere else between SIOF transmit/receive FIFO accesses.
5. Transmit/receive FIFO underflow

   If the transmit/receive FIFO underflows during a transmit/receive operation, control of the SIOF's transmit/receive FIFO may fail and data may be lost.

   To prevent this, either set a watermark so that an underflow does not occur, or execute a transmit reset (TXRST) or receive reset (RXRST) when an empty interrupt is generated.
6. Transmit/receive reset execution

   When using the SIOF again after a transmit/receive operation ends, or after erroneous operation occurs, first execute a transmit reset (TXRST) or receive reset (RXRST).

Section 18 Ethernet Controller (EtherC)

This LSI has an on-chip Ethernet controller (EtherC) conforming to the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the Ethernet controller (EtherC) to perform transmission and reception of Ethernet/IEEE802.3 frames. The LSI has two MAC layer interface ports (hereafter referred to as port 0 and port 1), both of which can be made to perform transmission and reception independently. This Ethernet controller also has an on-chip TSU (Transfer Switching Unit) which controls transferring, allowing mutual transfer of data between MAC layer controllers of ports 0 and 1. This TSU has a 32-entry CAM (Content Addressable Memory) and two external CAM interface input pins for determining whether to receive or transfer packets input to both Ethernet controllers. The TSU also has a total 6-kbyte transfer FIFO for retaining packets to be transferred, allowing allocation of transfer FIFO capacity to be set freely for the transfer conditions of port 0 to 1 and port 1 to 0. The Ethernet controller is connected to the Ethernet Direct Memory Access Controller (E-DMAC) for Ethernet controller inside the LSI, and carries out high-speed data transfer to and from the memory.

Figure 18.1 shows a configuration of the EtherC.

18.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps receive/transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard MII (Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Ethernet frame relay function by the TSU
- Qtag addition and deletion functions conforming to IEEE802.1Q specifications (when frame relay is performed by the TSU)
- MAC address filtering function by the multicast (group) address
- Ethernet frame receive and transfer control functions by the CAM (Content Addressable Memory) interface signals input externally

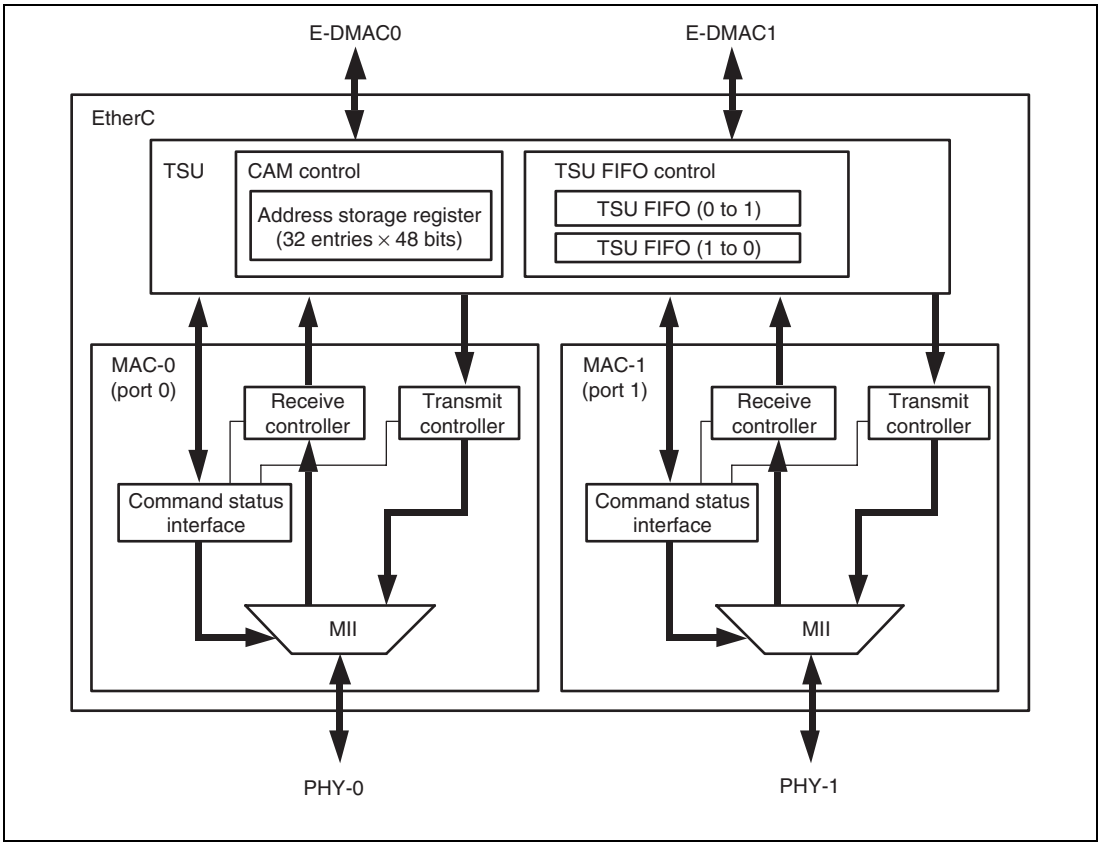


Figure 18.1 Configuration of EtherC

18.2 Input/Output Pins

Table 18.1 lists the pin configuration of the EtherC.

Table 18.1 Pin Configuration

Name	Port	Abbreviation	I/O	Function
Transmit clock	0	TX-CLK0*[1]	I	TX-EN, ETXD3 to ETXD0, TX-ER timing reference signal
Receive clock	0	RX-CLK0*[1]	I	RX-DV, ERXD3 to ERXD0, RX-ER timing reference signal
Transmit enable	0	TX-EN0*[1]	O	Indicates that transmit data is ready on ETXD3 to ETXD0
Transmit data	0	ETXD03 to ETXD00*[1]	O	4-bit transmit data
Transmit error	0	TX-ER0*[1]	O	Notifies PHY_LSI of error during transmission
Receive data valid	0	RX-DV0*[1]	I	Indicates that valid receive data is on ERXD3 to ERXD0
Receive data	0	ERXD03 to ERXD00*[1]	I	4-bit receive data
Receive error	0	RX-ER0*[1]	I	Identifies error state occurred during data reception
Carrier detection	0	CRS0*[1]	I	Carrier detection signal
Collision detection	0	COL0*[1]	I	Collision detection signal
Management data clock	0	MDC0*[1]	O	Reference clock signal for information transfer via MDIO
Management data I/O	0	MDIO0*[1]	I/O	Bidirectional signal for exchange of management information between this LSI and PHY
Link status	0	LNKSTA0	I	Inputs link status from PHY
General-purpose external output	0	EXOUT0	O	Signal indicating value of register-bit (ECMR0-ELB)
Wake-On-LAN	0	WOL0	O	Signal indicating reception of Magic Packet
Transmit clock	1	TX-CLK1*[1]	I	TX-EN, ETXD3 to ETXD0, TX-ER timing reference signal
Receive clock	1	RX-CLK1*[1]	I	RX-DV, ERXD3 to ERXD0, RX-ER timing reference signal

Name	Port	Abbreviation	I/O	Function
Transmit enable	1	TX-EN1*[1]	O	Indicates that transmit data is ready on ETXD3 to ETXD0
Transmit data	1	ETXD13 to ETXD10*[1]	O	4-bit transmit data
Transmit error	1	TX-ER1*[1]	O	Notifies PHY-LSI of error during transmission
Receive data valid	1	RX-DV1*[1]	I	Indicates that valid receive data is on ERXD3 to ERXD0
Receive data	1	ERXD13 to ERXD10*[1]	I	4-bit receive data
Receive error	1	RX-ER1*[1]	I	Identifies error state occurred during data reception
Carrier detection	1	CRS1*[1]	I	Carrier detection signal
Collision detection	1	COL1*[1]	I	Collision detection signal
Management data clock	1	MDC1*[1]	O	Reference clock signal for information transfer via MDIO
Management data I/O	1	MDIO1*[1]	I/O	Bidirectional signal for exchange of management information between this LSI and PHY
Link status	1	LKNSTA1	I	Inputs link status from PHY
General-purpose external output	1	EXOUT1	O	Signal indicating value of register-bit (ECMR1-ELB)
Wake-On-LAN	1	WOL1	O	Signal indicating reception of Magic Packet
CAM input 0	—	CAMSEN0*[2]	I	CAM interface signal input 0
CAM input 1	—	CAMSEN1*[2]	I	CAM interface signal input 1
Bus release request	—	$\overline{\text{ARBUSY}}$*[3]	O	Signal indicating bus release request when the threshold value set for the data volume in the receive FIFO has been exceeded

Notes: 1. MII signal conforming to IEEE802.3u

2. The CAM input signal function is set by the CAMSEL03 to CAMSEL00 and CAMSEL13 to CAMSEL10 in the TSU_FWSLC register.
3. Refer to section 19, Ethernet Controller Direct Memory Access Controller (E-DMAC) and section 19.2.18, Overflow Alert FIFO Threshold Register (FCFTR).

18.3 Register Descriptions

The EtherC has the following registers. The last number of the abbreviation of the MAC layer interface control register corresponds to the number of the two MAC layer interfaces (MAC-0 or MAC-1). Some numbers have been omitted in the text. For details on addresses and access sizes of registers, see section 24, List of Registers.

Reset Register:

- Software reset register (ARSTR)

MAC Layer Interface Control Registers:

Port 0

- EtherC mode register (ECMR0)
- EtherC status register (ECSR0)
- EtherC interrupt permission register (ECSIPR0)
- PHY interface register (PIR0)
- MAC address high register (MAHR0)
- MAC address low register (MALR0)
- Receive frame length register (RFLR0)
- PHY status register (PSR0)
- Transmit retry over counter register (TROCR0)
- Delayed collision detect counter register (CDCR0)
- Lost carrier counter register (LCCR0)
- Carrier not detect counter register (CNDCR0)
- CRC error frame receive counter register (CEFCR0)
- Frame receive error counter register (FRECR0)
- Too-short frame receive counter register (TSFRCR0)
- Too-long frame receive counter register (TLFRCR0)
- Residual-bit frame receive counter register (RFCR0)
- Multicast address frame receive counter register (MAFCR0)
- IPG register (IPGR0)

Port 1

- EtherC mode register (ECMR1)
- EtherC status register (ECSR1)
- EtherC interrupt permission register (ECSIPR1)
- PHY interface register (PIR1)
- MAC address high register (MAHR1)
- MAC address low register (MALR1)
- Receive frame length register (RFLR1)
- PHY status register (PSR1)
- Transmit retry over counter register (TROCR1)
- Delayed collision detect counter register (CDCR1)
- Lost carrier counter register (LCCR1)
- Carrier not detect counter register (CNDCR1)
- CRC error frame receive counter register (CEFCR1)
- Frame receive error counter register (FRECR1)
- Too-short frame receive counter register (TSFRCR1)
- Too-long frame receive counter register (TLFRCR1)
- Residual-bit frame receive counter register (RFCR1)
- Multicast address frame receive counter register (MAFCR1)
- IPG register (IPGR1)

TSU Control Registers:

- TSU counter reset register (TSU_CTRST)
- Relay enable register (Port 0 to 1) (TSU_FWEN0)
- Relay enable register (Port 1 to 0) (TSU_FWEN1)
- Relay FIFO size select register (TSU_FCM)
- Relay FIFO overflow alert set register (port 0) (TSU_BSYSL0)
- Relay FIFO overflow alert set register (port 1) (TSU_BSYSL1)
- Transmit/relay priority control mode register (port 0) (TSU_PRISL0)
- Transmit/relay priority control mode register (port 1) (TSU_PRISL1)
- Receive/relay function set register (port 0 to 1) (TSU_FWSL0)
- Receive/relay function set register (port 1 to 0) (TSU_FWSL1)
- Relay function set register (common) (TSU_FWSLC)
- Qtag addition/deletion set register (port 0 to 1) (TSU_QTAGM0)
- Qtag addition/deletion set register (port 1 to 0) (TSU_QTAGM1)

- Relay status register (TSU_FWSR)
- Relay status interrupt mask register (TSU_FWINMK)
- Added Qtag value set register (port 0 to 1) (TSU_ADQT0)
- Added Qtag value set register (port 1 to 0) (TSU_ADQT1)
- CAM entry table busy register (TSU_ADSBSY)
- CAM entry table enable register (TSU_TEN)
- CAM entry table POST1 to POST4 registers (TSU_POST1 to TSU_POST4)
- CAM entry table 0 to 31 H registers (TSU_ADRH0 to TSU_ADRH31)
- CAM entry table 0 to 31 L registers (TSU_ADRL0 to TSU_ADRL31)
- Transmit frame counter register (port 0) (normal transmission only) (TXNLCR0)
- Transmit frame counter register (port 0) (normal and error transmission) (TXALCR0)
- Receive frame counter register (port 0) (normal reception only) (RXNLCR0)
- Receive frame counter register (port 0) (normal and error reception) (RXALCR0)
- Relay frame counter register (port 1 to 0) (normal relay only) (FWNLCR0)
- Relay frame counter register (port 1 to 0) (normal and error relay) (FWALCR0)
- Transmit frame counter register (port 1) (normal transmission only) (TXNLCR1)
- Transmit frame counter register (port 1) (normal and error transmission) (TXALCR1)
- Receive frame counter register (port 1) (normal reception only) (RXNLCR1)
- Receive frame counter register (port 1) (normal and error reception) (RXALCR1)
- Relay frame counter register (port 0 to 1) (normal relay only) (FWNLCR1)
- Relay frame counter register (port 0 to 1) (normal and error relay) (FWALCR1)

18.3.1 Software Reset Register (ARSTR)

ARSTR resets all modules (EtherC and E-DMAC) related to the Ethernet. By writing 1 to the ARST bit in ARSTR, a software reset is issued to all modules related to the Ethernet (for 64 cycles at external bus clock Bϕ). The ARST bit is always read as 0. While a software reset is issued, register access to all modules related to the Ethernet is prohibited.

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	ARST	0	R/W	Software Reset When written with 1, a software reset is issued to all modules related to the Ethernet (for 64 cycles at external bus clock Bϕ). Writing 0 does not affect this bit. This bit is always read as 0. While a software reset is issued, register access to all modules related to the Ethernet is prohibited. The following registers are not initialized by a software reset. TSU_ADRH0 to TSU_ADRH31, TSU_ADRL0 to TSU_ADRL31, TXNLCR0, TXNLCR1, TXALCR0, TXALCR1, RXNLCR0, RXNLCR1, RXALCR0, RXALCR1, FWNLCR0, FWNLCR1, FWALCR0, FWALCR1

18.3.2 EtherC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register and specifies the operating mode of the Ethernet controller. The settings in this register are normally made in the initialization process following a reset.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

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	MCT	0	R/W	Multicast Address Frame Receive Mode 0: Frames other than the multicast address set by the CAM entry table 0 to 31 (H/L) registers are received. However, if the on-chip CAM entry table reference is disabled, all multicast address frames are received. 1: Only the multicast address set by the CAM entry table 0 to 31 (H/L) registers is received.
12	PRCEF	0	R/W	CRC Error Frame Reception Enable 0: A receive frame including a CRC error is received as a frame with an error. 1: A receive frame including a CRC error is received as a frame without an error. When this bit is cleared to 0, the CRC error is reflected in ECSR of the E-DMAC and the status of the receive descriptor. When this bit is set to 1, a frame is received as a normal frame.
11	—	0	R	Reserved
10	—	0	R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection by hardware to allow activation from the Ethernet. 0: Magic Packet detection is not enabled 1: Magic Packet detection is enabled
8	—	0	R	Reserved
7	—	0	R	These bits are always read as 0. The write value should always be 0.
6	RE	0	R/W	Reception Enable If a switch is made from receive function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receive function will be enabled until reception of the corresponding frame is completed. 0: Receive function is disabled 1: Receive function is enabled
5	TE	0	R/W	Transmission Enable If a switch is made from transmit function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmit function will be enabled until transmission of the corresponding frame is completed. 0: Transmit function is disabled 1: Transmit function is enabled
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is performed. 1: Data loopback is performed inside the MAC in the EtherC.

Bit	Bit Name	Initial Value	R/W	Description
2	ELB	0	R/W	External Loop Back Mode This bit value is output directly to this LSI's general-purpose external output pin (EXOUT). This bit is used for loopback mode directives, etc., in the PHY-LSI, using the EXOUT pin. In order for PHY-LSI loopback to be implemented using this function, the PHY-LSI must have a pin corresponding to the EXOUT pin. 0: Low-level output from the EXOUT pin 1: High-level output from the EXOUT pin
1	DM	0	R/W	Duplex Mode Specifies the EtherC transfer method. 0: Half-duplex transfer is specified 1: Full-duplex transfer is specified
0	PRM	0	R/W	Promiscuous Mode Setting this bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.). 0: EtherC performs normal operation 1: EtherC performs promiscuous mode operation

18.3.3 EtherC Status Register (ECSR)

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

The interrupts generated due to this status register are indicated in each ECI bit in EESR of the E-DMAC0 derived from port0 and the E-DMAC1 derived from port1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from the PHY-LSI has changed from high to low or low to high. However, signal changes may be detected at the timing at which the LNKSTA function was selected using PACR of PFC. To check the current Link state, refer to the LMON bit in the PHY status register (PSR). 0: Change in the LNKSTA signal has not been detected 1: Change in the LNKSTA signal has been detected (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detected on the line. 0: Magic Packet has not been detected 1: Magic Packet has been detected

Bit	Bit Name	Initial Value	R/W	Description
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs before the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used. 0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line

18.3.4 EtherC Interrupt Permission Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	LCHNGIP	0	R/W	LINK Signal Changed Interrupt Enable 0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled
1	MPDIP	0	R/W	Magic Packet Detection Interrupt Enable 0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled
0	ICDIP	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

18.3.5 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY-LSI registers via the MII.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDI	Undefined	R	MII Management Data-In Indicates the level of the MDIO pin.
2	MDO	0	R/W	MII Management Data-Out Outputs the value set to this bit from the MDIO pin, when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with respect to the MII. 0: Read direction is indicated 1: Write direction is indicated
0	MDC	0	R/W	MII Management Data Clock Outputs the value set to this bit from the MDC pin and supplies the MII with the management data clock. For the method of accessing the MII registers, see section 18.4.6, Accessing MII Registers.

18.3.6 MAC Address High Register (MAHR)

MAHR is a 32 -bit readable/writable register that specifies the upper 32 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MA47 to MA16	All 0	R/W	MAC Address Bits These bits are used to set the upper 32 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'01234567.

18.3.7 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MA15 to MA0	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'000089AB.

18.3.8 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register and it specifies the maximum frame length (in bytes) that can be received by this LSI. The settings in this register must not be changed while the receiving function is enabled.

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	RFL11 to RFL0	All 0	R/W	Receive Frame Length 11 to 0 The frame length described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data is not included in the transfer. When data that exceeds the specified value is received, the part of the data that exceeds the specified value is discarded. H'000 to H'5EE: 1,518 bytes H'5EF: 1,519 bytes H'5F0: 1,520 bytes : : : : H'7FF: 2,047 bytes H'800 to H'FFF: 2,048 bytes

18.3.9 PHY Status Register (PSR)

PSR is a read-only register that can read interface signals from the PHY-LSI.

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	LMON	0	R	LNKSTA Pin Status The Link status can be read by connecting the Link signal output from the PHY-LSI to the LNKSTA pin. For the polarity, refer to the PHY-LSI specifications to be connected.

18.3.10 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer. When 16 transmission attempts have failed, TROCR is incremented by 1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TROC31 to TROC0	All 0	R/W	Transmit Retry Over Count These bits indicate the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.

18.3.11 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 32-bit counter that indicates the number of all delayed collisions that accured on the line after the start of data transmission. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	COSDC31 to COSDC0	All 0	R/W	Delayed Collision Detect Count These bits indicate the number of all delayed collisions after the start of data transmission.

18.3.12 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during data transmission. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by writing to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC31 to LCC0	All 0	R/W	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

18.3.13 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier could not be detected while the preamble was being sent. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNDC31 to CNDC0	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times the carrier was not detected.

18.3.14 CRC Error Frame Receive Counter Register (CEFCR)

CEFCR is a 32-bit counter that indicates the number of times a frame with a CRC error was received. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CEFC31 to CEFC0	All 0	R/W	CRC Error Frame Count These bits indicate the count of CRC error frames received.

18.3.15 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames for which a receive error was indicated by the RX-ER input pin from the PHY-LSI. FRECR is incremented each time the RX-ER pin becomes active. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FREC31 to FREC0	All 0	R/W	Frame Receive Error Count These bits indicate the count of errors during frame reception.

18.3.16 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames of fewer than 64 bytes that have been received. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC31 to TSFC0	All 0	R/W	Too-Short Frame Receive Count These bits indicate the count of frames received with a length of less than 64 bytes.

18.3.17 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 32-bit counter that indicates the number of frames received with a length exceeding the value specified by the receive frame length register (RFLR). When the value in this register reaches H'FFFFFFFF, the count is halted. TLFRCR is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame counter register (RFCR). The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TLFC31 to TLFC0	All 0	R/W	Too-Long Frame Receive Count These bits indicate the count of frames received with a length exceeding the value in RFLR.

18.3.18 Residual-Bit Frame Receive Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC31 to RFC0	All 0	R/W	Residual-Bit Frame Count These bits indicate the count of frames received containing residual bits.

18.3.19 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specified multicast address. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC31 to MAFC0	All 0	R/W	Multicast Address Frame Count These bits indicate the count of multicast frames received.

18.3.20 IPG Register (IPGR)

IPGR sets the IPG (Inter Packet Gap). This register must not be changed while the transmitting and receiving functions of the EtherC mode register (ECMR) are enabled. (For details, refer to section 18.4.8, Operation by IPG Setting.)

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 0	IPG4 to IPG0	H′13	R/W	Inter Packet Gap Sets the IPG value every 4-bit time. H′00: 20-bit time H′01: 24-bit time : : H′13: 96-bit time (Default) : : H′1F: 144-bit time

18.3.21 TSU Counter Reset Register (TSU_CTRST)

TSU_CTRST clears the transmit, receive, and transfer frame counters to 0.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CTRST	0	R/W	TSU Counter Reset When 1 is written to this bit, the values of registers TXNCR0/1, TXALCR0/1, RXNLCR0/1, RXALCR0/1, FWNLCR0/1, and FWALCR0/1 are cleared to 0. Writing 0 does not affect this bit. These bits are always read as 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.22 Relay Enable Register (Port 0 to 1) (TSU_FWEN0)

TSU_FWEN0 enables or disables relay operations from the MAC-0 to MAC-1 (writing to the relay FIFO).

Bit	Bit Name	Initial Value	R/W	Description
31	FWEN0	0	R/W	Port 0 to 1 Relay Operation Enable 0: Port 0 to 1 relay is disabled 1: Port 0 to 1 relay is enabled When the value of the FCM2 to FCM0 in the TSU FIFO size select register TSU_FCM is set to H′4, setting this bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.23 Relay Enable Register (Port 1 to 0) (TSU_FWEN1)

TSU_FWEN1 enables or disables relay operations from the MAC-1 to MAC-0 (writing to the relay FIFO).

Bit	Bit Name	Initial Value	R/W	Description
31	FWEN1	0	R/W	Port 1 to 0 Relay Operation Enable 0: Port 1 to 0 relay is disabled 1: Port 1 to 0 relay is enabled When the value of the FCM2 to FCM0 in the TSU FIFO size select register TSU_FCM is set to H′3, setting this bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.24 Relay FIFO Size Select Register (TSU_FCM)

TSU_FCM selects the size of the TSU FIFO, used for relay operations between the MAC-0 and MAC-1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	FCM2 to FCM0	All 0	R/W	TSU FIFO Size H′0: Port 0 to 1: 3 kbytes Port 1 to 0: 3 kbytes H′1: Port 0 to 1: 4 kbytes Port 1 to 0: 2 kbytes H′2: Port 0 to 1 : 5 kbytes Port 1 to 0: 1 kbyte H′3: Port 0 to 1: 6 kbytes Port 1 to 0: Not used H′4: Port 0 to 1: Not used Port 1 to 0: 6 kbytes H′5: Port 0 to 1: 1 kbyte Port 1 to 0: 5 kbytes H′6: Port 0 to 1: 2 kbytes Port 1 to 0: 4 kbytes H′7: Setting prohibited Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

18.3.25 Relay FIFO Overflow Alert Set Register (Port 0) (TSU_BSYSL0)

The TSU has an alert function, which informs the MAC-0 and MAC-1 that writing to the TSU FIFO will be disabled when the data volume written in the TSU FIFO during relay operations exceeds a certain threshold. TSU_BSYSL0 sets the threshold of the TSU FIFO when the TSU alerts the MAC-0 that writing to the TSU FIFO will be disabled during relay operations.

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	BSYSL05 to BSYSL00	All 1	R/W	Sets the threshold of the port 0 to 1 TSU FIFO capacity in 256-byte units when the TSU alerts the MAC-0 that writing in the TSU FIFO will be disabled during relay operations. H′00: 0 byte H′01: 256 bytes H′02: 512 bytes : : H′16: 5632 bytes H′17: 5888 bytes Settings are disabled for H′18 to H′3F. (Alert is not always carried out.) When H′00 is set, the TSU always alerts the MAC-0 that writing to the TSU FIFO will be disabled. When the value set is above the port 0 to 1 transfer FIFO capacity set by the FCM2 to FCM0 in TSU_FCM, the TSU does not alert the MAC-0 that writing to the TSU FIFO will be disabled. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1). When the enable bit of relay operations (the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1) is cleared to 0, the TSU stops alerting the MAC-0 that writing to the TSU FIFO will be disabled.

18.3.26 Relay FIFO Overflow Alert Set Register (Port 1) (TSU_BSYSL1)

The TSU has an alert function, which informs the MAC-0 and MAC-1 that writing to the TSU FIFO will be disabled when the data volume written in the TSU FIFO during relay operations exceeds a certain threshold. TSU_BSYSL1 sets the threshold of the TSU FIFO when the TSU alerts the MAC-1 to writing to the TSU FIFO will be disabled during relay operations.

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	BSYSL15 to BSYSL10	All 1	R/W	Sets the threshold of the port 1 to 0 TSU FIFO capacity in 256-byte units when the TSU alerts the MAC-1 that writing in the TSU FIFO will be disabled during relay operations. H′00: 0 byte H′01: 256 bytes H′02: 512 bytes : : H′16: 5632 bytes H′17: 5888 bytes Settings are disabled for H′18 to H′3F. (Alert is not always carried out.) When H′00 is set, the TSU always alerts the MAC-1 that writing to the transfer FIFO will be disabled. When the value set is above the port 1 to 0 TSU FIFO capacity set by the FCM2 to FCM0 in TSU_FCM, the TSU does not alert the MAC-1 to writing that the TSU FIFO will be disabled. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1). When the enable bit of relay operations (the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1) is cleared to 0, the TSU stops alerting the MAC-1 to writing to the TSU FIFO will be disabled.

18.3.27 Transmit/Relay Priority Control Mode Register (Port 0) (TSU_PRISL0)

TSU_PRISL0 sets the priority control mode when the transmission request from the E-DMAC to MAC-0 come into collision with port 1 to 0 relay operations. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 12	PRIMD02 to PRIMD00	All 0	R/W	Sets the priority control mode of MAC-0 transmission and port 1 to 0 relay operations. H′0: Round robin H′1: Transmission priority H′2: Relay priority H′4: Round robin, however switched to relay priority when TSU FIFO use amount exceeds the set value of PRISL07 to PRISL00 H′5: Transmission priority, however switched to relay priority when TSU FIFO use amount exceeds the set value of PRISL07 to PRISL00 Others: 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 to 0	PRISL07 to PRISL00	All 0	R/W	Sets the threshold of the port 1 to 0 TSU FIFO capacity in 64-byte units in the event switching to relay priority when PRIMD02 to PRIMD00 are set to H′4 or H′5. H′00: 0 byte H′01: 64 bytes H′02: 128 bytes : : H′5E: 6016 bytes H′5F: 6080 bytes Settings are disabled for H′60 to H′FF. When set to H′00, relay always takes priority. When the value set is above the port 1 to 0 TSU FIFO capacity set by the FCM2 to FCM0 in TSU_FCM, if the PRIMD02 to PRIMD00 are H′4, round robin will always be set. If the PRIMD02 to PRIMD00 are H′5, transmission always takes priority.

18.3.28 Transmit/Relay Priority Control Mode Register (Port 1) (TSU_PRISL1)

TSU_PRISL1 sets the priority control mode when the transmission request from the E-DMAC to MAC-1 come into collision with port 0 to 1 relay operations. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	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
14 to 12	PRIMD12 to PRIMD10	All 0	R/W	Sets the priority control mode of MAC-1 transmission and port 0 to 1 relay operations. H′0: Round robin H′1: Transmission priority H′2: Relay priority H′4: Round robin, however switched to relay priority when TSU FIFO use amount exceeds the set value of PRISL17 to PRISL10 H′5: Transmission priority, however switched to relay priority when TSU FIFO use amount exceeds the set value of PRISL17 to PRISL10 Others: Setting prohibited
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	PRISL17 to PRISL10	All 0	R/W	Sets the threshold value of the port 0 to 1 TSU FIFO capacity in 64-byte units in the event switching to relay priority when PRIMD12 to PRIMD10 are set to H′4 or H′5. H′00: 0 byte H′01: 64 bytes H′02: 128 bytes : : H′5E: 6016 bytes H′5F: 6080 bytes Settings are disabled for H′60 to H′FF. When set to H′00, relay always takes priority. When the value set is above the port 0 to 1 TSU FIFO capacity set by the FCM2 to FCM0 in TSU_FCM, if the PRIMD12 to PRIMD10 are H′4, round robin will always be set. If the PRIMD12 to PRIMD10 are H′5, transmission always takes priority.

18.3.29 Receive/Relay Function Set Register (Port 0 to 1) (TSU_FWSL0)

TSU_FWSL0 sets the processing method of each frame in port 0 reception and port 0 to 1 relay operations. In receiving a frame, the processing method can be determined by referring to the CAM evaluation results when the multicast frame and the destination are other than this LSI. (For details, refer to section 18.4.4, CAM Function.) Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

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	FW40	0	R/W	Sets the processing method when frames from port 0 are addressed to this LSI 0: Frames are not relayed 1: Frames are relayed to port 1
10	FW30	0	R/W	Sets the processing method when frames from port 0 are Broadcast. 0: Frames are not relayed 1: Frames are relayed to port 1
9	FW20	0	R/W	Sets the processing method when frames from port 0 are multicast. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1
8	FW10	0	R/W	Sets the processing method when frames from port 0 are addressed to other than this LSI. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1

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

18.3.30 Receive/Relay Function Set Register (Port 1 to 0) (TSU_FWSL1)

TSU_FWSL1 sets the processing method of each frame in port 1 reception and port 1 to 0 relay operations. In receiving a frame, the processing method can be determined by referring to the CAM evaluation results when the multicast frame and the destination are other than this LSI. (For details, refer to section 18.4.4, CAM Function.) Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

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	FW41	0	R/W	Sets the processing method when frames from port 1 are addressed to this LSI 0: Frames are not relayed 1: Frames are relayed to port 0
10	FW31	0	R/W	Sets the processing method when frames from port 1 are Broadcast. 0: Frames are not relayed 1: Frames are relayed to port 0
9	FW21	0	R/W	Sets the processing method when frames from port 1 are multicast. 0: CAM hit: Frames are relayed to port 0 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 0

Bit	Bit Name	Initial Value	R/W	Description
8	FW11	0	R/W	Sets the processing method when frames from port 1 are addressed to other than this LSI. 0: CAM hit: Frames are relayed to port 0 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 0
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.31 Relay Function Set Register (Common) (TSU_FWSLC)

When the CAM is used, the referred area in the CAM entry table (partially or wholly) can be specified by the TSU_POST1 to TSU_POST4 registers. When the CAM is installed outside this LSI, the evaluation results of the external CAM can be referred by input on the CAMSEN0 and CAMSEN1 pins. (For details, refer to section 18.4.4, CAM Function.) TSU_FWSLC enables settings by the TSU_POST1 to TSU_POST4 registers and conditions for referring signals on the CAMSEN0 and CAMSEN1 pins. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

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	POSTENU	0	R/W	Enables the settings of the POST field of CAM entry tables 0 to 15 (settings by the TSU_POST1 and TSU_POST2 registers). 0: Disables the settings of the POST field. (The CAM entry table is referred only in port 0 reception.) 1: Enables the settings of the POST field. (The CAM entry table reference conditions follow the POST field settings.)
12	POSTENL	0	R/W	Enables the settings of the POST field of CAM entry tables 16 to 31 (settings by the TSU_POST3 and TSU_POST4 registers). 0: Disables the settings of the POST field. (The CAM entry table is referred only in port 1 reception.) 1: Enables the settings of the POST field. (The CAM entry table reference conditions follow the POST field settings.)
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	CAMSEL03	1	R/W	These bits set the conditions for referring signals on the CAMSEN0 pin. By setting multiple bits to 1, multiple conditions can be selected. CAMSEL03: Refers signals on the CAMSEN0 pin in port 0 reception CAMSEL02: Refers signals on the CAMSEN0 pin in port 0 to 1 relay CAMSEL01: Refers signals on the CAMSEN0 pin in port 1 reception CAMSEL00: Refers signals on the CAMSEN0 pin in port 1 to 0 relay
6	CAMSEL02	0	R/W	
5	CAMSEL01	0	R/W	
4	CAMSEL00	0	R/W	
3	CAMSEL13	0	R/W	These bits set the conditions for referring signals on the CAMSEN1 pin. By setting multiple bits to 1, multiple conditions can be selected. CAMSEL13: Refers signals on the CAMSEN1 pin in port 0 reception CAMSEL12: Refers signals on the CAMSEN1 pin in port 0 to 1 relay CAMSEL11: Refers signals on the CAMSEN1 pin in port 1 reception CAMSEL10: Refers signals on the CAMSEN1 pin in port 1 to 0 relay
2	CAMSEL12	0	R/W	
1	CAMSEL11	1	R/W	
0	CAMSEL10	0	R/W	

18.3.32 Qtag Addition/Deletion Set Register (Port 0 to 1) (TSU_QTAGM0)

TSU_QTAGM0 sets the functions adding Qtag from the normal Ethernet frames (no Qtag) to IEEE802.1Q frames (with Qtag) and deleting Qtags from IEEE802.1Q frames (with Qtag) to normal Ethernet frames (no Qtag) during port 0 to 1 relay operations. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

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.
1,0	QTAGM01, QTAGM00	All 0	R/W	These bits set Qtag adding and deleting functions during port 0 to 1 relay operations. H′0: No Qtag adding and deleting functions H′1: No Qtag adding and deleting functions (same as H′0) H′2: Deletes Qtag from frames with Qtag H′3: Adds Qtag to frames with no Qtag Writing to this register is prohibited, after transfer operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

18.3.33 Qtag Addition/Deletion Set Register (Port 1 to 0) (TSU_QTAGM1)

TSU_QTAGM1 sets the functions adding Qtag from the normal Ethernet frames (no Qtag) to IEEE802.1Q frames (with Qtag) and deleting Qtags from IEEE802.1Q frames (with Qtag) to normal Ethernet frames (no Qtag) during port 1 to 0 relay operations. Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

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, 0	QTAGM11, QTAGM10	All 0	R/W	These bits set Qtag adding and deleting functions during port 1 to 0 relay operations. H′0: No Qtag adding and deleting functions H′1: No Qtag adding and deleting functions (same as H′0) H′2: Deletes Qtag from frames with Qtag H′3: Adds Qtag to frames with no Qtag Writing to this register is prohibited, after transfer operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

18.3.34 Relay Status Register (TSU_FWSR)

TSU_FWSR is a 32-bit readable/writable register that indicates the status during relay operations. By setting the TSU status interrupt mask register (TSU_FWINMK), this status can be notified to the CPU as an interrupt source. The status bit set to 1 will be cleared to 0 by writing 1 to corresponding bit. (The status bit retains the value until it is cleared to 0.)

Interrupts generated due to this status register is EINT2. For details on the priority order of interrupts, refer to section 8.3.5, Interrupt Exception Handling and Priority in section 8, Interrupt Controller (INTC).

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	TINT40	0	R/W	MAC-0 Carrier Not Detect Set to 1 when a carrier not detect has occured in the MAC-0
26	TINT30	0	R/W	MAC-0 Carrier Lost Set to 1 when a carrier is lost during data transmission in the MAC-0
25	TINT20	0	R/W	MAC-0 Collision Detect Set to 1 when a collision of frames is detected in the MAC-0
24	TINT10	0	R/W	MAC-0 Transmission Time Out Set to 1 when frames were unable to be transmitted in 16 transmission attempts including the retransfer in the MAC-0
23	OVF0	0	R/W	Port 0 to 1 TSU FIFO Overflow Detect Set to 1 when a port 0 to 1 TSU FIFO overflow has occured
22	RBSY0	0	R/W	MAC-0 Overflow Alert Signal Output Set to 1 when the threshold of TSU_BSYSL0 is valid and exceeded
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	Description
20	RINT50	0	R/W	MAC-0 Residual Bit Frame Receive Set to 1 when frames containing residual bits (less than an 8-bit unit) are received in the MAC-0
19	RINT40	0	R/W	MAC-0 Exceeding Byte Frame Receive Set to 1 when frames exceeding the value set by RFLR0 are received in the MAC-0
18	RINT30	0	R/W	MAC-0 Less 64-Byte Frame Receive Set to 1 when frames with a length of less than 64 bytes are received in the MAC-0
17	RINT20	0	R/W	MAC-0 Frame Receive Error Set to 1 when a receive error is detected on the RX-ER pin input from the PHY in the MAC-0
16	RINT10	0	R/W	MAC-0 CRC Error Frame Receive Set to 1 when a receive frame results in a CRC error in the MAC-0
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	TINT41	0	R/W	MAC-1 Carrier Not Detect Set to 1 when a carrier not detect has occured in the MAC-1
10	TINT31	0	R/W	MAC-1 Carrier Lost Set to 1 when a carrier is lost during data transmission in the MAC-1
9	TINT21	0	R/W	MAC-1 Collision Detect Set to 1 when a collision of frames is detected in the MAC-1
8	TINT11	0	R/W	MAC-1 Transmission Time Out Set to 1 when frames were unable to be transmitted in 16 transmission attempts including the retransfer in the MAC-1
7	OVF1	0	R/W	Port 1 to 0 TSU FIFO Overflow Detect Set to 1 when a port 1 to 0 TSU FIFO overflow has occured

Bit	Bit Name	Initial Value	R/W	Description
6	RBSY1	0	R/W	MAC-1 Overflow Alert Signal Output Set to 1 when the threshold of TSU_BSYSL1 is valid and exceeded
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	RINT51	0	R/W	MAC-1 Residual Bit Frame Receive Set to 1 when frames containing residual bits (less than an 8-bit unit) are received in the MAC-1
3	RINT41	0	R/W	MAC-1 Exceeding Byte Frame Receive Set to 1 when frames exceeding the value set by RFLR1 are received in the MAC-1
2	RINT31	0	R/W	MAC-1 Less 64-Byte Frame Receive Set to 1 when frames with a length of less than 64 bytes are received in the MAC-1
1	RINT21	0	R/W	MAC-1 Frame Receive Error Set to 1 when a receive error is detected on the RX-ER pin input from the PHY in the MAC-1
0	RINT11	0	R/W	MAC-1 CRC Error Frame Receive Set to 1 when a receive frame results in a CRC error in the MAC-1

18.3.35 Relay Status Interrupt Mask Register (TSU_FWINMK)

TSU_FWINMK is a 32-bit readable/writable register that sets the interrupt mask for status bits in TSU_FWSR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27	TINTM40	0	R/W	MAC-0 Carrier Not Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

Bit	Bit Name	Initial Value	R/W	Description
26	TINTM30	0	R/W	MAC-0 Carrier Lost Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
25	TINTM20	0	R/W	MAC-0 Collision Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
24	TINTM10	0	R/W	MAC-0 Transmission Time Out Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
23	OVFM0	0	R/W	Port 0 to 1 TSU FIFO Overflow Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
22	RBSYM0	0	R/W	MAC-0 Overflow Alert Signal Output Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	RINTM50	0	R/W	MAC-0 Residual Bit Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
19	RINTM40	0	R/W	MAC-0 Exceeding Byte Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
18	RINTM30	0	R/W	MAC-0 Less 64-Byte Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
17	RINTM20	0	R/W	MAC-0 Frame Receive Error Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

Bit	Bit Name	Initial Value	R/W	Description
16	RINTM10	0	R/W	MAC-0 CRC Error Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	TINTM41	0	R/W	MAC-1 Carrier Not Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
10	TINTM31	0	R/W	MAC-1 Carrier Lost Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
9	TINTM21	0	R/W	MAC-1 Collision Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
8	TINTM11	0	R/W	MAC-1 Transmission Time Out Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
7	OVFM1	0	R/W	Port 1 to 0 TSU FIFO Overflow Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
6	RBSYM1	0	R/W	MAC-1 Overflow Alert Signal Output Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	RINTM51	0	R/W	MAC-1 Residual Bit Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

Bit	Bit Name	Initial Value	R/W	Description
3	RINTM41	0	R/W	MAC-1 Exceeding Byte Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
2	RINTM31	0	R/W	MAC-1 Less 64-Byte Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
1	RINTM21	0	R/W	MAC-1 Frame Receive Error Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
0	RINTM11	0	R/W	MAC-1 CRC Error Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

18.3.36 Added Qtag Value Set Register (Port 0 to 1) (TSU_ADQT0)

TSU_ADQT0 sets Qtag data to be added in the conversion of normal Ethernet frames (without Qtag) to IEEE802.1Q frames (with Qtag) in port 0 to 1 relay operations (when setting the QTAGM01 to QTAGM00 bits in TSU_QTAGM0 to H′3 in the use of the Qtag addition function). Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	QTAG031 to QTAG016	H′8100	R/W	Be sure to set the value of the upper 16 bits (QTAG031 to QTAG016) as H′8100 (indicates that it is the Qtag extension frame format). The value read is H′8100.
15 to 13	QTAG015 to QTAG013	H′0	R/W	Priority Setting (PRT) These bits set the processing priority of frames with Qtag. For details on the settings, refer to the specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	QTAG011 to QTAG000	H′000	R/W	V-LAN ID Setting (VID) These bits set the flames with Qtag to be used in the systems supporting V-LAN. For details on settings, refer to the specifications on Qtag control specified in IEEE802.1Q.

18.3.37 Added Qtag Value Set Register (Port 1 to 0) (TSU_ADQT1)

TSU_ADQT1 sets Qtag data to be added in the conversion of normal Ethernet frames (without Qtag) to IEEE802.1Q frames (with Qtag) in port 1 to 0 relay operations (when setting the QTAGM11 to QTAGM10 bits in TSU_QTAGM1 to H′3 in the use of the Qtag addition function). Writing to this register is prohibited, after relay operations have been enabled once (after the FWEN0 in TSU_FWEN0 or the FWEN1 in TSU_FWEN1 is set to 1).

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	QTAG131 to QTAG116	H′8100	R/W	Be sure to set the value of the upper 16 bits (QTAG131 to QTAG116) as H′8100 (indicates that it is the Qtag extension frame format). The value read is H′8100.
15 to 13	QTAG115 to QTAG113	H′0	R/W	Priority Setting (PRT) These bits set the processing priority of frames with Qtag. For details on the settings, refer to the specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	QTAG111 to QTAG100	H′000	R/W	V-LAN ID Setting (VID) These bits set the flames with Qtag to be used in the systems supporting V-LAN. For details on settings, refer to the specifications on Qtag control specified in IEEE802.1Q.

18.3.38 CAM Entry Table Busy Register (TSU_ADSBSY)

When CAM entry table registers (TSU_ADRH0 to TSU_ADRH31, TSU_ADRL0 to TSU_ADRL31) are set by register writing, the ADSBSY bit in this register is set to 1 (when the process of reflecting the contents of the CAM entry table register in the CAM controller is completed inside the TSU, the ADSBSY bit is automatically restored to 0).

Accessing to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while the ADSBSY bit in this register is set to 1. This register is a read-only status register, and writing to this register is prohibited.

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	ADSBSY	0	R	CAM Entry Table Setting Busy When TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 are set by register writing, the ADSBSY bit is set to 1. When the process of reflecting the contents of the CAM entry table register in the CAM controller is completed inside the TSU, the ADSBSY bit is automatically restored to 0. Accessing to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while this bit is set to 1. Writing to this register is also prohibited.

18.3.39 CAM Entry Table Enable Register (TSU_TEN)

TSU_TEN enables or disables to refer CAM Entry Table registers (TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31).

Bit	Bit Name	Initial Value	R/W	Description
31	TEN0	0	R/W	CAM Entry Table 0 (TSU_ADRH0 and TSU_ADRL0) Setting 0: Disabled 1: Enabled
30	TEN1	0	R/W	CAM Entry Table 1 (TSU_ADRH1 and TSU_ADRL1) Setting 0: Disabled 1: Enabled
29	TEN2	0	R/W	CAM Entry Table 2 (TSU_ADRH2 and TSU_ADRL2) Setting 0: Disabled 1: Enabled
28	TEN3	0	R/W	CAM Entry Table 3 (TSU_ADRH3 and TSU_ADRL3) Setting 0: Disabled 1: Enabled
27	TEN4	0	R/W	CAM Entry Table 4 (TSU_ADRH4 and TSU_ADRL4) Setting 0: Disabled 1: Enabled
26	TEN5	0	R/W	CAM Entry Table 5 (TSU_ADRH5 and TSU_ADRL5) Setting 0: Disabled 1: Enabled
25	TEN6	0	R/W	CAM Entry Table 6 (TSU_ADRH6 and TSU_ADRL6) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
24	TEN7	0	R/W	CAM Entry Table 7 (TSU_ADRH7 and TSU_ADRL7) Setting 0: Disabled 1: Enabled
23	TEN8	0	R/W	CAM Entry Table 8 (TSU_ADRH8 and TSU_ADRL8) Setting 0: Disabled 1: Enabled
22	TEN9	0	R/W	CAM Entry Table 9 (TSU_ADRH9 and TSU_ADRL9) Setting 0: Disabled 1: Enabled
21	TEN10	0	R/W	CAM Entry Table 10 (TSU_ADRH10 and TSU_ADRL10) Setting 0: Disabled 1: Enabled
20	TEN11	0	R/W	CAM Entry Table 11 (TSU_ADRH11 and TSU_ADRL11) Setting 0: Disabled 1: Enabled
19	TEN12	0	R/W	CAM Entry Table 12 (TSU_ADRH12 and TSU_ADRL12) Setting 0: Disabled 1: Enabled
18	TEN13	0	R/W	CAM Entry Table 13 (TSU_ADRH13 and TSU_ADRL13) Setting 0: Disabled 1: Enabled
17	TEN14	0	R/W	CAM Entry Table 14 (TSU_ADRH14and TSU_ADRL14) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
16	TEN15	0	R/W	CAM Entry Table 15 (TSU_ADRH10 and TSU_ADRL15) Setting 0: Disabled 1: Enabled
15	TEN16	0	R/W	CAM Entry Table 16 (TSU_ADRH16 and TSU_ADRL16) Setting 0: Disabled 1: Enabled
14	TEN17	0	R/W	CAM Entry Table 17 (TSU_ADRH17 and TSU_ADRL17) Setting 0: Disabled 1: Enabled
13	TEN18	0	R/W	CAM Entry Table 18 (TSU_ADRH18 and TSU_ADRL18) Setting 0: Disabled 1: Enabled
12	TEN19	0	R/W	CAM Entry Table 19 (TSU_ADRH19 and TSU_ADRL19) Setting 0: Disabled 1: Enabled
11	TEN20	0	R/W	CAM Entry Table 20 (TSU_ADRH20 and TSU_ADRL20) Setting 0: Disabled 1: Enabled
10	TEN21	0	R/W	CAM Entry Table 21 (TSU_ADRH21 and TSU_ADRL21) Setting 0: Disabled 1: Enabled
9	TEN22	0	R/W	CAM Entry Table 22 (TSU_ADRH22 and TSU_ADRL22) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
8	TEN23	0	R/W	CAM Entry Table 23 (TSU_ADRH23 and TSU_ADRL23) Setting 0: Disabled 1: Enabled
7	TEN24	0	R/W	CAM Entry Table 24 (TSU_ADRH24 and TSU_ADRL24) Setting 0: Disabled 1: Enabled
6	TEN25	0	R/W	CAM Entry Table 25 (TSU_ADRH20 and TSU_ADRL25) Setting 0: Disabled 1: Enabled
5	TEN26	0	R/W	CAM Entry Table 26 (TSU_ADRH20 and TSU_ADRL26) Setting 0: Disabled 1: Enabled
4	TEN27	0	R/W	CAM Entry Table 27 (TSU_ADRH27 and TSU_ADRL27) Setting 0: Disabled 1: Enabled
3	TEN28	0	R/W	CAM Entry Table 28 (TSU_ADRH28 and TSU_ADRL28) Setting 0: Disabled 1: Enabled
2	TEN29	0	R/W	CAM Entry Table 29 (TSU_ADRH29 and TSU_ADRL29) Setting 0: Disabled 1: Enabled
1	TEN30	0	R/W	CAM Entry Table 30 (TSU_ADRH30 and TSU_ADRL30) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
0	TEN31	0	R/W	CAM Entry Table 31 (TSU_ADRH31 and TSU_ADRL31) Setting 0: Disabled 1: Enabled

18.3.40 CAM Entry Table POST1 Register (TSU_POST1)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST1 specifies the conditions for referring to TSU_ADRH0 to TSU_ADRH7 and TSU_ADRL0 to TSU_ADRL7. The settings of this register are valid when the POSENU bit in TSU_FWSLC is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST03 to POST00	All 0	R/W	These bits set the conditions for referring to the CAM entry table 0. By setting multiple bits to 1, multiple conditions can be selected. POST03: The CAM entry table 0 is referred in port 0 reception. POST02: The CAM entry table 0 is referred in port 0 to 1 relay. POST01: The CAM entry table 0 is referred in port 1 reception. POST00: The CAM entry table 0 is referred in port 1 to 0 relay.
27 to 24	POST13 to POST10	All 0	R/W	These bits set the conditions for referring to the CAM entry table 1. By setting multiple bits to 1, multiple conditions can be selected. POST13: The CAM entry table 1 is referred in port 0 reception. POST12: The CAM entry table 1 is referred in port 0 to 1 relay. POST11: The CAM entry table 1 is referred in port 1 reception. POST10: The CAM entry table 1 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST23 to POST20	All 0	R/W	These bits set the conditions for referring to the CAM entry table 2. By setting multiple bits to 1, multiple conditions can be selected. POST23: The CAM entry table 2 is referred in port 0 reception. POST22: The CAM entry table 2 is referred in port 0 to 1 relay. POST21: The CAM entry table 2 is referred in port 1 reception. POST20: The CAM entry table 2 is referred in port 1 to 0 relay.
19 to 16	POST33 to POST30	All 0	R/W	These bits set the conditions for referring to the CAM entry table 3. By setting multiple bits to 1, multiple conditions can be selected. POST33: The CAM entry table 3 is referred in port 0 reception. POST32: The CAM entry table 3 is referred in port 0 to 1 relay. POST31: The CAM entry table 3 is referred in port 1 reception. POST30: The CAM entry table 3 is referred in port 1 to 0 relay.
15 to 12	POST43 to POST40	All 0	R/W	These bits set the conditions for referring to the CAM entry table 4. By setting multiple bits to 1, multiple conditions can be selected. POST43: The CAM entry table 4 is referred in port 0 reception. POST42: The CAM entry table 4 is referred in port 0 to 1 relay. POST41: The CAM entry table 4 is referred in port 1 reception. POST40: The CAM entry table 4 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST53 to POST50	All 0	R/W	These bits set the conditions for referring to the CAM entry table 5. By setting multiple bits to 1, multiple conditions can be selected. POST53: The CAM entry table 5 is referred in port 0 reception. POST52: The CAM entry table 5 is referred in port 0 to 1 relay. POST51: The CAM entry table 5 is referred in port 1 reception. POST50: The CAM entry table 5 is referred in port 1 to 0 relay.
7 to 4	POST63 to POST60	All 0	R/W	These bits set the conditions for referring to the CAM entry table 6. By setting multiple bits to 1, multiple conditions can be selected. POST63: The CAM entry table 6 is referred in port 0 reception. POST62: The CAM entry table 6 is referred in port 0 to 1 relay. POST61: The CAM entry table 6 is referred in port 1 reception. POST60: The CAM entry table 6 is referred in port 1 to 0 relay.
3 to 0	POST73 to POST70	All 0	R/W	These bits set the conditions for referring to the CAM entry table 7. By setting multiple bits to 1, multiple conditions can be selected. POST73: The CAM entry table 7 is referred in port 0 reception. POST72: The CAM entry table 7 is referred in port 0 to 1 relay. POST71: The CAM entry table 7 is referred in port 1 reception. POST70: The CAM entry table 7 is referred in port 1 to 0 relay.

18.3.41 CAM Entry Table POST2 Register (TSU_POST2)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST2 specifies the conditions for referring to TSU_ADRH8 to TSU_ADRH15 and TSU_ADRL8 to TSU_ADRL15. The settings of this register are valid when the POSENU bit in TSU_FWSLC is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST83 to POST80	All 0	R/W	These bits set the conditions for referring to the CAM entry table 8. By setting multiple bits to 1, multiple conditions can be selected. POST83: The CAM entry table 8 is referred in port 0 reception. POST82: The CAM entry table 8 is referred in port 0 to 1 relay. POST81: The CAM entry table 8 is referred in port 1 reception. POST80: The CAM entry table 8 is referred in port 1 to 0 relay.
27 to 24	POST93 to POST90	All 0	R/W	These bits set the conditions for referring to the CAM entry table 9. By setting multiple bits to 1, multiple conditions can be selected. POST93: The CAM entry table 9 is referred in port 0 reception. POST92: The CAM entry table 9 is referred in port 0 to 1 relay. POST91: The CAM entry table 9 is referred in port 1 reception. POST90: The CAM entry table 9 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST103 to POST100	All 0	R/W	These bits set the conditions for referring to the CAM entry table 10. By setting multiple bits to 1, multiple conditions can be selected. POST103: The CAM entry table 10 is referred in port 0 reception. POST102: The CAM entry table 10 is referred in port 0 to 1 relay. POST101: The CAM entry table 10 is referred in port 1 reception. POST100: The CAM entry table 10 is referred in port 1 to 0 relay.
19 to 16	POST113 to POST110	All 0	R/W	These bits set the conditions for referring to the CAM entry table 11. By setting multiple bits to 1, multiple conditions can be selected. POST113: The CAM entry table 11 is referred in port 0 reception. POST112: The CAM entry table 11 is referred in port 0 to 1 relay. POST111: The CAM entry table 11 is referred in port 1 reception. POST110: The CAM entry table 11 is referred in port 1 to 0 relay.
15 to 12	POST123 to POST120	All 0	R/W	These bits set the conditions for referring to the CAM entry table 12. By setting multiple bits to 1, multiple conditions can be selected. POST123: The CAM entry table 12 is referred in port 0 reception. POST122: The CAM entry table 12 is referred in port 0 to 1 relay. POST121: The CAM entry table 12 is referred in port 1 reception. POST120: The CAM entry table 12 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST133 to POST130	All 0	R/W	These bits set the conditions for referring to the CAM entry table 13. By setting multiple bits to 1, multiple conditions can be selected. POST133: The CAM entry table 13 is referred in port 0 reception. POST132: The CAM entry table 13 is referred in port 0 to 1 relay. POST131: The CAM entry table 13 is referred in port 1 reception. POST130: The CAM entry table 13 is referred in port 1 to 0 relay.
7 to 4	POST143 to POST140	All 0	R/W	These bits set the conditions for referring to the CAM entry table 14. By setting multiple bits to 1, multiple conditions can be selected. POST143: The CAM entry table 14 is referred in port 0 reception. POST142: The CAM entry table 14 is referred in port 0 to 1 relay. POST141: The CAM entry table 14 is referred in port 1 reception. POST140: The CAM entry table 14 is referred in port 1 to 0 relay.
3 to 0	POST153 to POST150	All 0	R/W	These bits set the conditions for referring to the CAM entry table 15. By setting multiple bits to 1, multiple conditions can be selected. POST153: The CAM entry table 15 is referred in port 0 reception. POST152: The CAM entry table 15 is referred in port 0 to 1 relay. POST151: The CAM entry table 15 is referred in port 1 reception. POST150: The CAM entry table 15 is referred in port 1 to 0 relay.

18.3.42 CAM Entry Table POST3 Register (TSU_POST3)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST3 specifies the conditions for referring to TSU_ADRH16 to TSU_ADRH23 and TSU_ADRL16 to TSU_ADRL23. The settings of this register are valid when the POSENU bit in TSU_FWSLC is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST163 to POST160	All 0	R/W	These bits set the conditions for referring to the CAM entry table 16. By setting multiple bits to 1, multiple conditions can be selected. POST163: The CAM entry table 16 is referred in port 0 reception. POST162: The CAM entry table 16 is referred in port 0 to 1 relay. POST161: The CAM entry table 16 is referred in port 1 reception. POST160: The CAM entry table 16 is referred in port 1 to 0 relay.
27 to 24	POST173 to POST170	All 0	R/W	These bits set the conditions for referring to the CAM entry table 17. By setting multiple bits to 1, multiple conditions can be selected. POST173: The CAM entry table 17 is referred in port 0 reception. POST172: The CAM entry table 17 is referred in port 0 to 1 relay. POST171: The CAM entry table 17 is referred in port 1 reception. POST170: The CAM entry table 17 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST183 to POST180	All 0	R/W	These bits set the conditions for referring to the CAM entry table 18. By setting multiple bits to 1, multiple conditions can be selected. POST183: The CAM entry table 18 is referred in port 0 reception. POST182: The CAM entry table 18 is referred in port 0 to 1 relay. POST181: The CAM entry table 18 is referred in port 1 reception. POST180: The CAM entry table 18 is referred in port 1 to 0 relay.
19 to 16	POST193 to POST190	All 0	R/W	These bits set the conditions for referring to the CAM entry table 19. By setting multiple bits to 1, multiple conditions can be selected. POST193: The CAM entry table 19 is referred in port 0 reception. POST192: The CAM entry table 19 is referred in port 0 to 1 relay. POST191: The CAM entry table 19 is referred in port 1 reception. POST190: The CAM entry table 19 is referred in port 1 to 0 relay.
15 to 12	POST203 to POST200	All 0	R/W	These bits set the conditions for referring to the CAM entry table 20. By setting multiple bits to 1, multiple conditions can be selected. POST203: The CAM entry table 20 is referred in port 0 reception. POST202: The CAM entry table 20 is referred in port 0 to 1 relay. POST201: The CAM entry table 20 is referred in port 1 reception. POST200: The CAM entry table 20 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST213 to POST210	All 0	R/W	These bits set the conditions for referring to the CAM entry table 21. By setting multiple bits to 1, multiple conditions can be selected. POST213: The CAM entry table 21 is referred in port 0 reception. POST212: The CAM entry table 21 is referred in port 0 to 1 relay. POST211: The CAM entry table 21 is referred in port 1 reception. POST210: The CAM entry table 21 is referred in port 1 to 0 relay.
7 to 4	POST223 to POST220	All 0	R/W	These bits set the conditions for referring to the CAM entry table 22. By setting multiple bits to 1, multiple conditions can be selected. POST223: The CAM entry table 22 is referred in port 0 reception. POST222: The CAM entry table 22 is referred in port 0 to 1 relay. POST221: The CAM entry table 22 is referred in port 1 reception. POST220: The CAM entry table 22 is referred in port 1 to 0 relay.
3 to 0	POST233 to POST230	All 0	R/W	These bits set the conditions for referring to the CAM entry table 23. By setting multiple bits to 1, multiple conditions can be selected. POST233: The CAM entry table 23 is referred in port 0 reception. POST232: The CAM entry table 23 is referred in port 0 to 1 relay. POST231: The CAM entry table 23 is referred in port 1 reception. POST230: The CAM entry table 23 is referred in port 1 to 0 relay.

18.3.43 CAM Entry Table POST4 Register (TSU_POST4)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST4 specifies the conditions for referring to TSU_ADRH24 to TSU_ADRH31 and TSU_ADRL24 to TSU_ADRL31. The settings of this register are valid when the POSENU bit in TSU_FWSLC is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST243 to POST240	All 0	R/W	These bits set the conditions for referring to the CAM entry table 24. By setting multiple bits to 1, multiple conditions can be selected. POST243: The CAM entry table 24 is referred in port 0 reception. POST242: The CAM entry table 24 is referred in port 0 to 1 relay. POST241: The CAM entry table 24 is referred in port 1 reception. POST240: The CAM entry table 24 is referred in port 1 to 0 relay.
27 to 24	POST253 to POST250	All 0	R/W	These bits set the conditions for referring to the CAM entry table 25. By setting multiple bits to 1, multiple conditions can be selected. POST253: The CAM entry table 25 is referred in port 0 reception. POST252: The CAM entry table 25 is referred in port 0 to 1 relay. POST251: The CAM entry table 25 is referred in port 1 reception. POST250: The CAM entry table 25 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST263 to POST260	All 0	R/W	These bits set the conditions for referring to the CAM entry table 26. By setting multiple bits to 1, multiple conditions can be selected. POST263: The CAM entry table 26 is referred in port 0 reception. POST262: The CAM entry table 26 is referred in port 0 to 1 relay. POST261: The CAM entry table 26 is referred in port 1 reception. POST260: The CAM entry table 26 is referred in port 1 to 0 relay.
19 to 16	POST273 to POST270	All 0	R/W	These bits set the conditions for referring to the CAM entry table 27. By setting multiple bits to 1, multiple conditions can be selected. POST273: The CAM entry table 27 is referred in port 0 reception. POST272: The CAM entry table 27 is referred in port 0 to 1 relay. POST271: The CAM entry table 27 is referred in port 1 reception. POST270: The CAM entry table 27 is referred in port 1 to 0 relay.
15 to 12	POST283 to POST280	All 0	R/W	These bits set the conditions for referring to the CAM entry table 28. By setting multiple bits to 1, multiple conditions can be selected. POST283: The CAM entry table 28 is referred in port 0 reception. POST282: The CAM entry table 28 is referred in port 0 to 1 relay. POST281: The CAM entry table 28 is referred in port 1 reception. POST280: The CAM entry table 28 is referred in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST293 to POST290	All 0	R/W	These bits set the conditions for referring to the CAM entry table 29. By setting multiple bits to 1, multiple conditions can be selected. POST293: The CAM entry table 29 is referred in port 0 reception. POST292: The CAM entry table 29 is referred in port 0 to 1 relay. POST291: The CAM entry table 29 is referred in port 1 reception. POST290: The CAM entry table 29 is referred in port 1 to 0 relay.
7 to 4	POST303 to POST300	All 0	R/W	These bits set the conditions for referring to the CAM entry table 30. By setting multiple bits to 1, multiple conditions can be selected. POST303: The CAM entry table 30 is referred in port 0 reception. POST302: The CAM entry table 30 is referred in port 0 to 1 relay. POST301: The CAM entry table 30 is referred in port 1 reception. POST300: The CAM entry table 30 is referred in port 1 to 0 relay.
3 to 0	POST313 to POST310	All 0	R/W	These bits set the conditions for referring to the CAM entry table 31. By setting multiple bits to 1, multiple conditions can be selected. POST313: The CAM entry table 31 is referred in port 0 reception. POST312: The CAM entry table 31 is referred in port 0 to 1 relay. POST311: The CAM entry table 31 is referred in port 1 reception. POST310: The CAM entry table 31 is referred in port 1 to 0 relay.

18.3.44 CAM Entry Table 0 to 31 H Registers (TSU_ADRH0 to TSU_ADRH31)

TSU_ADRH0 to TSU_ADRH31 are entry tables referred by the CAM in reception and relay. This register sets the upper 32 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered. To refer to input signals on the CAMSEN0 and CAMSEN 1 pins, do not set the same MAC address set by this register to the entry tables of the external CAM.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ADRHn31 to ADRHn0 (n: 0 to 31)	All 0	R/W	MAC Address Bit These bits set the upper 32 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), H′01234567 is set to this register.

Notes: Set the CAM entry table as follows:

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC address by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC address by TSU_ADRL0 to TSU_ADRL31.

18.3.45 CAM Entry Table 0 to 31 L Registers (TSU_ADRL0 to TSU_ADRL31)

TSU_ADRL0 to TSU_ADRL31 are entry tables referred by the CAM in reception and relay. This register sets the lower 16 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered. To refer to input signals on the CAMSEN0 and CAMSEN 1 pins, do not set the same MAC address set by this register to the entry tables of the external CAM.

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 0	ADRLn15 to ADRLn0 (n: 0 to 31)	All 0	R/W	MAC Address Bit These bits set the lower 16 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), H′000089AB is set to this register.

Notes: Set the CAM entry table as follows:

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC address by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC address by TSU_ADRL0 to TSU_ADRL31.

18.3.46 Transmit Frame Counter Register (Port 0) (Normal Transmission Only) (TXNLCR0)

TXNLCR0 is a 32-bit counter indicating the number of frames successfully transmitted in MAC-0. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC031 to NTC000	All 0	R	Port 0 Transmit Frame Counter Bit These bits indicate the number of frames successfully transmitted.

18.3.47 Transmit Frame Counter Register (Port 0) (Normal and Error Transmission) (TXALCR0)

TXALCR0 is a 32-bit counter indicating the number of frames successfully transmitted and frames transmitted with error in MAC-0. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TC031 to TC000	All 0	R	Port 0 Transmit Frame Counter Bit These bits indicate the number of frames successfully transmitted and frames transmitted with error.

18.3.48 Receive Frame Counter Register (Port 0) (Normal Reception Only) (RXNLCR0)

RXNLCR0 is a 32-bit counter indicating the number of frames successfully received in MAC-0. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC031 to NRC000	All 0	R	Port 0 Receive Frame Counter Bit These bits indicate the number of frames successfully received.

18.3.49 Receive Frame Counter Register (Port 0) (Normal and Error Reception) (RXALCR0)

RXALCR0 is a 32-bit counter indicating the number of frames successfully received and frames received with error in MAC-0. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RC031 to RC000	All 0	R	Port 0 Receive Frame Counter Bit These bits indicate the number of frames successfully received and frames received with error.

18.3.50 Relay Frame Counter Register (Port 1 to 0) (Normal Relay Only) (FWNLCR0)

FWNLCR0 is a 32-bit counter indicating the number of frames successfully relayed in port 1 to 0 relay operations. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC031 to NFC000	All 0	R	Port 1 to 0 Relay Frame Counter Bit These bits indicate the number of frames successfully relayed.

18.3.51 Relay Frame Counter Register (Port 1 to 0) (Normal and Error Relay) (FWALCR0)

FWALCR0 is a 32-bit counter indicating the number of frames successfully relayed and frames relayed with error in port 1 to 0 relay operations. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FC031 to FC000	All 0	R	Port 1 to 0 Relay Frame Counter Bit These bits indicate the number of frames successfully relayed and frames relayed with error.

18.3.52 Transmit Frame Counter Register (Port 1) (Normal Transmission Only) (TXNLCR1)

TXNLCR1 is a 32-bit counter indicating the number of frames successfully transmitted in MAC-1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC131 to NTC100	All 0	R	Port 1 Transmit Frame Counter Bit These bits indicate the number of frames successfully transmitted.

18.3.53 Transmit Frame Counter Register (Port 1) (Normal and Error Transmission) (TXALCR1)

TXALCR1 is a 32-bit counter indicating the number of frames successfully transmitted and frames transmitted with error in MAC-1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TC131 to TC100	All 0	R	Port 1 Transmit Frame Counter Bit These bits indicate the number of frames successfully transmitted and frames transmitted with error.

18.3.54 Receive Frame Counter Register (Port 1) (Normal Reception Only) (RXNLCR1)

RXNLCR1 is a 32-bit counter indicating the number of frames successfully received in MAC-1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC131 to NRC100	All 0	R	Port 1 Receive Frame Counter Bit These bits indicate the number of frames successfully received.

18.3.55 Receive Frame Counter Register (Port 1) (Normal and Error Reception) (RXALCR1)

RXALCR1 is a 32-bit counter indicating the number of frames successfully received and frames received with error in MAC-1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RC131 to RC100	All 0	R	Port 1 Receive Frame Counter Bit These bits indicate the number of frames successfully received and frames received with error.

18.3.56 Relay Frame Counter Register (Port 0 to 1) (Normal Relay Only) (FWNLCR1)

FWNLCR1 is a 32-bit counter indicating the number of frames successfully relayed in port 0 to 1 relay operations. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC131 to NFC100	All 0	R	Port 0 to 1 Relay Frame Counter Bit These bits indicate the number of frames successfully relayed.

18.3.57 Relay Frame Counter Register (Port 0 to 1) (Normal and Error Relay) (FWALCR1)

FWALCR1 is a 32-bit counter indicating the number of frames successfully relayed and frames relayed with error in port 0 to 1 relay operations. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a read to this register. This register cannot be written.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FC131 to FC100	All 0	R	Port 0 to 1 Relay Frame Counter Bit These bits indicate the number of frames successfully relayed and frames relayed with error.

18.4 Operation

The following outlines the operations of the Ethernet controller (EtherC).

Automatic Ethernet Frame Transfer Function by Hardware: Each MAC controller can transmit and receive independently using two ports of MAC controllers. Furthermore, relay between the two MAC controllers can be performed by hardware using the on-chip TSU of the EtherC. TSU selects one of the following processes depending on the MAC address of the destination of the Ethernet frame input to the MAC controller according to on the settings of the CAM and registers TSU_FWSL0/1, and TSU_FWSLC; 1) reception, 2) relay, 3) reception and relay, and 4) discard. This setting can be performed independently for each port at the receive and relay sides by means of registers TSU_TEN and TSU_POST1 to TSU_POST4. It also has a 6-kbyte TSU FIFO for temporarily retaining the frames relayed. This TSU FIFO can vary capacity allotment with port 0 to 1 transfer and port 1 to 0 transfer using the TSU FIFO size select register (TSU_FCM).

TSU FIFO Overflow Prevention Function: By supporting relay operations, the MAC controller needs to transmit relay frames other than transmit frames requested by the E-DMAC normally. Arbitration is carried out between these two frames. The procedure of arbitration is specified by registers TSU_PRISL0 and TSU_PRISL1. It has a function which relays frames of the TSU FIFO with priority when the using rate of the TSU FIFO exceeds the value set by registers TSU_PRISL0 and TSU_PRISL1, thus preventing frame losses by TSU FIFO overflow.

QoS (IEEE802.1Q) Frame Transmit/Receive, Relay Function: QoS frames can be transmitted and received. At the using relay function, if the Ethernet device connected to one of the MAC controllers cannot transmit/receive QoS frames, this LSI can convert to the normal IEEE802.3 frames and relay it.

Figure 18.2 shows the data path and outline of various settings.

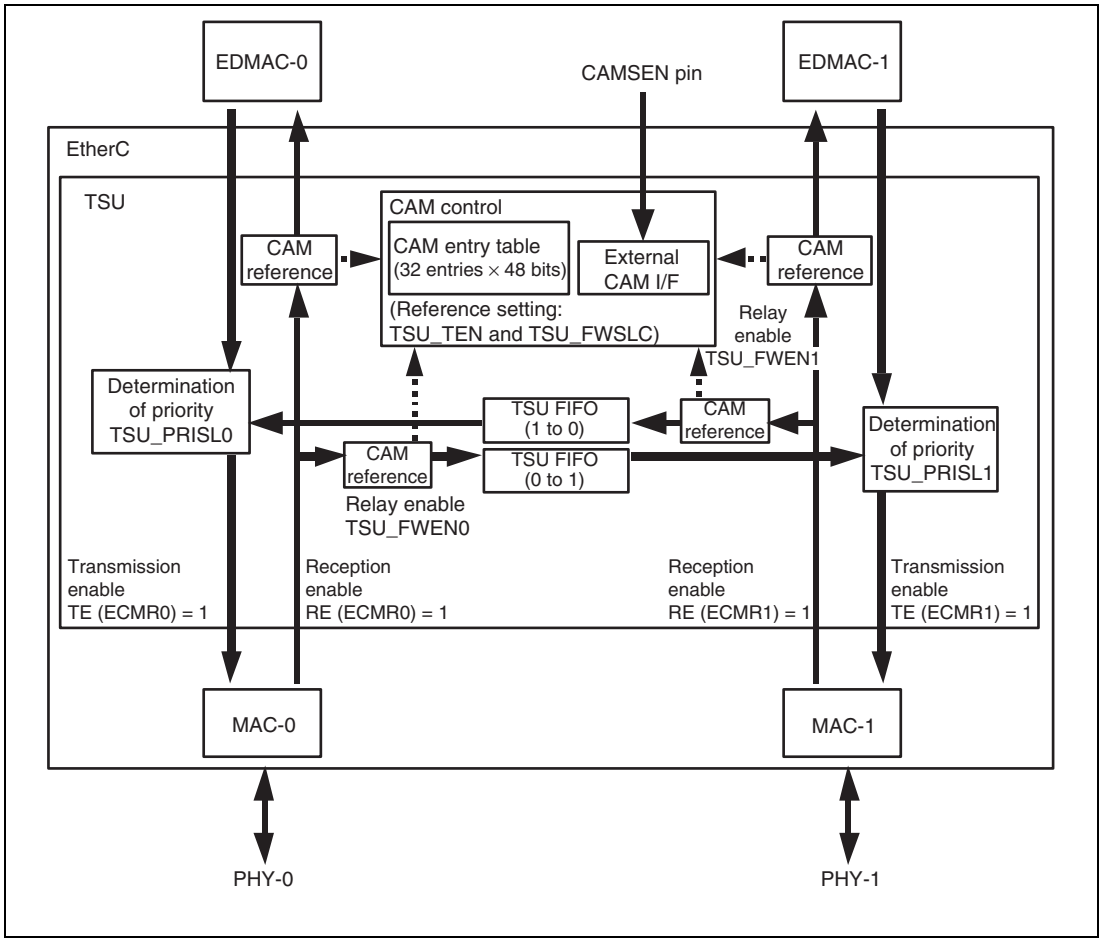
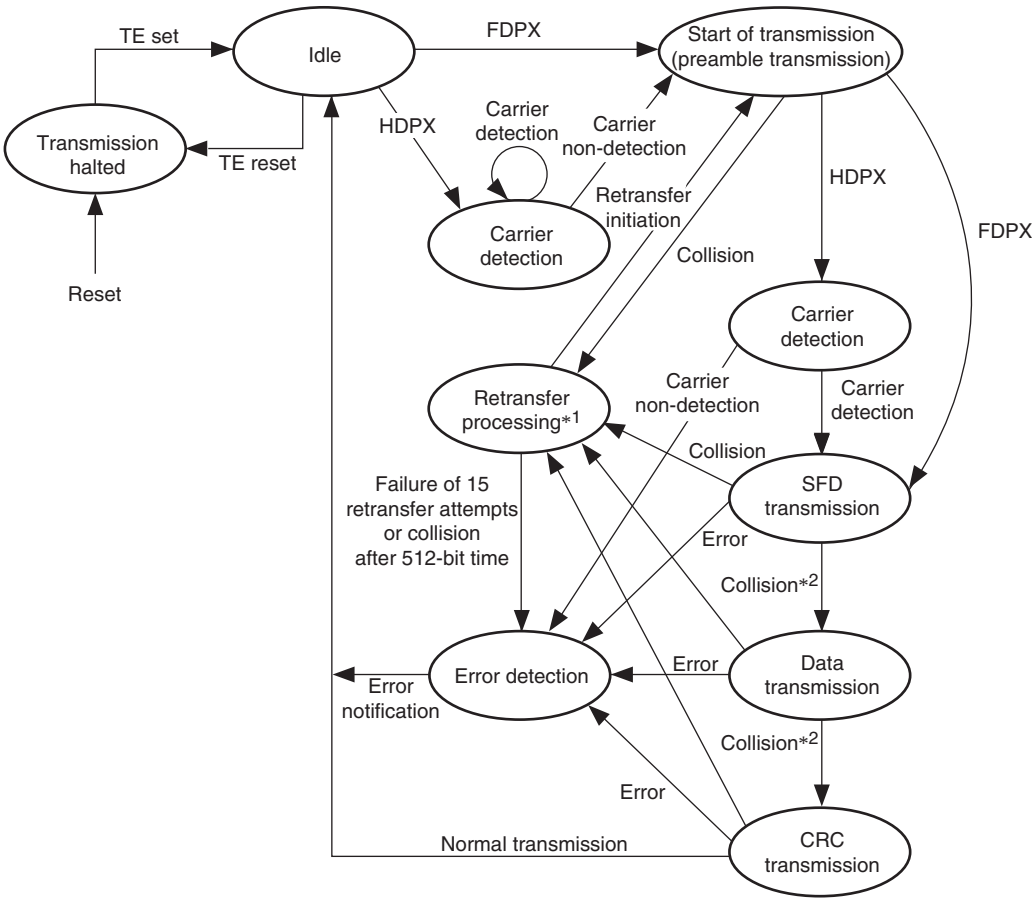


Figure 18.2 EtherC Data Path and Various Settings

18.4.1 Transmission

The EtherC transmitter assembles the transmit data on the frame and outputs to MII when there is a transmit request from the E-DMAC. The data transmitted via the MII is transmitted to the lines by PHY-LSI. Figure 18.3 shows the status change of the Ether-C transmitter. This operation is the same between ports 0 and 1. The priority of the process when transmit frame from E-DMAC and relay frame transmission collide can be set by the Transmit/Relay Priority Control Mode register (TSU_PRISL0/1).



Legend
FDPX: Full-duplex
HDPX: Half-duplex

Notes:
1. Transmission retry processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm.
2. Transmission is retried only when data of 512 bits or less (including the preamble and SFD)is transmitted. When a collision is detected during the transmission of data greater than 512 bits, only jam is transmitted and transmission based on the back-off algorithm is not retried.

Figure 18.3 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit E-DMAC, the EtherC sends the preamble after a transmission delay equivalent to the frame interval time. If full-duplex transfer is

selected, which does not require carrier detection, the preamble is sent as soon as a transmit request is issued by the E-DMAC.

3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit E-DMAC generates a transmission complete interrupt (TC). If a collision or the carrier-not-detected state occurs during data transmission, these are reported as interrupt sources.

4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmitting.

18.4.2 Reception

The EtherC receiver separates the frame from the MII into preamble, SFD, data and CRC, and the fields from DA (destination address) to the CRC data are transferred to the receive E-DMAC. Figure 18.4 shows the state transitions of the EtherC receiver. These operations are the same for ports 0 and 1. In frame processing during reception, CAM evaluation can be referenced. (When using the CAM function, refer to section 18.4.4, CAM Function.)

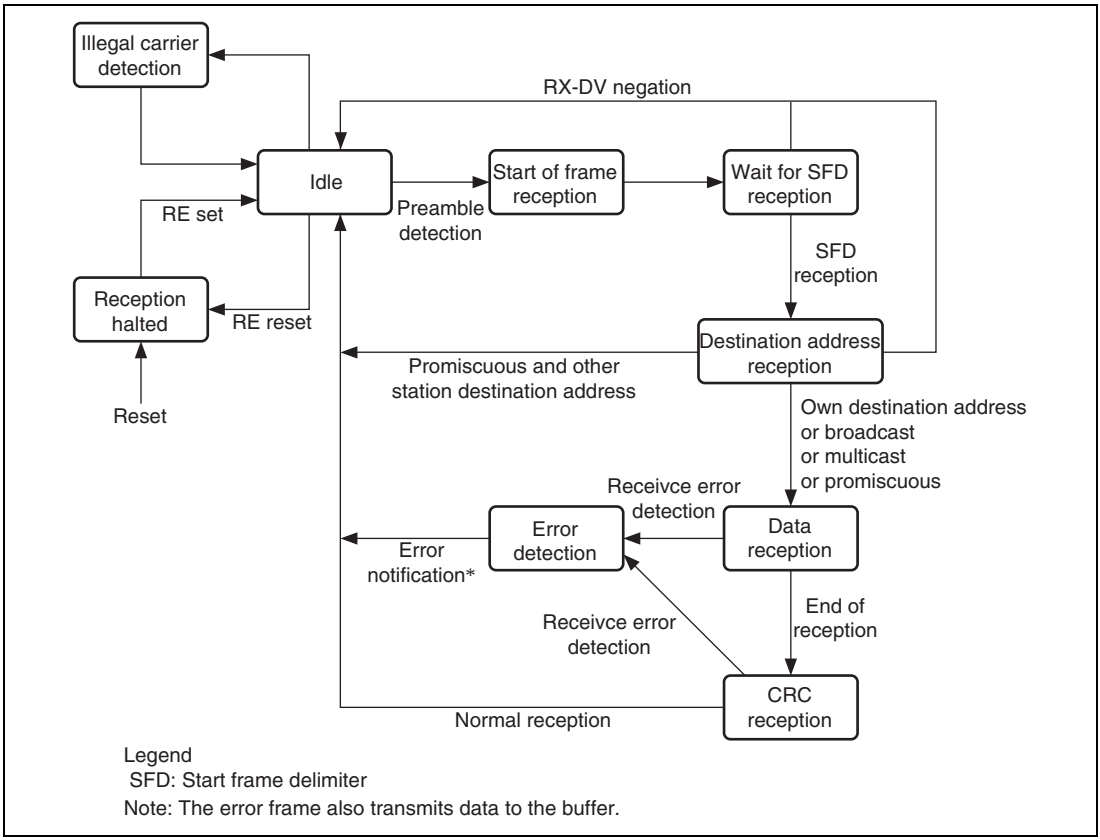


Figure 18.4 EtherC Receiver State Transmissions

1. When the receive enable (RE) bit is set, the receiver enters the receive idle state.
2. When an SFD (start frame delimiter) is detected after a receive packet preamble, the receiver starts receive processing. Discards a frame with an invalid pattern.
3. In normal mode, if the destination address matches the receiver's own address, or if broadcast or multicast transmission or promiscuous mode is specified, the receiver starts data reception.
4. Following data reception from the MII, the receiver carries out a CRC check. The result is indicated as a status bit in the descriptor after the frame data has been written to memory. Reports an error status in the case of an abnormality.
5. After one frame has been received, if the receive enable bit is set (RE = 1) in the EtherC mode register, the receiver prepares to receive the next frame.

18.4.3 Relay

EtherC has a function to relay frames received from the MII of either MAC-0 or MAC-1 to the other MAC. When relay is enabled, frames input from the MII are sent to both the TSU FIFO and receive E-DMAC, and determined independently whether to receive or not by the receive E-DMAC and whether to relay or not by the TSU. (Refer to figure 18.2.) To execute relay, specify both MAC controllers as promiscuous mode, and the same MAC address in both MAC controllers (hereafter this MAC address is referred to as MAC address of this LSI). The setting of the transfer frame processing (relayed/discarded) is carried by the TSU_FWSL0 and TSU_FWSL1. Frames passing the TSU FIFO during relaying are sent to the PHY_LSI from MAC-1 in MAC-0 to MAC-1 relay, from MAC-0 in MAC1 to MAC0 relay via the MII. At this time, collision with the relay frames from the E-DMAC may occur. The priority of the process when collision occurs can be set by TSU_PRISL0/1. For multicast frames and frames their destinations are other than this LSI, the CAM evaluation in frame relay processing can be referenced (for details on the CAM function, refer to section 18.4.4, CAM Function). Table 18.2 shows the settings of the relay frame processing (without CAM).

Table 18.2 Transfer Frame Processing (Without CAM)

Name	TSU-FWSL	Frame Processing
Frame for this LSI	FW40/1 = 0	Discarded
	FW40/1 = 1	Relayed
Broadcast frame	FW30/1 = 0	Discarded
	FW30/1 = 1	Relayed
Multicast frame	FW20/1 = 0	Discarded
	FW20/1 = 1	Relayed
Frames to destinations other than this LSI	FW10/1 = 0	Discarded
	FW10/1 = 1	Relayed

18.4.4 CAM Function

Frames input to the MAC are grouped into the following four types; unicast for this LSI, broadcast, multicast, and unicast to other destinations. Of this, the MAC addresses of unicast for this LSI and broadcast are fixed, and determination is carried out only by register settings. Consequently, only multicast and unicast to other destinations determine whether to receive or not and whether to relay or not by using the CAM (unicast frames whose destination MAC addresses match this LSI are called unicast frames to this LSI, and those that do not are called unicast frames to other destinations).

Furthermore, with the EtherC, the evaluation of receive and relay of unicast to other destinations and multicast frames by using CAM are performed by referencing the registered MAC addresses of the CAM entry table in the EtherC and the CAM logic connected externally via the CAMSEN0 and CAMSEN1 pins. By using this function, receive FIFO overflow can be prevented caused by accumulation of frame data not required for reception, and CPU processing for determining receive can be reduced.

The POST table is composed of 4 bits, and each bit corresponds to port 0 reception, port 1 reception, port 0 to 1 relay, and port 1 to 0 relay. When the corresponding bit is set to 1, the CAM evaluation results are used for determining receive and relay. In other words, when the corresponding bit of the POST table is cleared to 0, receive and relay evaluation will be the same as when CAM is not used shown in table 18.2. The difference between the on-chip CAM entry table and externally connected CAM logic lies in how the POST table is set. In the internal CAM entry table, there are 32 POST tables (same as the number of entries) and the POST table can be set for each entry. The internal CAM entry table has 32 entries and 32 POST tables, and the POST table can be specified in each entry. The external connection CAM logic configuration is based on pins because POST tables (total of 2) are allocated to the CAMSEN0 and CAMSEN1 pins.

When On-Chip CAM Entry Table is Used: The on-chip CAM has entry tables which can register the MAC address of 32 entries, the details of which can be set by TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31. The setting to enable/disable referencing of the on-chip CAM entry table is carried out by the CAM entry table enable setting register which sets whether to perform CAM evaluation or not, and the CAM entry table POST setting register for setting whether to use the CAM determination results for determining receive or relay. When on-chip CAM entry table referencing during receive is enabled, the destination address in the frame and MAC address registered in the CAM entry table are compared, and it is determined whether to transfer the frames input to the MAC to E-DMAC (have E-DMAC receive the frames) or discard the frames. When relaying and CAM entry table referencing during relay are both enabled, whether to transfer or discard multicast frames and frames for destinations other than this LSI can be determined by comparing the destination address in the frame and MAC address registered in the CAM entry table. Table 18.3 shows the processing method of frames (receive or discard) in MAC0 to E-DMAC0 and MAC1 to E-DMAC1 reception, while table 18.4 shows the processing for frames in MAC0 to MAC1 and MAC1 to MAC0 relay (relay or discard).

Table 18.3 Reception Frame Process

CAM Entry Table Referencing Results	Frame	Normal Mode		Promiscuous Mode	
		MCT = 0	MCT = 1	MCT = 0	MCT = 1
CAM hit (when addresses match)	Frame to this LSI	Discarded		Discarded	
	Broadcast frame	Discarded		Discarded	
	Multicast frame	Discarded	Received	Discarded	Received
	Frames to destinations other than this LSI	Received		Discarded	
CAM mishit (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Discarded
	Frames to destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Receive when CAM mishit/
1: Receive when CAM hit)

Table 18.4 Relay Frame Process (With CAM)

Frame	Relay Function Setting Register Bit	CAM Hit	CAM Mishit
Multicast frame	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed
Frames to destinations other than this LSI	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed

Note: CAM can be referenced only for multicast frames and frames to destinations other than this LSI. The processing of frames to this LSI and broadcast frames conforms to the values of the relay function setting register regardless of CAM reference.

When External CAM Logic is Used: In addition to the on-chip CAM entry table, use of the CAMSEN0 and CAMSEN1 pins allows referencing of evaluation results of the external CAM logic connected externally to this LSI for frame processing evaluation. This function externally connects the CAM logic for comparing the destination address in receive frames, and receives the results of comparing destination addresses corresponding to the signals (RXD3 to RXD0) input from the MII to determine whether to receive or discard the corresponding frame. Figure 18.5

shows the connection example of the external CAM logic while figure 18.6 shows the timing conditions of the external CAM signal.

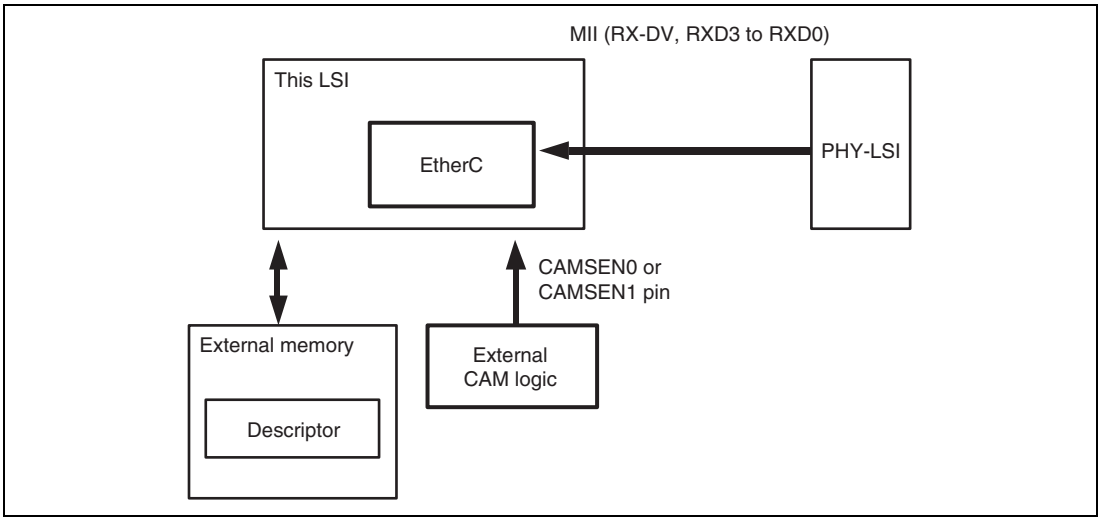


Figure 18.5 Example of External CAM Connection

The setting on whether to enable or disable the referencing of external CAM logic evaluation results by the CAMSEN0 and CAMSEN1 pins is carried out by the transfer function setting register (common) (TSU_FWSLC). When referencing of the CAMSEN0 and CAMSEN1 pins is enabled during receive, it is determined whether to send or discard the frames input from to MAC-0/1 to E-DMAC0/1 (have E-DMAC receive the frames) according to the value of the CAMSEN0 or CAMSEN1 pin. When relaying and CAMSEN0/1 pin referencing are enabled at the same time, the transfer or discard of multicast frames and frames to destinations other than this LSI can be determined by the value of the CAMSEN0 and CAMSEN1 pins.

Table 18.5 shows the processing method (receive or discard) for frames in MAC0 to E-DMAC0 or MAC1 to E-DMAC1 reception, while Table 18.6 shows the processing method (receive or discard) for frames in MAC0 to MAC1 or MAC1 to MAC relay. The external CAM logic is memorized with MAC addresses different from the CAM entry table in this LSI. When the MAC address received from the PHY matches the destination address memorized in the external CAM logic, the CAMSEN0 or CAMSEN1 pin is asserted*. EtherC receives or discards the frames when CAMSEN0/1 was asserted according to the settings in table 18.5.

Figure 18.6 shows the valid range of CAMSEN0/1 assertion for the corresponding receive frames.

With EtherC, before storage of receive frames in the FIFO of E-DMAC/TSU is started, there is a need to determine receive frame processing. The time limit for determining this processing is within 52 clocks from RX_DV assertion.

Note: * Do not memorize MAC addresses overlapping with the internal CAM entry table of this LSI during external CAM logic. If the CAMSEN0 or CAMSEN1 pin is asserted at the same time as CAM hit occurs for the internal CAM entry table, evaluation may not performed correctly.

Table 18.5 Receive Frame Process (When External CAM Logic is Used)

CAMSEN0 or CAMSEN1 Pin	Frame	Normal Mode		Promiscuous Mode	
		MCT = 0	MCT = 1	MCT = 0	MCT = 1
Assertion (when addresses match)	Frame to this LSI	Discarded		Discarded	
	Broadcast frame	Discarded		Discarded	
	Multicast frame	Discarded	Received	Discarded	Received
	Frames to destinations other than this LSI	Received		Discarded	
Negation (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Discarded
	Frames to destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Received when CAM mishit/
1: Received when CAM hit)

Table 18.6 Relay Frame Process (When External CAM Logic is Used)

Frame	Relay Function Setting Register bit	CAMSEN0 or CAMSEN1 Pin Assertion	CAMSEN0 or CAMSEN1 Pin Negation
Multicast frame	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed
Frames to destinations other than this LSI	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed

Note: CAM can be referenced only for multicast frames and frames to destinations other than this LSI. The processing of frames to this LSI and broadcast frames conforms to the values of the relay function setting register regardless of CAM reference.

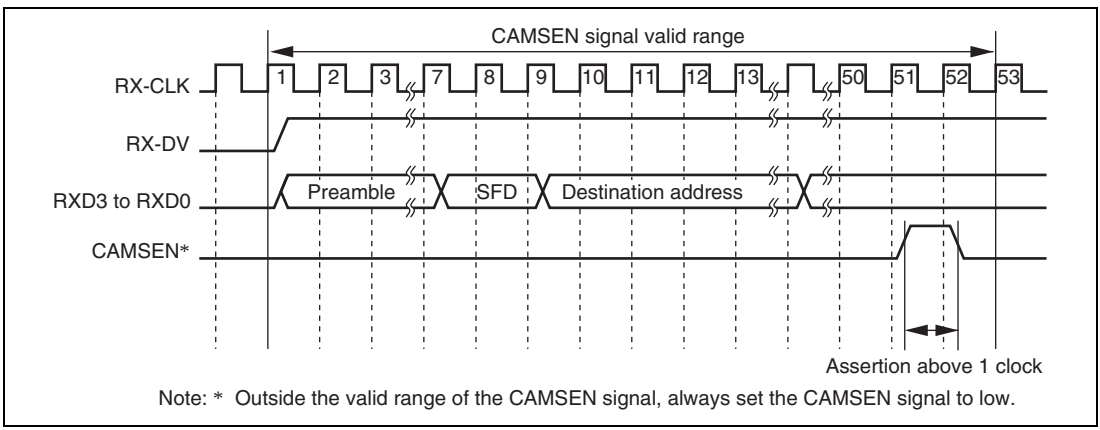


Figure 18.6 External CAM Signal Timing

18.4.5 MII Frame Timing

Each MII Frame timing is shown in figure 18.7.

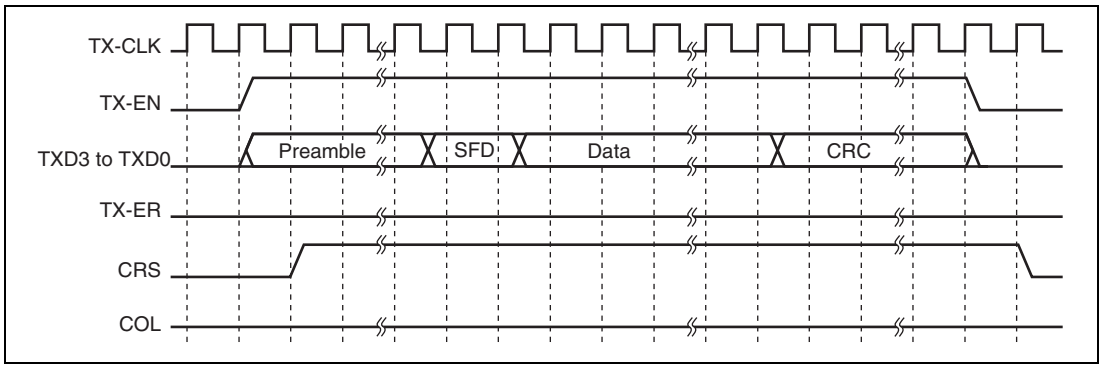


Figure 18.7 (1) MII Frame Transmit Timing (Normal Transmission)

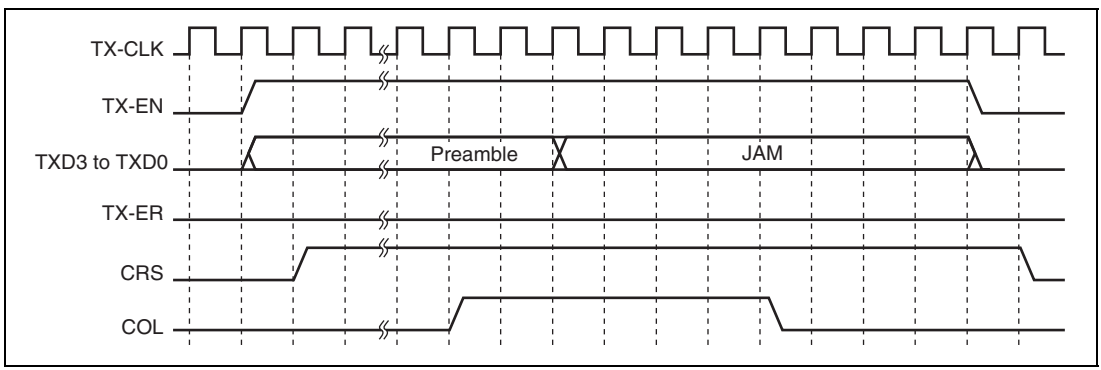


Figure 18.7 (2) MII Frame Transmit Timing (Collision)

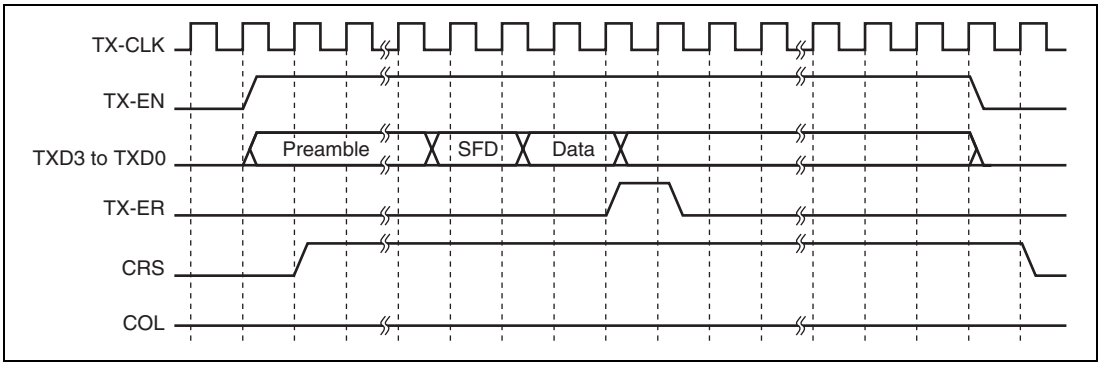


Figure 18.7 (3) MII Frame Transmit Timing (Transmit Error)

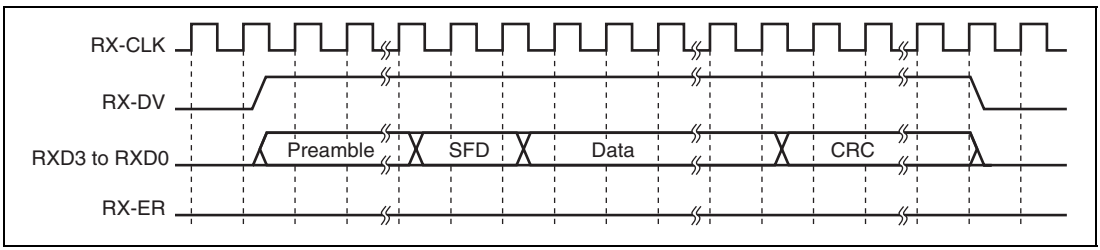


Figure 18.7 (4) MII Frame Receive Timing (Normal Reception)

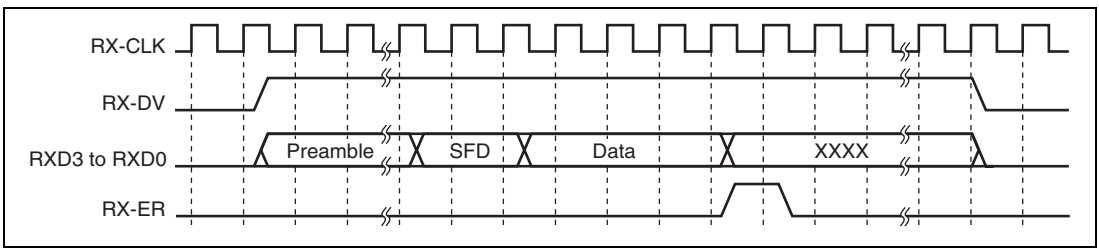


Figure 18.7 (5) MII Frame Receive Timing (Reception Error (1))

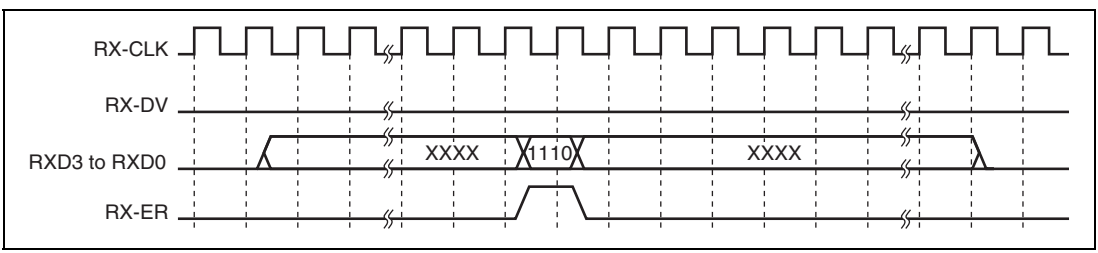


Figure 18.7 (6) MII Fame Receive Timing (Reception Error (2))

18.4.6 Accessing MII Registers

MII registers in the PHY-LSI are accessed via this LSI's PHY interface register (PIR). Connection is made as a serial interface in accordance with the MII frame format specified in IEEE802.3u.

MII Management Frame Format: The format of an MII management frame is shown in figure 18.8. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]
PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI address.
REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(b) Read: Bus already released in TA; control unnecessary

Figure 18.8 MII Management Frame Format

MII Register Access Procedure: The program accesses MII registers via the PHY interface register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 18.9 shows the MII register access timing. The timing will differ depending on the PHY-LSI type.

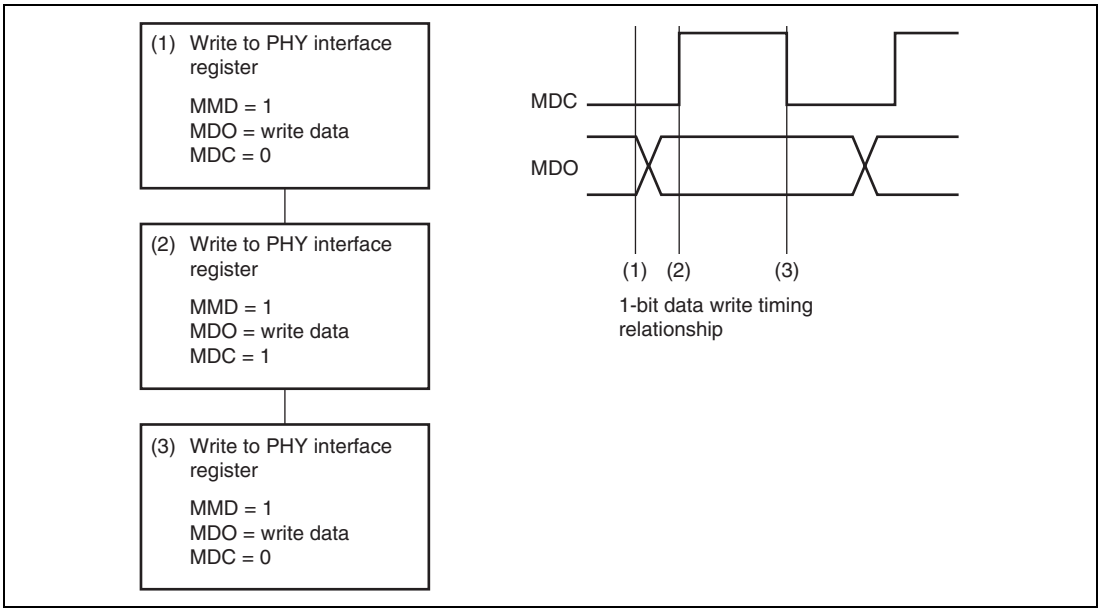
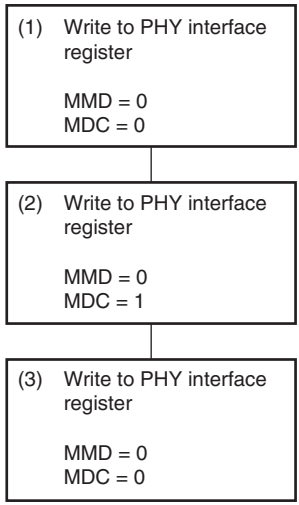


Figure 18.9 (1) 1-Bit Data Write Flowchart



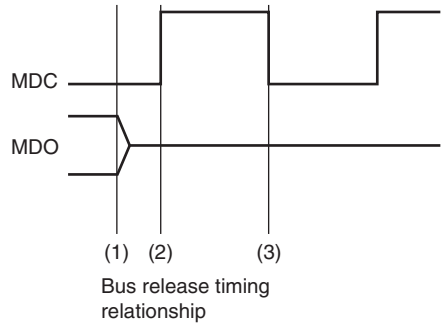
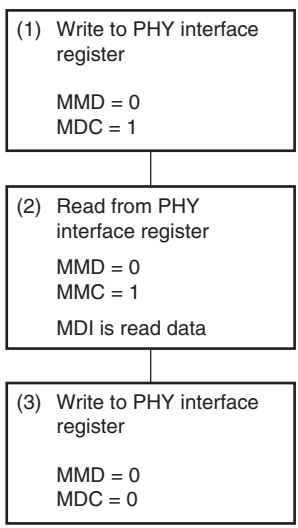


Figure 18.9 (2) Bus Release Flowchart (TA in Read in Figure 18.8)



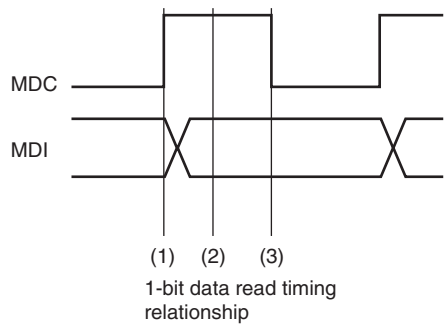


Figure 18.9 (3) 1-Bit Data Read Flowchart

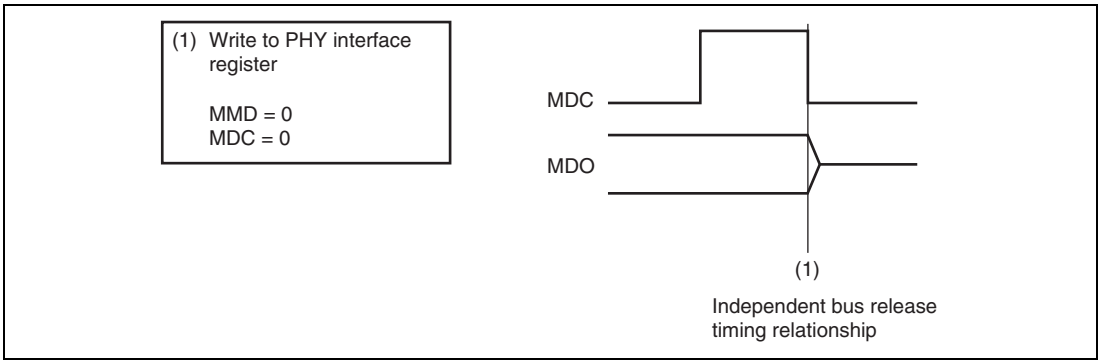


Figure 18.9 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 18.8)

18.4.7 Magic Packet Detection

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

With a Magic Packet, reception is performed regardless of the destination address. As a result, this function is valid, and the WOL pin enabled, only in the case of a match with the destination address specified by the format in the Magic Packet. Further information on Magic Packets can be found in the technical documentation published by AMD Corporation.

The procedure for using the WOL function with this LSI is as follows.

1. Disable interrupt source output by means of the various interrupt enable/mask registers.
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECMR).
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt enable register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to module standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin notifies peripheral LSIs that the Magic Packet has been detected.

18.4.8 Operation by IPG Setting

The EtherC has a function to change the non-transmission period IPG (Inter Packet Gap) between transmit frames. By changing the set values of the IPG setting register (IPGR), the transmission efficiency can be raised and lowered from the standard value. IPG settings are prescribed in IEEE802.3 standards. When changing settings, adequately check that the respective devices can operate smoothly on the same network.

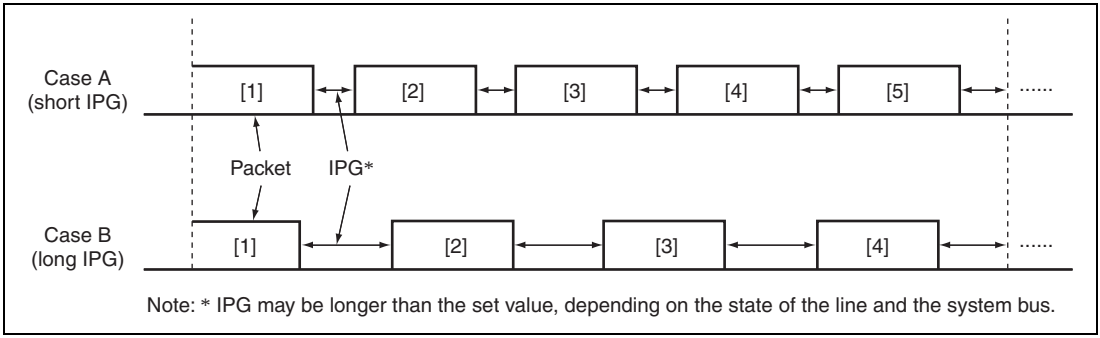


Figure 18.10 Changing IPG and Transmission Efficiency

18.4.9 Direction for IEEE802.1Q Qtag

The EtherC supports IEEE802.1Q frame processing. It can add or delete Qtags to or from frames processed in relay. This function can also transmit and receive QoS frames. During relaying, if the Ethernet device connected to one MAC controller cannot transmit or receive QoS frames, it can be converted to the normal IEEE802.3 frames and relayed in this LSI. Whether to add or delete Qtags is determined by the added Qtag value set register (TSU_ADQT0/1). Figure 18.11 shows an outline of the Qtag add function while figure 18.12 shows the comparison between the normal Ethernet frames and IEEE802.1Q frames (with Qtag). For details on setting Qtag, refer to the specifications on Qtag control specified in IEEE802.1Q.

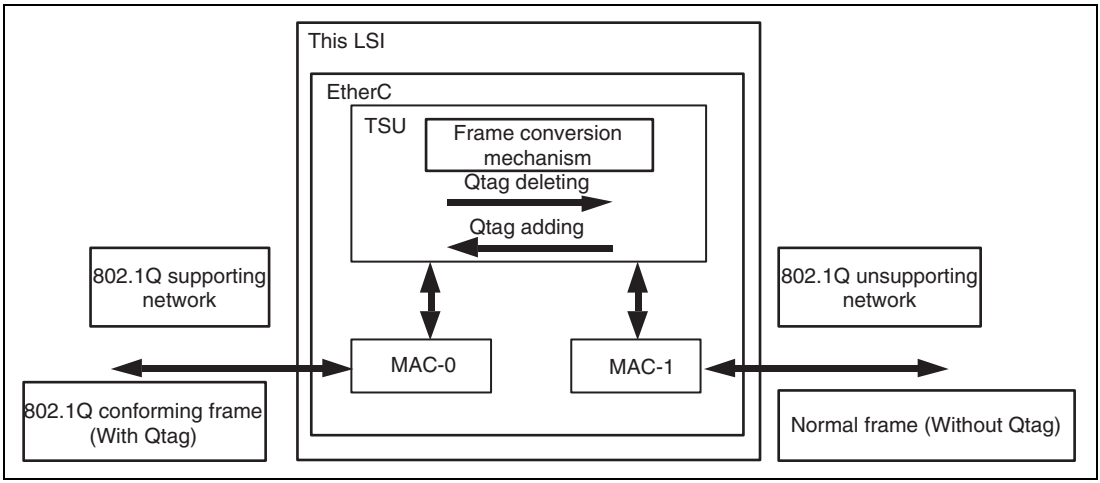


Figure 18.11 Diagram of Qtag Additional Functions

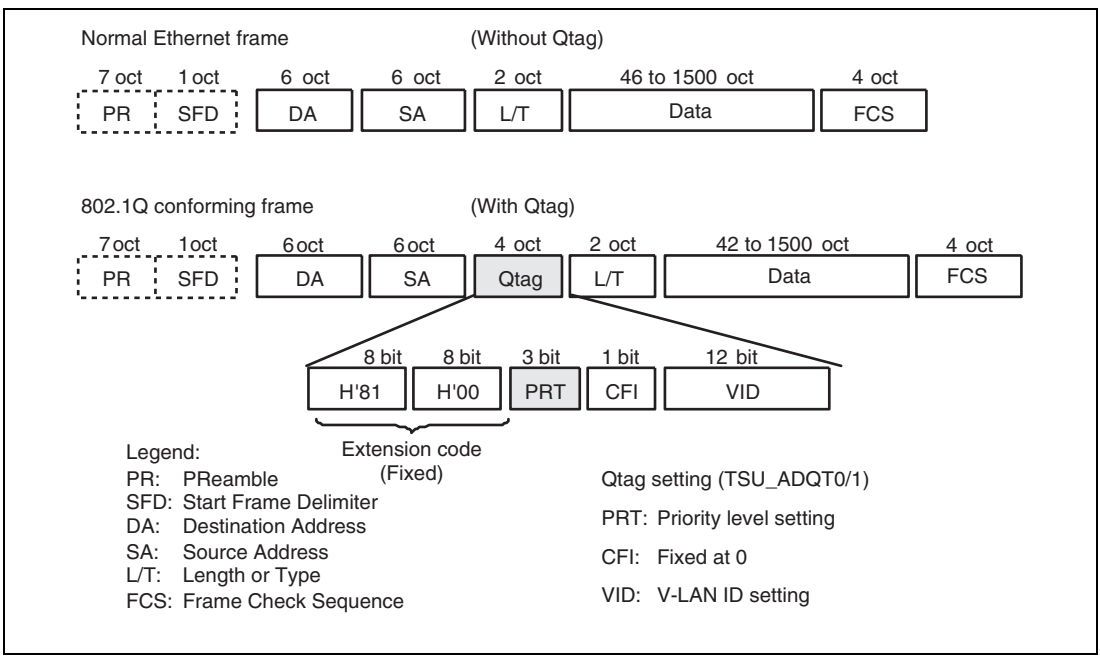


Figure 18.12 Comparison of Normal Ethernet Frame and IEEE802.1Q Frame (with Qtag)

18.5 Connection to LSI

Figure 18.13 shows the example of connection to a DP83847 (National Semiconductor Corporation).

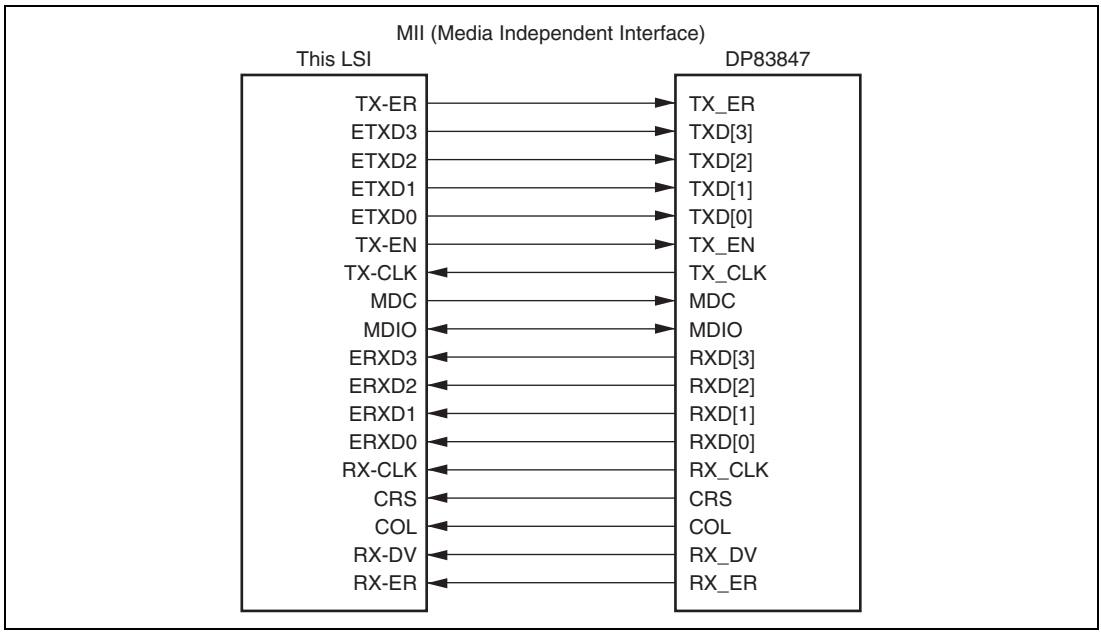


Figure 18.13 Example of Connection to DP83847

Section 19 Ethernet Controller Direct Memory Access Controller (E-DMAC)

This LSI has an on-chip two-channel direct memory access controller (E-DMAC0/1) directly connected to the Ethernet controller (EtherC). By using the DMAC contained in the E-DMAC, the E-DMAC transfers transmit/receive data between the transmit/receive FIFO in the E-DMAC and a user-specified data storage destination (buffer) by DMA transfer. At DMA transfer, information referenced by the E-DMAC is referred to as a transmit/receive descriptor, and the user places this descriptor in memory.

This function reduces the load on the CPU and enables efficient data transfer control to be achieved. The E-DMAC0 and E-DMAC1 control the data transmission/reception from the MAC-0 and MAC-1 of EtherC respectively. (Hereafter the channel controlled by the E-DMAC0 is channel 0. The channel controlled by the E-DMAC1 is channel 1.)

Figure 19.1 shows the configuration of the E-DMAC, and the descriptors and transmit/receive buffers in memory.

19.1 Features

The E-DMAC has the following features:

- Contains two-channel independent transmit/receive DMAC
- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of DMA block transfer (16-byte units)
- Supports single-frame/single-descriptor operation and single-frame/multi-frame (multi-buffer) operation

Note: The E-DMAC cannot access peripheral modules.

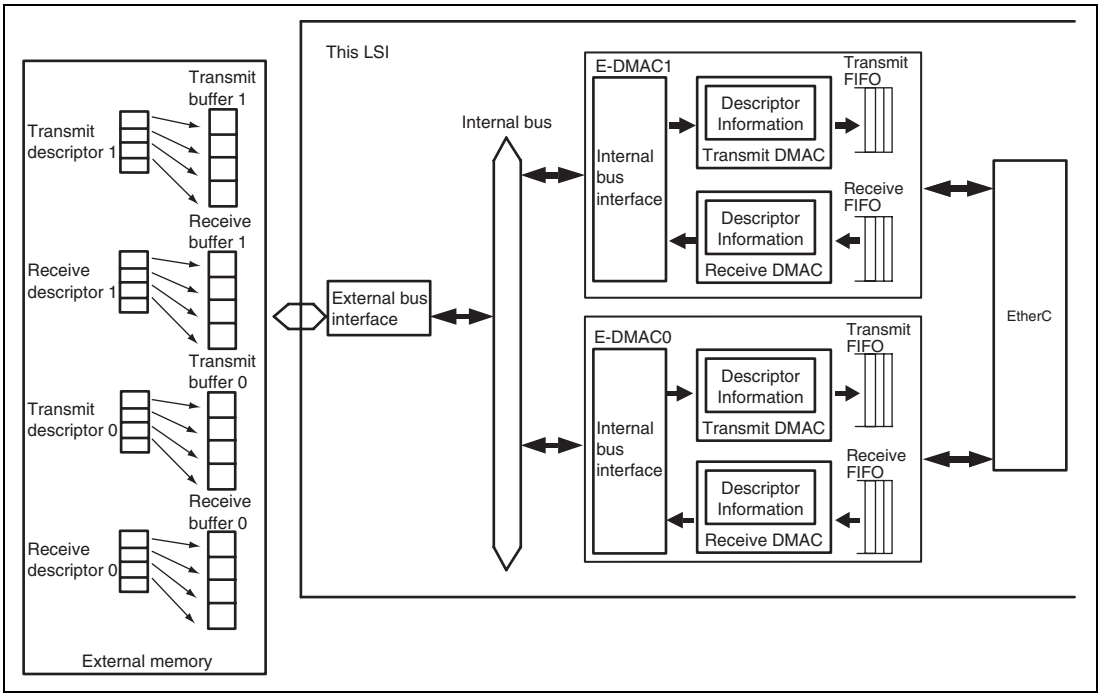


Figure 19.1 Configuration of E-DMAC, and Descriptors and Buffers

19.2 Register Descriptions

The E-DMAC has the following registers. The number at the end of the register abbreviation represents the number of corresponding E-DMAC (E-DMAC0 or E-DMAC1). In this section, some numbers are not mentioned. For addresses and access sizes of these registers, see section 24, List of Registers.

Channel 0:

- E-DMAC mode register (EDMR0)
- E-DMAC transmit request register (EDTRR0)
- E-DMAC receive request register (EDRRR0)
- Transmit descriptor list address register (TDLAR0)
- Receive descriptor list address register (RDLAR0)
- EtherC/E-DMAC status register (EESR0)
- EtherC/E-DMAC status interrupt permission register (EESIPR0)
- Transmit/receive status copy enable register (TRSCER0)

- Receive missed-frame counter register (RMFCR0)
- Transmit FIFO threshold register (TFTR0)
- FIFO depth register (FDR0)
- Receiving method control register (RMCR0)
- E-DMAC operation control register (EDOCR0)
- Receive buffer write address register (RBWAR0)
- Receive descriptor fetch address register (RDFAR0)
- Transmit buffer read address register (TBRAR0)
- Transmit descriptor fetch address register (TDFAR0)
- Overflow alert FIFO threshold register (FCFTR0)
- Transmit interrupt register (TRIMD0)

Channel 1:

- E-DMAC mode register (EDMR1)
- E-DMAC transmit request register (EDTRR1)
- E-DMAC receive request register (EDRRR1)
- Transmit descriptor list address register (TDLAR1)
- Receive descriptor list address register (RDLAR1)
- EtherC/E-DMAC status register (EESR1)
- EtherC/E-DMAC status interrupt permission register (EESIPR1)
- Transmit/receive status copy enable register (TRSCER1)
- Receive missed-frame counter register (RMFCR1)
- Transmit FIFO threshold register (TFTR1)
- FIFO depth register (FDR1)
- Receiving method control register (RMCR1)
- E-DMAC operation control register (EDOCR1)
- Receive buffer write address register (RBWAR1)
- Receive descriptor fetch address register (RDFAR1)
- Transmit buffer read address register (TBRAR1)
- Transmit descriptor fetch address register (TDFAR1)
- Overflow alert FIFO threshold register (FCFTR1)
- Transmit interrupt register (TRIMD1)

19.2.1 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies E-DMAC resetting and transmit/receive descriptor length. This register is to be set before the TR bit in EDTRR or the RR bit in EDRRR is set to 1. If a software reset is executed with this register during data transmission, abnormal data may be transmitted on the line. Execute a software reset with this register before specifying transmit/receive descriptor length and modifying the settings of TDLAR, RDLAR, and so forth, the setting of ECMR (EtherC mode register), and the settings of registers related to E-DMAC and EtherC operation. The time required for completion of EtherC and E-DMAC initialization from a software reset with this register is 64 cycles of the internal bus clock Bϕ. Therefore, registers of the EtherC and E-DMAC should be accessed after 64 cycles of the internal bus clock Bϕ has elapsed.

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	DL1	0	R/W	Descriptor Length
4	DL0	0	R/W	These bits specify the descriptor length. (See section 19.3.1, Descriptors and Descriptor List.) 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Reserved (setting prohibited)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	SWR	0	R/W	Software Reset By writing 1 to this bit, the registers of the E-DMAC other than TDLAR, RDLAR, and RMFCR and the registers of EtherC other than TSU-related registers can be initialized. (The registers whose names start with TSU_ are not initialized.) The SWR bit in EDMR0 initializes the EDMAC0 and MAC-0 registers in the EtherC. The SWR bit in EDMR1 initializes EDMAC1 and MAC-1 registers in the EtherC. When transfer operation is enabled by specifying the relay enable register (Port 0 to 1) (TSU_FWEN0) and the relay enable register (Port 1 to 0) (TSU_FWEN1) in the EtherC, software reset should not be performed by using this bit. While a software reset is issued (64 cycles of the internal bus clock Bϕ), accesses to the all Ethernet-related registers are prohibited. Software reset period (example): When Bϕ = 100 MHz: 0.64 μS When Bϕ = 66 MHz: 0.97 μS When Bϕ = 50 MHz: 1.28 μS When Bϕ = 33 MHz: 1.94 μS This bit is always read as 0. 1: EtherC and E-DMAC are reset (when writing)

19.2.2 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. After writing 1 to the TR bit in this register, the E-DMAC reads the transmit descriptor at the address specified by TDLAR. If the TACT bit of this descriptor is set to 1 (valid), transmit DMA transfer by the E-DMAC starts. When DMA transfer based on the first transmit descriptor is completed, the E-DMAC reads the next transmit descriptor. If the TACT bit of that descriptor is set to 1 (valid), the E-DMAC continues transmit DMA operation. If the TACT bit of a transmit descriptor is cleared to 0 (invalid), the E-DMAC clears the TR bit and stops transmit DMAC operation.

For details of writing to the TR bit, see section 19.4.1, Using of EDTRR and EDRRR.

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	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does not stop transmission. Termination of transmission is controlled by the TACT bit of the transmit descriptor. 1: Transmit DMA operation being performed by the E-DMAC. After writing 1 to this bit, the E-DMAC starts reading a transmit descriptor.

19.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. After writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor at the address specified by RDLAR. If the RACT bit of this descriptor is set to 1 (valid), and the receive FIFO holds a receive frame, the E-DMAC starts receive DMA transfer. When DMA transfer based on the first receive descriptor is completed, the E-DMAC reads the next receive descriptor. If the RACT bit of that descriptor is set to 1 (valid), the E-DMAC continues receive DMA operation. However, if the receive FIFO holds no receive data, the E-DMAC places receive DMA operation in the standby state. If the RACT bit of the receive descriptor is cleared to 0 (invalid), the E-DMAC clears the RR bit and stops receive DMAC operation.

For details of writing to the RR bit, see section 19.4.1, Using of EDTRR and EDRRR.

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	RR	0	R/W	Receive Request 0: The receive function is disabled* 1: A receive descriptor is read, and the E-DMAC is ready to receive

Note: * If the receive function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the E-DMAC can not operate successfully. In this case, to make the E-DMAC reception enabled again, execute a software reset by the SWR bit in EDMR0 (EDMR1). To make the E-DMAC reception disabled without executing a software reset, specify the RE bit in ECMR0 (ECMR1). Next, after the E_DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receive function of this register.

19.2.4 Transmit Descriptor List Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transmit descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bit in EDMR. This register must not be written to during transmission. Modifications to this register should only be made while transmission is disabled by the TR bit (= 0) in the E-DMAC transmit request register (EDTRR).

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA31 to TDLA0	All 0	R/W	Transmit Descriptor Start Address The lower bits are set as follows according to the specified descriptor length. 16-byte boundary: TDLA3 and TDLA0 = 0000 32-byte boundary: TDLA4 and TDLA0 = 00000 64-byte boundary: TDLA5 and TDLA0 = 000000

19.2.5 Receive Descriptor List Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bit in EDMR. This register must not be written to during reception. Modifications to this register should only be made while reception is disabled by the RR bit (= 0) in the E-DMAC Receive Request Register (EDRRR).

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDLA31 to RDLA0	All 0	R/W	Receive Descriptor Start Address The lower bits are set as follows according to the specified descriptor length. 16-byte boundary: RDLA3 and RDLA0 = 0000 32-byte boundary: RDLA4 and RDLA0 = 00000 64-byte boundary: RDLA5 and RDLA0 = 000000

19.2.6 EtherC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that shows communications status information on the E-DMAC in combination with the EtherC. The information in this register is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a read-only bit and not to be cleared by writing 1) and are not affected by writing 0. Each interrupt source can also be masked by means of the corresponding bit in the EtherC/E-DMAC status interrupt permission register (EESIPR).

The interrupts generated by this register are EINT0 for channel 0 and EINT1 for channel 1. For interrupt priorities, see section 8, Interrupt Controller (INTC) and section 8.3.5, Interrupt Exception Handling and Priority.

The EINT2 is an interrupt generated by the TSU_FNSR in the EtherC.

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.

Bit	Bit Name	Initial Value	R/W	Description
30	TWB	0	R/W	Write-Back Complete Indicates that write-back from the E-DMAC to the corresponding descriptor has completed. This operation is enabled when the TIS bit in TRIMD is set to 1. 0: Write-back has not completed, or no transmission directive 1: Write-back has completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABT	0	R/W	Transmit Abort Detection Indicates that the EtherC aborts transmitting a frame because of failures during transmitting the frame. 0: Frame transmission has not been aborted or no transmit directive 1: Frame transmit has been aborted
25	RABT	0	R/W	Receive Abort Detection Indicates that the EtherC aborts receiving a frame because of failures during receiving the frame. 0: Frame reception has not been aborted or no receive directive 1: Frame receive has been aborted
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the receive FIFO frame counter has overflowed. 0: Receive frame counter has not overflowed 1: Receive frame counter overflows

Bit	Bit Name	Initial Value	R/W	Description
23	ADE	0	R/W	Address Error Indicates that the memory address that the E-DMAC tried to transfer is found illegal. 0: Illegal memory address not detected (normal operation) 1: Illegal memory address detected Note: When an address error is detected, the E-DMAC halts transmitting/receiving. To resume the operation, execute a software reset with the SWR bit in EDMR.
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source of an ECSR interrupt in the EtherC is cleared, this bit is also cleared. 0: EtherC status interrupt source has not been detected 1: EtherC status interrupt source has been detected
21	TC	0	R/W	Frame Transmit Complete Indicates that all the data specified by the transmit descriptor has been transmitted from the EtherC. This bit is set to 1, assuming the completion of transmission, when transmission of one frame is completed in single-frame/single-descriptor operation or when the last data of a frame has been transmitted and the transmit descriptor valid bit (TACT) of the next descriptor is not set in for the processing of multi-buffer frame based on single-frame/multi-descriptor operation. After frame transmission, the E-DMAC writes the transmission status back to the relevant descriptor. 0: Transfer not complete, or no transfer directive 1: Transfer complete

Bit	Bit Name	Initial Value	R/W	Description
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmit descriptor valid bit (TACT) of a transmit descriptor read by the E-DMAC is not set if the previous descriptor does not represent the end of a frame for the processing of multi-buffer frame based on the single-frame/multi-descriptor. As a result, an incomplete frame may be transmitted. 0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected When transmission descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, the address that is stored in the transmit descriptor list address register (TDLAR) is transmitted first.
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred
18	FR	0	R/W	Frame Reception Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received. 0: Frame not received 1: Frame received

Bit	Bit Name	Initial Value	R/W	Description
17	RDE	0	R/W	Receive Descriptor Empty Indicates that the RACT bit of a receive descriptor read by the E-DMAC for receive DMA is cleared to 0 (invalid). When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RACT bit (cleared to 0) of the receive descriptor to 1 and writing 1 to the RR bit in EDRRR. 0: Receive descriptor active bit RACT = 1 detected 1: Receive descriptor active bit RACT = 0 detected
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowed during frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status during preamble transmission. 0: A carrier is detected when transmission starts 1: A carrier is not detected
10	DLC	0	R/W	Detect Loss of Carrier Indicates that loss of the carrier has been detected during frame transmission. 0: Loss of carrier not detected 1: Loss of carrier detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been detected during frame transmission. 0: Delayed collision not detected 1: Delayed collision detected

Bit	Bit Name	Initial Value	R/W	Description
8	TRO	0	R/W	Transmit Retry Over Indicates that a retry-over condition has occurred during frame transmission. Total 16 transmission retries including 15 retries based on the back-off algorithm are failed after the EtherC transmission starts. 0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected
7	RMAF	0	R/W	Receive Multicast Address Frame 0: Multicast address frame has not been received 1: Multicast address frame has been received
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that the frame more than the number of receive frame length upper limit set by RFLR has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error not detected 1: PHY-LSI receive error detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error not detected 1: CRC error detected

19.2.7 EtherC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the EtherC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

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	TWBIP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Enable 0: Transmit abort detection interrupt is disabled 1: Transmit abort detection interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detection Interrupt Enable 0: Receive abort detection interrupt is disabled 1: Receive abort detection interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Enable 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	ADEIP	0	R/W	Address Error Interrupt Enable 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Interrupt Enable 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Enable 0: Frame transmit complete interrupt is disabled 1: Frame transmit complete interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Enable 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Received Interrupt Enable 0: Frame received interrupt is disabled 1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Receive FIFO overflow interrupt is disabled 1: Receive FIFO overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Enable 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Enable 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Enable 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Enable 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Enable 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Enable 0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Enable 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Enable 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Enable 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame 0: CRC error on received frame interrupt is disabled 1: CRC error on received frame interrupt is enabled

19.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER indicates whether multicast address frame receive status information reported by bit 7 in EESR is reflected in the RFE bit in the corresponding descriptor (for details of descriptor descriptions, see section 19.3.1, Descriptors and Descriptor List).

The RMAFCE bit in this register corresponds to bit 7 in EESR. When the RMAFCE bit is cleared to 0, the receive status (bit 7 in EESR) is reflected in the RFE bit in the receive descriptor. When this bit is set to 1, the status is not reflected in the descriptor even if the corresponding source occurs. The RMAFCE bit is cleared to 0 after a power-on reset and manual reset.

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	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.2.9 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames missed (discarded, and not transferred to the receive buffer) during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in this register reaches H′FFFF, counting-up is halted. When this register is read, the counter value is cleared to 0. Write operations to this register have no effect.

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 0	MFC15 to MFC0	All 0	R	Missed-Frame Counter Indicate the number of frames that are discarded and not transferred to the receive buffer during reception.

19.2.10 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is 4 times the set value. The EtherC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified by this register, when the transmit FIFO is full, or when 1-frame write is executed. When setting this register, do so in the transmission-halt state.

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TFT10 to TFT0	All 0	R/W	Transmit FIFO threshold When setting a transmit FIFO, the FIFO must be set to a smaller value than the specified value of the FIFO capacity by FDR. H′00: Store and forward modes H′01 to H′0C: Setting prohibited H′0D: 52 bytes H′0E: 56 bytes : : H′1F: 124 bytes H′20: 128 bytes : : H′3F: 252 bytes H′40: 256 bytes : : H′7F: 508 bytes H′80: 512 bytes : : H′FF: 1020 bytes H′100: 1024 bytes : : H′1FF: 2044 bytes H′200: 2048 bytes

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

19.2.11 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the size of the transmit and receive FIFOs.

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	TFD2 to TFD0	All 1	R/W	Transmit FIFO Size Specifies 256 bytes to 2 kbytes in 256-byte units as the size of the transmit FIFO. The setting must not be changed after transmission/reception has started.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	RFD2 to RFD0	All 1	R/W	Receive FIFO Size Specifies 256 bytes to 2 kbytes in 256-byte units as the size of the receive FIFO. The setting must not be changed after transmission/reception has started.

19.2.12 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RE bit in ECMR when a frame is received. This register must be set during the receiving-halt state.

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	RNC	0	R/W	Receive Enable Control Sets whether to continue frame reception. 0: Upon completion of reception of one frame, the E-DMAC writes receive status to the descriptor and clears the RR bit in EDRRR to 0 1: Upon completion of reception of one frame, the E-DMAC writes (writes back) receive status to the descriptor. In addition, the E-DMAC reads the next descriptor and prepares for the reception of the next frame

19.2.13 E-DMAC Operation Control Register (EDOCR)

EDOCR is a 32-bit readable/writable register that specifies the control methods used in E-DMAC operation.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	FEC	0	R/W	FIFO Error Control Specifies E-DMAC operation when transmit FIFO underflow or receive FIFO overflow occurs. 0: E-DMAC operation continues when underflow or overflow occurs 1: E-DMAC operation halts when underflow or overflow occurs
2	AEC	0	R/W	Address Error Control Indicates detection of an illegal memory address in an attempted E-DMAC transfer. 0: Illegal memory address not detected (normal operation) 1: Indicates that E-DMAC operation is halted because an illegal memory address is detected. When 0 is written to this bit, the E-DMAC resumes operation
1 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.2.14 Receive Buffer Write Address Register (RBWAR)

RBWAR stores the address of data to be written in the receiving buffer when the E-DMAC writes data to the receiving buffer. Which addresses in the receiving buffer are processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address that the E-DMAC is actually processing may be different from the value read from this register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBWA31 to RBWA0	All 0	R	Receiving-Buffer Write Address These bits can only be read. Writing is prohibited.

19.2.15 Receive Descriptor Fetch Address Register (RDFAR)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFA31 to RDFA0	All 0	R	Receiving-Descriptor Fetch Address These bits can only be read. Writing is prohibited.

19.2.16 Transmit Buffer Read Address Register (TBRAR)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBRA31 to TBRA0	All 0	R	Transmission-Buffer Read Address These bits can only be read. Writing is prohibited.

19.2.17 Transmit Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmission descriptor. Which transmission descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFA31 to TDFA0	All 0	R	Transmission-Descriptor Fetch Address These bits can only be read. Writing is prohibited.

19.2.18 Overflow Alert FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC. The threshold can be specified by the size of the receive FIFO data (RFD2 to RFD0) and the number of receive frames (RFF2 to RFF0).

If the same receive FIFO size as set by the FIFO size register (FDR) is set when flow control is turned on according to the RFD setting condition, flow control is turned on with (FIFO data size – 64) bytes. For instance, when RFD in FDR = 7 and RFD in FCFTR = 7, flow control is turned on when (2048 – 64) bytes of data is stored in the receive FIFO. The value set in the RFD bits in this register should be equal to or less than those in FDR.

Flow control is turned on when any of the setting conditions of the RFF2 to RFF0 bits or the RFD2 to RFD0 bits is satisfied. Flow control is turned off when none of the conditions is satisfied (release).

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 17 16	RFF2 RFF1 RFF0	1 1 1	R/W R/W R/W	Receive FIFO Overflow Alert Signal Output Threshold 000: When one receive frame has been stored in the receive FIFO 001: When two receive frames have been stored in the receive FIFO : : 110: When seven receive frames have been stored in the receive FIFO 111: When eight receive frames have been stored in the receive FIFO
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 1 0	RFD2 RFD1 RFD0	1 1 1	R/W R/W R/W	Receive FIFO Overflow Alert Signal Output Threshold 000: When (256 – 32) bytes of data is stored in the receive FIFO 001: When (512 – 32) bytes of data is stored in the receive FIFO : : 110: When (1792 – 32) bytes of data is stored in the receive FIFO 111: When (2048 – 64) bytes of data is stored in the receive FIFO

19.2.19 Transmit Interrupt Register (TRIMD)

TRIMD is a 32-bit readable/writable register that specifies whether or not to notify write-back completion for each frame using the TWB bit in EESR and an interrupt on transmit operations.

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	TIS	0	R/W	Transmit Interrupt Setting 0: Write-backed completion for each frame using the TWB bit in EESR is notified 1: Write-back completion for each frame is not notified

19.3 Operation

Using its direct memory access (DMA) function, the E-DMAC performs DMA transfer of transmit/receive data between a Ethernet frame transmission/reception data storage destination of user- specified (accessible memory space: transmit buffer/receive buffer) and the transmit/receive FIFO in the E-DMAC. (The user cannot read and write data in the transmit/receive FIFO directly via the CPU).

To enable the E-DMAC to perform DMA transfer, information (data) including a transmit/receive data storage address and so forth, referred to as a descriptor, is required. Before Ethernet frame transmission/reception, the E-DMAC reads descriptor information, then reads transmit data from the transmit buffer or writes receive data to the receive buffer according to the read descriptor information. By arranging multiple descriptors as a descriptor row (list) (to be placed in a readable/writable memory space), multiple Ethernet frames can be transmitted or received continuously.

19.3.1 Descriptors and Descriptor List

Two types of descriptors are available: transmit descriptors and receive descriptors. The E-DMAC automatically starts reading a transmit/receive descriptor when the TR bit in EDTRR is set to 1 or the RR bit in EDRRR is set to 1. In a transmit/receive descriptor, the user stores information about DMA transfer of transmit/receive data. After completion of Ethernet frame transmission/reception, the E-DMAC disables the descriptor valid/invalid bit and reflects the result of transmission/reception in the status bits.

Descriptors are placed in a readable/writable memory space. The address of the start descriptor (descriptor to be read first by the E-DMAC) is set in TDLAR/RDLAR. When multiple descriptors are prepared as a descriptor row (descriptor list), the descriptors are placed in continuous (memory) addresses according to the descriptor length set in the DL0 and DL1 bits in EDMR.

The E-DMAC consists of two systems: system 0 and system 1. The DMAC for transmission and the DMAC for reception operate independently of each other, and the DMAC for system 0 and the DMAC for system 1 operate independently of each other. For normal E-DMAC operation, place descriptors for transmission and reception and descriptors for system 0 and system 1 in those address spaces that do not overlap.

(1) **Transmit Descriptor**

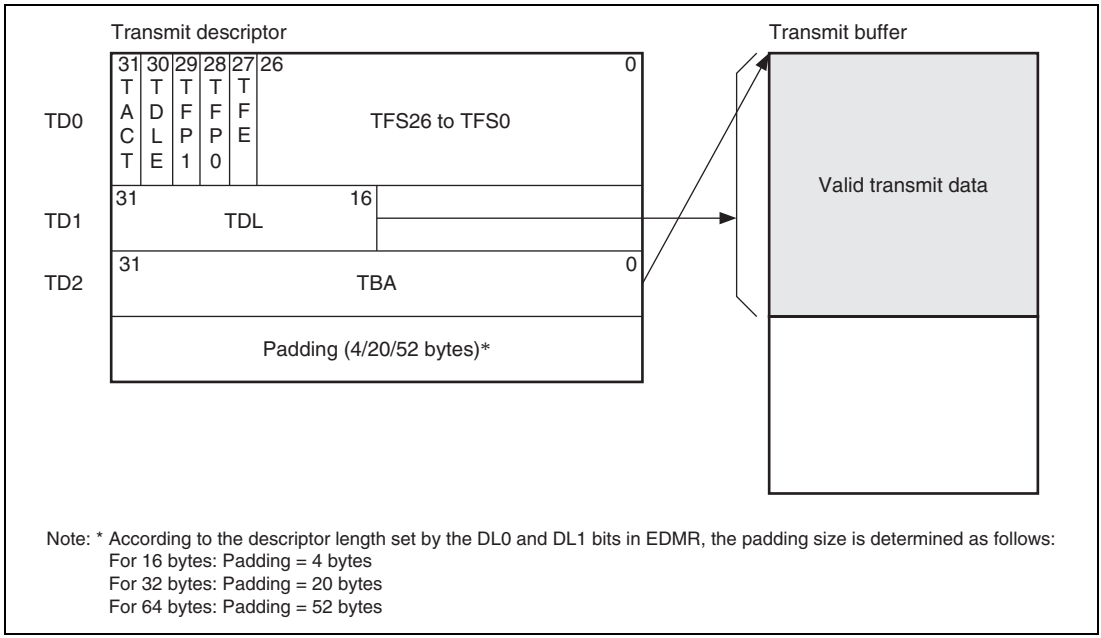
Figure 19.2 shows the configuration of a transmit descriptor and the relationship with a transmit buffer.

The data of a transmit descriptor consists of TD0, TD1, TD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR. In the figure, TBA (bits 31 to 0 in TD2) indicates the start address of a transmit buffer, and TDL (bits 31 to 16 in TD1) indicates the valid data length of the transmit buffer.

TD0 indicates whether the transmit descriptor is valid or invalid as well as information about the descriptor configuration and status. TD1 indicates the length of data in a transmit buffer to be transferred according to the specification of the descriptor. TD2 indicates the start address of a transmit buffer that holds data to be transferred.

Depending on the descriptor specification, one transmit descriptor can specify all transmit data of one frame (single-frame/single-buffer) or multiple descriptors can specify the transmit data of one frame (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation, the data portion that is used in a fixed manner in each Ethernet frame transmission can be referenced by multiple descriptors. For example, multiple descriptors can share the destination address and

transmit source address in an Ethernet frame, and the remaining data can be stored in each separate buffer.



Note: * According to the descriptor length set by the DL0 and DL1 bits in EDMR, the padding size is determined as follows:
For 16 bytes: Padding = 4 bytes
For 32 bytes: Padding = 20 bytes
For 64 bytes: Padding = 52 bytes

Figure 19.2 Relationship between Transmit Descriptor and Transmit Buffer

(a) Transmit Descriptor 0 (TD0)

Before the TR bit in EDTRR is set to 1, the user sets the descriptor valid/invalid bit and sets other descriptor configuration. After completion of Ethernet frame transmission, the E-DMAC disables the descriptor valid/invalid bit and writes status information. This operation is referred to as write-back.

When using TD0, the user should write desired values to bits 31 to 28 according to the descriptor configuration. Write 0 to bits 27 to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Valid/Invalid Indicates whether the corresponding descriptor is valid or invalid. To make this bit valid, store transmit data in a transmit buffer (user-specified transmit data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 upon completion of data transfer. 0: Indicates that the transmit descriptor is invalid Indicates the initial setting state, the state after 0 is written, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because of completion of the processing of the E-DMAC data transfer. If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the TR bit in EDTRR to 0, and halts transfer operation related to transmission by the E-DMAC. 1: Indicates that the transmit descriptor is valid After the user writes 1 to this bit, this bit indicates that data is not transferred yet or data is being transferred. When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid.

Bit	Bit Name	Initial Value	R/W	Description
30	TDLE	0	R/W	Transmit Descriptor List End Indicates whether the corresponding descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor Upon completion of transfer of the corresponding descriptor, the E-DMAC reads the next one in the list of continuous descriptors. 1: Last descriptor Upon completion of transfer of the corresponding descriptor, the E-DMAC reads the descriptor placed at the address indicated by TDLAR.

Bit	Bit Name	Initial Value	R/W	Description
29	TFP1	0	R/W	Transmit Frame Position 1, 0
28	TFP0	0	R/W	Indicates whether information of the corresponding descriptor represents information about the start, middle, or end of the transmit frame. 00: The information of the descriptor represents information about the middle of the frame. 01: The information of the descriptor represents information about the end of the frame. 10: The information of the descriptor represents information about the start of the frame. 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)). Reference: When one frame is divided for use, the method of specifying this bit for a descriptor row according to the number of divisions is described below. [For single-frame/single-descriptor operation] First descriptor: TFP[1:0] = 11 [For single-frame/two-descriptor operation] First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 01 [For single-frame/three-descriptor operation] First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 00 Third descriptor: TFP[1:0] = 01 When the number of divisions is large, a descriptor row is configured by adding intermediate descriptors with TFP[1:0] = 00.

Bit	Bit Name	Initial Value	R/W	Description
27	TFE	0	R/W	Transmit Frame Error Occurrence Indicates that an error occurred in the transmit frame. The errors occurred in TFS8 (bit 8), or TFS3 to TFS0 (bits 3 to 0).
26 to 0	TFS26 to TFS0	All 0	R/W	Transmit Frame Status Indicate the status of the corresponding frame. A bit below, when set to 1, indicates the occurrence of the corresponding event. If the events of TFS8, or TFS3 to TFS0 occur, frames are incompletely transmitted. TFS26 to TFS9: Reserved (The write value should always be 0.) TFS8: Transmit abort detected Note: This bit is set when any bit of TFS3 to TFS0 is set. TFS7 to TFS4: Reserved (The write value should always be 0.) TFS3: Failure to detect the carrier at the start of transmission (corresponding to the CND bit in EESR) TFS2: Loss of the carrier during transmission (corresponding to the DLC bit in EESR) TFS1: Late (delayed) collision (corresponding to the CD bit in EESR) TFS0: Transmit retry over (corresponding to the TRO bit in EESR)

(b) Transmit Descriptor 1 (TD1)

TD1 indicates the data length of the transmit buffer used by the corresponding descriptor.

The user should set TD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TDL	All 0	R/W	Transmit Buffer Data Length (in bytes) Indicate the data length of the corresponding transmit buffer in bytes.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(c) Transmit Descriptor 2 (TD2)

TD2 indicates the start address of the corresponding 32-bit width transmit buffer. An address value on a longword boundary should be specified.

The user should set TD2 before the start of a read by the E-DMAC.

(2) Receive Descriptor

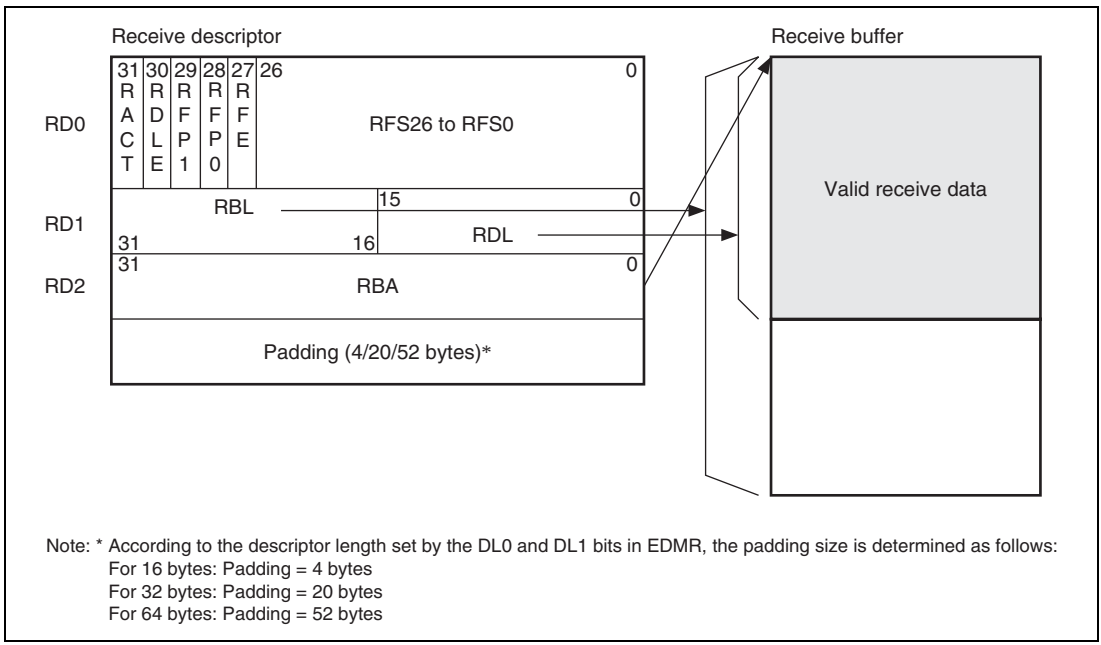
Figure 19.3 shows the relationship between a receive descriptor and receive buffer.

The data of a receive descriptor consists of RD0, RD1, RD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR. In the figure, RBA (bits 31 to 0 in RD2) indicates the start address of a receive buffer. RBL (bits 31 to 16 in RD1) indicates the usable data length of the receive buffer. RDL (bits 15 to 0 in RD1) indicates the data length of a received frame.

RD0 indicates whether the receive descriptor is valid or invalid as well as information about descriptor configuration and status. RD1 indicates the length (storage destination size) of data in the receive buffer to be received according to the specification of the descriptor. RD2 indicates the start address of the receive buffer for storing receive data.

Depending on the descriptor specification, one receive descriptor can specify the storing of all receive data of one frame in a receive buffer (single-frame/single-buffer) or multiple descriptors can specify the storing of the receive data of one frame in receive buffers (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation, suppose that a row of multiple descriptors (descriptor list) is prepared, RBL of each descriptor is 500 bytes, and a 1514-byte

Ethernet frame is received. In such a case, the received Ethernet frame is transferred sequentially to buffers, 500 bytes for each buffer, starting with the first descriptor. Only the last 14 bytes are transferred to the fourth buffer. When a frame longer than RBL of a descriptor is received, the E-DMAC transfers the remaining data to the receive buffer by using the subsequent descriptors. As an example of efficient single-frame/multi-buffer operation, information items on different processing layers in an Ethernet frame can be separated from each other by using different buffers. For example, the destination address, transmit source address, and type field data in an Ethernet frame can be stored in buffer 1 (with RBL set to 14 bytes) and the remaining data can be stored in buffer 2 (with RBL set to 1500 bytes). All receive frames, of course, can be stored in a single buffer if multiple descriptors are prepared and RBL of each descriptor is set to more than 1514 bytes (maximum Ethernet frame length).



Note: * According to the descriptor length set by the DL0 and DL1 bits in EDMR, the padding size is determined as follows:
For 16 bytes: Padding = 4 bytes
For 32 bytes: Padding = 20 bytes
For 64 bytes: Padding = 52 bytes

Figure 19.3 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

The user sets the descriptor valid/invalid bit and sets whether the descriptor represents the end of the descriptor list in RD0 before the RR bit in EDRRR is set to 1 and the start of a read by the E-DMAC. After completion of receive DMA transfer of an Ethernet frame by the E-DMAC, the E-DMAC disables the descriptor valid/invalid bit and writes status information. This operation is referred to as write-back.

When using RD0, the user should write desired values to bits 31 and 30 according to the descriptor configuration. Write 0 to bits 29 to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Valid/Invalid Indicates whether the corresponding descriptor is valid or invalid. To make this bit valid, prepare a receive buffer (user-specified receive data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 upon completion of data transfer. 0: Indicates that the receive descriptor is invalid Indicates the initial setting state, the state after 0 is written, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because of completion of the processing of the E-DMAC data transfer. If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the RR bit in EDRRR to 0, and halts transfer operation related to reception by the E-DMAC. 1: Indicates that the receive descriptor is valid Indicates that data is not transferred yet after the user writes 1 to this bit, or that data is being transferred. When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid.

Bit	Bit Name	Initial Value	R/W	Description
30	RDLE	0	R/W	Receive Descriptor List End Indicates whether the corresponding descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor Upon completion of transfer of the corresponding descriptor, the E-DMAC reads the next one in the list of continuous descriptors. 1: Last descriptor Upon completion of transfer of the corresponding descriptor, the E-DMAC reads the descriptor placed at the address indicated by RDLAR.

Bit	Bit Name	Initial Value	R/W	Description
29	RFP1	0	R/W	Receive Frame Position 1, 0
28	RFP0	0	R/W	The E-DMAC indicates by write-back operation whether information of the corresponding descriptor represents information about the start, middle, or end of the receive frame. 00: The information of the descriptor represents information about the middle of the frame. 01: The information of the descriptor represents information about the end of the frame. 10: The information of the descriptor represents information about the start of the frame. 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)). Reference: The relationship between a frame after reception of one frame and a descriptor is described below. [For single-frame/single-descriptor operation] First descriptor: RFP[1:0] = 11 [For single-frame/two-descriptor operation] First descriptor: RFP[1:0] = 10 Second descriptor: RFP[1:0] = 01 [For single-frame/three-descriptor operation] First descriptor: RFP[1:0] = 10 Second descriptor: RFP[1:0] = 00 Third descriptor: RFP[1:0] = 01 When the number of divisions is large, a descriptor row is configured by adding intermediate descriptors with RFP[1:0] = 00.

Bit	Bit Name	Initial Value	R/W	Description
27	RFE	0	R/W	Receive Frame Error Occurrence Indicates that an error occurred in the receive frame. The errors occurred in RFS8 (bit 8), or RFS3 to RFS0 (bits 3 to 0). TRSCER can specify whether the multicast address frame receive information is reflected in this bit or not.
26 to 0	RFS26 to RFS0	All 0	R/W	Receive Frame Status Indicate the status of the corresponding frame. A bit below, when set to 1, indicates the occurrence of the corresponding event. If the events of RFS8, or RFS4 to RFS0 occur, frames are incompletely received. RFS26 to RFS10: Reserved (The write value should always be 0) RFS9: Receive FIFO overflow (corresponding to the RFOF bit in EESR) RFS8: Receive abort detected Note: This bit is set when any bit of RFS3 to RFS0 is set. RFS7: Multicast address frame received (corresponding to the RMAF bit in EESR) RFS6 and RFS5: Reserved (The write value should always be 0) RFS4: residual-bit frame receive error (corresponding to the RRF bit in EESR) RFS3: Too-long frame receive error (corresponding to the RTLF bit in EESR) RFS2: Too-short frame receive error (corresponding to the RTSF bit in EESR) RFS1: PHY-LSI receive error (corresponding to the PRE bit in EESR) RFS0: CRC error on receive frame (corresponding to the CERF bit in EESR)

(b) Receive Descriptor 1 (RD1)

In RD1, the user specifies the data length of a receive buffer usable by the corresponding descriptor. After reception of a frame, RD1 indicates the length of a frame received by the E-DMAC.

The user should set RD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Data Length (in bytes, to be specified on 16-byte boundary) Set the length of data that can be received by the corresponding receive buffer in bytes. Set a receive buffer length on a 16-byte boundary (with bits 19 to 16 cleared to 0). In single-frame/single-buffer (descriptor) operation, the maximum receive frame length excluding CRC data is 1514 bytes. When specifying a receive buffer length, set 1520 bytes (H'05F0), which is determined considering the maximum receive frame length and a 16-byte boundary.
15 to 0	RDL	All 0	R	Receive Data Length Indicate the data length of a receive frame stored in the receive buffer. Receive data transferred to the receive buffer does not include CRC data (4 bytes) placed at the end of a frame. As a receive frame length, the number of bytes (valid data bytes) not including CRC data are reported. In single-frame/multi-buffer (descriptor) operation, only the receive data length of the last descriptor is valid. The receive data length of an intermediate descriptor has no meaning.

(c) Receive Descriptor 2 (RD2)

RD2 indicates the start address of the corresponding 32-bit width receive buffer. Set the start address of a receive buffer on a longword boundary. When an SDRAM is connected, set the start address of a receive buffer on a 16-byte boundary.

The user should set RD2 before the start of a read by the E-DMAC.

19.3.2 Transmission

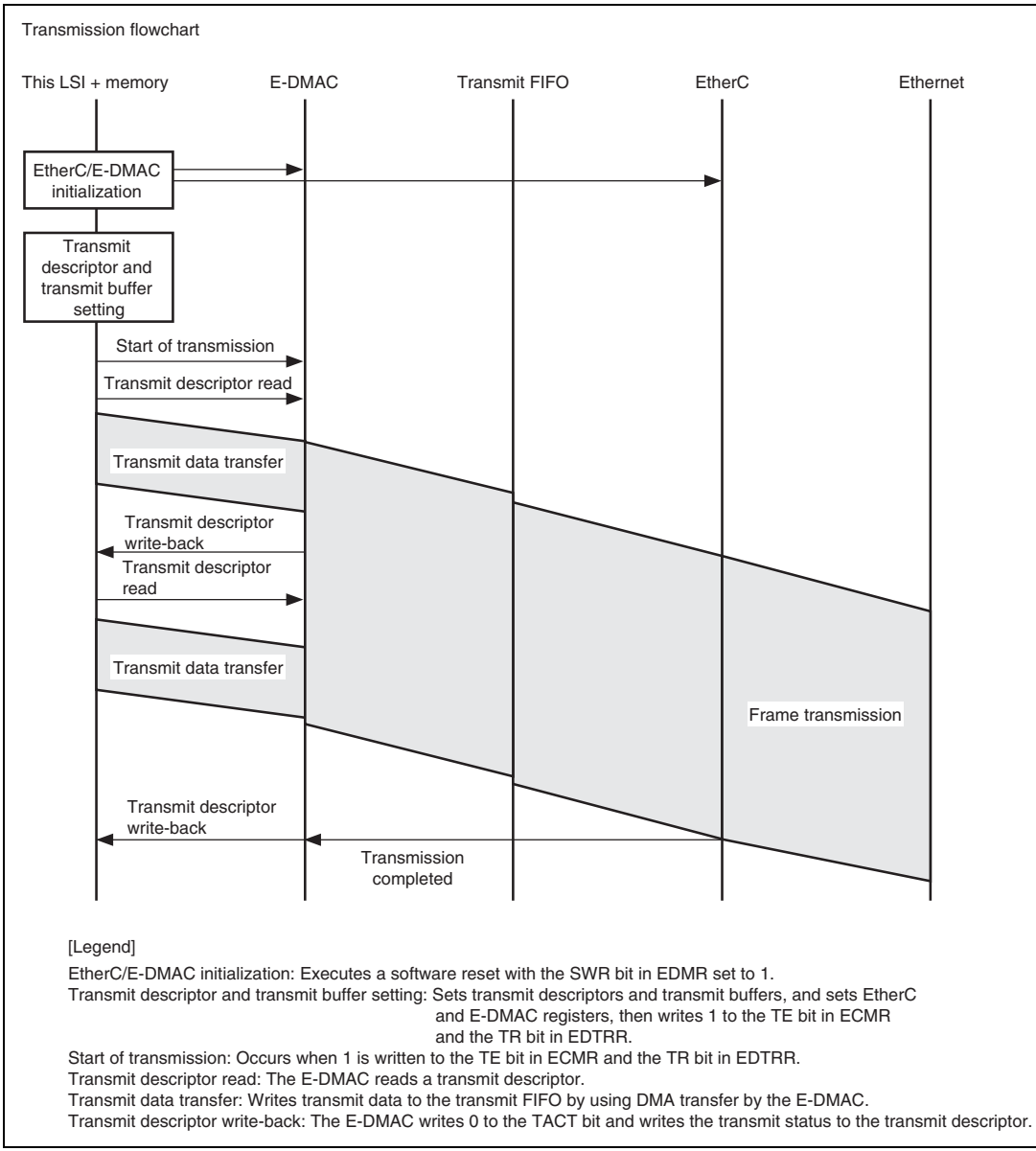
When 1 is written to the transmit request bit (TR) in the E-DMAC transmit request register (EDTRR) while the TE bit in ECMR is set to 1, the E-DMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by the transmit descriptor start address register (TDLAR) at the initial start time). If the TACT bit of the read descriptor is set to 1 (valid), the E-DMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 for transfer to the EtherC. The EtherC creates a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TFP value.

1. TFP = 00 or 10 (frame continuation):

   Descriptor write-back (writing 0 to the TACT bit) is performed after DMA transfer.
2. TFP = 01 or 11 (frame end):

   Descriptor write-back (writing 0 to the TACT bit and writing status) is performed after completion of frame transmission.

As long as the TACT bit of a read descriptor is set to 1 (valid), the reading of E-DMAC descriptors and the transmission of frames continue. When a descriptor with the TACT bit cleared to 0 (invalid) is read, the E-DMAC clears the TR bit in EDTRR to 0 and completes transmit processing.



[Legend]

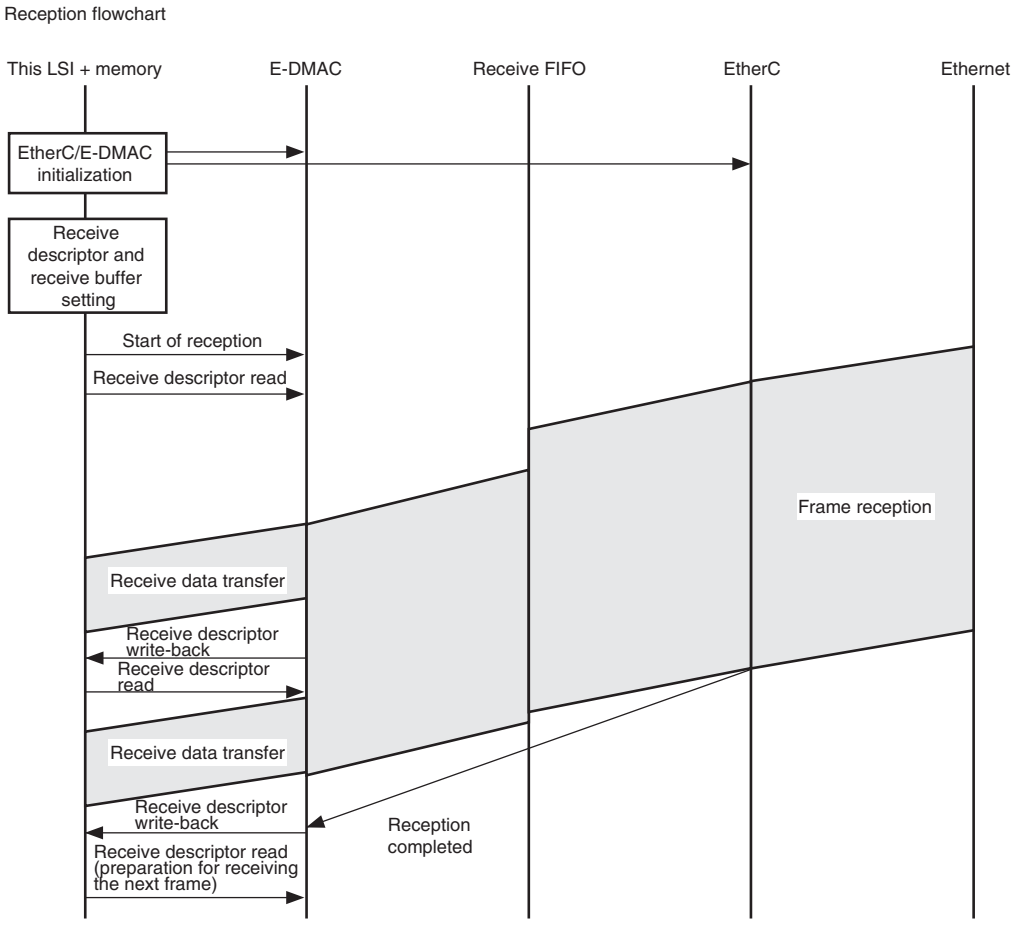
EtherC/E-DMAC initialization: Executes a software reset with the SWR bit in EDMR set to 1.
Transmit descriptor and transmit buffer setting: Sets transmit descriptors and transmit buffers, and sets EtherC and E-DMAC registers, then writes 1 to the TE bit in ECMR and the TR bit in EDTRR.
Start of transmission: Occurs when 1 is written to the TE bit in ECMR and the TR bit in EDTRR.
Transmit descriptor read: The E-DMAC reads a transmit descriptor.
Transmit data transfer: Writes transmit data to the transmit FIFO by using DMA transfer by the E-DMAC.
Transmit descriptor write-back: The E-DMAC writes 0 to the TACT bit and writes the transmit status to the transmit descriptor.

Figure 19.4 Sample Transmission Flowchart (Single-Frame/Two-Descriptor)

19.3.3 Reception

When 1 is written to the receive request bit (RR) in the E-DMAC receive request register (EDRRR) while the RE bit in ECMR is set to 1, the E-DMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by the receive descriptor start address register (RDLAR) at the initial start time) then enters the receive standby state. When the EtherC receives a frame for this LSI (with an address enabled for reception by this LSI), the EtherC stores the receive data in the receive FIFO. The receive data is transferred to the receive buffer specified by RD2 according to the receive descriptor with the RACT bit set to 1 (valid). If the data length of a received frame is longer than the buffer length specified by RD1, the E-DMAC performs a write-back operation to the descriptor (with RFP set to 10 or 00) when the buffer becomes full, then reads the next descriptor. The E-DMAC then continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is suspended because of a certain kind of error, the E-DMAC performs write-back to the relevant descriptor (with RFP set to 11 or 01), and then ends the receive processing. The E-DMAC then reads the next descriptor and enters the receive standby state again.

To receive frames continuously, the receive enable control bit (RNC) must be set to 1 in the receive method control register (RMCR). The initial value is 0.



[Legend]

EtherC/E-DMAC initialization: Executes a software reset with the SWR bit in EDMR set to 1.
Receive descriptor and receive buffer setting: Sets receive descriptors and receive buffers, and sets EtherC and E-DMAC registers, then writes 1 to the RE bit in ECMR and the RR bit in EDRRR.
Start of reception: Occurs when 1 is written to the RE bit in ECMR and the RR bit in EDRRR.
Receive descriptor read: The E-DMAC reads a receive descriptor.
Receive data transfer: Writes receive data from the receive FIFO to the receive buffer by using DMA transfer by the E-DMAC.
Receive descriptor write-back: The E-DMAC writes 0 to the RACT bit and writes the receive status to the receive descriptor.

Figure 19.5 Sample Reception Flowchart (Single-Frame/Two-Descriptor)

19.3.4 Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/ Multi-Descriptor)

(1) Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in figure 19.6 is carried out by the E-DMAC.

In the figure where the transmit descriptor is shown as inactive (TACT bit = 0), buffer data has already been transmitted normally, and where the transmit descriptor is shown as active (TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first descriptor part where the transmit descriptor is active (TACT bit = 1), transmission is halted, and the TACT bit cleared to 0, immediately. The next descriptor is then read, and the position within the transmit frame is determined on the basis of bits TFP1 and TFP0 (continuing [B′00] or end [B′01]). In the case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descriptor is read immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to 0, but write-back is also performed to the TFE and TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the final descriptor. If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the final descriptor write-back.

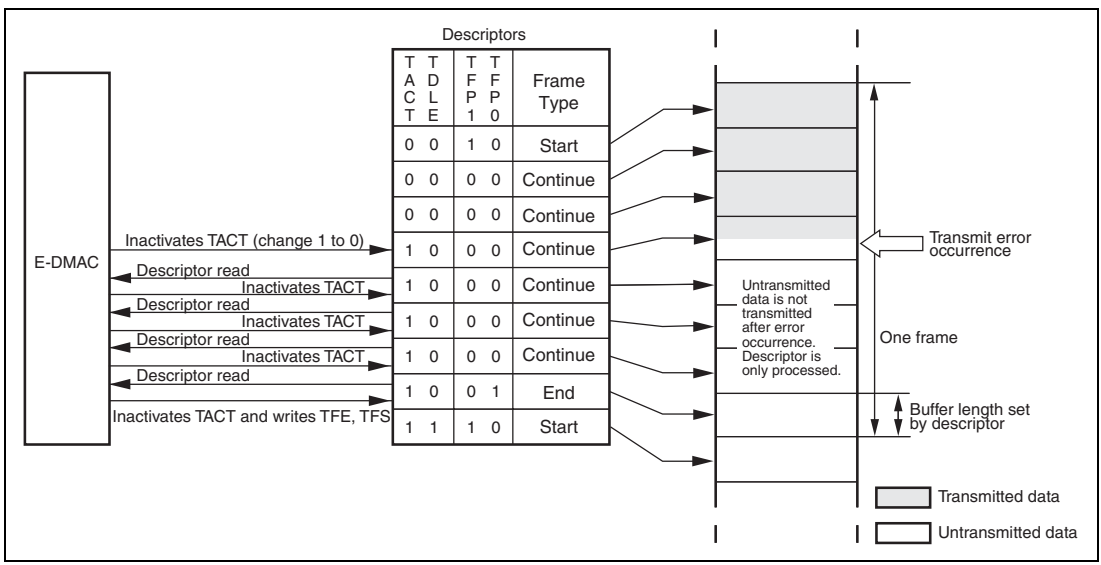


Figure 19.6 E-DMAC Operation after Transmit Error

(2) Receive Processing in Case of Multi-Buffer Frame

If an error occurs during reception in the case of a multi-buffer frame where a receive frame is divided for storage in multiple buffers, the E-DMAC performs the processing shown in figure 19.7.

In the figure, the invalid receive descriptors (with the RACT bit cleared to 0) represent the normal reception of data to be stored in buffers, and the valid receive descriptors (with the RACT bit set to 1) represent unreceived buffers. If a frame receive error occurs with a descriptor shown in the figure, the status is written back to the corresponding descriptor.

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

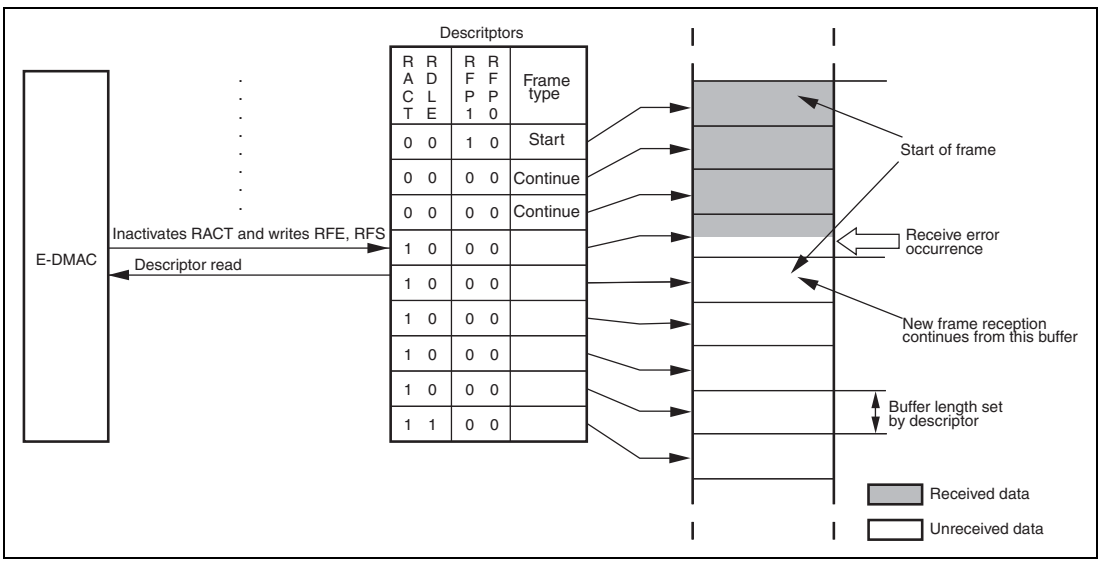


Figure 19.7 E-DMAC Operation after Receive Error

19.3.5 Receive FIFO Overflow Alert Signal ($\overline{\text{ARBUSY}}$)

The E-DMAC outputs the receive FIFO overflow alert signal ($\overline{\text{ARBUSY}}$) to the EtherC to support flow control function conforming to IEEE802.3x of the EtherC. The $\overline{\text{ARBUSY}}$ signal synchronized with the bus clock (B clock) signal is also output to an external pin of this LSI.

When the capacity of data received in receive FIFO or the number of receive frames reach the threshold (RFF2 to RFF0, or RFD2 to RFD0) specified in FCFTR in E-DMAC, $\overline{\text{ARBUSY}}$ is valid.

The threshold is the value less than the overflow value: 2048 – 64, 1792 – 32, 1536 – 32, and 256 – 32 bytes.

Figure 19.9 shows the configuration of the receive FIFO overflow alert signal ($\overline{\text{ARBUSY}}$) output.

As shown in figure 19.9, because the $\overline{\text{ARBUSY}}$ signal passes through the system clock synchronization circuit, it is behind the receive FIFO overflow alert signal received in EtherC.

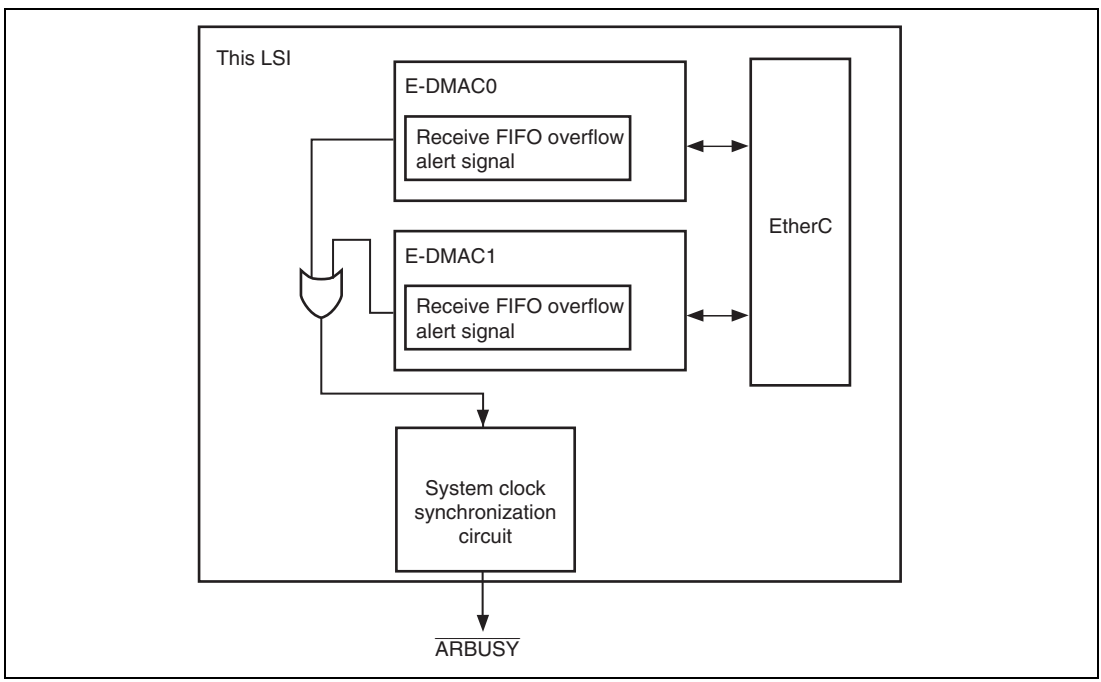


Figure 19.8 Configuration of $\overline{\text{ARBUSY}}$

(1) Operation of Receive FIFO Overflow Alert Signal

Receive FIFO overflow alert signal is asserted when the number of the receive data accumulated in the receive FIFO is equal to or more than the threshold set in the overflow alert FIFO threshold register (FCFTR) (1). After that, when the number of the accumulated receive data drops below (threshold – 32) bytes, the signal is negated (2).

Here, threshold values are as follows: 2048 – 64, 1792 – 32,…. Therefore, the signal-negated values are as follows: 2048 – 96, 1792 – 64,….

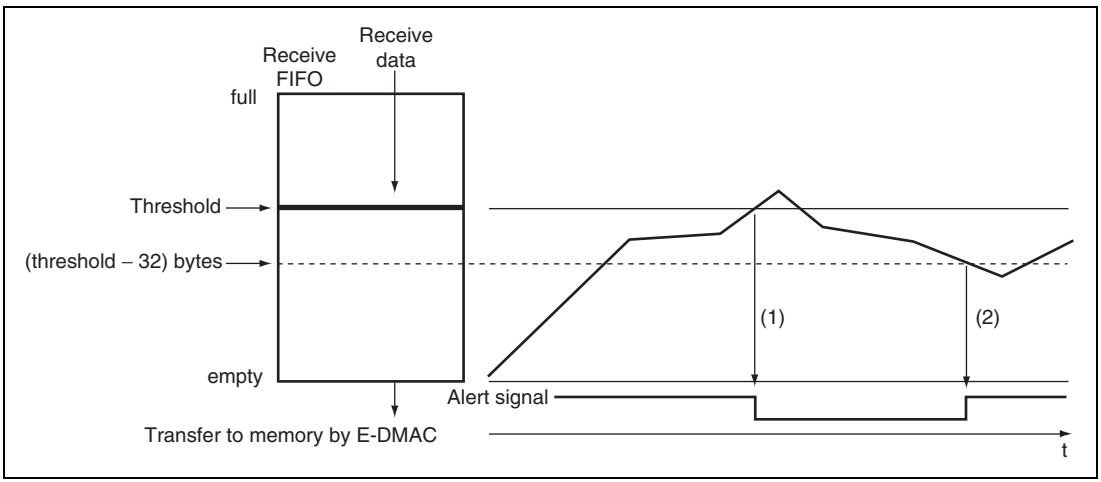
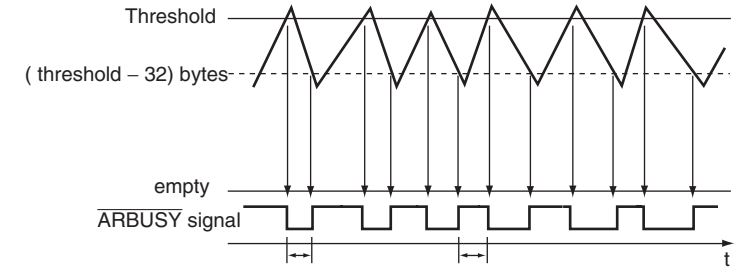


Figure 19.9 Summary of Receive FIFO Overflow Alert Signal

(a) Receive FIFO Overflow Alert Signal Changing

The receive FIFO in the E-DMAC can perform writing (reception) data from the Ethernet line and reading data from the system simultaneously. Therefore during system operation, receive data of FIFO is always increased or decreased. If the change is performed near the threshold, the receive FIFO overflow alert signal may be seen as shown in figure 19.11. Minimum of receive data changes depending on the number of FIFO read cycles and rate of Ethernet line (10 to 100 Mbs).



Minimum value (low): 150 nsec (Exernal bus operated at 66.66MHz and 16-byte burst transfer (5 cycle × 2))

Minimum value (high): 2560 nsec (100 Base-T (100M Ether) operation)
(Time for receiving 32-byte data)

Figure 19.10 $\overline{\text{ARBUSY}}$ Signal Change and Minimum Pulse Width Depending on Increase and Decrease of FIFO

19.4 Usage Notes

19.4.1 Using of EDTRR and EDRRR

[Problems]

While the Ethernet functions are being used, the TR bit in EDTRR or the RR bit in EDRRR is cleared to 0 to stop the E-DAMC functions if the descriptor valid bit is invalid. When the request bit (TR or RR) is cleared by the E-DMAC and the request bit (TR or RR) is set by the user's firmware simultaneously, transmission or reception may not be started even if the request bit (TR or RR) is set to 1.

[Occurring Condition]

When the user's firmware tries to set the request bit (TR or RR), while the request bit (TR or RR) is 1.

[Avoiding Methods]

To prevent the simultaneous occurrence of the request bit (TR or RR) being cleared by the E-DMAC and the request bit (TR or RR) being set by the user's firmware, the user's firmware should set the request bit (TR or RR) after confirming that it is cleared by the E-DMAC.

The methods to clear the RR bit with E-DMAC are as follows.

(1) Confirmation of the TR bit

As a direct method, it is possible to confirm by reading the TR bit in EDTRR as 0.

As an indirect method, it is possible to confirm by reading the TDE bit in EESR as 1.

(2) Confirmation of the RR bit

As a direct method, it is possible to confirm by reading the RR bit in EDRRR as 0.

As an indirect method, it is possible to confirm by reading the RDE bit in EESR as 0.

19.4.2 Endian Support in E-DMAC

When the external memory is accessed through the E-DMAC, big endian is supported but little endian is not supported. Therefore, if the external memory is accessed through the E-DMAC in little endian mode, data format should be converted from big endian mode to little endian mode through software.

Section 20 IP Security Accelerator (IPSEC)

This section will be made available on conclusion of a nondisclosure agreement.

For details, contact your Renesas sales agency.

Section 21 Pin Function Controller (PFC)

21.1 Overview

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the input/output direction. The pin function and input/output direction can be selected for each pin individually without regard to the operating mode of the chip. Tables 21.1 and 21.2 list the multiplexed pins.

Table 21.1 List of Multiplexed Pins (1)

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 input/output (port)	SIOFSYNC0 input/output (SIOF0)
A	PTA6 input/output (port)	TXD_SIO0 output (SIOF0)
A	PTA5 input/output (port)	RXD_SIO0 input (SIOF0)
A	PTA4 input/output (port)	SIOMCLK0 input (SIOF0)
A	PTA3 input/output (port)	SCK_SIO0 input/output (SIOF0)
A	PTA2 input/output (port)	SCIF0CK input/output (SCIF0)
A	PTA1 input/output (port)	TXD0 output (SCIF0)
A	PTA0 input/output (port)	RXD0 input (SCIF0)
B	PTB7 input/output (port)	$\overline{\text{RTS0}}$ output (SCIF0)
B	PTB6 input/output (port)	$\overline{\text{CTS0}}$ input (SCIF0)
B	PTB5 input/output (port)	SCIF1CK input/output (SCIF1)
B	PTB4 input/output (port)	TXD1 output (SCIF1)
B	PTB3 input/output (port)	RXD1 input (SCIF1)
B	PTB2 input/output (port)	$\overline{\text{RTS1}}$ output (SCIF1)
B	PTB1 input/output (port)	$\overline{\text{CTS1}}$ input (SCIF1)
B	PTB0 input/output (port)	Reserved (setting prohibited)*
C	PTC7 input/output (port)	$\overline{\text{IOIS16}}$ input (BSC)
C	PTC6 input/output (port)	$\overline{\text{CE2B}}$ output (BSC)
C	PTC5 input/output (port)	$\overline{\text{CE2A}}$ output (BSC)
C	PTC4 input/output (port)	SIOFSYNC1 input/output (SIOF1)

Port	Port Function (Related Module)	Other Function (Related Module)
C	PTC3 input/output (port)	TXD_SIO1 output (SIOF1)
C	PTC2 input/output (port)	RXD_SIO1 input (SIOF1)
C	PTC1 input/output (port)	SIOMCLK1 input (SIOF1)
C	PTC0 input/output (port)	SCK_SIO1 input/output (SIOF1)

Note: * When the register is set to reserved, the operation is not guaranteed.

Table 21.2 List of Multiplexed Pins (2)

Ethernet controller Function	Other Function (Related Module)
EXOUT1 output	TEND1 output (DMAC)
CAMSEN1 input	IRQ5 input (INTC)
EXOUT0 output	TEND0 output (DMAC)
CAMSEN0 input	IRQ4 input (INTC)

21.2 Register Configuration

The registers of the pin function controller are shown below.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Ethernet controller pin control register (PETCR)

21.3 Register Descriptions

21.3.1 Port A Control Register (PACR)

PACR is a 16-bit readable/writable register that selects the pin functions. PACR is initialized to H′AAAA by a power-on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PA7MD1	1	R/W	Modes PA7 to PA0 Control
14	PA7MD0	0	R/W	The combination of PAnMD1 and PAnMD0 (n = 0 to 7) selects the pin functions and control input pull-up MOS.
13	PA6MD1	1	R/W	
12	PA6MD0	0	R/W	00: Other function (see table 21.1)
11	PA5MD1	1	R/W	01: Port output
10	PA5MD0	0	R/W	10: Port input (pull-up MOS: on)
9	PA4MD1	1	R/W	11: Port input (pull-up MOS: off)
8	PA4MD0	0	R/W	
7	PA3MD1	1	R/W	
6	PA3MD0	0	R/W	
5	PA2MD1	1	R/W	
4	PA2MD0	0	R/W	
3	PA1MD1	1	R/W	
2	PA1MD0	0	R/W	
1	PA0MD1	1	R/W	
0	PA0MD0	0	R/W	

21.3.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin functions. PBCR is initialized to H′AAAA by a power-on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	1	R/W	Modes PB7 to PB0 Control
14	PB7MD0	0	R/W	The combination of PBnMD1 and PBnMD0 (n = 0 to 7) selects the pin functions and controls input pull-up MOS.
13	PB6MD1	1	R/W	
12	PB6MD0	0	R/W	00: Other function (n = 1 to 7) or reserved (n = 0) (see table 21.1)
11	PB5MD1	1	R/W	
10	PB5MD0	0	R/W	01: Port output
9	PB4MD1	1	R/W	10: Port input (pull-up MOS: on)
8	PB4MD0	0	R/W	11: Port input (pull-up MOS: off)
7	PB3MD1	1	R/W	
6	PB3MD0	0	R/W	
5	PB2MD1	1	R/W	
4	PB2MD0	0	R/W	
3	PB1MD1	1	R/W	
2	PB1MD0	0	R/W	
1	PB0MD1	1	R/W	
0	PB0MD0	0	R/W	

21.3.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects the pin functions. PCCR is initialized to H′AAAA by a power-on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PC7MD1	1	R/W	Modes PC7 to PC0 Control
14	PC7MD0	0	R/W	The combination of PCnMD1 and PCnMD0 (n = 0 to 7) selects the pin functions and controls input pull-up MOS.
13	PC6MD1	1	R/W	
12	PC6MD0	0	R/W	00: Other function (see table 21.1.)
11	PC5MD1	1	R/W	01: Port output
10	PC5MD0	0	R/W	10: Port input (pull-up MOS: on)
9	PC4MD1	1	R/W	11: Port input (pull-up MOS: off)
8	PC4MD0	0	R/W	
7	PC3MD1	1	R/W	
6	PC3MD0	0	R/W	
5	PC2MD1	1	R/W	
4	PC2MD0	0	R/W	
3	PC1MD1	1	R/W	
2	PC1MD0	0	R/W	
1	PC0MD1	1	R/W	
0	PC0MD0	0	R/W	

21.3.4 Ethernet Controller Pin Control Register (PETCR)

PETCR is a 16-bit readable/writable register that selects the pin functions. PETCR is initialized to H′AAAA by a power-on reset, but is not initialized by a manual reset, in standby mode, or in sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
15	PET3MD	1	R/W	Controls output of EXOUT1 (Ethernet controller function) and TEND1 (other function). 0: TEND1 (other function) is selected. 1: EXOUT1 (Ethernet controller function) is selected.
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13	PET2MD	1	R/W	Controls input of CAMSEN1 (Ethernet controller function) and IRQ5 (other function). 0: IRQ5 (other function) is selected. 1: CAMSEN1 (Ethernet controller function) is selected.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	PET1MD	1	R/W	Controls output of EXOUT0 (Ethernet controller function) and TEND0 (other function). 0: TEND0 (other function) is selected. 1: EXOUT0 (Ethernet controller function) is selected.
10	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
9	PET0MD	1	R/W	Controls input of CAMSEN0 (Ethernet controller function) and IRQ4 (other function). 0: IRQ4 (other function) is selected. 1: CAMSEN0 (Ethernet controller function) is selected.

Bit	Bit Name	Initial Value	R/W	Description
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Section 22 I/O Ports

22.1 Overview

This LSI has three 8-bit ports (ports A to C). All port pins are multiplexed with other pin functions (the pin function controller (PFC) handles the selection of pin functions and pull-up MOS control). Each port has a data register which stores data for the pins.

22.2 Register Descriptions

22.2.1 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores data for pins PTA7 to PTA0. Bits PA7DT to PA0DT correspond to pins PTA7 to PTA0. PADR is initialized to H′00 by a power-on reset but is not initialized by a manual reset, in standby mode, or sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DT	0	R/W	When the pin function is general output port, if the port is read, the value of the corresponding PADR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 22.1 shows the function of PADR.
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

Table 22.1 Port A Data Register (PADR) Read/Write Operations

PAnMD1	PAnMD0	Pin State	Read	Write
0	0	Other function	PADR value	Value is written to PADR, but does not affect pin state.
	1	Output	PADR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PADR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PADR, but does not affect pin state.

[Legend]
n = 0 to 7

22.2.2 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores the data for pins PTB7 to PTB0. Bits PB7DT to PB0DT correspond to the pins PTB7 to PTB0. PBDR is initialized to H′00 by a power-on reset but is not initialized by a manual reset, in standby mode, or sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DT	0	R/W	When the pin function is general output port, if the port is read, the value of the corresponding PBDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Tables 22.2 and 22.3 show the function of PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

Table 22.2 Port B Data Register (PBDR) Read/Write Operations (1)

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Other function	PBDR value	Value is written to PBDR, but does not affect pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PBDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PBDR, but does not affect pin state.

[Legend]

n = 1 to 7

Table 22.3 Port B Data Register (PBDR) Read/Write Operations (2)

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Reserved*	PBDR value	Value is written to PBDR, but does not affect pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin status	Value is written to PBDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin status	Value is written to PBDR, but does not affect pin state.

[Legend]

n = 0

Note: * When this pin is specified as a reserved pin, its operation is not guaranteed.

22.2.3 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores the data for pins PTC7 to PTC0. Bits PC7DT to PC0DT correspond to the pins PTC7 to PTC0. PCDR is initialized to H′00 by a power-on reset but is not initialized by a manual reset, in standby mode, or sleep mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	0	R/W	When the pin function is general output port, if the port is read, the value of the corresponding PCDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 22.4 shows the function of PCDR.
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

Table 22.4 Port C Data Register (PCDR) Read/Write Operations

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Value is written to PCDR, but does not affect pin state.
	1	Output	PCDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PCDR, but does not affect pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PCDR, but does not affect pin state.

[Legend]
n = 0 to 7

Section 23 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) and advanced user debugger (AUD) for a boundary scan function and emulator support.

This section describes the H-UDI. The AUD is a function exclusively for use by an emulator. Refer to the User's Manual for the relevant emulator for details of the AUD.

23.1 Features

The H-UDI (User debugging interface) is a serial I/O interface which conforms to JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in this LSI supports a boundary scan mode, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

Figure 23.1 shows a block diagram of the H-UDI.

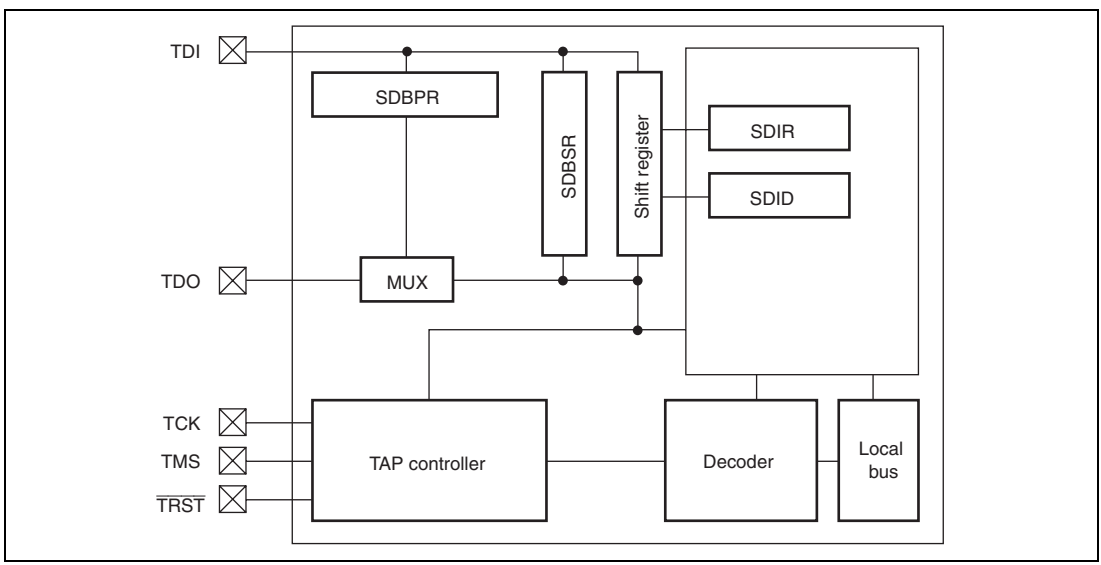


Figure 23.1 Block Diagram of H-UDI

23.2 Input/Output Pins

Table 23.1 shows the pin configuration of the H-UDI.

Table 23.1 Pin Configuration

Pin Name	Input/Output	Description
TCK	Input	Serial Data Input/Output Clock Pin Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a constant period when power is turned on regardless of using the H-UDI function. As the same as the $\overline{\text{RESETP}}$ pin, the $\overline{\text{TRST}}$ pin should be driven low at the power-on reset state and driven high after the power-on reset state is released. This is different from the JTAG standard. See section 23.4.2, Reset Configuration, for more information.
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The data output timing depends on the command type set in the SDIR. See section 23.4.3 TDO Output Timing, for more information.
$\overline{\text{ASEMD0}}$	Input	ASE Mode Select Pin If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered; if a high level is input, normal mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RESETP}}$ negation.

Pin Name	Input/Output	Description
$\overline{\text{ASEBRKAK}}$	Output	Dedicated emulator pin
$\overline{\text{AUDSYNC}}$	Output	
AUDATA3 to 0	Output	
AUDCK	Output	

23.3 Register Descriptions

The H-UDI has the following registers. Refer the section 24, List of Registers, for the addresses and access size for registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

23.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the bypass mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined but SDBPR is initialized to 0 if the TAP is in Capture-DR state.

23.3.2 Instruction Register (SDIR)

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TI7 to TI5	All 1	R	Test Instruction 7 to 0
12	TI4	0	R	The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 23.2.
11 to 8	TI3 to TI0	All 1	R	
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 23.2 H-UDI Commands

Bits 15 to 8								
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	Description
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PRELOAD
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Initial value)
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than the above								Reserved

23.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/output pins of this LSI. The initial value is undefined. SDBSR cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary scan test conforming to the JTAG standard can be carried out. Table 23.3 shows the correspondence between this LSI's pins and boundary scan register bits.

Table 23.3 This LSI's Pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
	from TDI		334	D2	OUT
362	$\overline{\text{BREQ}}$	IN	333	D3	OUT
361	$\overline{\text{WAIT}}$	IN	332	D4	OUT
360	D0	IN	331	D5	OUT
359	D1	IN	330	D6	OUT
358	D2	IN	329	D7	OUT
357	D3	IN	328	D8	OUT
356	D4	IN	327	D9	OUT
355	D5	IN	326	D10	OUT
354	D6	IN	325	D11	OUT
353	D7	IN	324	D12	OUT
352	D8	IN	323	D13	OUT
351	D9	IN	322	D14	OUT
350	D10	IN	321	D15	OUT
349	D11	IN	320	$\overline{\text{WE0}}$($\overline{\text{BE0}}$)/DQMLL	OUT
348	D12	IN	319	$\overline{\text{WE1}}$($\overline{\text{BE1}}$)/DQMLU/$\overline{\text{WE}}$	OUT
347	D13	IN	318	RD/$\overline{\text{WR}}$	OUT
346	D14	IN	317	$\overline{\text{CAS}}$	OUT
345	D15	IN	316	CKE	OUT
344	$\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$/$\overline{\text{ARBUSY}}$	OUT	315	$\overline{\text{RAS}}$	OUT
343	$\overline{\text{BACK}}$	OUT	314	$\overline{\text{CS2}}$	OUT
342	$\overline{\text{CS0}}$	OUT	313	$\overline{\text{CS3}}$	OUT
341	$\overline{\text{CS4}}$	OUT	312	A0	OUT
340	$\overline{\text{CS5A}}$	OUT	311	A1	OUT
339	$\overline{\text{CS6A}}$	OUT	310	A2	OUT
338	$\overline{\text{RD}}$	OUT	309	A3	OUT
337	$\overline{\text{BS}}$	OUT	308	A4	OUT
336	D0	OUT	307	A5	OUT
335	D1	OUT	306	A6	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
305	A7	OUT	273	RD/$\overline{WR}$	Control
304	A8	OUT	272	$\overline{CAS}$	Control
303	A9	OUT	271	CKE	Control
302	A10	OUT	270	$\overline{RAS}$	Control
301	A11	OUT	269	$\overline{CS2}$	Control
300	A12	OUT	268	$\overline{CS3}$	Control
299	$\overline{REFOUT}$/$\overline{IRQOUT}$/$\overline{ARBUSY}$	Control	267	A0	Control
298	$\overline{BACK}$	Control	266	A1	Control
297	$\overline{CS0}$	Control	265	A2	Control
296	$\overline{CS4}$	Control	264	A3	Control
295	$\overline{CS5A}$	Control	263	A4	Control
294	$\overline{CS6A}$	Control	262	A5	Control
293	$\overline{RD}$	Control	261	A6	Control
292	$\overline{BS}$	Control	260	A7	Control
291	D0	Control	259	A8	Control
290	D1	Control	258	A9	Control
289	D2	Control	257	A10	Control
288	D3	Control	256	A11	Control
287	D4	Control	255	A12	Control
286	D5	Control	254	D16	IN
285	D6	Control	253	D17	IN
284	D7	Control	252	D18	IN
283	D8	Control	251	D19	IN
282	D9	Control	250	D20	IN
281	D10	Control	249	D21	IN
280	D11	Control	248	D22	IN
279	D12	Control	247	D23	IN
278	D13	Control	246	D24	IN
277	D14	Control	245	D25	IN
276	D15	Control	244	D26	IN
275	$\overline{WE0}$($\overline{BE0}$)/DQMLL	Control	243	D27	IN
274	$\overline{WE1}$($\overline{BE1}$)/DQMLU/$\overline{WE}$	Control	242	D28	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
241	D29	IN	210	D21	OUT
240	D30	IN	209	D22	OUT
239	D31	IN	208	D23	OUT
238	PTB0	IN	207	D24	OUT
237	PTB1/$\overline{\text{CTS1}}$	IN	206	D25	OUT
236	PTB2/$\overline{\text{RTS1}}$	IN	205	D26	OUT
235	PTB3/RXD1	IN	204	D27	OUT
234	PTB4/TXD1	IN	203	D28	OUT
233	PTB5/SCIF1CK	IN	202	D29	OUT
232	PTB6/$\overline{\text{CTS0}}$	IN	201	D30	OUT
231	PTB7/$\overline{\text{RTS0}}$	IN	200	D31	OUT
230	PTA0/RXD0	IN	199	A18	OUT
229	PTA1/TXD0	IN	198	A19	OUT
228	PTA2/SCIF0CK	IN	197	A20	OUT
227	PTA3/SCK_SIO0	IN	196	A21	OUT
226	PTA4/SIOMCLK0	IN	195	A22	OUT
225	PTA5/RXD_SIO0	IN	194	A23	OUT
224	PTA6/TXD_SIO0	IN	193	A24	OUT
223	PTA7/SIOFSYNC0	IN	192	A25	OUT
222	A13	OUT	191	PTB0	OUT
221	A14	OUT	190	PTB1/$\overline{\text{CTS1}}$	OUT
220	A15	OUT	189	PTB2/$\overline{\text{RTS1}}$	OUT
219	A16	OUT	188	PTB3/RXD1	OUT
218	A17	OUT	187	PTB4/TXD1	OUT
217	$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$	OUT	186	PTB5/SCIF1CK	OUT
216	$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$	OUT	185	PTB6/$\overline{\text{CTS0}}$	OUT
215	D16	OUT	184	PTB7/$\overline{\text{RTS0}}$	OUT
214	D17	OUT	183	PTA0/RXD0	OUT
213	D18	OUT	182	PTA1/TXD0	OUT
212	D19	OUT	181	PTA2/SCIF0CK	OUT
211	D20	OUT	180	PTA3/SCK_SIO0	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
179	PTA4/SIOMCLK0	OUT	148	A22	Control
178	PTA5/RXD_SIO0	OUT	147	A23	Control
177	PTA6/TXD_SIO0	OUT	146	A24	Control
176	PTA7/SIOFSYNC0	OUT	145	A25	Control
175	A13	Control	144	PTB0	Control
174	A14	Control	143	PTB1/$\overline{\text{CTS1}}$	Control
173	A15	Control	142	PTB2/$\overline{\text{RTS1}}$	Control
172	A16	Control	141	PTB3/RXD1	Control
171	A17	Control	140	PTB4/TXD1	Control
170	$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$	Control	139	PTB5/SCIF1CK	Control
169	$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$	Control	138	PTB6/$\overline{\text{CTS0}}$	Control
168	D16	Control	137	PTB7/$\overline{\text{RTS0}}$	Control
167	D17	Control	136	PTA0/RXD0	Control
166	D18	Control	135	PTA1/TXD0	Control
165	D19	Control	134	PTA2/SCIF0CK	Control
164	D20	Control	133	PTA3/SCK_SIO0	Control
163	D21	Control	132	PTA4/SIOMCLK0	Control
162	D22	Control	131	PTA5/RXD_SIO0	Control
161	D23	Control	130	PTA6/TXD_SIO0	Control
160	D24	Control	129	PTA7/SIOFSYNC0	Control
159	D25	Control	128	CRS1	IN
158	D26	Control	127	COL1	IN
157	D27	Control	126	TX-CLK1	IN
156	D28	Control	125	RX-ER1	IN
155	D29	Control	124	RX-CLK1	IN
154	D30	Control	123	RX-DV1	IN
153	D31	Control	122	ERXD10	IN
152	A18	Control	121	ERXD11	IN
151	A19	Control	120	ERXD12	IN
150	A20	Control	119	ERXD13	IN
149	A21	Control	118	MDIO1	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
117	LNKSTA1	IN	86	TX-EN0	OUT
116	CAMSEN1/IRQ5	IN	85	TX-ER0	OUT
115	CRS0	IN	84	MDC0	OUT
114	COL0	IN	83	MDIO0	OUT
113	TX-CLK0	IN	82	WOL0	OUT
112	RX-ER0	IN	81	EXOUT0/TEND0	OUT
111	RX-CLK0	IN	80	ETXD13	Control
110	RX-DV0	IN	79	ETXD12	Control
109	ERXD00	IN	78	ETXD11	Control
108	ERXD01	IN	77	ETXD10	Control
107	ERXD02	IN	76	TX-EN1	Control
106	ERXD03	IN	75	TX-ER1	Control
105	MDIO0	IN	74	MDC1	Control
104	LNKSTA0	IN	73	MDIO1	Control
103	CAMSEN0/IRQ4	IN	72	WOL1	Control
102	MD4	IN	71	EXOUT1	Control
101	MD5	IN	70	ETXD03	Control
100	ETXD13	OUT	69	ETXD02	Control
99	ETXD12	OUT	68	ETXD01	Control
98	ETXD11	OUT	67	ETXD00	Control
97	ETXD10	OUT	66	TX-EN0	Control
96	TX-EN1	OUT	65	TX-ER0	Control
95	TX-ER1	OUT	64	MDC0	Control
94	MDC1	OUT	63	MDIO0	Control
93	MDIO1	OUT	62	WOL0	Control
92	WOL1	OUT	61	EXOUT0	Control
91	EXOUT1/TEND1	OUT	60	NMI	IN
90	ETXD03	OUT	59	IRQ0/$\overline{\text{IRL0}}$	IN
89	ETXD02	OUT	58	IRQ1/$\overline{\text{IRL1}}$	IN
88	ETXD01	OUT	57	IRQ2/$\overline{\text{IRL2}}$	IN
87	ETXD00	OUT	56	IRQ3/$\overline{\text{IRL3}}$	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
55	DREQ0	IN	26	PTC4/SIOFSYNC1	OUT
54	DREQ1	IN	25	PTC5/$\overline{\text{CE2A}}$	OUT
53	PTC0/SCK_SIO1	IN	24	PTC6/$\overline{\text{CE2B}}$	OUT
52	PTC1/SIOMCLK1	IN	23	PTC7/$\overline{\text{IOIS16}}$	OUT
51	PTC2/RXD_SIO1	IN	22	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$	OUT
50	PTC3/TXD_SIO1	IN	21	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$	OUT
49	PTC4/SIOFSYNC1	IN	20	$\overline{\text{ASEBRKAK}}$	Control
48	PTC5/$\overline{\text{CE2A}}$	IN	19	$\overline{\text{AUDSYNC}}$	Control
47	PTC6/$\overline{\text{CE2B}}$	IN	18	AUDCK	Control
46	PTC7/$\overline{\text{IOIS16}}$	IN	17	AUDATA3	Control
45	MD0	IN	16	AUDATA2	Control
44	MD1	IN	15	AUDATA1	Control
43	MD2	IN	14	AUDATA0	Control
42	MD3	IN	13	STATUS0	Control
41	$\overline{\text{ASEBRKAK}}$	OUT	12	STATUS1	Control
40	$\overline{\text{AUDSYNC}}$	OUT	11	DACK0	Control
39	AUDCK	OUT	10	DACK1	Control
38	AUDATA3	OUT	9	PTC0/SCK_SIO1	Control
37	AUDATA2	OUT	8	PTC1/SIOMCLK1	Control
36	AUDATA1	OUT	7	PTC2/RXD_SIO1	Control
35	AUDATA0	OUT	6	PTC3/TXD_SIO1	Control
34	STATUS0	OUT	5	PTC4/SIOFSYNC1	Control
33	STATUS1	OUT	4	PTC5/$\overline{\text{CE2A}}$	Control
32	DACK0	OUT	3	PTC6/$\overline{\text{CE2B}}$	Control
31	DACK1	OUT	2	PTC7/$\overline{\text{IOIS16}}$	Control
30	PTC0/SCK_SIO1	OUT	1	$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$	Control
29	PTC1/SIOMCLK1	OUT	0	$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$	Control
28	PTC2/RXD_SIO1	OUT	To TDO		
27	PTC3/TXD_SIO1	OUT			

Note: Control is an active-low signal.
When Control is driven low, the corresponding pin is driven by the value of OUT.

23.3.4 ID Register (SDID)

The ID register (SDID) is a 32-bit read-only register in which SDIDH and SDIDL are connected. Each register is a 16-bit that can be read by CPU.

The IDCODE command is set from the H-UDI pin. This register can be read from the TDO when the TAP state is Shift-DR. Writing is disabled.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID31 to DID0	Refer to description	R	Device ID31 to 0 Device ID register that is stipulated by JTAG. Device ID in this LSI is H'001E200F. Upper four bits may be changed by the chip version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

23.4 Operation

23.4.1 TAP Controller

Figure 23.2 shows the internal states of the TAP controller. State transitions basically conform with the JTAG standard.

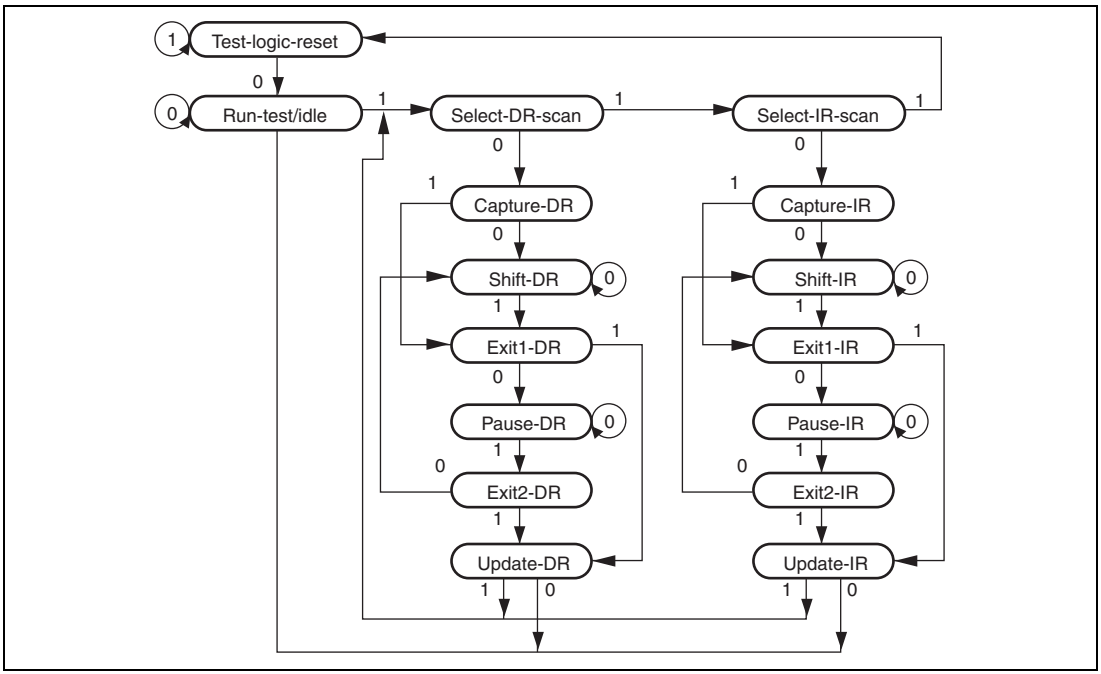


Figure 23.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on change timing of the TDO value, see section 23.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

23.4.2 Reset Configuration

Table 23.4 Reset Configuration

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RESETP}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Normal reset and H-UDI reset*4
		H	Normal reset*4
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	In ASE user mode*3: Normal reset In ASE break mode*3: $\overline{\text{RESETP}}$ assertion is masked
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode

2. In ASE mode, reset hold is enabled by driving the $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins low for a constant cycle. In this state, the CPU does not activate, even if $\overline{\text{RESETP}}$ is driven high. When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does not activate. The reset hold state is canceled by the following conditions:
   - Another $\overline{\text{RESETP}}$ assertion (power-on reset)
   - $\overline{\text{TRST}}$ reassertion
3. ASE mode is classified into two modes; ASE break mode to execute the firmware program of an emulator and ASE user mode to execute the user program.
4. Make sure the $\overline{\text{TRST}}$ pin is low when the power is turned on.

23.4.3 TDO Output Timing

The timing of data output from the TDO is switched by the command type set in the SDIR. The timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HIGHZ, SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG standard. When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI interrupt) are set, TDO is output at the TCK rising edge earlier than the JTAG standard by a half cycle.

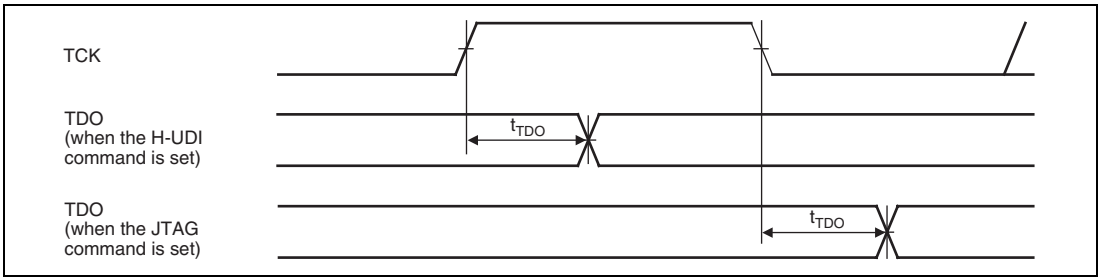


Figure 23.3 H-UDI Data Transfer Timing

23.4.4 H-UDI Reset

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

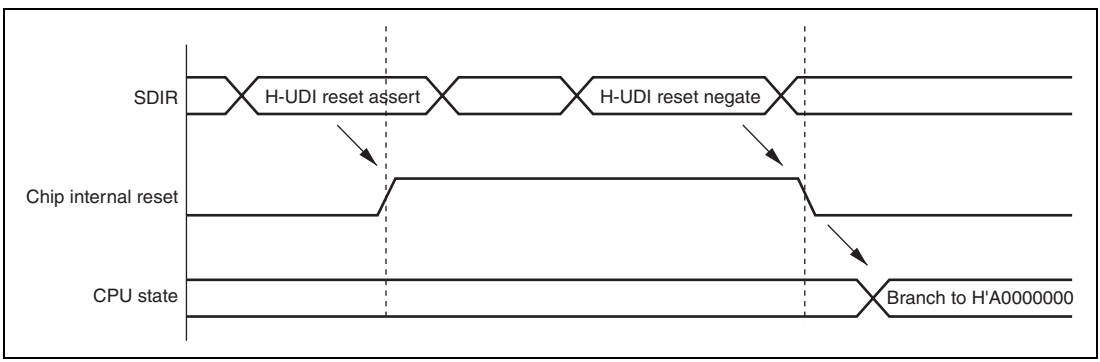


Figure 23.4 H-UDI Reset

23.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in the SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a branch to an address based on the VBR value plus offset, and with return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode.

23.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in the boundary scan mode stipulated by JTAG.

23.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

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

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this LSI's internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is executing, this LSI's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction. The upper four bits of the instruction code are 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when the this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The upper four bits of the instruction code are B'0000.

IDCODE: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the IDCODE mode stipulated by JTAG. When the H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

CLAMP, HIGHZ: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the CLAMP or HIGHZ mode stipulated by JTAG.

23.5.2 Points for Attention

1. Boundary scan mode does not cover clock-related signals (EXTAL, EXTAL2, XTAL, XTAL2, CKIO, CKIO2).
2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$).
4. Boundary scan mode does not cover the $\overline{\text{ASEMD0}}$ pin.
5. When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RESETP}}$ pin low.
6. When a boundary scan test for other than BYPASS and IDCODE is carried out, fix the $\overline{\text{ASEMD0}}$ pin high.

23.6 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not re-issued from the H-UDI. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect chip operations is once set.
2. In standby mode, the H-UDI function cannot be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

23.7 Advanced User Debugger (AUD)

The AUD is a function only for an emulator. For details on the AUD, refer to each emulator's user's manual.

Section 24 List of Registers

The address map gives information on the on-chip I/O registers and is configured as described below.

Register Addresses (by functional module, in order of the corresponding section numbers):

- Descriptions by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

Register Bits:

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- When registers consist of 16 or 32 bits, bits are described from the MSB side. The order in which bytes are described is on the presumption of a big-endian system.

Register States in Each Operating Mode:

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

24.1 Register Addresses (by functional module, in order of the corresponding section numbers)

Entries under Access Size indicates number of bits.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
INTEVT	Exception handling	L	H'FFFF FFD8	32	32
INTEVT2		L	H'A400 0000	32	32
TRA		L	H'FFFF FFD0	32	32
EXPEVT		L	H'FFFF FFD4	32	32
TEA		L	H'FFFF FFFC	32	32
MMUCR	MMU	L	H'FFFF FFE0	32	32
PTEH		L	H'FFFF FFF0	32	32
PTEL		L	H'FFFF FFF4	32	32
TTB		L	H'FFFF FFF8	32	32
CCR1	Cache	L	H'FFFF FFEC	32	32
CCR2		L	H'A400 00B0	32	32
CCR3		L	H'A400 00B4	32	32
IPRA	INTC	P	H'A414 FEE2	16	16
IPRB		P	H'A414 FEE4	16	16
IPRC		P	H'A414 0016	16	16
IPRD		P	H'A414 0018	16	16
IPRE		P	H'A414 001A	16	16
IPRF		P	H'A408 0000	16	16
IPRG		P	H'A408 0002	16	16
IPRH		P	H'A408 0004	16	16
IPRI		P	H'A408 0006	16	16
ICR0		P	H'A414 FEE0	16	16
ICR1		P	H'A414 0010	16	16

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
IRR0	INTC	P	H'A414 0004	8	8
IRR1		P	H'A414 0006	8	8
IRR2		P	H'A414 0008	8	8
IRR3		P	H'A414 000A	8	8
IRR4		P	H'A414 000C	8	8
IRR5		P	H'A408 0020	8	8
IRR7		P	H'A408 0024	8	8
IRR8		P	H'A408 0026	8	8
BARA	UBC	L	H'A4FF FFB0	32	32
BAMRA		L	H'A4FF FFB4	32	32
BBRA		L	H'A4FF FFB8	16	16
BARB		L	H'A4FF FFA0	32	32
BAMRB		L	H'A4FF FFA4	32	32
BBRB		L	H'A4FF FFA8	16	16
BDRB		L	H'A4FF FF90	32	32
BDMRB		L	H'A4FF FF94	32	32
BRCR		L	H'A4FF FF98	32	32
BETR		L	H'A4FF FF9C	16	16
BRSR		L	H'A4FF FFAC	32	32
BRDR		L	H'A4FF FFBC	32	32
BASRA		L	H'FFFF FFE4	8	8
BASRB		L	H'FFFF FFE8	8	8
STBCR	Power-down mode	P	H'A415 FF82	8	8
STBCR2		P	H'A415 FF88	8	8
STBCR3		P	H'A40A 0000	8	8
FRQCR	CPG	P	H'A415 FF80	16	16
WTCNT		P	H'A415 FF84	8	8/16*[4]
WTCSR		P	H'A415 FF86	8	8/16*[4]
CMNCR	BSC	I	H'A4FD 0000	32	32
CS0BCR		I	H'A4FD 0004	32	32
CS2BCR		I	H'A4FD 0008	32	32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
CS3BCR	BSC	I	H'A4FD 000C	32	32
CS4BCR		I	H'A4FD 0010	32	32
CS5ABCR		I	H'A4FD 0014	32	32
CS5BBCR		I	H'A4FD 0018	32	32
CS6ABCR		I	H'A4FD 001C	32	32
CS6BBCR		I	H'A4FD 0020	32	32
CS0WCR		I	H'A4FD 0024	32	32
CS2WCR		I	H'A4FD 0028	32	32
CS3WCR		I	H'A4FD 002C	32	32
CS4WCR		I	H'A4FD 0030	32	32
CS5AWCR		I	H'A4FD 0034	32	32
CS5BWCR		I	H'A4FD 0038	32	32
CS6AWCR		I	H'A4FD 003C	32	32
CS6BWCR		I	H'A4FD 0040	32	32
SDCR		I	H'A4FD 0044	32	32
RTCSR		I	H'A4FD 0048	32	32
RTCNT		I	H'A4FD 004C	32	32
RTCOR		I	H'A4FD 0050	32	32
SDMR2		I	H'A4FD 4xxx	—	16
SDMR3		I	H'A4FD 5xxx	—	16
SAR_0	DMAC	P	H'A401 0020	32	16/32
DAR_0		P	H'A401 0024	32	16/32
DMATCR_0		P	H'A401 0028	32	16/32
CHCR_0		P	H'A401 002C	32	8/16/32
SAR_1		P	H'A401 0030	32	16/32
DAR_1		P	H'A401 0034	32	16/32
DMATCR_1		P	H'A401 0038	32	16/32
CHCR_1		P	H'A401 003C	32	8/16/32
SAR_2		P	H'A401 0040	32	16/32
DAR_2		P	H'A401 0044	32	16/32
DMATCR_2		P	H'A401 0048	32	16/32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
CHCR_2	DMAC	P	H'A401 004C	32	8/16/32
SAR_3		P	H'A401 0050	32	16/32
DAR_3		P	H'A401 0054	32	16/32
DMATCR_3		P	H'A401 0058	32	16/32
CHCR_3		P	H'A401 005C	32	8/16/32
SAR_4		P	H'A401 0070	32	16/32
DAR_4		P	H'A401 0074	32	16/32
DMATCR_4		P	H'A401 0078	32	16/32
CHCR_4		P	H'A401 007C	32	8/16/32
SAR_5		P	H'A401 0080	32	16/32
DAR_5		P	H'A401 0084	32	16/32
DMATCR_5		P	H'A401 0088	32	16/32
CHCR_5		P	H'A401 008C	32	8/16/32
DMAOR		P	H'A401 0060	16	8/16
DMARS0		P	H'A409 0000	16	16
DMARS1		P	H'A409 0004	16	16
DMARS2		P	H'A409 0008	16	16
TSTR	TMU	P	H'A412 FE92	8	8
TCOR0		P	H'A412 FE94	32	32
TCNT0		P	H'A412 FE98	32	32
TCR0		P	H'A412 FE9C	16	16
TCOR1		P	H'A412 FEA0	32	32
TCNT1		P	H'A412 FEA4	32	32
TCR1		P	H'A412 FEA8	16	16
TCOR2		P	H'A412 FEAC	32	32
TCNT2		P	H'A412 FEB0	32	32
TCR2		P	H'A412 FEB4	16	16
R64CNT	RTC	P	H'A413 FEC0	8	8
RSECCNT		P	H'A413 FEC2	8	8
RMINCNT		P	H'A413 FEC4	8	8
RHRCNT		P	H'A413 FEC6	8	8

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
RWKCNT	RTC	P	H'A413 FEC8	8	8
RDAYCNT		P	H'A413 FECA	8	8
RMONCNT		P	H'A413 FECC	8	8
RYRCNT		P	H'A413 FECE	16	16
RSECAR		P	H'A413 FED0	8	8
RMINAR		P	H'A413 FED2	8	8
RHRAR		P	H'A413 FED4	8	8
RWKAR		P	H'A413 FED6	8	8
RDAYAR		P	H'A413 FED8	8	8
RMONAR		P	H'A413 FEDA	8	8
RCR1		P	H'A413 FEDC	8	8
RCR2		P	H'A413 FEDE	8	8
RYRAR		P	H'A413 FEE0	16	16
RCR3		P	H'A413 FEE4	8	8
SCSMR_0	SCIF	P	H'A440 0000	16	16
SCBRR_0		P	H'A440 0004	8	8
SCSCR_0		P	H'A440 0008	16	16
SCFTDR_0		P	H'A440 000C	8	8
SCFSR_0		P	H'A440 0010	16	16
SCFRDR_0		P	H'A440 0014	8	8
SCFCR_0		P	H'A440 0018	16	16
SCFDR_0		P	H'A440 001C	16	16
SCLSR_0		P	H'A440 0024	16	16
SCSMR_1		P	H'A441 0000	16	16
SCBRR_1		P	H'A441 0004	8	8
SCSCR_1		P	H'A441 0008	16	16
SCFTDR_1		P	H'A441 000C	8	8
SCFSR_1		P	H'A441 0010	16	16
SCFRDR_1		P	H'A441 0014	8	8
SCFCR_1		P	H'A441 0018	16	16

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
SCFDR_1	SCIF	P	H'A441 001C	16	16
SCLSR_1		P	H'A441 0024	16	16
SIMDR_0	SIOF	P	H'A442 0000	16	16
SISCR_0		P	H'A442 0002	16	16
SITDAR_0		P	H'A442 0004	16	16
SIRDAR_0		P	H'A442 0006	16	16
SICDAR_0		P	H'A442 0008	16	16
SICTR_0		P	H'A442 000C	16	16
SIFCTR_0		P	H'A442 0010	16	16
SISTR_0		P	H'A442 0014	16	16
SIIER_0		P	H'A442 0016	16	16
SITDR_0		P	H'A442 0020	32	32
SIRDR_0		P	H'A442 0024	32	32
SITCR_0		P	H'A442 0028	32	32
SIRCR_0		P	H'A442 002C	32	32
SIMDR_1		P	H'A443 0000	16	16
SISCR_1		P	H'A443 0002	16	16
SITDAR_1		P	H'A443 0004	16	16
SIRDAR_1		P	H'A443 0006	16	16
SICDAR_1		P	H'A443 0008	16	16
SICTR_1		P	H'A443 000C	16	16
SIFCTR_1		P	H'A443 0010	16	16
SISTR_1		P	H'A443 0014	16	16
SIIER_1		P	H'A443 0016	16	16
SITDR_1		P	H'A443 0020	32	32
SIRDR_1		P	H'A443 0024	32	32
SITCR_1		P	H'A443 0028	32	32
SIRCR_1		P	H'A443 002C	32	32
ECMR0	EtherC (MAC-0)	I	H'A700 0160	32	32
ECSR0		I	H'A700 0164	32	32
ECSIPR0		I	H'A700 0168	32	32

Abbreviation	Module*1	Bus*2	Address	Size (bit)	Access Size (bit)*3
PIR0	EtherC (MAC-0)	I	H'A700 016C	32	32
MAHR0		I	H'A700 0170	32	32
MALR0		I	H'A700 0174	32	32
RFLR0		I	H'A700 0178	32	32
PSR0		I	H'A700 017C	32	32
TROCR0		I	H'A700 0180	32	32
CDCR0		I	H'A700 0184	32	32
LCCR0		I	H'A700 0188	32	32
CNDCR0		I	H'A700 018C	32	32
CEFCR0		I	H'A700 0194	32	32
FRECR0		I	H'A700 0198	32	32
TSFRCR0		I	H'A700 019C	32	32
TLFRCR0		I	H'A700 01A0	32	32
RFCR0		I	H'A700 01A4	32	32
MAFCR0		I	H'A700 01A8	32	32
IPGR0		I	H'A700 01B4	32	32
ECMR1	EtherC (MAC-1)	I	H'A700 0560	32	32
ECSR1		I	H'A700 0564	32	32
ECSIPR1		I	H'A700 0568	32	32
PIR1		I	H'A700 056C	32	32
MAHR1		I	H'A700 0570	32	32
MALR1		I	H'A700 0574	32	32
RFLR1		I	H'A700 0578	32	32
PSR1		I	H'A700 057C	32	32
TROCR1		I	H'A700 0580	32	32
CDCR1		I	H'A700 0584	32	32
LCCR1		I	H'A700 0588	32	32
CNDCR1		I	H'A700 058C	32	32
CEFCR1		I	H'A700 0594	32	32
FRECR1		I	H'A700 0598	32	32
TSFRCR1		I	H'A700 059C	32	32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
TLFRCR1	EtherC (MAC-1)	I	H'A700 05A0	32	32
RFCR1		I	H'A700 05A4	32	32
MAFCR1		I	H'A700 05A8	32	32
IPGR1		I	H'A700 05B4	32	32
ARSTR	EtherC	I	H'A700 0800	32	32
TSU_CTRST	EtherC (TSU)	I	H'A700 0804	32	32
TSU_FWEN0		I	H'A700 0810	32	32
TSU_FWEN1		I	H'A700 0814	32	32
TSU_FCM		I	H'A700 0818	32	32
TSU_BSYSL0		I	H'A700 0820	32	32
TSU_BSYSL1		I	H'A700 0824	32	32
TSU_PRISL0		I	H'A700 0828	32	32
TSU_PRISL1		I	H'A700 082C	32	32
TSU_FWSL0		I	H'A700 0830	32	32
TSU_FWSL1		I	H'A700 0834	32	32
TSU_FWSLC		I	H'A700 0838	32	32
TSU_QTAGM0		I	H'A700 0840	32	32
TSU_QTAGM1		I	H'A700 0844	32	32
TSU_ADQT0		I	H'A700 0848	32	32
TSU_ADQT1		I	H'A700 084C	32	32
TSU_FWSR		I	H'A700 0850	32	32
TSU_FWINMK		I	H'A700 0854	32	32
TSU_ADSBSY		I	H'A700 0860	32	32
TSU_TEN		I	H'A700 0864	32	32
TSU_POST1		I	H'A700 0870	32	32
TSU_POST2		I	H'A700 0874	32	32
TSU_POST3		I	H'A700 0878	32	32
TSU_POST4		I	H'A700 087C	32	32
TXNLCR0		I	H'A700 0880	32	32
TXALCR0		I	H'A700 0884	32	32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
RXNLCR0	EtherC (TSU)	I	H'A700 0888	32	32
RXALCR0		I	H'A700 088C	32	32
FWNLCR0		I	H'A700 0890	32	32
FWALCR0		I	H'A700 0894	32	32
TXNLCR1		I	H'A700 08A0	32	32
TXALCR1		I	H'A700 08A4	32	32
RXNLCR1		I	H'A700 08A8	32	32
RXALCR1		I	H'A700 08AC	32	32
FWNLCR1		I	H'A700 08B0	32	32
FWALCR1		I	H'A700 08B4	32	32
TSU_ADRH0		I	H'A700 0900	32	32
TSU_ADRL0		I	H'A700 0904	32	32
TSU_ADRH1		I	H'A700 0908	32	32
TSU_ADRL1		I	H'A700 090C	32	32
TSU_ADRH2		I	H'A700 0910	32	32
TSU_ADRL2		I	H'A700 0914	32	32
TSU_ADRH3		I	H'A700 0918	32	32
TSU_ADRL3		I	H'A700 091C	32	32
TSU_ADRH4		I	H'A700 0920	32	32
TSU_ADRL4		I	H'A700 0924	32	32
TSU_ADRH5		I	H'A700 0928	32	32
TSU_ADRL5		I	H'A700 092C	32	32
TSU_ADRH6		I	H'A700 0930	32	32
TSU_ADRL6		I	H'A700 0934	32	32
TSU_ADRH7		I	H'A700 0938	32	32
TSU_ADRL7		I	H'A700 093C	32	32
TSU_ADRH8		I	H'A700 0940	32	32
TSU_ADRL8		I	H'A700 0944	32	32
TSU_ADRH9		I	H'A700 0948	32	32
TSU_ADRL9		I	H'A700 094C	32	32
TSU_ADRH10		I	H'A700 0950	32	32

Abbreviation	Module*1	Bus*2	Address	Size (bit)	Access Size (bit)*3
TSU_ADRL10	EtherC (TSU)	I	H'A700 0954	32	32
TSU_ADRH11		I	H'A700 0958	32	32
TSU_ADRL11		I	H'A700 095C	32	32
TSU_ADRH12		I	H'A700 0960	32	32
TSU_ADRL12		I	H'A700 0964	32	32
TSU_ADRH13		I	H'A700 0968	32	32
TSU_ADRL13		I	H'A700 096C	32	32
TSU_ADRH14		I	H'A700 0970	32	32
TSU_ADRL14		I	H'A700 0974	32	32
TSU_ADRH15		I	H'A700 0978	32	32
TSU_ADRL15		I	H'A700 097C	32	32
TSU_ADRH16		I	H'A700 0980	32	32
TSU_ADRL16		I	H'A700 0984	32	32
TSU_ADRH17		I	H'A700 0988	32	32
TSU_ADRL17		I	H'A700 098C	32	32
TSU_ADRH18		I	H'A700 0990	32	32
TSU_ADRL18		I	H'A700 0994	32	32
TSU_ADRH19		I	H'A700 0998	32	32
TSU_ADRL19		I	H'A700 099C	32	32
TSU_ADRH20		I	H'A700 09A0	32	32
TSU_ADRL20		I	H'A700 09A4	32	32
TSU_ADRH21		I	H'A700 09A8	32	32
TSU_ADRL21		I	H'A700 09AC	32	32
TSU_ADRH22		I	H'A700 09B0	32	32
TSU_ADRL22		I	H'A700 09B4	32	32
TSU_ADRH23		I	H'A700 09B8	32	32
TSU_ADRL23		I	H'A700 09BC	32	32
TSU_ADRH24		I	H'A700 09C0	32	32
TSU_ADRL24		I	H'A700 09C4	32	32
TSU_ADRH25		I	H'A700 09C8	32	32
TSU_ADRL25		I	H'A700 09CC	32	32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
TSU_ADRH26	EtherC (TSU)	I	H'A700 09D0	32	32
TSU_ADRL26		I	H'A700 09D4	32	32
TSU_ADRH27		I	H'A700 09D8	32	32
TSU_ADRL27		I	H'A700 09DC	32	32
TSU_ADRH28		I	H'A700 09E0	32	32
TSU_ADRL28		I	H'A700 09E4	32	32
TSU_ADRH29		I	H'A700 09E8	32	32
TSU_ADRL29		I	H'A700 09EC	32	32
TSU_ADRH30		I	H'A700 09F0	32	32
TSU_ADRL30		I	H'A700 09F4	32	32
TSU_ADRH31		I	H'A700 09F8	32	32
TSU_ADRL31		I	H'A700 09FC	32	32
EDMR0	E-DMAC0	I	H'A700 0000	32	32
EDTRR0		I	H'A700 0004	32	32
EDRRR0		I	H'A700 0008	32	32
TDLAR0		I	H'A700 000C	32	32
RDLAR0		I	H'A700 0010	32	32
EESR0		I	H'A700 0014	32	32
EESIPR0		I	H'A700 0018	32	32
TRSCER0		I	H'A700 001C	32	32
RMFCR0		I	H'A700 0020	32	32
TFTR0		I	H'A700 0024	32	32
FDR0		I	H'A700 0028	32	32
RMCR0		I	H'A700 002C	32	32
EDOCR0		I	H'A700 0030	32	32
FCFTR0		I	H'A700 0034	32	32
TRIMD0		I	H'A700 003C	32	32
RBWAR0		I	H'A700 0040	32	32
RDFAR0		I	H'A700 0044	32	32

Abbreviation	Module*[1]	Bus*[2]	Address	Size (bit)	Access Size (bit)*[3]
TBRAR0	E-DMAC0	I	H'A700 004C	32	32
TDFAR0		I	H'A700 0050	32	32
EDMR1	E-DMAC1	I	H'A700 0400	32	32
EDTRR1		I	H'A700 0404	32	32
EDRRR1		I	H'A700 0408	32	32
TDLAR1		I	H'A700 040C	32	32
RDLAR1		I	H'A700 0410	32	32
EESR1		I	H'A700 0414	32	32
EESIPR1		I	H'A700 0418	32	32
TRSCER1		I	H'A700 041C	32	32
RMFCR1		I	H'A700 0420	32	32
TFTR1		I	H'A700 0424	32	32
FDR1		I	H'A700 0428	32	32
RMCR1		I	H'A700 042C	32	32
EDOCR1		I	H'A700 0430	32	32
FCFTR1		I	H'A700 0434	32	32
TRIMD1		I	H'A700 043C	32	32
RBWAR1		I	H'A700 0440	32	32
RDFAR1		I	H'A700 0444	32	32
TBRAR1		I	H'A700 044C	32	32
TDFAR1		I	H'A700 0450	32	32
PACR	PFC	P	H'A405 0100	16	16
PBCR		P	H'A405 0102	16	16
PCCR		P	H'A405 0104	16	16
PETCR		P	H'A405 0106	16	16
PADR	I/O port	P	H'A405 0120	8	8
PBDR		P	H'A405 0122	8	8
PCDR		P	H'A405 0124	8	8
SDIR	H-UDI	P	H'A410 0200	16	16
SDID/SDIDH		P	H'A410 0214	32/16	32/16
SDIDL		P	H'A410 0216	16	16

Notes: 1. Module:

MMU:	Memory Management Unit
INTC:	Interrupt Controller
UBC:	User Break Controller
CPG:	Clock Pulse Generator
BSC:	Bus State Controller
DMAC:	Direct Memory Access Controller
TMU:	Timer Unit
RTC:	Realtime Clock
SCIF0:	Serial Communication Interface with FIFO 0
SCIF1:	Serial Communication Interface with FIFO 1
SIOF0:	Serial I/O with FIFO 0
SIOF1:	Serial I/O with FIFO 1
EtherC (MAC-0):	Ethernet Controller 0
EtherC (MAC-1):	Ethernet Controller 1
EtherC (TSU):	Transfer Unit for Ethernet Controller
E-DMAC0:	Ethernet Controller Direct Memory Access Controller 0
E-DMAC1:	Ethernet Controller Direct Memory Access Controller 1
IPSEC:	IP Security Accelerator
PFC:	Pin Function Controller
H-UDI:	User Debugging Interface

2. Bus:
   - L: Connected to the CPU, DSP, CCN, Cache, MMU, and UBC.
   - I: Connected to the BSC, CCN, Cache, DMAC, E-DMAC0, E-DMAC1, and IPSEC.
   - P: Connected to the BSC and peripheral modules (RTC, TMU, SCIF0, SCIF1, SIOF0, SIOF1, DMAC, PORT, INTC, H-UDI, CPG).
3. The access size indicates the size when accessing (read/write) the control registers. If an access is performed in the different size as shown above, the result is not correct data.
4. 16 bits when writing and 8 bits when reading.

24.2 Register Bits

Register bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TRA	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	TRA	TRA	
	TRA	TRA	TRA	TRA	TRA	TRA	—	—	
EXPEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	EXPEVT	EXPEVT	EXPEVT	EXPEVT	
	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	EXPEVT	
INTEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	INTEVT	INTEVT	INTEVT	INTEVT	
	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	INTEVT	
INTEVT2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	INTEVT2	INTEVT2	INTEVT2	INTEVT2	
	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	INTEVT2	—	—	
TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
	TEA	TEA	TEA	TEA	TEA	TEA	TEA	TEA	
PTEH	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	MMU
	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	
	VPN	VPN	VPN	VPN	VPN	VPN	—	—	
	ASID	ASID	ASID	ASID	ASID	ASID	ASID	ASID	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
PTEL	—	—	—	PPN	PPN	PPN	PPN	PPN	MMU
	PPN	PPN	PPN	PPN	PPN	PPN	PPN	PPN	
	PPN	PPN	PPN	PPN	PPN	PPN	—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									
MMUCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
CCR1	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	CB	WT	CE	
CCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LE	
	—	—	—	—	—	—	W3LOAD	W3LOCK	
	—	—	—	—	—	—	W2LOAD	W2LOCK	
CCR3	—	—	—	—	—	—	—	—	
	CSIZE7	CSIZE6	CSIZE5	CSIZE4	CSIZE3	CSIZE2	CSIZE1	CSIZE0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
IPRA	TMU0				TMU1				INTC
	TMU2				RTC				
IPRB	WDT				REF				
	—	—	—	—	—	—	—	—	
IPRC	IRQ3				IRQ2				
	IRQ1				IRQ0				

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
IPRD	—	—	—	—	—	—	—	—	INTC
	IRQ5				IRQ4				
IPRE	DMAC(1)					SCIF0			
	SCIF1				—	—	—	—	
IPRF	IPSEC					DMAC(2)			
	—	—	—	—	—	—	—	—	
IPRG	E-DMAC(1)					E-DMAC(2)			
	E-DMAC(3)				—	—	—	—	
IPRH	—	—	—	—	—	—	—	—	
	—	—	—	—		SIOF0			
IPRI	—	—	—	—	—	—	—	—	
		SIOF1			—	—	—	—	
ICR0	NMIL	—	—	—	—	—	—	NMIE	
	—	—	—	—	—	—	—	—	
ICR1	MAI	IRQLVL	BLMASK	IRLSEN	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IRR0	—	—	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	
IRR1	TXI0R	BRI0R	RXI0R	ERI0R	DEI3R	DEI2R	DEI1R	DEI0R	
IRR2	—	—	—	—	TXI1R	BRI1R	RXI1R	ERI1R	
IRR3	—	—	—	—	—	CUIR	PRIR	ATIR	
IRR4	—	TUNI2R	TUNI1R	TUNI0R	ITIR	—	—	RCMIR	
IRR5	IPSECIR	—	DEI5R	DEI4R	—	EINT2R	EINT1R	EINT0R	
IRR7	CCI0R	RXI0R	TXI0R	ERI0R	—	—	—	—	
IRR8	CCI1R	RXI1R	TXI1R	ERI1R	—	—	—	—	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	UBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	UBC
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BBRB	—	—	—	—	—	—	XYE	XYS	
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	BASMA	BASMB	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	
BETR	—	—	—	—	BET11	BET10	BET9	BET8	
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	UBC
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0	
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0	
STBCR	STBY	—	—	—	—	MSTP2	MSTP1	—	Power-down mode
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	—	MSTP3	
STBCR3	MSTP37	—	—	—	MSTP33	MEST32	MSTP31	MSTP30	
FRQCR	—	—	—	CKOEN	—	—	STC1	STC0	CPG
	—	—	IFC1	IFC0	—	PFC2	PFC1	PFC0	
WTCNT									
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	WAITSEL	BSD	—	MAP	BLOCK	DPRTY1	DPRTY0	DMAIW2	
	DMAIW1	DMAIW0	DMAIWA	—	ENDIAN	CK2DRV	HIZMEM	HIZCNT	
CSnBCR (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B)	—	IWW2	IWW1	IWW0	IWRWD2	IWRWD1	IWRWD0	IWRWS2	
	IWRWS1	IWRWS0	IWRRD2	IWRRD1	IWRRD0	IWRRS2	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
CS0WCR*[1]	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	—	—	—	BW1	BW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	—	—	—	—	—	W3	W2	W1	
	—	—	—	—	—	W3	W2	W1	
	WR0	WM	—	—	—	—	HW1	HW0	
	W0	WM	—	—	—	—	—	—	
	W0	WM	—	—	—	—	—	—	
CS2WCR*[2]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	—	—	—	—	—	—	—	A2CL1	
	WR0	WM	—	—	—	—	—	—	
	A2CL0	—	—	—	—	—	—	—	
CS3WCR*[2]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS3WCR*[2]	—	—	—	—	—	WR3	WR2	WR1	
	—	TRP1	TRP0	—	TRCD1	TRCD0	—	A3CL1	
	WR0	WM	—	—	—	—	—	—	
	A3CL0	—	—	TRWL1	TRWL0	—	TRC1	TRC0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
CS4WCR*[3]	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	BEN	—	—	BW1	BW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	—	—	—	SW1	SW0	PCW3W3	PCW2W2	PCW1W1	
	WR0	WM	—	—	—	—	HW1	HW0	
	PCW0W0	WM	—	—	—	—	HW1	HW0	
CS5A WCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR*[5]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	SA1	SA0	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	WR0	WM	—	—	—	—	HW1	HW0	
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	
CS6AWCR*[4]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6BWCR*[6]	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	SA1	SA0	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	WR0	WM	—	—	—	—	HW1	HW0	
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
SDCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	A2ROW1	A2ROW0	—	A2COL1	A2COL0	
	—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0	
RTCSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CMF	CMIE	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0	
RTCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RTCOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SAR_n (n = 0 to 5)									DMAC
DAR_n (n = 0 to 5)									
DMATCR_n (n = 0 to 5)									

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
CHCR_n (n = 0, 1)	—	—	—	—	—	—	—	—	DMAC
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
CHCR_m (m = 2 to 5)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	—	TB	TS1	TS0	IE	TE	DE	
DMAOR	—	—	—	—	—	—	PR1	PR0	
	—	—	—	—	—	AE	NMIF	DME	
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0	
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0	
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0	
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0	
DMARS2	C5MID5	C5MID4	C5MID3	C5MID2	C5MID1	C5MID0	C5RID1	C5RID0	
	C4MID5	C4MID4	C4MID3	C4MID2	C4MID1	C4MID0	C4RID1	C4RID0	
TSTR	—	—	—	—	—	STR2	STR1	STR0	TMU
TCRn (n = 0 to 2)	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0	
TCORn (n = 0 to 2)									
TCNTn (n = 0 to 2)									

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	RTC
RSECCNT	—	10-unit of second			1-unit of second				
RMINCNT	—	10-unit of minute			1-unit of minute				
RHRCNT	—	—	10-unit of hour		1-unit of hour				
RWKCNT	—	—	—	—	—		Day of week code		
RDAYCNT	—	—	10-unit of date		1-unit of date				
RMONCNT	—	—	—	10-unit of month	1-unit of month				
RYRCNT	1000-unit of year				100-unit of year				
	10-unit of year				1-unit of year				
RSECAR	ENB	10-unit of second			1-unit of second				
RMINAR	ENB	10-unit of minute			1-unit of minute				
RHRAR	ENB	—	10-unit of hour		1-unit of hour				
RWKAR	ENB	—	—	—	—		Day of week code		
RDAYAR	ENB	—	10-unit of date		1-unit of date				
RMONAR	ENB	—	—	10-unit of month	1-unit of month				
RYRAR	1000-unit of year				100-unit of year				
	10-unit of year				1-unit of year				
RCR1	CF	—	—	CIE	AIE	—	—	AF	
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	
RCR3	YAEN	—	—	—	—	—	—	—	
SCFRDR_n (n = 0, 1)									SCIF
SCFTDR_n (n = 0, 1)									
SCSMR_n (n = 0, 1)	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCSCR_n (n = 0, 1)	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
SCFSR_n (n = 0, 1)	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	SCIF
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCBRR_n (n = 0, 1)									
SCFCR_n (n = 0, 1)	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_n (n = 0, 1)	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCLSR_n (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SIMDR_n (n = 0, 1)	TRMD1	TRMD0	—	REDG	FL3	FL2	FL1	FL0	SIOF
	TXDIZ	LSBF	RCIM	—	—	—	—	—	
SISCR_n (n = 0, 1)	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0	
	—	—	—	—	—	BRDV2	BRDV1	BRDV0	
SITDAR_n (n = 0, 1)	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0	
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0	
SIRDAR_n (n = 0, 1)	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0	
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0	
SICDAR_n (n = 0, 1)	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0	
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0	
SICTR_n (n = 0, 1)	SCKE	FSE	—	—	—	—	TXE	RXE	
	—	—	—	—	—	—	TXRST	RXRST	
SIFCTR_n (n = 0, 1)	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0	
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0	
SISTR_n (n = 0, 1)	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	
	—	—	—	FSERR	TFOVR	TFUDR	RFUDR	RFOVR	
SIIER_n (n = 0, 1)	—	TCRDYE	TFEMPE	TDREQE	—	RCRDYE	RFFULE	RDREQE	
	—	—	—	FSERRE	TFOVRE	TFUDRE	RFUDRE	RFOVRE	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
SITDR_n (n = 0, 1)	SITDL15	SITDL14	SITDL13	SITDL12	SITDL11	SITDL10	SITDL9	SITDL8	SIOF
	SITDL7	SITDL6	SITDL5	SITDL4	SITDL3	SITDL2	SITDL1	SITDL0	
	SITDR15	SITDR14	SITDR13	SITDR12	SITDR11	SITDR10	SITDR9	SITDR8	
	SITDR7	SITDR6	SITDR5	SITDR4	SITDR3	SITDR2	SITDR1	SITDR0	
SIRDR_n (n = 0, 1)	SIRDL15	SIRDL14	SIRDL13	SIRDL12	SIRDL11	SIRDL10	SIRDL9	SIRDL8	
	SIRDL7	SIRDL6	SIRDL5	SIRDL4	SIRDL3	SIRDL2	SIRDL1	SIRDL0	
	SIRDR15	SIRDR14	SIRDR13	SIRDR12	SIRDR11	SIRDR10	SIRDR9	SIRDR8	
	SIRDR7	SIRDR6	SIRDR5	SIRDR4	SIRDR3	SIRDR2	SIRDR1	SIRDR0	
SITCR_n (n = 0, 1)	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08	
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00	
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18	
	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10	
SIRCR_n (n = 0, 1)	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08	
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00	
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18	
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10	
ARSTR	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ARST	
ECMRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	MCT	PRCEF	—	—	MPDE	—	
	—	RE	TE	—	ILB	ELB	DM	PRM	
ECSRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	LCHNG	MPD	ICD	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
ECSIPRn (n = 0, 1)	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	LCHNGIP	MPDIP	ICDIP	
PIRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	MDI	MDO	MMD	MDC	
MAHRn (n = 0, 1)	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40	
	MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32	
	MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24	
	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16	
MALRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	
	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	
RFLRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	RFL11	RFL10	RFL9	RFL8	
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0	
PSRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LMON	
TROCRn (n = 0, 1)	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24	
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16	
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8	
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
CDCRn (n = 0, 1)	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC24	EtherC
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC16	
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC8	
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC0	
LCCRn (n = 0, 1)	LCC31	LCC30	LCC29	LCC28	LCC27	LCC26	LCC25	LCC24	
	LCC23	LCC22	LCC21	LCC20	LCC19	LCC18	LCC17	LCC16	
	LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8	
	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0	
CNDCRn (n = 0, 1)	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC24	
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC16	
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8	
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0	
CEFCRn (n = 0, 1)	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC24	
	CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC16	
	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8	
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0	
FRECRn (n = 0, 1)	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24	
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16	
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8	
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0	
TSFRCRn (n = 0, 1)	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24	
	TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16	
	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8	
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0	
TLFRCRn (n = 0, 1)	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24	
	TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16	
	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8	
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
RFCRn (n = 0, 1)	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24	EtherC
	RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16	
	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8	
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0	
MAFCRn (n = 0, 1)	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24	
	MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16	
	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8	
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0	
IPGRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0	
TSU_CTRST	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	CTRST	
	—	—	—	—	—	—	—	—	
TSU_FWEN0	FWEN0	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TSU_FWEN1	FWEN1	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TSU_FCM	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	FCM2	FCM1	FCM0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TSU_ BSYSL0	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	BSYSL05	BSYSL04	BSYSL03	BSYSL02	BSYSL01	BSYSL00	
TSU_ BSYSL1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	BSYSL15	BSYSL14	BSYSL13	BSYSL12	BSYSL11	BSYSL10	
TSU_ PRISL0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	PRIMD02	PRIMD01	PRIMD00	—	—	—	—	
	PRISL07	PRISL06	PRISL05	PRISL04	PRISL03	PRISL02	PRISL01	PRISL00	
TSU_ PRISL1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	PRIMD12	PRIMD11	PRIMD10	—	—	—	—	
	PRISL17	PRISL16	PRISL15	PRISL14	PRISL13	PRISL12	PRISL11	PRISL10	
TSU_ FWSL0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FW40	FW30	FW20	FW10	
	—	—	—	—	—	—	—	—	
TSU_ FWSL1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FW41	FW31	FW21	FW11	
	—	—	—	—	—	—	—	—	
TSU_ FWSLC	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	POSTENU	POSTENL	—	—	—	—	
	CAMSEL03	CAMSEL02	CAMSEL01	CAMSEL00	CAMSEL13	CAMSEL12	CAMSEL11	CAMSEL10	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TSU_ QTAGM0	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	QTAGM01	QTAGM00	
TSU_ QTAGM1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	QTAGM11	QTAGM10	
TSU_ FWSR	—	—	—	—	TINT40	TINT30	TINT20	TINT10	
	OVF0	RBSY0	—	RINT50	RINT40	RINT30	RINT20	RINT10	
	—	—	—	—	TINT41	TINT31	TINT21	TINT11	
	OVF1	RBSY1	—	RINT51	RINT41	RINT31	RINT21	RINT11	
TSU_ FWINMK	—	—	—	—	TINTM40	TINTM30	TINTM20	TINTM10	
	OVFM0	RBSYM0	—	RINTM50	RINTM40	RINTM30	RINTM20	RINTM10	
	—	—	—	—	TINTM41	TINTM31	TINTM21	TINTM11	
	OVFM1	RBSYM1	—	RINTM51	RINTM41	RINTM31	RINTM21	RINTM11	
TSU_ ADQT0	QTAG031	QTAG030	QTAG029	QTAG028	QTAG027	QTAG026	QTAG025	QTAG024	
	QTAG023	QTAG022	QTAG021	QTAG020	QTAG019	QTAG018	QTAG017	QTAG016	
	QTAG015	QTAG014	QTAG013	—	QTAG011	QTAG010	QTAG009	QTAG008	
	QTAG007	QTAG006	QTAG005	QTAG004	QTAG003	QTAG002	QTAG001	QTAG000	
TSU_ ADQT1	QTAG131	QTAG130	QTAG129	QTAG128	QTAG127	QTAG126	QTAG125	QTAG124	
	QTAG123	QTAG122	QTAG121	QTAG120	QTAG119	QTAG118	QTAG117	QTAG116	
	QTAG115	QTAG114	QTAG113	—	QTAG111	QTAG110	QTAG109	QTAG108	
	QTAG107	QTAG106	QTAG105	QTAG104	QTAG103	QTAG102	QTAG101	QTAG100	
TSU_ ADSBSY	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ADSBSY	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TSU_TEN	TEN0	TEN1	TEN2	TEN3	TEN4	TEN5	TEN6	TEN7	EtherC
	TEN8	TEN9	TEN10	TEN11	TEN12	TEN13	TEN14	TEN15	
	TEN16	TEN17	TEN18	TEN19	TEN20	TEN21	TEN22	TEN23	
	TEN24	TEN25	TEN26	TEN27	TEN28	TEN29	TEN30	TEN31	
TSU_ POST1	POST03	POST02	POST01	POST00	POST13	POST12	POST11	POST10	
	POST23	POST22	POST21	POST20	POST33	POST32	POST31	POST30	
	POST43	POST42	POST41	POST40	POST53	POST52	POST51	POST50	
	POST63	POST62	POST61	POST60	POST73	POST72	POST71	POST70	
TSU_ POST2	POST83	POST82	POST81	POST80	POST93	POST92	POST91	POST90	
	POST103	POST102	POST101	POST100	POST113	POST112	POST111	POST110	
	POST123	POST122	POST121	POST120	POST133	POST132	POST131	POST130	
	POST143	POST142	POST141	POST140	POST153	POST152	POST151	POST150	
TSU_ POST3	POST163	POST162	POST161	POST160	POST173	POST172	POST171	POST170	
	POST183	POST182	POST181	POST180	POST193	POST192	POST191	POST190	
	POST203	POST202	POST201	POST200	POST213	POST212	POST211	POST210	
	POST223	POST222	POST221	POST220	POST233	POST232	POST231	POST230	
TSU_ POST4	POST243	POST242	POST241	POST240	POST253	POST252	POST251	POST250	
	POST263	POST262	POST261	POST260	POST273	POST272	POST271	POST270	
	POST283	POST282	POST281	POST280	POST293	POST292	POST291	POST290	
	POST303	POST302	POST301	POST300	POST313	POST312	POST311	POST310	
TSU_ ADRHn (n = 0 to 31)	ADRHn31	ADRHn30	ADRHn29	ADRHn28	ADRHn27	ADRHn26	ADRHn25	ADRHn24	
	ADRHn23	ADRHn22	ADRHn21	ADRHn20	ADRHn19	ADRHn18	ADRHn17	ADRHn16	
	ADRHn15	ADRHn14	ADRHn13	ADRHn12	ADRHn11	ADRHn10	ADRHn9	ADRHn8	
	ADRHn7	ADRHn6	ADRHn5	ADRHn4	ADRHn3	ADRHn2	ADRHn1	ADRHn0	
TSU_ ADRLn (n = 0 to 31)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	ADRLn15	ADRLn14	ADRLn13	ADRLn12	ADRLn11	ADRLn10	ADRLn9	ADRLn8	
	ADRLn7	ADRLn6	ADRLn5	ADRLn4	ADRLn3	ADRLn2	ADRLn1	ADRLn0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TXNLCR0	NTC031	NTC030	NTC029	NTC028	NTC027	NTC026	NTC025	NTC024	EtherC
	NTC023	NTC022	NTC021	NTC020	NTC019	NTC018	NTC017	NTC016	
	NTC015	NTC014	NTC013	NTC012	NTC011	NTC010	NTC009	NTC008	
	NTC007	NTC006	NTC005	NTC004	NTC003	NTC002	NTC001	NTC000	
TXALCR0	TC031	TC030	TC029	TC028	TC027	TC026	TC025	TC024	
	TC023	TC022	TC021	TC020	TC019	TC018	TC017	TC016	
	TC015	TC014	TC013	TC012	TC011	TC010	TC009	TC008	
	TC007	TC006	TC005	TC004	TC003	TC002	TC001	TC000	
RXNLCR0	NRC031	NRC030	NRC029	NRC028	NRC027	NRC026	NRC025	NRC024	
	NRC023	NRC022	NRC021	NRC020	NRC019	NRC018	NRC017	NRC016	
	NRC015	NRC014	NRC013	NRC012	NRC011	NRC010	NRC009	NRC008	
	NRC007	NRC006	NRC005	NRC004	NRC003	NRC002	NRC001	NRC000	
RXALCR0	RC031	RC030	RC029	RC028	RC027	RC026	RC025	RC024	
	RC023	RC022	RC021	RC020	RC019	RC018	RC017	RC016	
	RC015	RC014	RC013	RC012	RC011	RC010	RC009	RC008	
	RC007	RC006	RC005	RC004	RC003	RC002	RC001	RC000	
FWNLCR0	NFC031	NFC030	NFC029	NFC028	NFC027	NFC026	NFC025	NFC024	
	NFC023	NFC022	NFC021	NFC020	NFC019	NFC018	NFC017	NFC016	
	NFC015	NFC014	NFC013	NFC012	NFC011	NFC010	NFC009	NFC008	
	NFC007	NFC006	NFC005	NFC004	NFC003	NFC002	NFC001	NFC000	
FWALCR0	FC031	FC030	FC029	FC028	FC027	FC026	FC025	FC024	
	FC023	FC022	FC021	FC020	FC019	FC018	FC017	FC016	
	FC015	FC014	FC013	FC012	FC011	FC010	FC009	FC008	
	FC007	FC006	FC005	FC004	FC003	FC002	FC001	FC000	
TXNLCR1	NTC131	NTC130	NTC129	NTC128	NTC127	NTC126	NTC125	NTC124	
	NTC123	NTC122	NTC121	NTC120	NTC119	NTC118	NTC117	NTC116	
	NTC115	NTC114	NTC113	NTC112	NTC111	NTC110	NTC109	NTC108	
	NTC107	NTC106	NTC105	NTC104	NTC103	NTC102	NTC101	NTC100	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TXALCR1	TC131	TC130	TC129	TC128	TC127	TC126	TC125	TC124	EtherC
	TC123	TC122	TC121	TC120	TC119	TC118	TC117	TC116	
	TC115	TC114	TC113	TC112	TC111	TC110	TC109	TC108	
	TC107	TC106	TC105	TC104	TC103	TC102	TC101	TC100	
RXNLCR1	NRC131	NRC130	NRC129	NRC128	NRC127	NRC126	NRC125	NRC124	
	NRC123	NRC122	NRC121	NRC120	NRC119	NRC118	NRC117	NRC116	
	NRC115	NRC114	NRC113	NRC112	NRC111	NRC110	NRC109	NRC108	
	NRC107	NRC106	NRC105	NRC104	NRC103	NRC102	NRC101	NRC100	
RXALCR1	RC131	RC130	RC129	RC128	RC127	RC126	RC125	RC124	
	RC123	RC122	RC121	RC120	RC119	RC118	RC117	RC116	
	RC115	RC114	RC113	RC112	RC111	RC110	RC109	RC108	
	RC107	RC106	RC105	RC104	RC103	RC102	RC101	RC100	
FWNLCR1	NFC131	NFC130	NFC129	NFC128	NFC127	NFC126	NFC125	NFC124	
	NFC123	NFC122	NFC121	NFC120	NFC119	NFC118	NFC117	NFC116	
	NFC115	NFC114	NFC113	NFC112	NFC111	NFC110	NFC109	NFC108	
	NFC107	NFC106	NFC105	NFC104	NFC103	NFC102	NFC101	NFC100	
FWALCR1	FC131	FC130	FC129	FC128	FC127	FC126	FC125	FC124	
	FC123	FC122	FC121	FC120	FC119	FC118	FC117	FC116	
	FC115	FC114	FC113	FC112	FC111	FC110	FC109	FC108	
	FC107	FC106	FC105	FC104	FC103	FC102	FC101	FC100	
EDMRn (n = 0, 1)	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	DL1	DL0	—	—	—	SWR	
EDTRRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	TR	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
EDRRRn (n = 0, 1)	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RR	
TDLARn (n = 0, 1)	TDLA31	TDLA30	TDLA29	TDLA28	TDLA27	TDLA26	TDLA25	TDLA24	
	TDLA23	TDLA22	TDLA21	TDLA20	TDLA19	TDLA18	TDLA17	TDLA16	
	TDLA15	TDLA14	TDLA13	TDLA12	TDLA11	TDLA10	TDLA9	TDLA8	
	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0	
RDLARn (n = 0, 1)	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24	
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16	
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8	
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0	
EESRn (n = 0, 1)	—	TWB	—	—	—	TABT	RABT	RFCOF	
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF	
	—	—	—	—	CND	DLC	CD	TRO	
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF	
EESIPRn (n = 0, 1)	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP	
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP	
TRSCERn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RMAFCE	—	—	—	—	—	—	—	
RMFCRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8	
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TFTRn (n = 0, 1)	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFT10	TFT9	TFT8	
	TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1	TFT0	
FDRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFD2	TFD1	TFD0	
	—	—	—	—	—	RFD2	RFD1	RFD0	
RMCRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RNC	
EDOCRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FEC	AEC	—	—	
RBWARn (n = 0, 1)	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24	
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16	
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8	
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0	
RDFARn (n = 0, 1)	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24	
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16	
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8	
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0	
TBRARn (n = 0, 1)	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA24	
	TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA16	
	TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA8	
	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0	

Register Abbreviation	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0	Module
TDFARn (n = 0, 1)	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA24	E-DMAC
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA16	
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8	
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0	
FCFTRn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFF2	RFF1	RFF0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFD2	RFD1	RFD0	
TRIMDn (n = 0, 1)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	TIS	
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	I/O port
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	H-UDI
	—	—	—	—	—	—	—	—	
SDID/ SDIDH	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16	
SDIDL	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	

Notes: 1. Bit names in the first row of CS0WCR show the names for the normal/byte-selection SRAM interface, in the second row for the burst ROM (asynchronous) interface, and in the third row for the burst ROM (synchronous) interface.
2. Bit names in the first rows of CS2WCR and CS3WCR show the names for the normal/byte-selection SRAM interface and in the second rows for the SDRAM interface.
3. Bit names in the first row of CS4WCR show the names for the normal/byte-selection SRAM interface and in the second row for the burst ROM (asynchronous) interface.
4. Bit names of CS5AWCR and CS6AWCR show the names for the normal/byte-selection SRAM interface.
5. Bit names in the first rows of CS5BWCR and CS6BWCR show the names for the normal/byte-selection SRAM interface and in the second rows for the PCMCIA interface.

24.3 Register States in Each Operating Mode

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
INTEVT	Initialized	Initialized	Retained	—	Retained	Exception handling
INTEVT2	Initialized	Initialized	Retained	—	Retained	
TRA	Initialized	Initialized	Retained	—	Retained	
EXPEVT	Initialized	Initialized	Retained	—	Retained	
TEA	Initialized	Initialized	Retained	—	Retained	
MMUCR	Initialized	Initialized	Retained	Retained	Retained	MMU
PTEH	Initialized	Initialized	Retained	Retained	Retained	
PTEL	Initialized	Initialized	Retained	Retained	Retained	
TTB	Initialized	Initialized	Retained	Retained	Retained	
CCR1	Initialized	Initialized	Retained	Retained	Retained	Cache
CCR2	Initialized	Initialized	Retained	Retained	Retained	
CCR3	Initialized	Initialized	Retained	Retained	Retained	
IPRA	Initialized	Initialized	Retained	—	Retained	INTC
IPRB	Initialized	Initialized	Retained	—	Retained	
IPRC	Initialized	Initialized	Retained	—	Retained	
IPRD	Initialized	Initialized	Retained	—	Retained	
IPRE	Initialized	Initialized	Retained	—	Retained	
IPRF	Initialized	Initialized	Retained	—	Retained	
IPRG	Initialized	Initialized	Retained	—	Retained	
IPRH	Initialized	Initialized	Retained	—	Retained	
IPRI	Initialized	Initialized	Retained	—	Retained	
ICR0	Initialized	Initialized	Retained	—	Retained	
ICR1	Initialized	Initialized	Retained	—	Retained	
IRR0	Initialized	Initialized	Retained	—	Retained	
IRR1	Initialized	Initialized	Retained	—	Retained	
IRR2	Initialized	Initialized	Retained	—	Retained	
IRR3	Initialized	Initialized	Retained	—	Retained	
IRR4	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
IRR5	Initialized	Initialized	Retained*[2]	—	Retained	INTC
IRR7	Initialized	Initialized	Retained	—	Retained	
IRR8	Initialized	Initialized	Retained	—	Retained	
BARA	Initialized	Initialized	Retained	Retained	Retained	UBC
BAMRA	Initialized	Initialized	Retained	Retained	Retained	
BBRA	Initialized	Initialized	Retained	Retained	Retained	
BARB	Initialized	Initialized	Retained	Retained	Retained	
BAMRB	Initialized	Initialized	Retained	Retained	Retained	
BBRB	Initialized	Initialized	Retained	Retained	Retained	
BDRB	Initialized	Initialized	Retained	Retained	Retained	
BDMRB	Initialized	Initialized	Retained	Retained	Retained	
BRCR	Initialized	Initialized	Retained	Retained	Retained	
BETR	Initialized	Initialized	Retained	Retained	Retained	
BRSR	Initialized	Initialized	Retained	Retained	Retained	
BRDR	Initialized	Initialized	Retained	Retained	Retained	
BASRA	Initialized	Initialized	Retained	Retained	Retained	
BASRB	Initialized	Initialized	Retained	Retained	Retained	
STBCR	Initialized	Retained	Retained	—	Retained	Power-down mode
STBCR2	Initialized	Retained	Retained	—	Retained	
STBCR3	Initialized	Retained	Retained	—	Retained	
FRQCR	Initialized	Retained	Retained	—	Retained	CPG
WTCNT	Initialized	Initialized	Retained	—	Retained	
WTCSR	Initialized	Initialized	Retained	—	Retained	
CMNCR	Initialized	Retained	Retained	—	Retained	BSC
CS0BCR	Initialized	Retained	Retained	—	Retained	
CS2BCR	Initialized	Retained	Retained	—	Retained	
CS3BCR	Initialized	Retained	Retained	—	Retained	
CS4BCR	Initialized	Retained	Retained	—	Retained	
CS5ABCR	Initialized	Retained	Retained	—	Retained	
CS5BBCR	Initialized	Retained	Retained	—	Retained	
CS6ABCR	Initialized	Retained	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
CS6BBCR	Initialized	Retained	Retained	—	Retained	BSC
CS0WCR	Initialized	Retained	Retained	—	Retained	
CS2WCR	Initialized	Retained	Retained	—	Retained	
CS3WCR	Initialized	Retained	Retained	—	Retained	
CS4WCR	Initialized	Retained	Retained	—	Retained	
CS5AWCR	Initialized	Retained	Retained	—	Retained	
CS5BWCR	Initialized	Retained	Retained	—	Retained	
CS6AWCR	Initialized	Retained	Retained	—	Retained	
CS6BWCR	Initialized	Retained	Retained	—	Retained	
SDCR	Initialized	Retained	Retained	—	Retained	
RTCSR	Initialized	Retained	Retained	—	Retained	
RTCNT	Initialized	Retained	Retained	—	Retained	
RTCOR	Initialized	Retained	Retained	—	Retained	
SAR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	DMAC
DAR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
DMATCR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
CHCR_n (n = 0 to 5)	Initialized	Initialized	Retained	Retained	Retained	
DMAOR	Initialized	Initialized	Retained	Retained	Retained	
DMARS0	Initialized	Initialized	Retained	Retained	Retained	
DMARS1	Initialized	Initialized	Retained	Retained	Retained	
DMARS2	Initialized	Initialized	Retained	Retained	Retained	
TSTR	Initialized	Initialized	Initialized*[3]	Initialized	Retained	TMU
TCOR_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	
TCNT_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	
TCR_n (n = 0 to 2)	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
R64CNT	Retained	Retained	Retained	Retained	Retained	RTC
RSECCNT	Retained	Retained	Retained	Retained	Retained	
RMINCNT	Retained	Retained	Retained	Retained	Retained	
RHRCNT	Retained	Retained	Retained	Retained	Retained	
RWKCNT	Retained	Retained	Retained	Retained	Retained	
RDAYCNT	Retained	Retained	Retained	Retained	Retained	
RMONCNT	Retained	Retained	Retained	Retained	Retained	
RYRCNT	Retained	Retained	Retained	Retained	Retained	
RSECAR	Retained*[4]	Retained	Retained	Retained	Retained	
RMINAR	Retained*[4]	Retained	Retained	Retained	Retained	
RHRAR	Retained*[4]	Retained	Retained	Retained	Retained	
RWKAR	Retained*[4]	Retained	Retained	Retained	Retained	
RDAYAR	Retained*[4]	Retained	Retained	Retained	Retained	
RMONAR	Retained*[4]	Retained	Retained	Retained	Retained	
RYRAR	Retained	Retained	Retained	Retained	Retained	
RCR1	Initialized	Initialized	Retained	Retained	Retained	
RCR2	Initialized	Initialized*[5]	Retained	Retained	Retained	
RCR3	Initialized	Retained	Retained	Retained	Retained	
SCSMR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	SCIF
SCBRR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCSCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFTDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFSR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFRDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SCFCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
SCFDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	SCIF
SCLSR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIMDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	SIOF
SISCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SITDAR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIRDAR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SICDAR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SICTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIFCTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SISTR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIIER_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SITDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIRDR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SITCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
SIRCR_n (n = 0, 1)	Initialized	Initialized	Retained	Retained	Retained	
ECMRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	EtherC
ECSRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
ECSIPRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
PIRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	EtherC
MAHRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
MALRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RFLRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
PSRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TROCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
CDCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
LCCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
CNDCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
CEFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
FRECRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TSFRCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TLFRCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
MAFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
IPGRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
ARSTR	Initialized	Initialized	Retained	—	Retained	
TSU_CTRST	Initialized	Initialized	Retained	—	Retained	
TSU_FWEN0	Initialized	Initialized	Retained	—	Retained	
TSU_FWEN1	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
TSU_FCM	Initialized	Initialized	Retained	—	Retained	EtherC
TSU_BSYSL0	Initialized	Initialized	Retained	—	Retained	
TSU_BSYSL1	Initialized	Initialized	Retained	—	Retained	
TSU_PRISL0	Initialized	Initialized	Retained	—	Retained	
TSU_PRISL1	Initialized	Initialized	Retained	—	Retained	
TSU_FWSL0	Initialized	Initialized	Retained	—	Retained	
TSU_FWSL1	Initialized	Initialized	Retained	—	Retained	
TSU_FWSLC	Initialized	Initialized	Retained	—	Retained	
TSU_QTAGM0	Initialized	Initialized	Retained	—	Retained	
TSU_QTAGM1	Initialized	Initialized	Retained	—	Retained	
TSU_ADQT0	Initialized	Initialized	Retained	—	Retained	
TSU_ADQT1	Initialized	Initialized	Retained	—	Retained	
TSU_FWSR	Initialized	Initialized	Retained	—	Retained	
TSU_FWINMK	Initialized	Initialized	Retained	—	Retained	
TSU_ADSBSY	Initialized	Initialized	Retained	—	Retained	
TSU_TEN	Initialized	Initialized	Retained	—	Retained	
TSU_POST1	Initialized	Initialized	Retained	—	Retained	
TSU_POST2	Initialized	Initialized	Retained	—	Retained	
TSU_POST3	Initialized	Initialized	Retained	—	Retained	
TSU_POST4	Initialized	Initialized	Retained	—	Retained	
TXNLCR0	Initialized	Initialized	Retained	—	Retained	
TXALCR0	Initialized	Initialized	Retained	—	Retained	
RXNLCR0	Initialized	Initialized	Retained	—	Retained	
RXALCR0	Initialized	Initialized	Retained	—	Retained	
FWNLCR0	Initialized	Initialized	Retained	—	Retained	
FWALCR0	Initialized	Initialized	Retained	—	Retained	
TXNLCR1	Initialized	Initialized	Retained	—	Retained	
TXALCR1	Initialized	Initialized	Retained	—	Retained	
RXNLCR1	Initialized	Initialized	Retained	—	Retained	
RXALCR1	Initialized	Initialized	Retained	—	Retained	
FWNLCR1	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
FWALCR1	Initialized	Initialized	Retained	—	Retained	EtherC
TSU_ADRHn (n = 0 to 31)	Initialized	Initialized	Retained	—	Retained	
TSU_ADRLn (n = 0 to 31)	Initialized	Initialized	Retained	—	Retained	
EDMRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	E-DMAC
EDTRRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
EDRRRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TDLARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RDLARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
EESRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
EESIPRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TRSCERn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RMFCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TFTRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
FDRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RMCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
EDOCRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RBWARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
RDFARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	

Register Abbreviation	Power-on Reset*[1]	Manual Reset*[1]	Software standby	Module Standby	Sleep	Module
TBRARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	E-DMAC
TDFARn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
FCFTRn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
TRIMDn (n = 0, 1)	Initialized	Initialized	Retained	—	Retained	
PACR	Initialized	Retained	Retained	—	Retained	PFC
PBCR	Initialized	Retained	Retained	—	Retained	
PCCR	Initialized	Retained	Retained	—	Retained	
PETCR	Initialized	Retained	Retained	—	Retained	
PADR	Initialized	Retained	Retained	—	Retained	I/O port
PBDR	Initialized	Retained	Retained	—	Retained	
PCDR	Initialized	Retained	Retained	—	Retained	
SDIR	Retained	Retained	Retained	Retained	Retained	H-UDI
SDID/SDIDH	Retained	Retained	Retained	Retained	Retained	
SDIDL	Retained	Retained	Retained	Retained	Retained	

Notes: 1. For the initial values of each register, see the sections for the corresponding modules. If the initial value is undefined, it is shown as initialized since the data is not retained.
2. Some bits are initialized in standby mode. See section 8, Interrupt Controller (INTC), for details.
3. If the multiplication rate of PLL1 is modified, this register is initialized.
4. Some bits are initialized by a power-on reset. See section 15, Realtime Clock (RTC), for details.
5. Some bits are initialized by a manual reset. See section 15, Realtime Clock (RTC), for details.

Section 25 Electrical Characteristics

25.1 Absolute Maximum Ratings

Table 25.1 shows the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage (I/O)	$V_{CC}Q$ $V_{CC}Q$-RTC	–0.3 to 4.6	V
Power supply voltage (internal)	V_{CC} V_{CC}-PLL1 V_{CC}-PLL2	–0.3 to 2.1	V
Input voltage	V_{in}	–0.3 to $V_{CC}Q$ + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{stg}	–55 to 125	°C

Caution:

- Operating the chip in excess of the absolute maximum rating may result in permanent damage.
- Order of turning on 1.5 V power (Vcc, Vcc-PLL1, Vcc-PLL2) and 3.3 V power (VccQ, VccQ-RTC):
  1. The 3.3 V power and the 1.5 V power should be turned on simultaneously or the 3.3 V power should be tuned on first. When the 3.3 V is turned on first, turn on the 1.5 V power within 1 ms. It is recommended that this interval will be as short as possible.
  2. Until voltage is applied to all power supplies and a low level is input at the $\overline{\text{RESETP}}$ pin, internal circuits remain unsettled, and so pin states are also undefined. The system design must ensure that these undefined states do not cause erroneous system operation.
  3. When the power is turned on, make sure that the voltage of the 1.5 V power is lower than that of the 3.3 V power.

- Power-off order
  1. In the reverse order of powering-on, first turn off the 1.5 V power, then turn off the 3.3 V power within 1 ms. It is recommended that this interval will be as short as possible.
  2. Pin states are undefined while only the 1.5 V power is off. The system design must ensure that these undefined states do not cause erroneous system operation.
  3. When the power is turned off, make sure that the voltage of the 1.5 V power is lower than that of the 3.3 V power.

Waveforms and recommended times at power on/off are shown in figure 25.1.

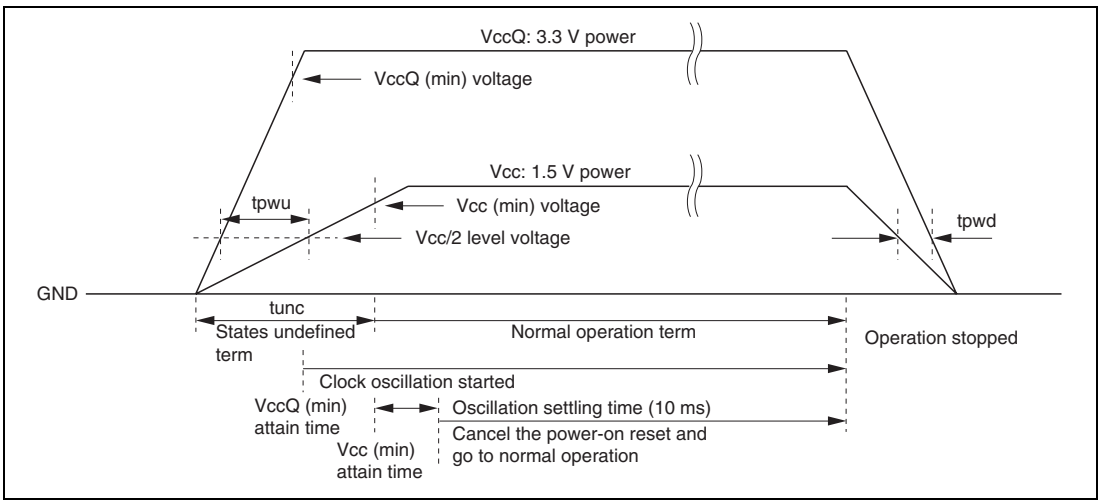


Figure 25.1 Power On/Off Sequence

Recommended Power On/Off Times

Item	Symbol	Max. Permitted Value	Unit
VccQ to Vcc power-on time interval	tpwu	1	ms
VccQ to Vcc power-off time interval	tpwd	1	ms
State undefined term	tunc	10	ms

Note: The recommended times shown above do not require strict settings.
The state undefined term indicates that pins are at the power rising stage. The pin state is stabilized at VccQ (min.) attain time. However, a power-on reset ($\overline{\text{RESETP}}$) is accepted successfully only after VccQ (min.) attain time and clock oscillation settling time. Set the state undefined term less than 10 ms.

25.2 DC Characteristics

Tables 25.2 and 25.3 list DC characteristics.

Table 25.2 DC Characteristics (1)

(Condition: Ta = –20 to 75°C)

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Power supply voltage		$V_{cc}Q$, $V_{cc}Q$-RTC	3.0	3.3	3.6	V	
		V_{cc}, V_{cc}-PLL1 V_{cc}-PLL2	1.4	1.5	1.6	V	
Current consumption	Normal operation	I_{cc}	—	250	330	mA	V_{cc} = 1.5 V Iϕ = 200 MHz
		$I_{cc}Q$	—	40	70	mA	Bϕ = 66.67 MHz
	In sleep mode*	I_{cc}	—	110	160	mA	$V_{cc}Q$ = 3.3 V
		$I_{cc}Q$	—	4	7		Bϕ = 66.67 MHz
	In standby mode	I_{cc}	—	1500	2600	µA	T_a = 25°C (RTC on) $V_{cc}Q$ = 3.3 V V_{cc} = 1.5 V
		$I_{cc}Q$	—	75	230		
		I_{cc}	—	1500	2600	µA	T_a = 25°C (RTC off) $V_{cc}Q$ = 3.3 V V_{cc} = 1.5 V
		$I_{cc}Q$	—	75	230		
Input leak current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $V_{cc}Q$ – 0.5 V
Three-state leak current	I/O, all output pins (off condition)	$\lvert I_{STI} \rvert$	—	—	1.0	µA	Vin = 0.5 to $V_{cc}Q$ – 0.5 V
Pull-up resistance	Port pin	R_{pull}	20	60	180	kΩ	
Pin capacitance	All pins	C	—	—	20	pF	

Note: * No external bus cycles except refresh cycles.

Table 25.2 DC Characteristics (2)

(Condition: Ta = –20 to 75°C)

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Input high voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI IRQ5 to IRQ0, MD5 to MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, CKIO	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	EXTAL2		—	—	—		When this pin is not connected to the crystal resonator, this pin should be connected to the $V_{CC}Q$ pin (pulled up).
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI IRQ5 to IRQ0, MD5 to MD0, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL, CKIO	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	EXTAL2		—	—	—		When this pin is not connected to the crystal resonator, this pin should be connected to the $V_{CC}Q$ pin (pulled up).
	Other input pins		–0.3	—	$V_{CC}Q \times 0.2$		
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V, I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	$V_{CC}Q$ = 3.0 V, I_{OL} = 2 mA

Notes: 1. Even when the RTC is not used, power must be supplied between $V_{CC}Q$-RTC and $V_{SS}Q$-RTC.

2. Current consumption values are for V_{IH} min. = $V_{CC}Q$ – 0.5 V and V_{IL} max. = 0.5 V with all output pins unloaded.

Table 25.3 Permitted Output Current Values

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, T_a = –20 to 75°C)

Item	Symbol	Min.	Typ.	Max.	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40	mA

Caution: To ensure LSI reliability, do not exceed the value for output current given in table 25.3.

25.3 AC Characteristics

In general, inputting for this LSI should be clock synchronous. Keep the setup and hold times for each input signal unless otherwise specified.

Table 25.4 Maximum Operating Frequencies

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, T_a = –20 to 75°C)

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU, cache (Iϕ)	f	33.34	—	200	MHz	
	External bus (Bϕ)		33.34	—	66.67		
	Peripheral module (Pϕ)		8.34	—	33.34		

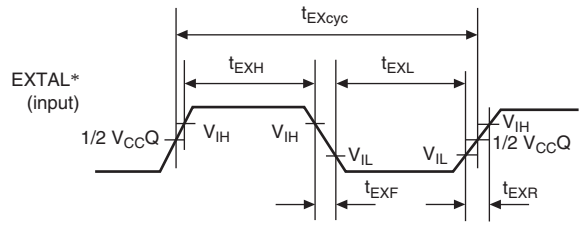
25.3.1 Clock Timing

Table 25.5 Clock Timing

(Condition: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C, maximum external bus operating frequency: 66.67 MHz)

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	66.67	MHz	25.2
EXTAL clock input cycle time	t_{EXcyc}	15	100	ns	
EXTAL clock input low pulse width	t_{EXL}	1.5	—		
EXTAL clock input high pulse width	t_{EXH}	1.5	—		
EXTAL clock input rise time	t_{EXR}	—	6		
EXTAL clock input fall time	t_{EXF}	—	6		
CKIO clock input frequency	f_{CKI}	33.34	66.67	MHz	25.3
CKIO clock input cycle time	t_{CKIcyc}	15	30	ns	
CKIO clock input low pulse width	t_{CKIL}	3	—		
CKIO clock input high pulse width	t_{CKIH}	3	—		
CKIO clock input rise time	t_{CKIR}	—	4		
CKIO clock input fall time	t_{CKIF}	—	4		
CKIO clock output frequency	f_{OP}	33.34	66.67	MHz	25.4
CKIO clock output cycle time	t_{cyc}	15	30	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—		
CKIO clock output high pulse width	t_{CKOH}	3	—		
CKIO clock output rise time	t_{CKOR}	—	4		
CKIO clock output fall time	t_{CKOF}	—	4		
CKIO2 clock output delay time	t_{CK2D}	—	2.5		
CKIO2 clock output rise time	t_{CK2OR}	—	7		
CKIO2 clock output fall time	t_{CK2OF}	—	7		
Power-on oscillation settling time	t_{OSC1}	10	—	ms	25.5
$\overline{\text{RESETP}}$ setup time	t_{RESPS}	20	—	ns	25.5
$\overline{\text{RESETP}}$ assert time	t_{RESPW}	20	—	t_{cyc}	25.5, 25.6
$\overline{\text{RESETM}}$ assert time	t_{RESMW}	20	—	t_{cyc}	25.6
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	25.6
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	25.7
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	25.8

Item	Symbol	Min.	Max.	Unit	Figure
PLL synchronization settling time 1	t_{PLL1}	100	—	µs	25.9, 25.10
PLL synchronization settling time 2	t_{PLL2}	100	—	µs	25.11
Interrupt determination time (RTC used and standby mode)	t_{IRLSTB}	100	—	µs	25.10



Note: * The clock input from the EXTAL pin.

Figure 25.2 EXTAL Clock Input Timing

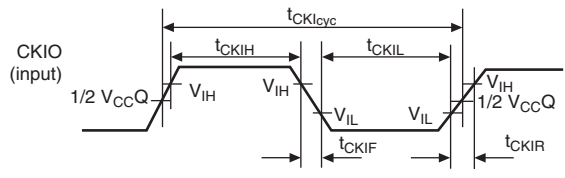


Figure 25.3 CKIO Clock Input Timing

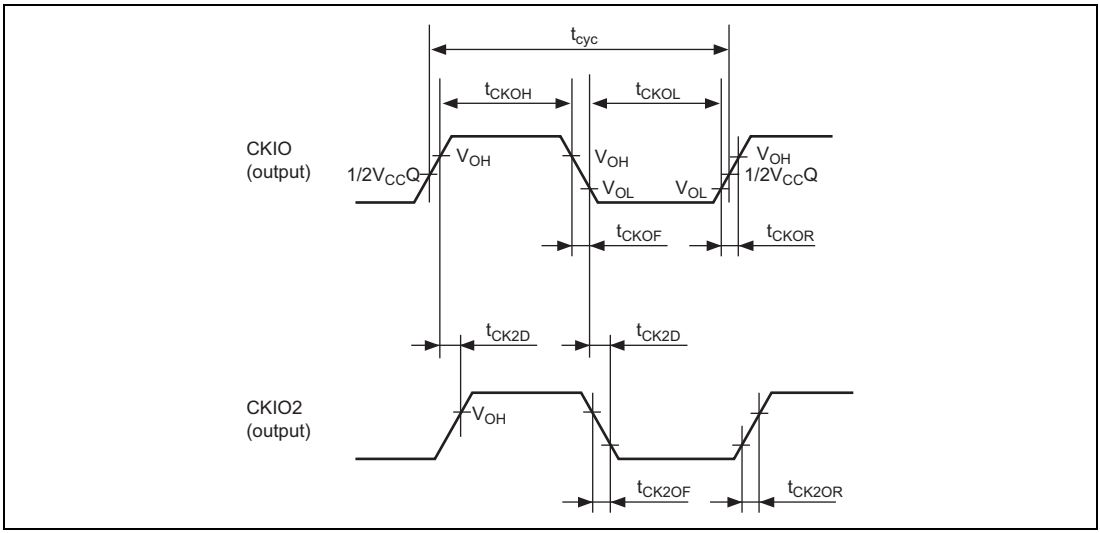


Figure 25.4 CKIO Clock Output Timing

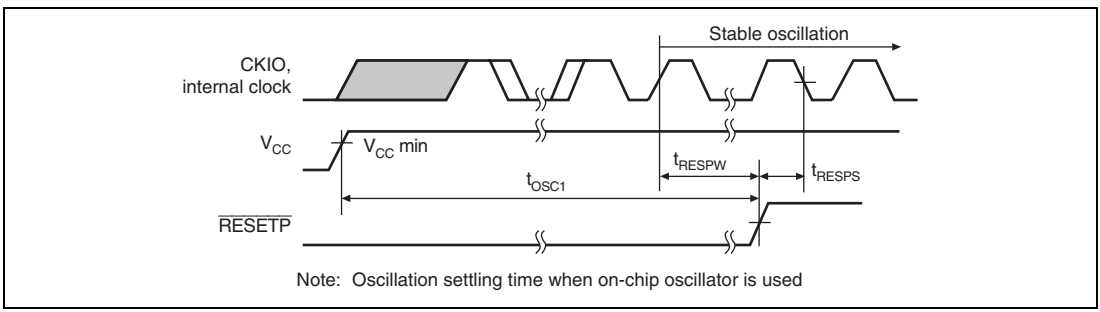


Figure 25.5 Power-On Oscillation Settling Time

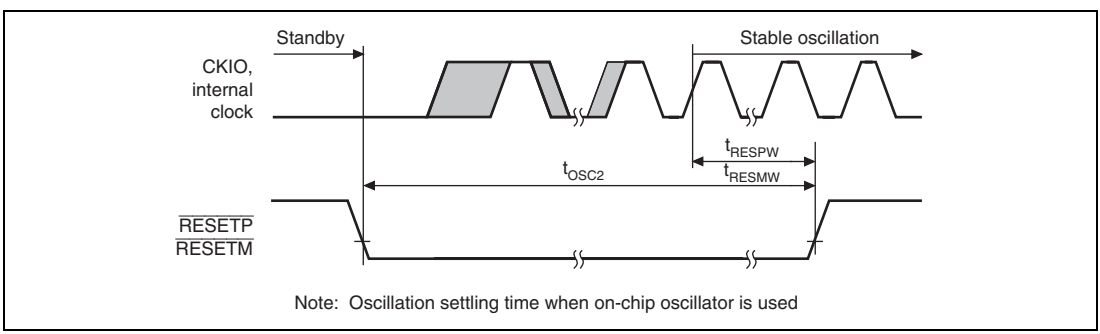
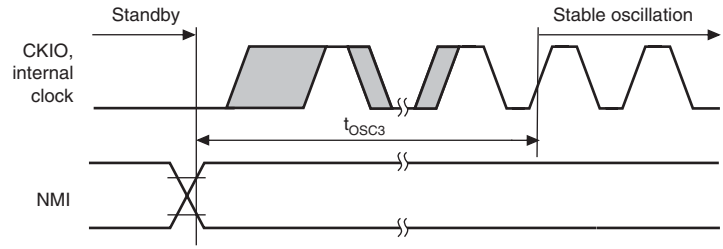
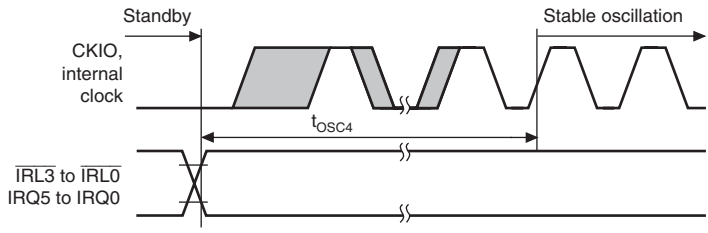


Figure 25.6 Oscillation Settling Time at Standby Return (Return by Reset)



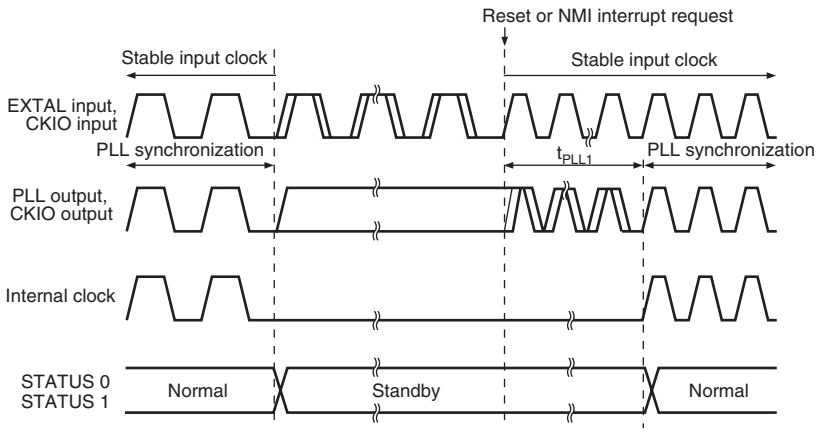
Note: Oscillation settling time when on-chip oscillator is used

Figure 25.7 Oscillation Settling Time at Standby Return (Return by NMI)



Note: Oscillation settling time when on-chip oscillator is used in oscillation stop mode

Figure 25.8 Oscillation Settling Time at Standby Return (Return by IRQ5 to IRQ0 and $\overline{IRL3}$ to $\overline{IRL0}$)



Note: PLL oscillation settling time when clock is input from EXTAL pin

Figure 25.9 PLL Synchronization Settling Time by Reset or NMI

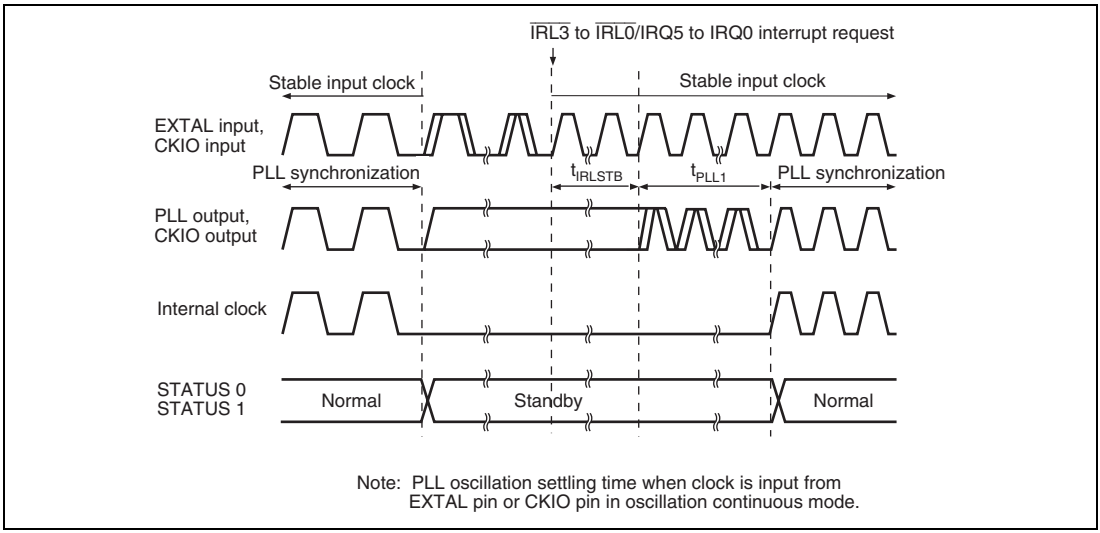


Figure 25.10 PLL Synchronization Settling Time by IRQ/IRL Interrupts

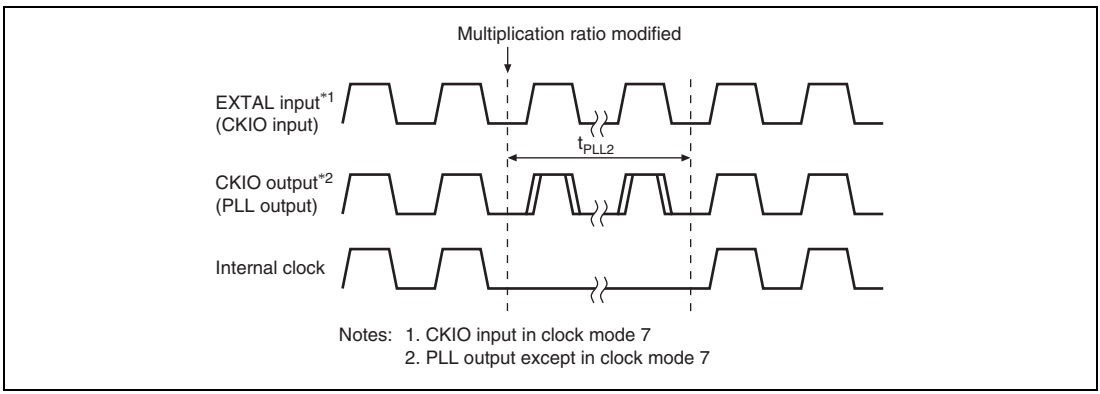


Figure 25.11 PLL Synchronization Settling Time when Frequency Multiplication Ratio Modified

25.3.2 Control Signal Timing

Table 25.6 Control Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

		66.67 MHz*[2]			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20*[3]	—	t_{cyc}	25.12
$\overline{\text{RESETP}}$ setup time*[1]	t_{RESPS}	20	—	ns	
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	20*[4]	—	t_{cyc}	
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	10	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	1/2 t_{cyc}+10	—		25.14
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1/2 t_{cyc}+3	—		
NMI setup time*[1]	t_{NMIS}	10	—		25.13
NMI hold time	t_{NMIH}	3	—		
IRQ5 to IRQ0 setup time*[1]	t_{IRQS}	10	—		
IRQ5 to IRQ0 hold time	t_{IRQH}	3	—		
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	1/2 t_{cyc}+13		25.14
STATUS1, STATUS0 delay time	t_{STD}	—	18		25.15
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOTD}	—	1/2 t_{cyc}+12		25.16
Bus tri-state delay time 1	t_{BOFF1}	0	30		25.14,
Bus tri-state delay time 2	t_{BOFF2}	0	30		25.15
Bus buffer-on time 1	t_{BON1}	0	30		
Bus buffer-on time 2	t_{BON2}	0	30		

Notes: t_{cyc} is the external bus clock cycle (B clock cycle).

1. $\overline{\text{RESETP}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock rise when the setup shown is kept. When the setup cannot be kept, detection can be delayed until the next clock rises.
2. The upper limit of the external bus clock is 66.67 MHz.
3. In standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). When the crystal oscillation continues or the clock multiplication ratio is changed in standby mode, $t_{RESPW} = t_{PLL1}$ (100 µs).
4. In standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). When the crystal oscillation continues or the clock multiplication ratio is changed in standby mode, $\overline{\text{RESETM}}$ must be kept low until STATUS (0-1) changes to reset (HH).

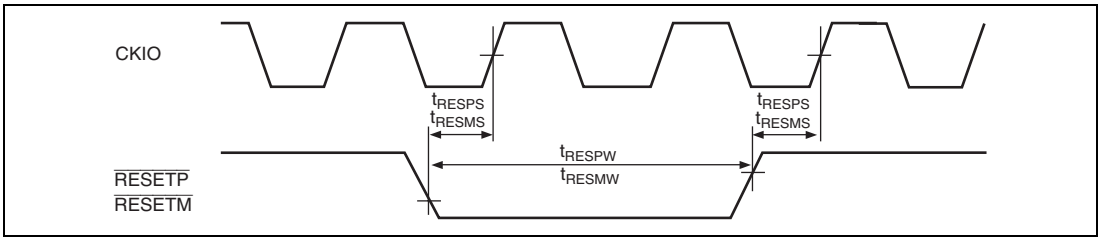


Figure 25.12 Reset Input Timing

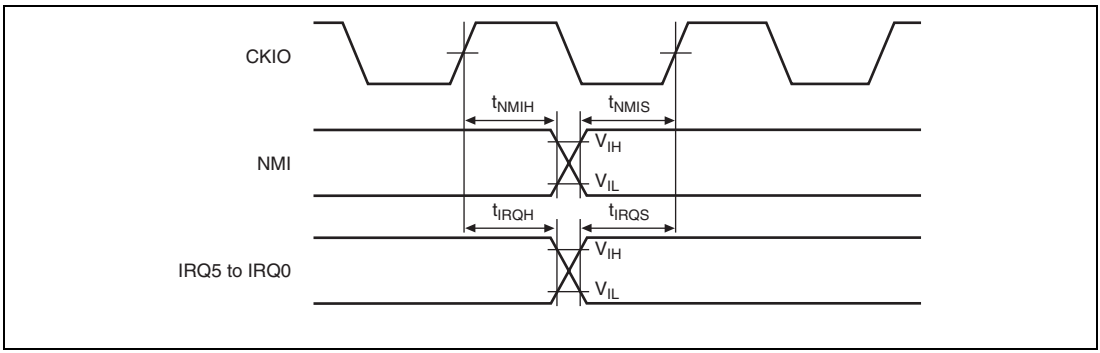


Figure 25.13 Interrupt Signal Input Timing

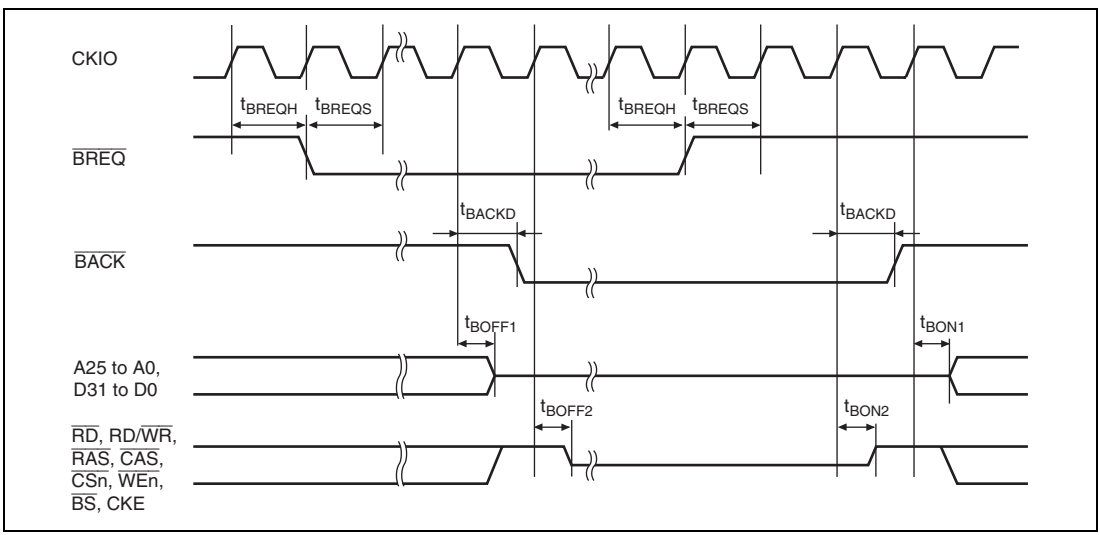


Figure 25.14 Bus Release Timing

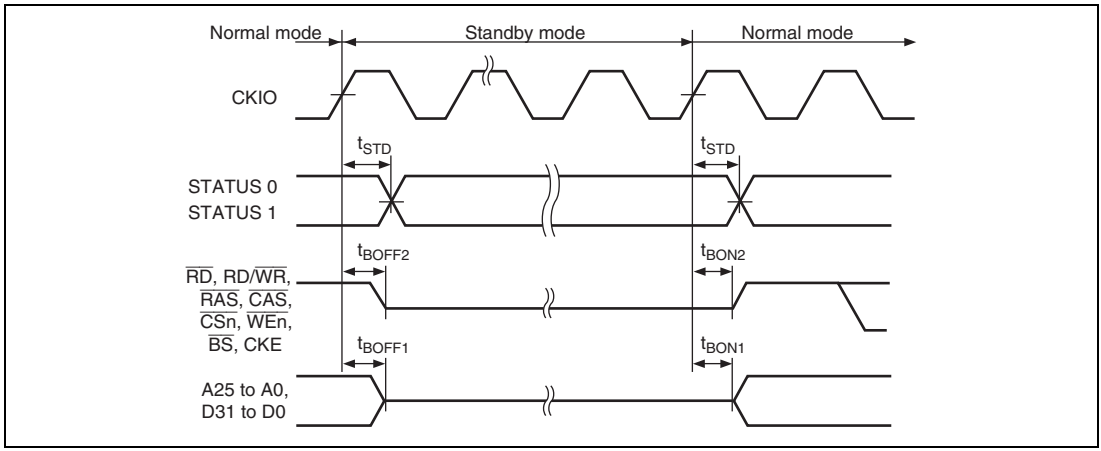


Figure 25.15 Pin Drive Timing at Standby

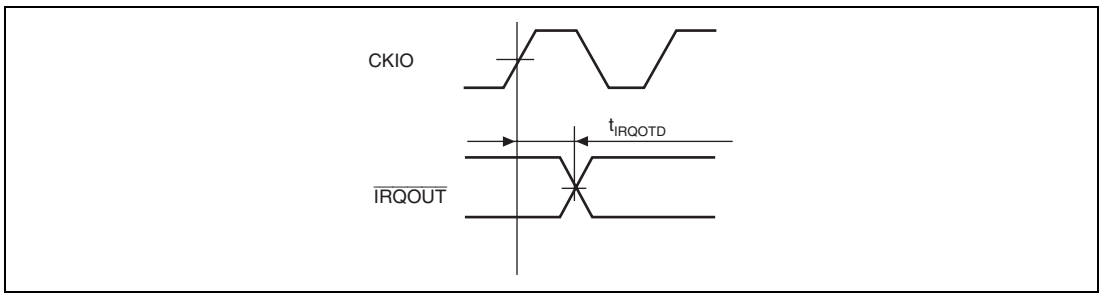


Figure 25.16 $\overline{\text{IRQOUT}}$ Output Delay Time

25.3.3 AC Bus Timing

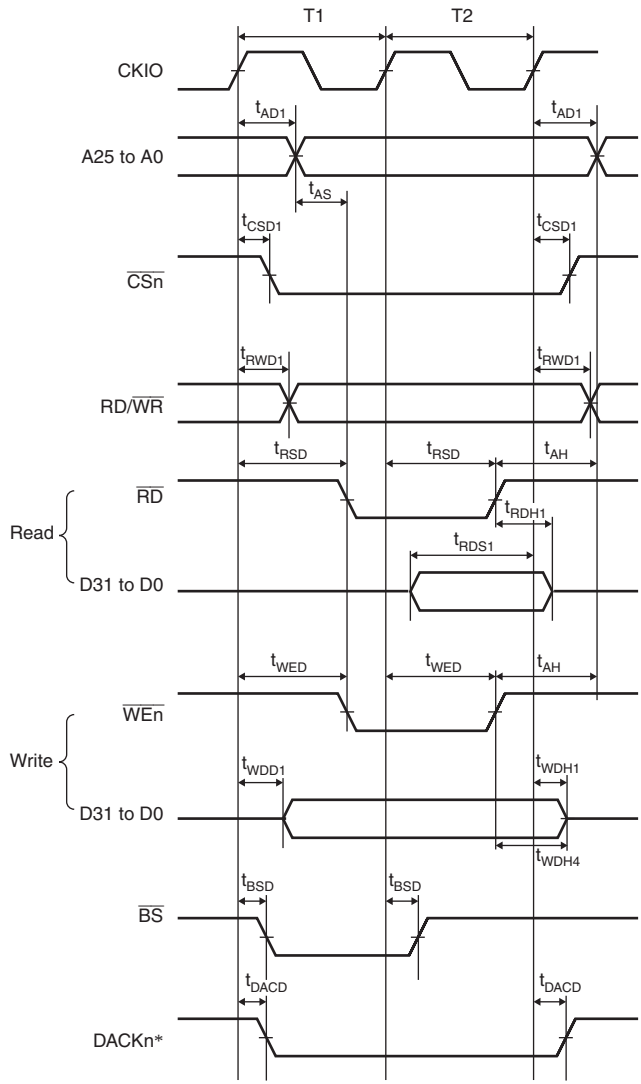
Table 25.7 Bus Timing (1)

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C, clock mode 0/1/2/4/5/6/7)

		66.67 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	12	ns	25.17 to 25.42
Address delay time 2	t_{AD2}	—	1/2 t_{cyc}+12		25.21
Address setup time	t_{AS}	0	—		25.17 to 25.20
Address hold time	t_{AH}	0	—		
$\overline{BS}$ delay time	t_{BSD}	—	10		25.17 to 25.35, 25.39 to 25.42
$\overline{CS}$ delay time 1	t_{CSD1}	1	10		25.17 to 25.42
Read/write delay time 1	t_{RWD1}	1	10		
Read strobe delay time	t_{RSD}	—	1/2 t_{cyc}+10		25.17 to 25.21, 25.39 to 25.40
Read data setup time 1	t_{RDS1}	1/2 t_{cyc}+6	—		25.17 to 25.20, 25.39 to 25.42
Read data setup time 2	t_{RDS2}	6	—		25.22 to 25.25, 25.30 to 25.32
Read data setup time 3	t_{RDS3}	1/2 t_{cyc}+6	—		25.21
Read data hold time 1	t_{RDH1}	0	—		25.17 to 25.20, 25.39 to 25.42
Read data hold time 2	t_{RDH2}	2	—		25.22 to 25.25, 25.30 to 25.32
Read data hold time 3	t_{RDH3}	0	—		25.21
Write enable delay time	t_{WED}	—	1/2 t_{cyc}+10		25.17 to 25.21, 25.39 to 25.40
Write data delay time 1	t_{WDD1}	—	12		25.17 to 25.20, 25.39 to 25.42
Write data delay time 2	t_{WDD2}	—	12		25.26 to 25.29, 25.33 to 25.35
Write data hold time 1	t_{WDH1}	1	—		25.17 to 25.20
Write data hold time 2	t_{WDH2}	1	—		25.26 to 25.29, 25.33 to 25.35

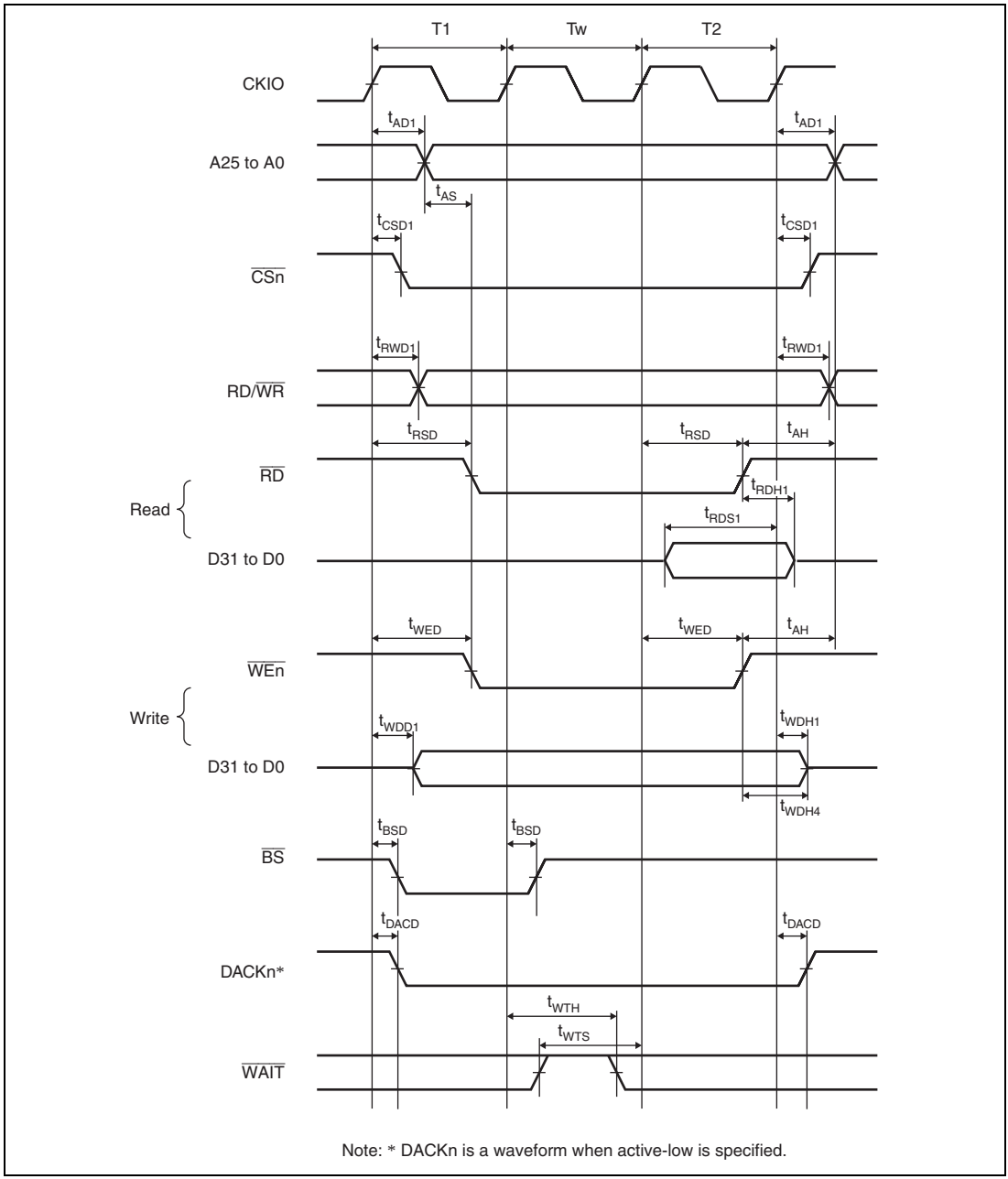
Item	Symbol	66.67 MHz Min.	66.67 MHz Max.	Unit	Figure
Write data hold time 4	t_{WDH4}	0	—	ns	25.17 to 25.20
Write data hold time 5	t_{WDH5}	1	—		25.39 to 25.42
$\overline{WAIT}$ setup time	t_{WTS}	1/2 t_{cyc}+6	—		25.18 to 25.21, 25.40,
$\overline{WAIT}$ hold time	t_{WTH}	1/2 t_{cyc}+2	—		25.42
$\overline{RAS}$ delay time 1	t_{RASD1}	1	10		25.22 to 25.38
$\overline{CAS}$ delay time 1	t_{CASD1}	1	10		
DQM delay time 1	t_{DQMD1}	1	10		25.22 to 25.35
CKE delay time 1	t_{CKED1}	1	10		25.37
DACK delay time	t_{DACD}	—	10		25.17 to 25.35
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	1/2 t_{cyc}+12		25.41, 25.42
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	1/2 t_{cyc}+12		
$\overline{IOIS16}$ setup time	t_{IO16S}	1/2 t_{cyc}+12	—		25.42
$\overline{IOIS16}$ hold time	t_{IO16H}	1/2 t_{cyc}+4	—		
$\overline{REFOUT}$ delay time	t_{REFOD}	—	1/2 t_{cyc}+12		25.43

25.3.4 Basic Timing



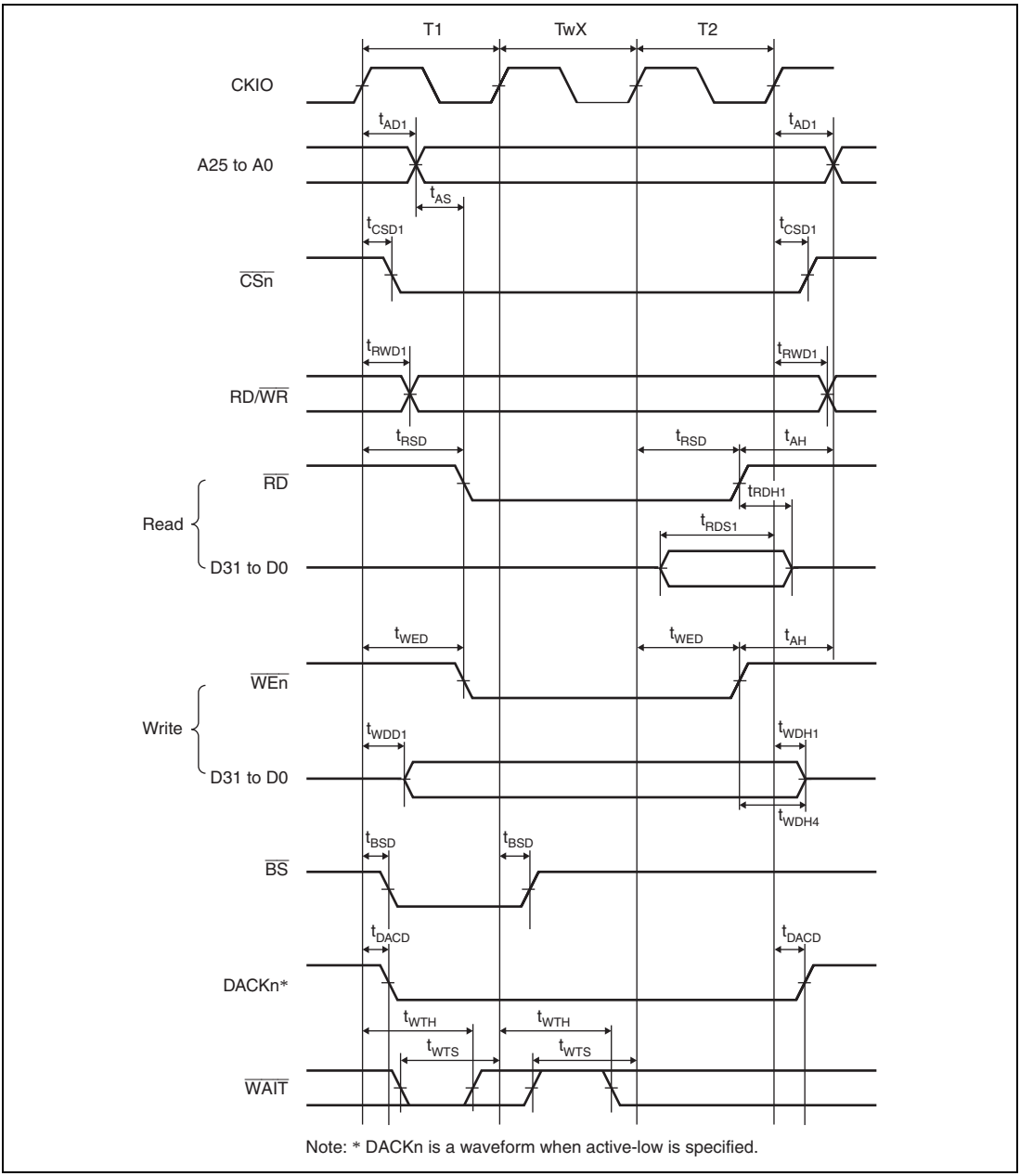
Note: * DACKn is a waveform when active-low is specified.

Figure 25.17 Basic Bus Cycle (No Wait)



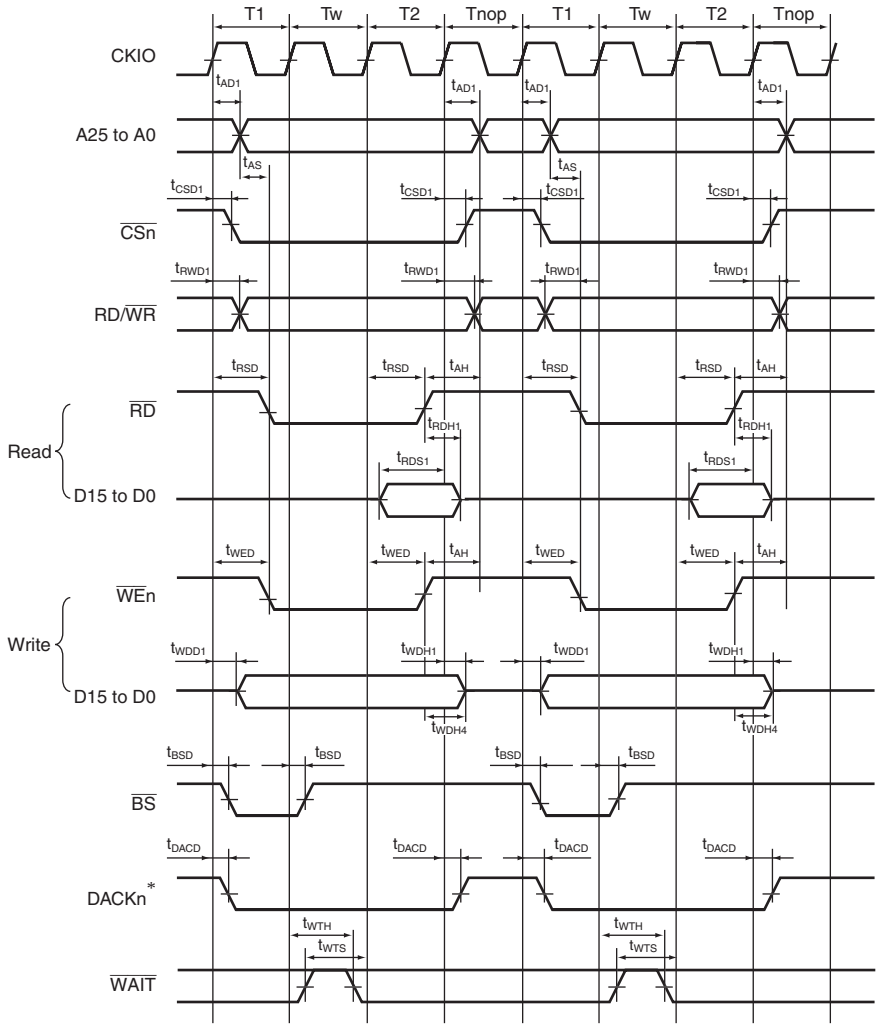
Note: * DACKn is a waveform when active-low is specified.

Figure 25.18 Basic Bus Cycle (One Software Wait)



Note: * DACKn is a waveform when active-low is specified.

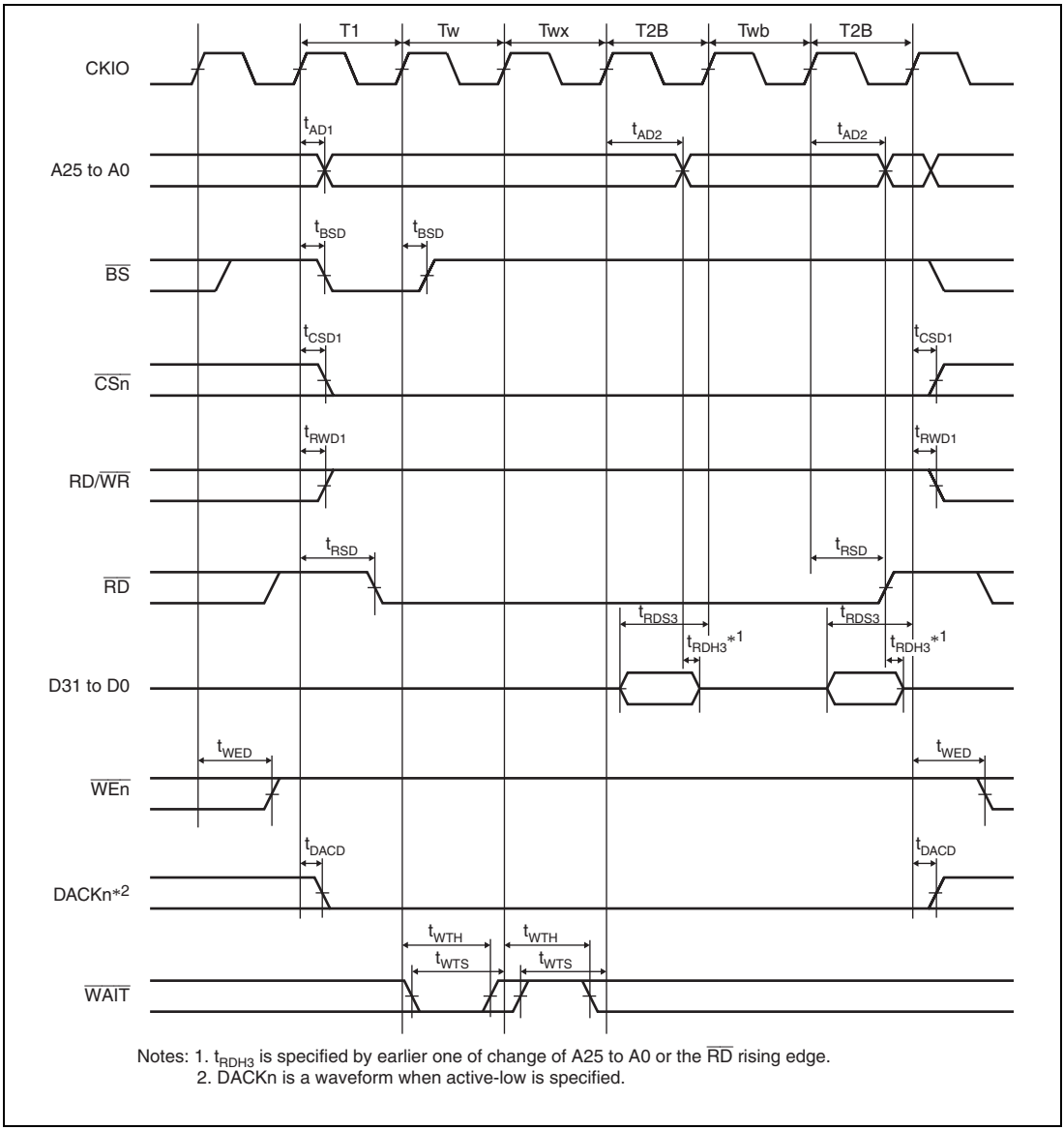
Figure 25.19 Basic Bus Cycle (One External Wait)



Note: * DACKn is a waveform when active-low is specified.

Figure 25.20 Basic Bus Cycle (One Software Wait, External Wait Enabled (WM bit = 0), No Idle Cycle Setting)

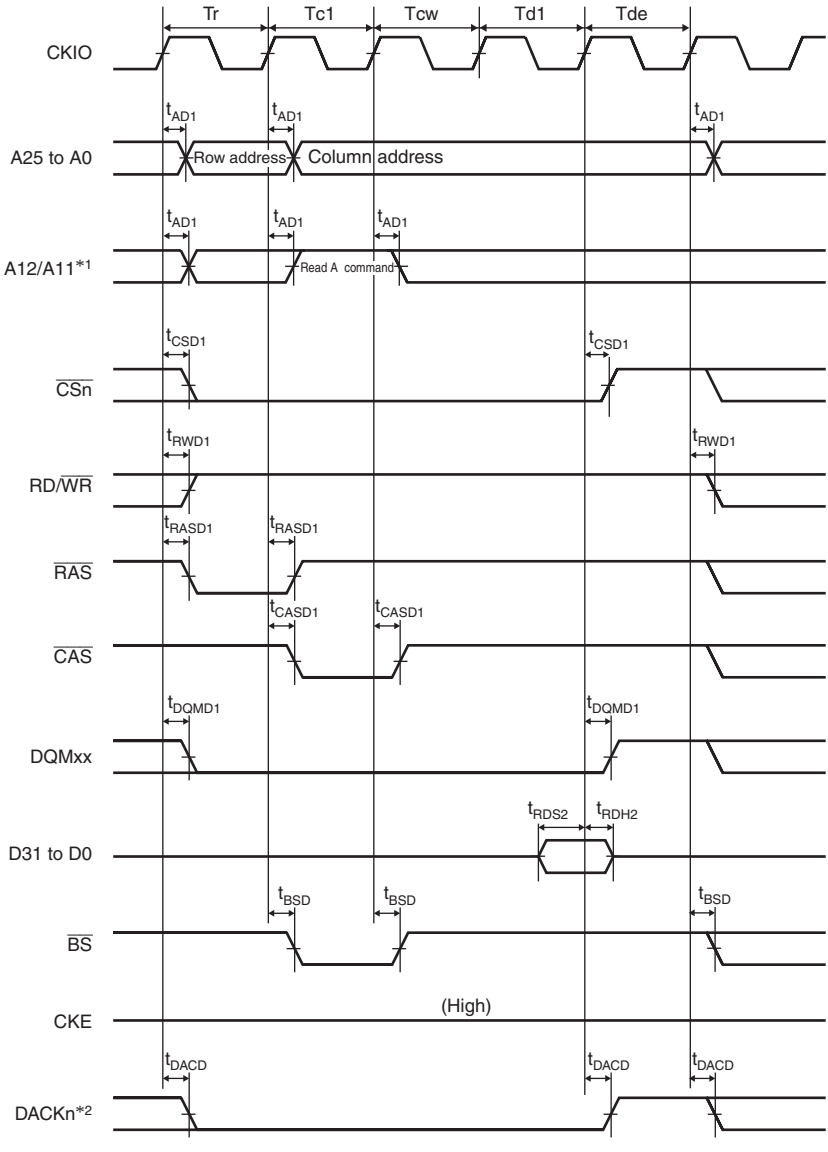
25.3.5 Burst ROM Timing



Notes: 1. t_{RDH3} is specified by earlier one of change of A25 to A0 or the $\overline{RD}$ rising edge.
2. DACKn is a waveform when active-low is specified.

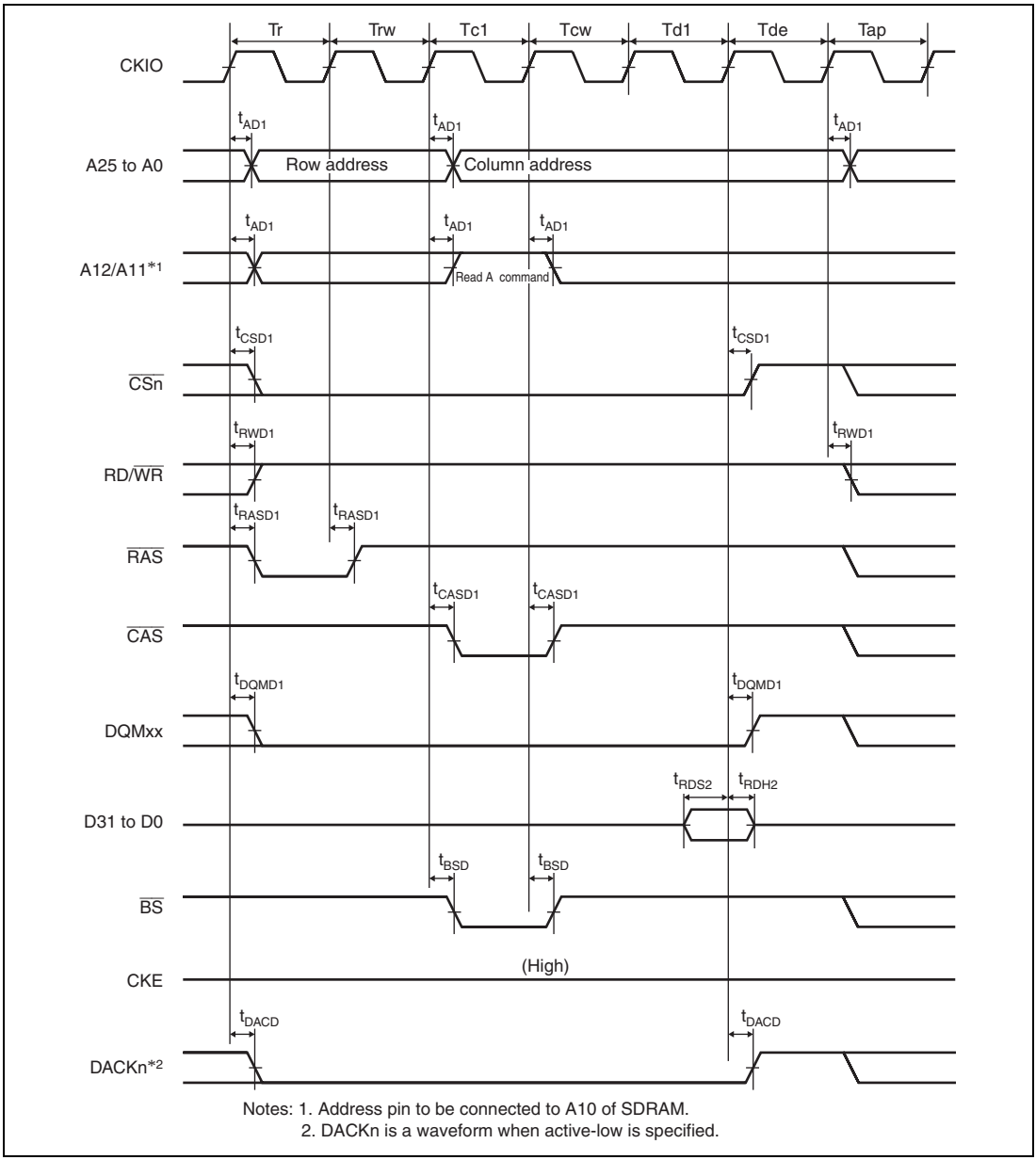
Figure 25.21 Burst ROM Read Cycle (One Access Wait, One External Wait, One Burst Wait, Two Bursts)

25.3.6 Synchronous DRAM Timing



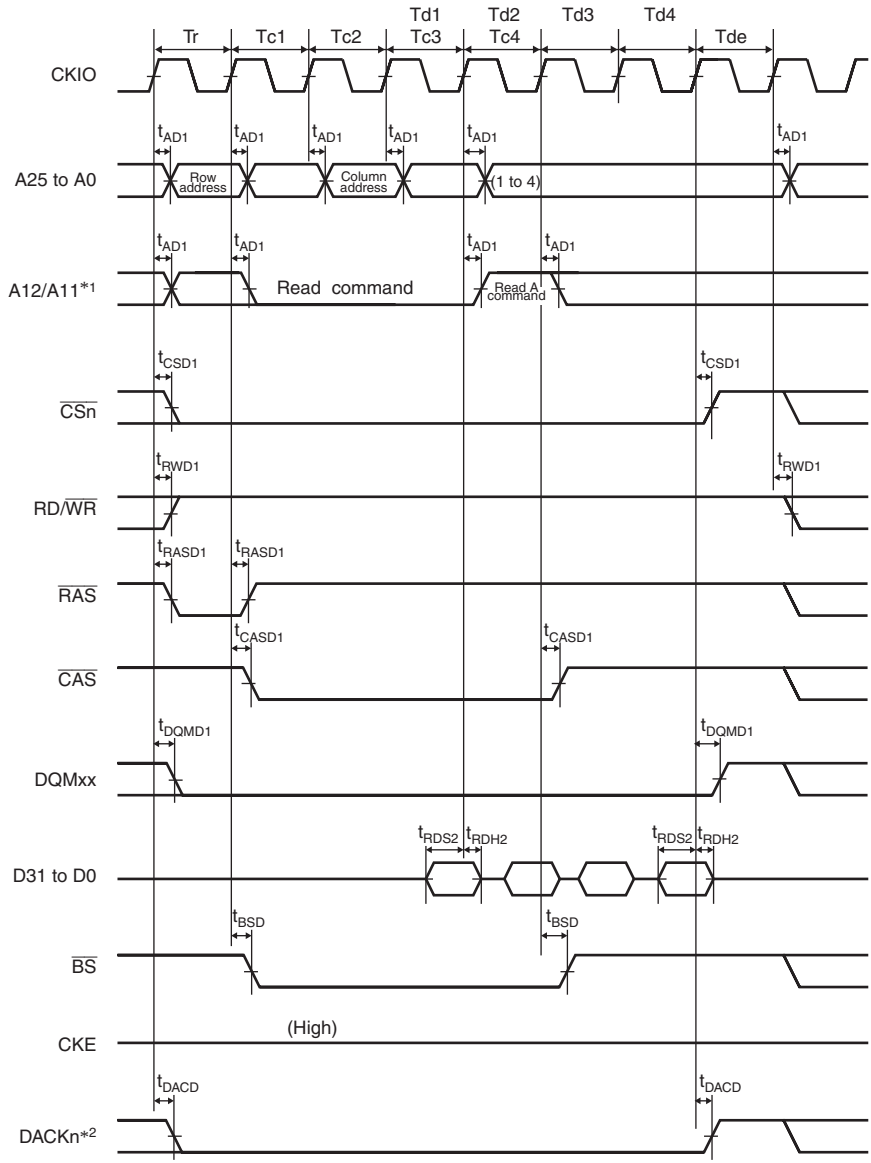
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.22 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 1 Cycle)



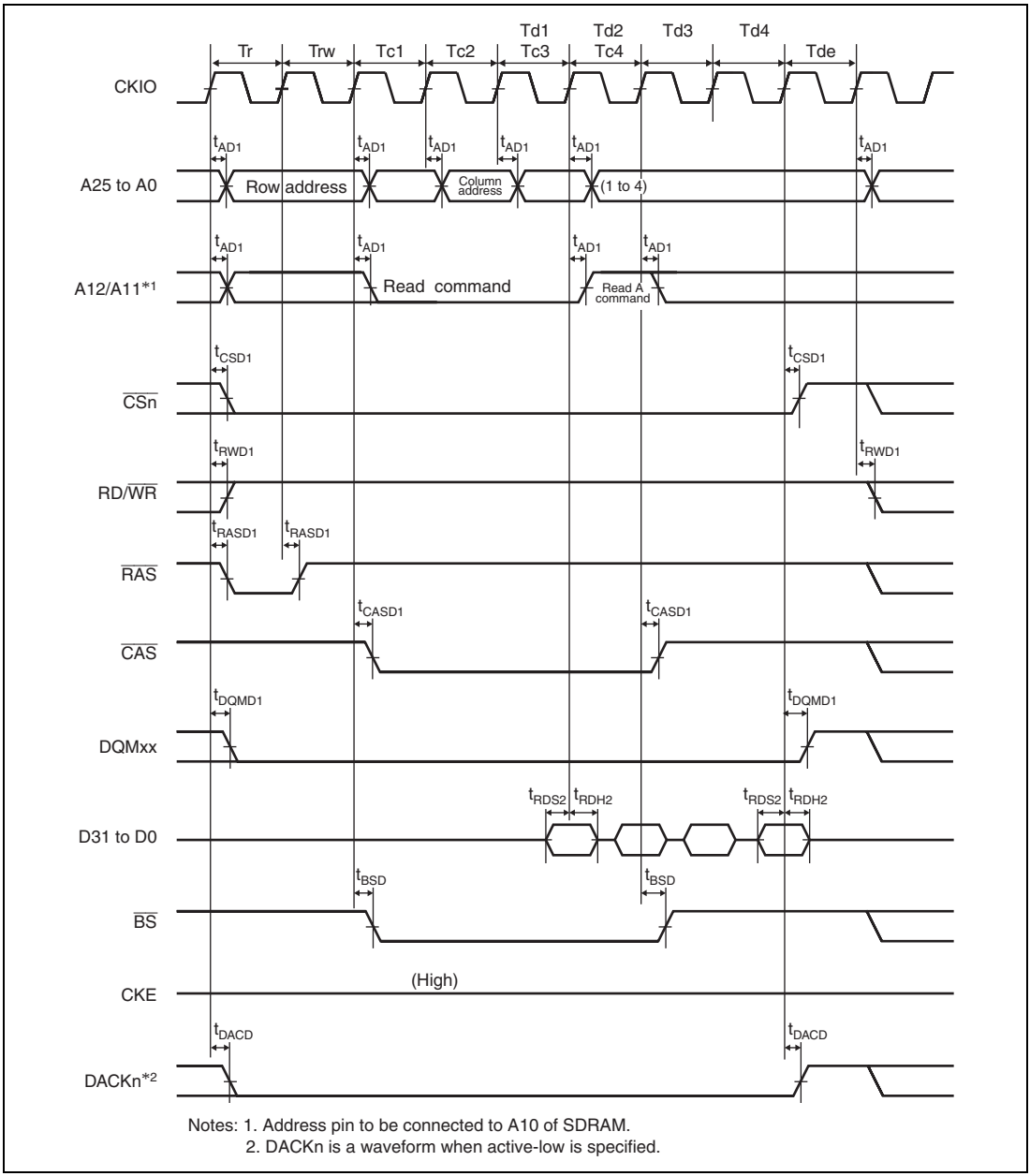
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 2 Cycle)



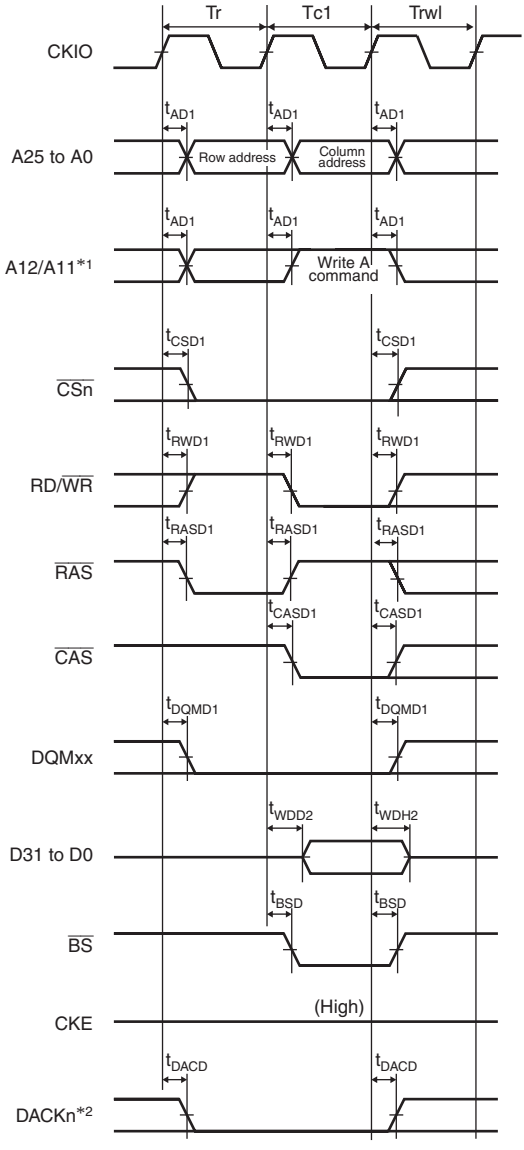
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4), (Auto Precharge, CAS Latency = 2, TRCD = 1 Cycle, TRP = 2 Cycle)



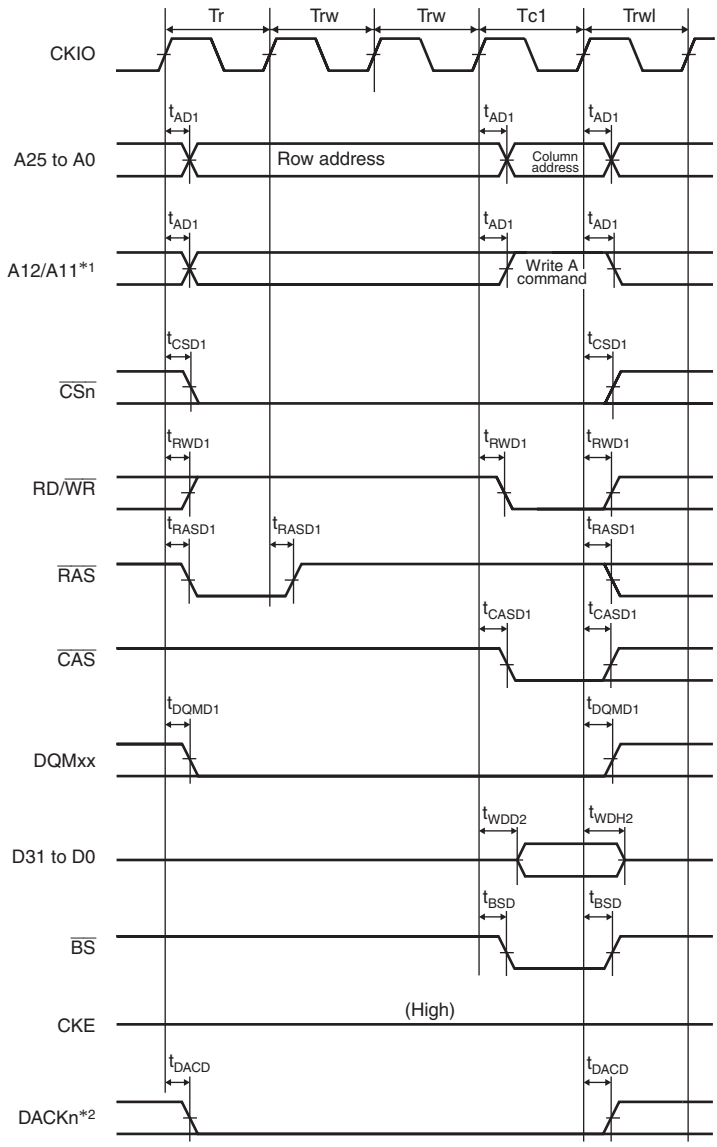
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4), (Auto Precharge, CAS Latency = 2, TRCD = 2 Cycle, TRP = 1 Cycle)



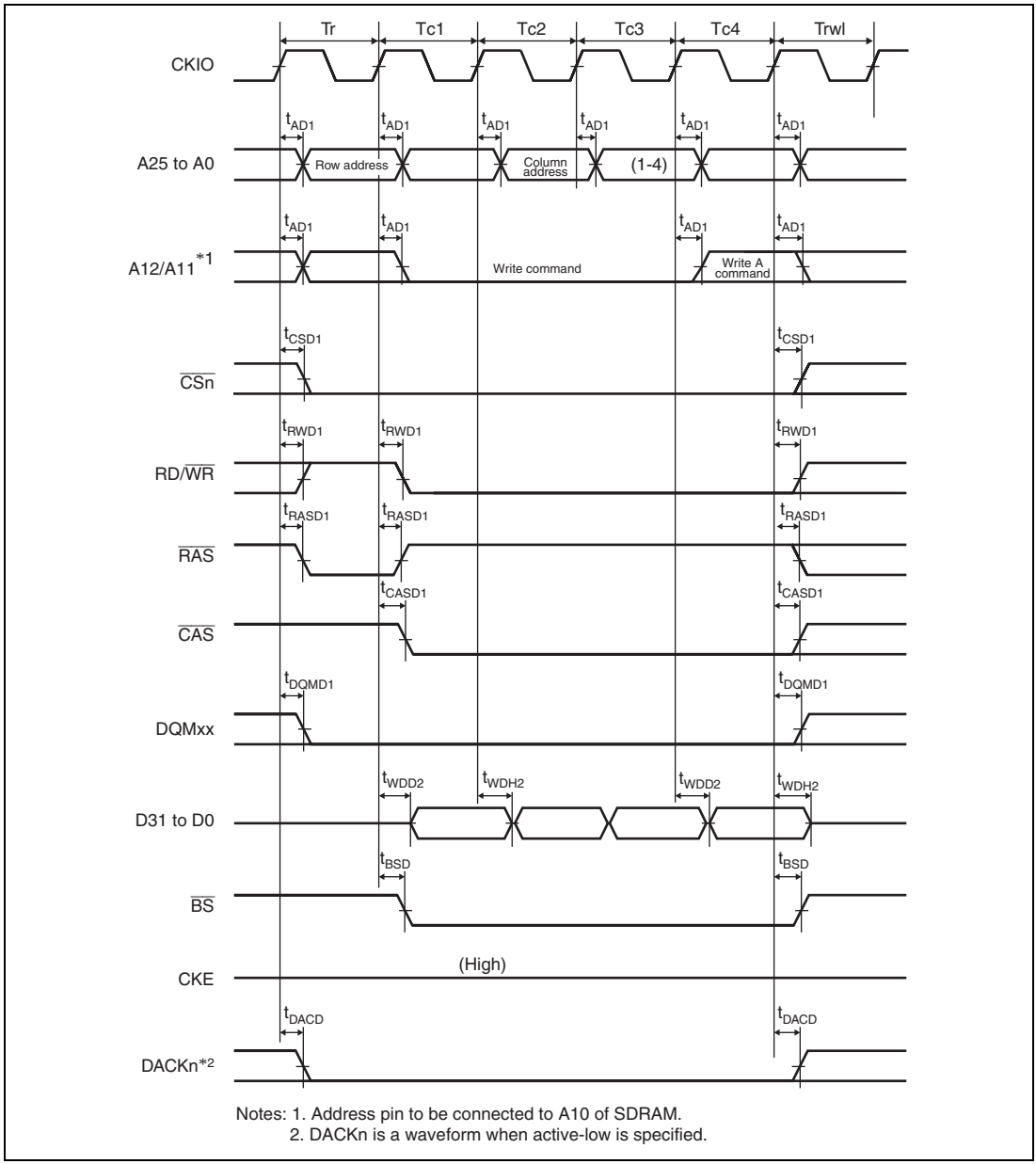
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.26 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 2 Cycle)



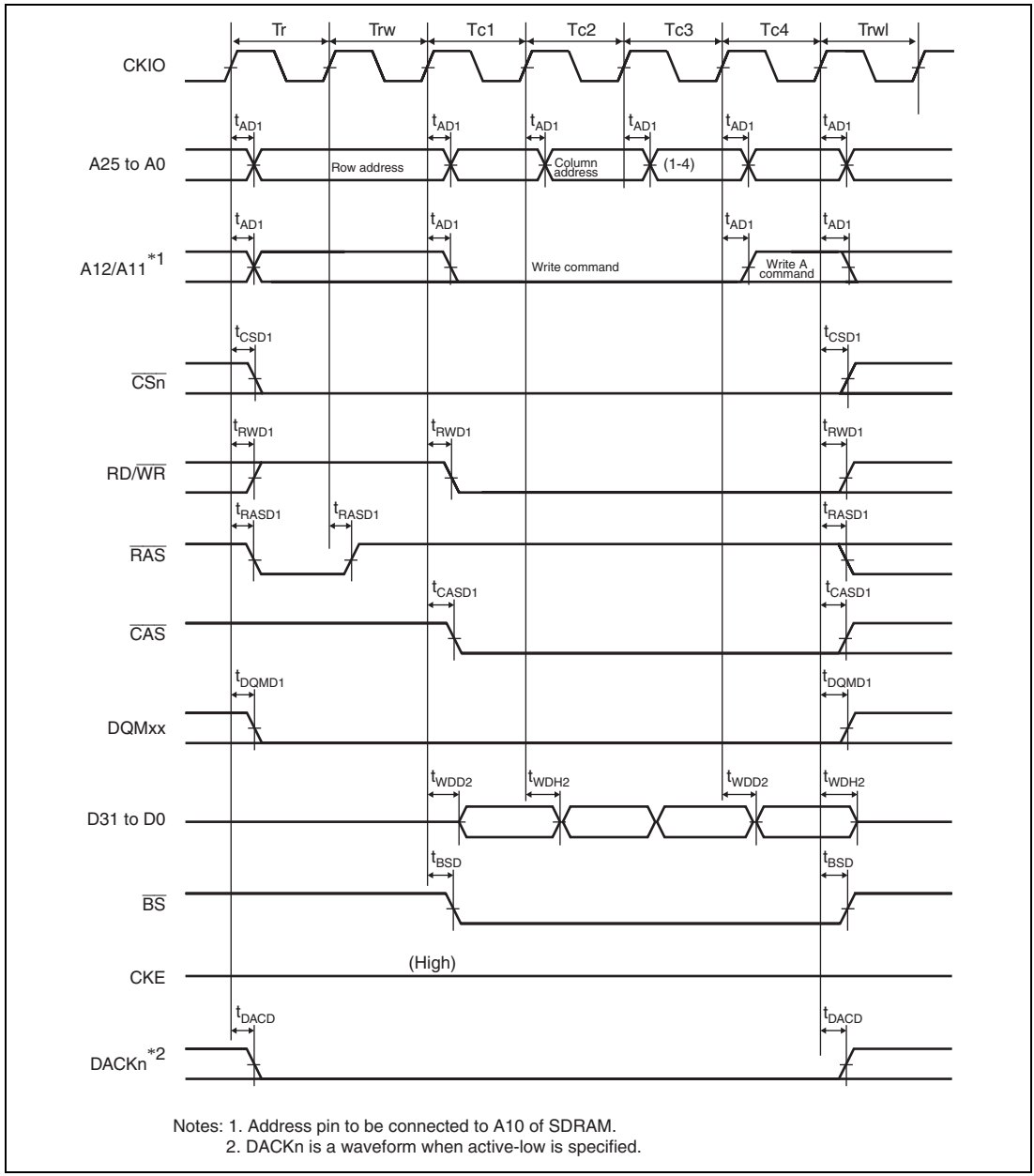
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRCD = 3 Cycle, TRWL = 2 Cycle)



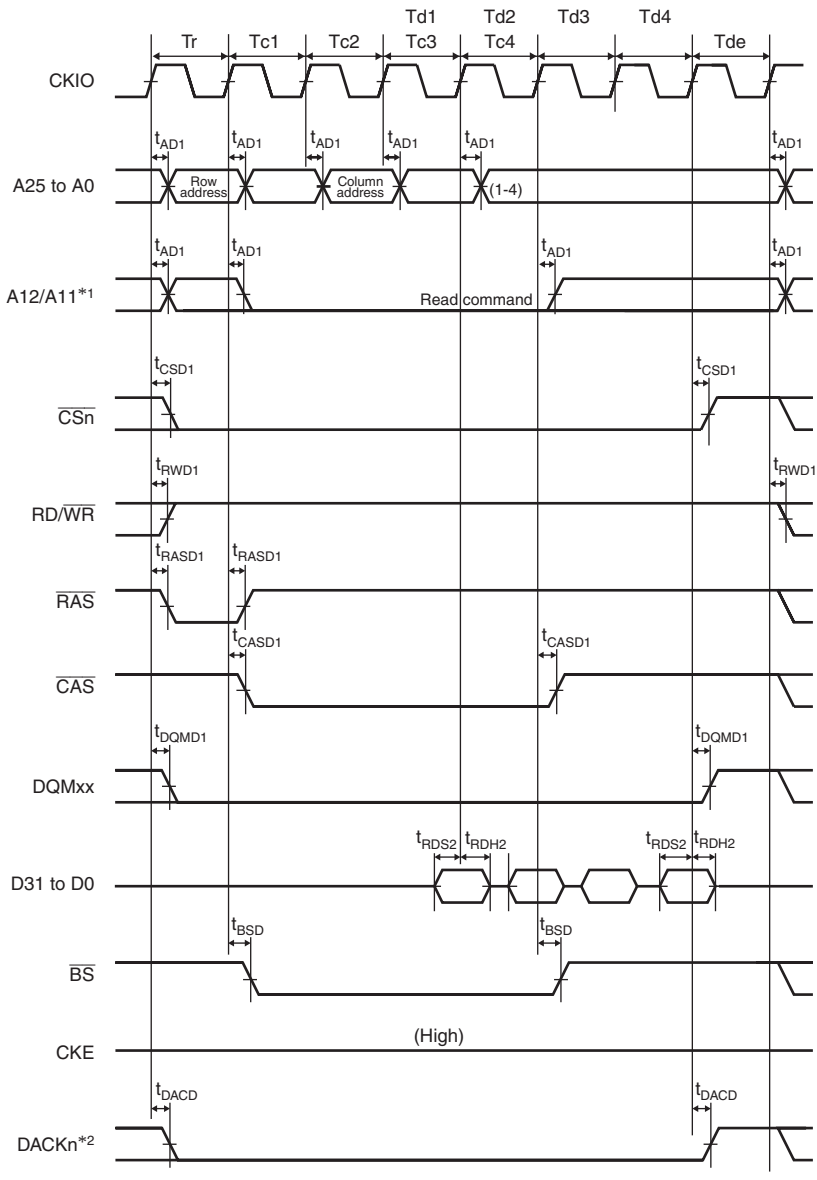
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4), (Auto Precharge, TRCD = 1 Cycle, TRWL = 2 Cycle)



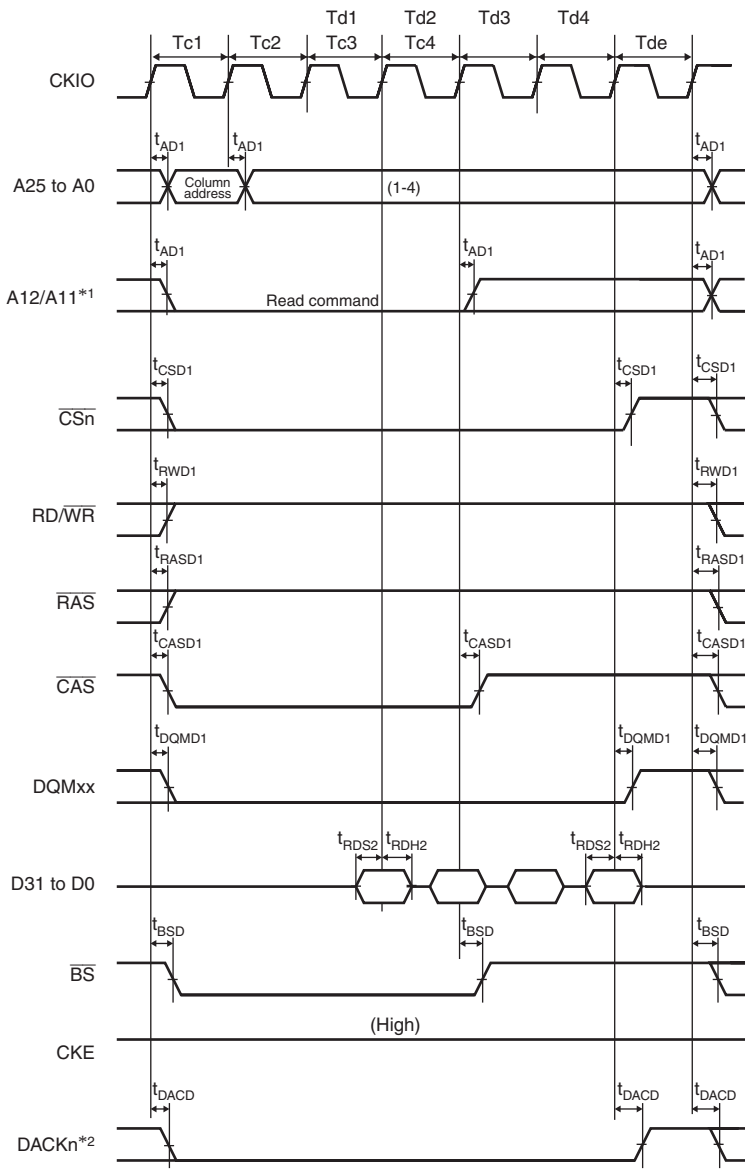
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4), (Auto Precharge, TRCD = 2 Cycle, TRWL = 2 Cycle)



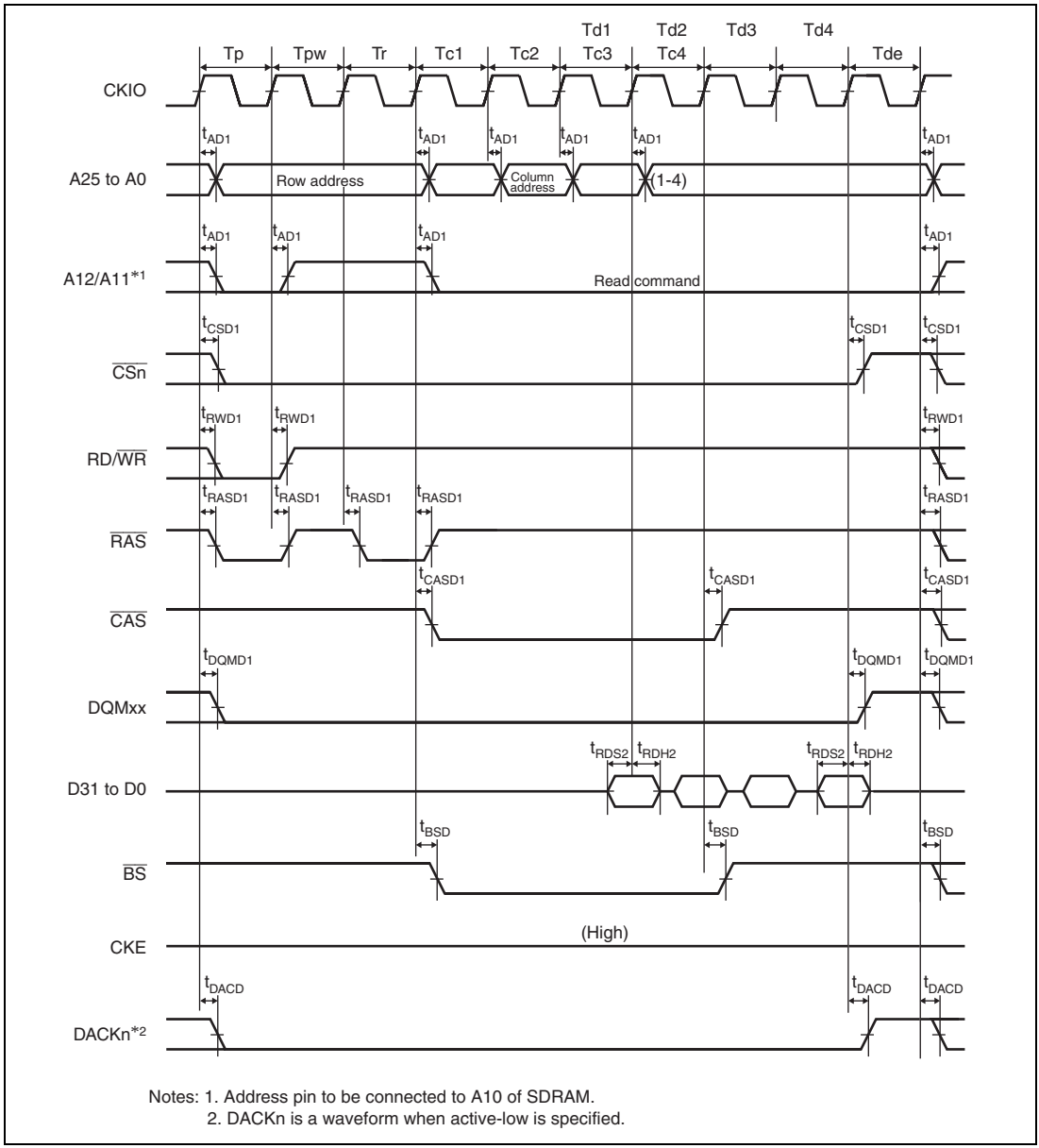
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.30 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode, ACTV + READ Commands, CAS Latency = 2, TRCD = 1 Cycle)



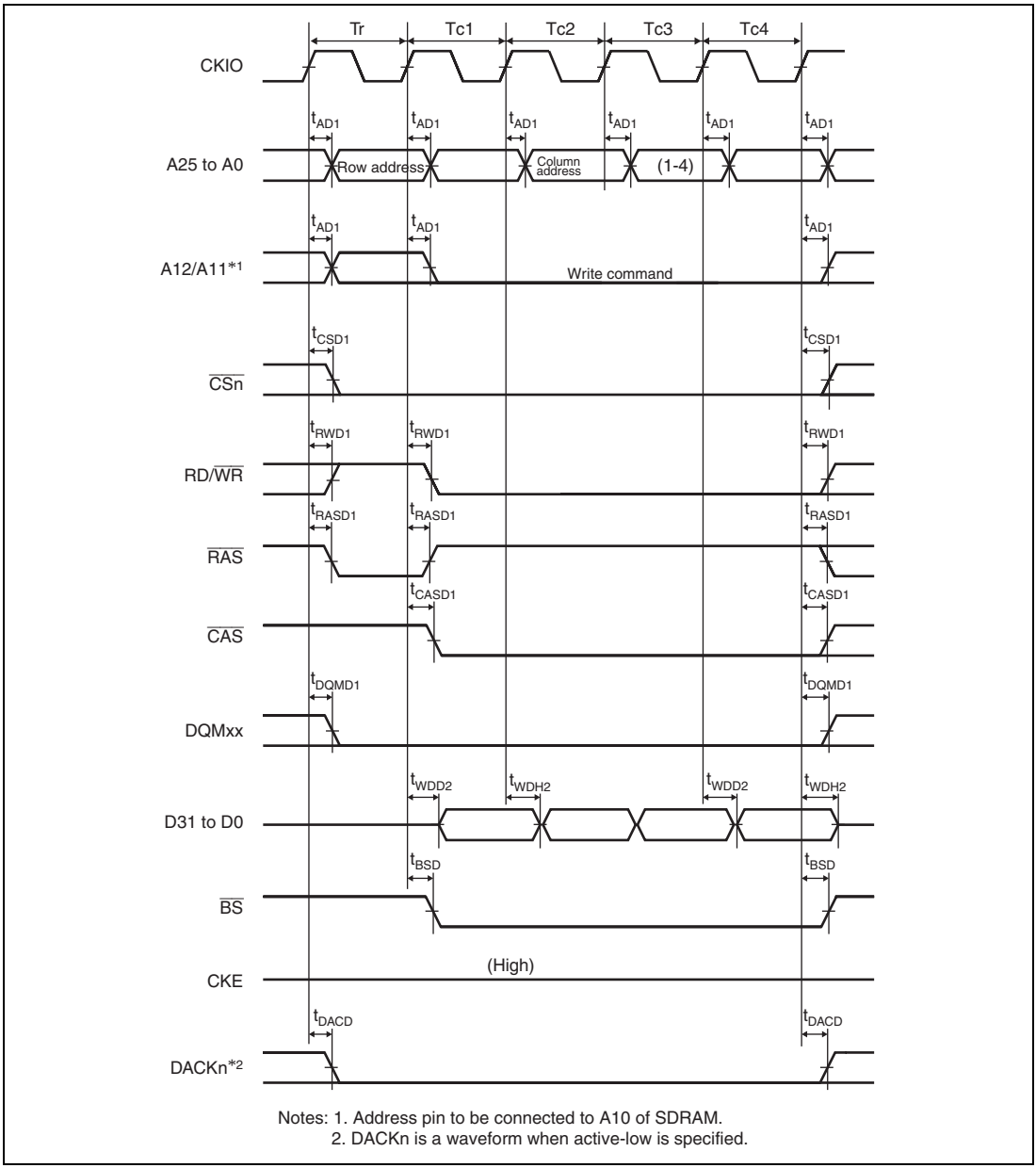
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.31 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode, READ Command, Same Row Address,
CAS Latency = 2, TRCD = 1 Cycle)



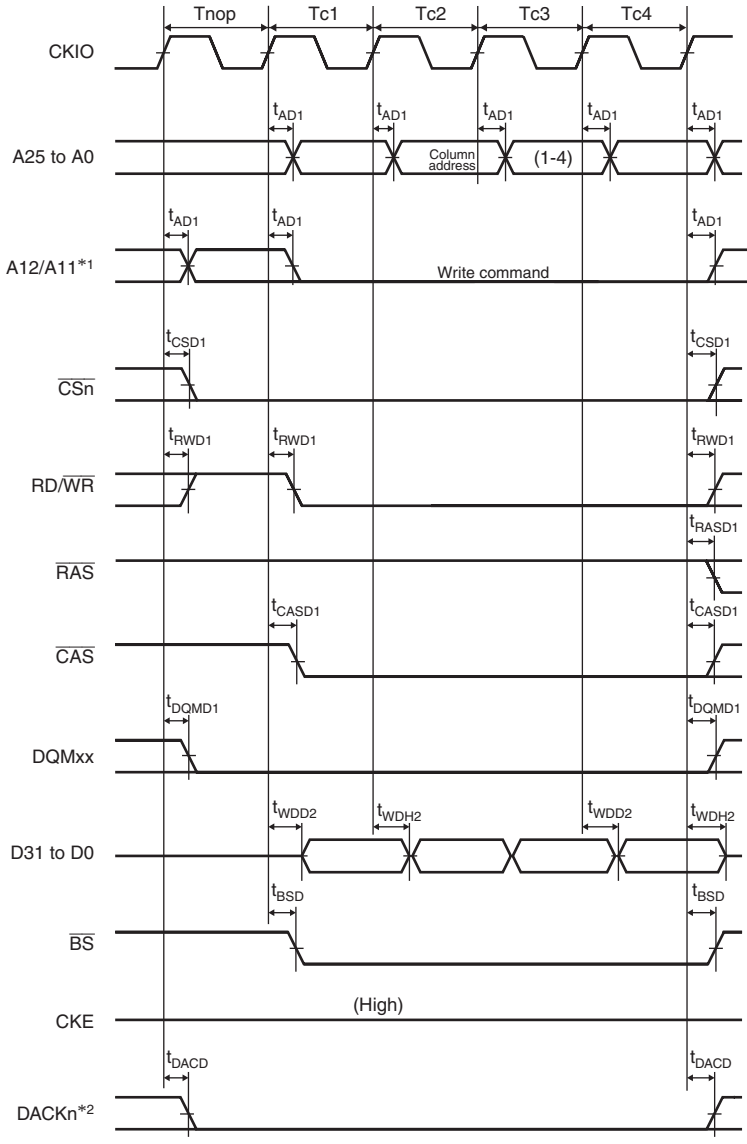
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.32 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode, PRE + ACTV + READ Commands,
Different Row Address, CAS Latency = 2, TRCD = 1 Cycle)



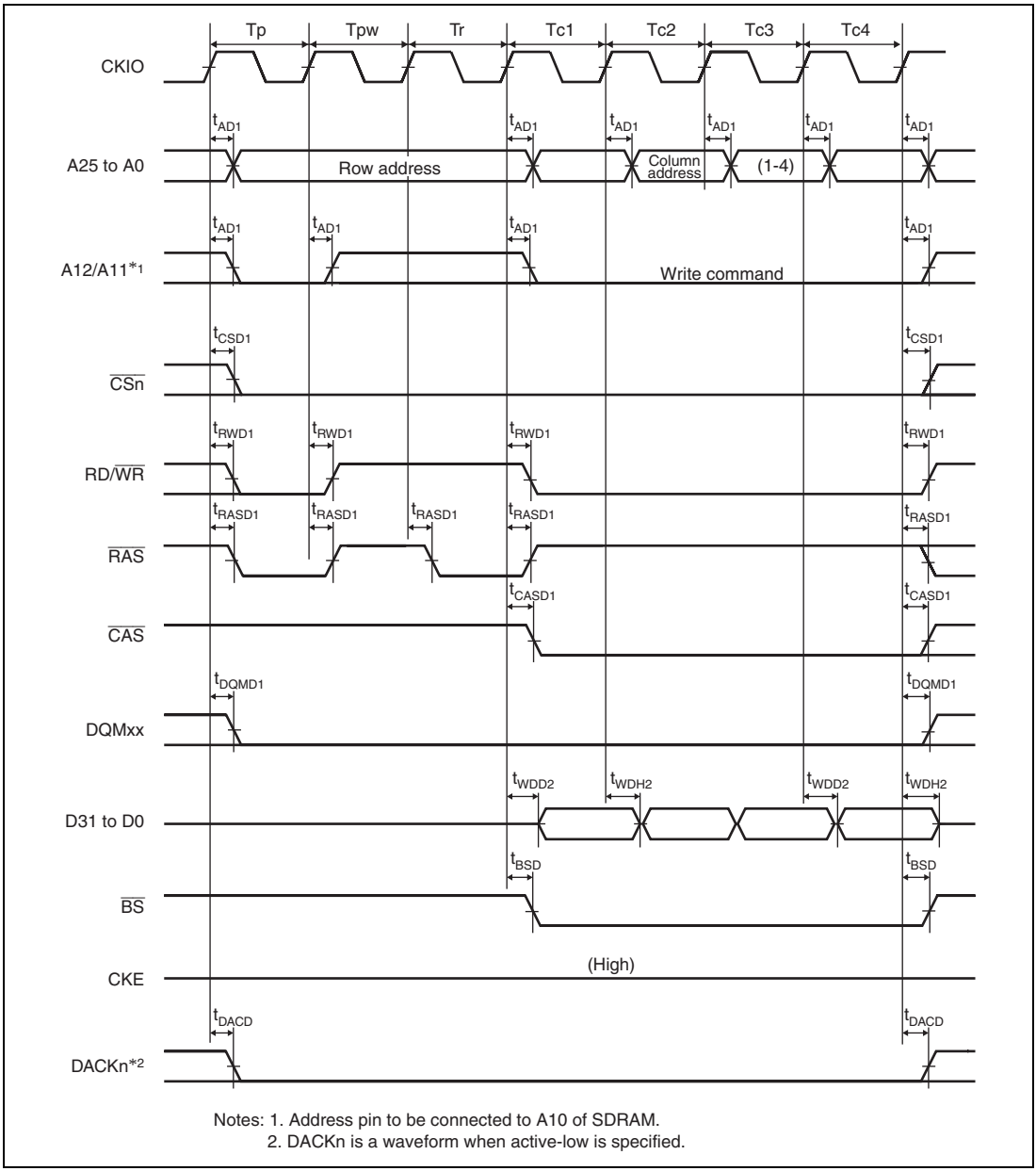
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.33 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4)
(Bank Active Mode, ACTV + WRITE Commands, TRCD = 1 Cycle, TRWL = 1 Cycle)



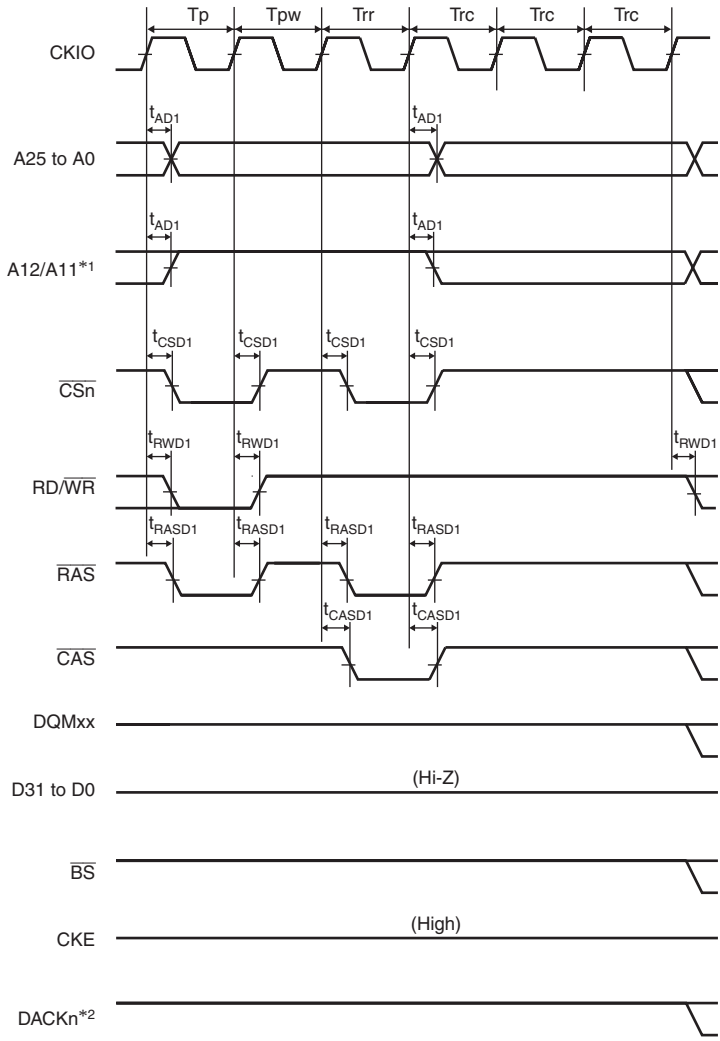
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.34 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode, WRITE Command, Same Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.35 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4)
(Bank Active Mode, PRE + ACTV + WRITE Commands,
Different Row Address, TRCD = 1 Cycle, TRWL = 1 Cycle)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.36 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle)

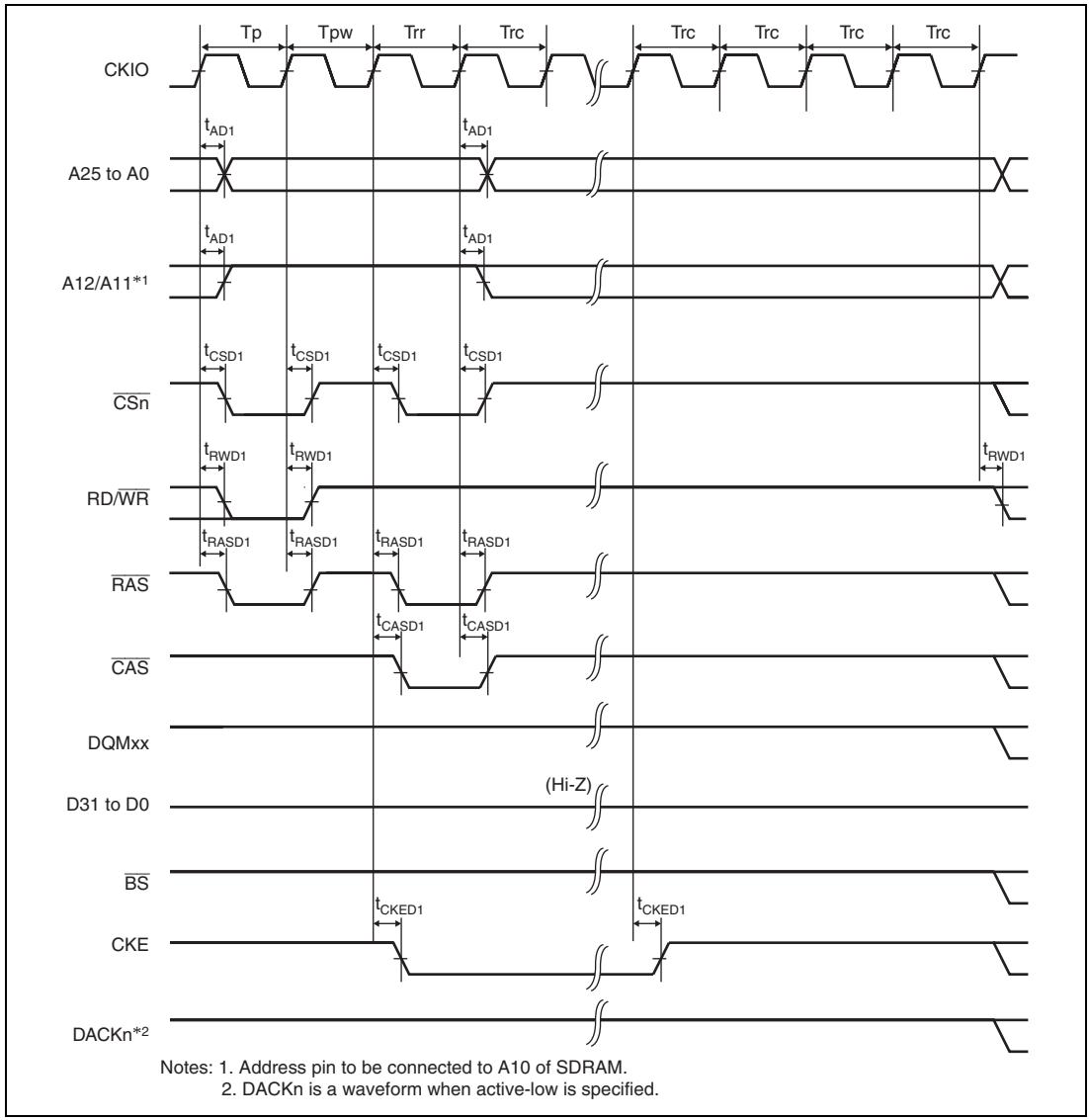
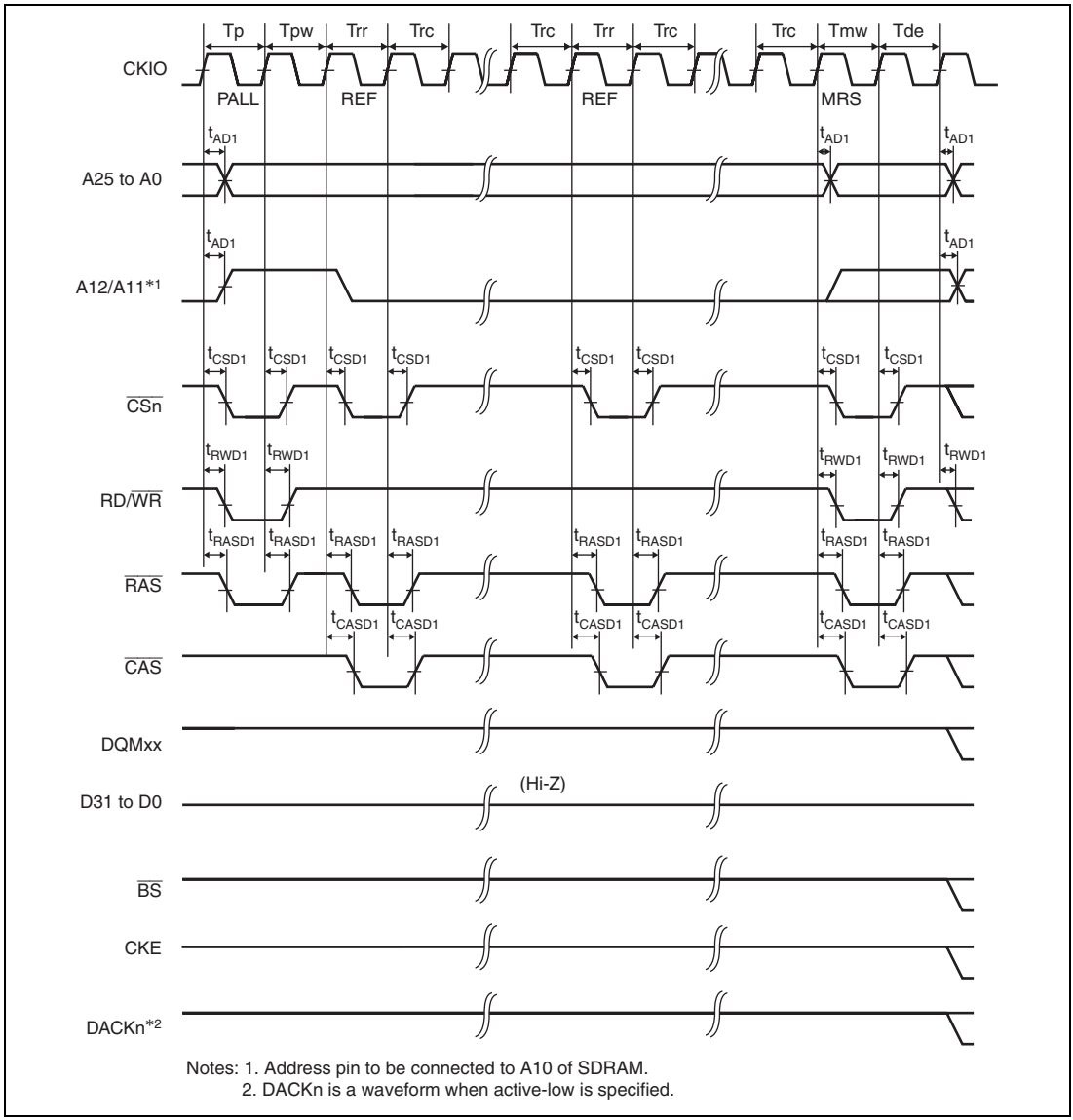


Figure 25.37 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.38 Synchronous DRAM Mode Register Write Timing (TRP = 2 Cycle)

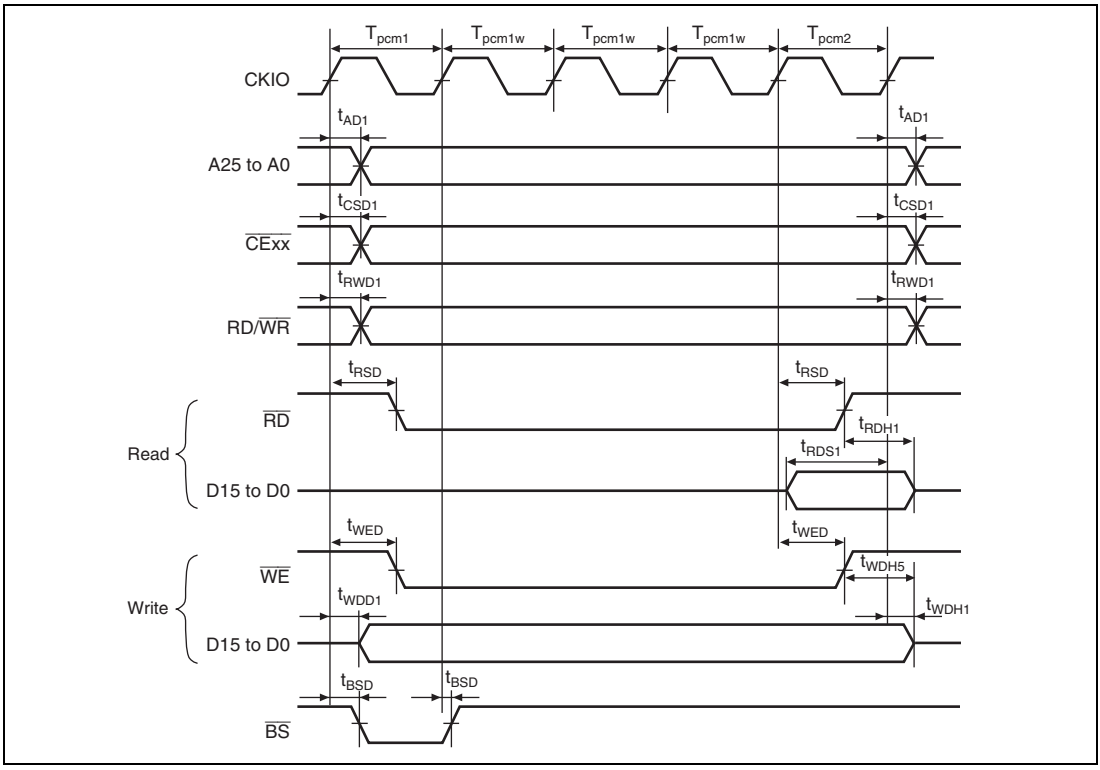


Figure 25.39 PCMCIA Memory Card Interface Bus Timing

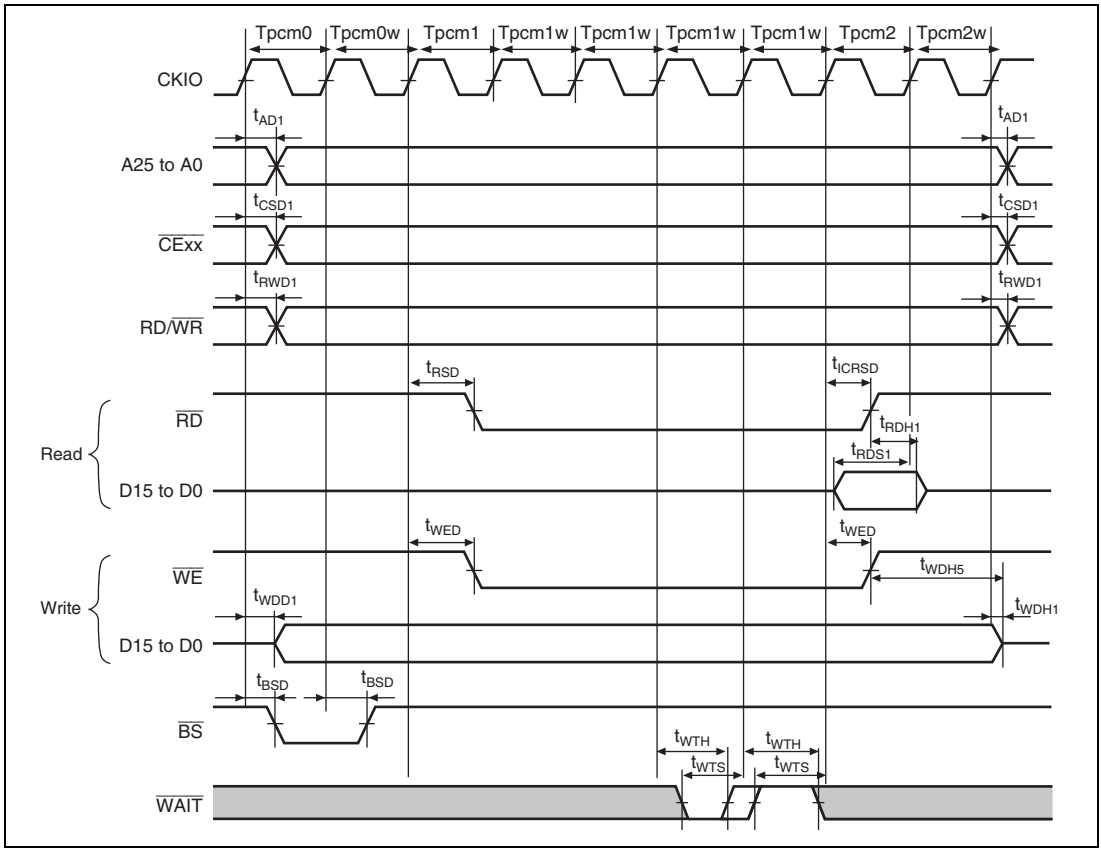


Figure 25.40 PCMCIA Memory Card Interface Bus Timing
(TED[3:0] = B'0010, TEH[3:0] = B'0001, One Software Wait, One Hardware Wait)

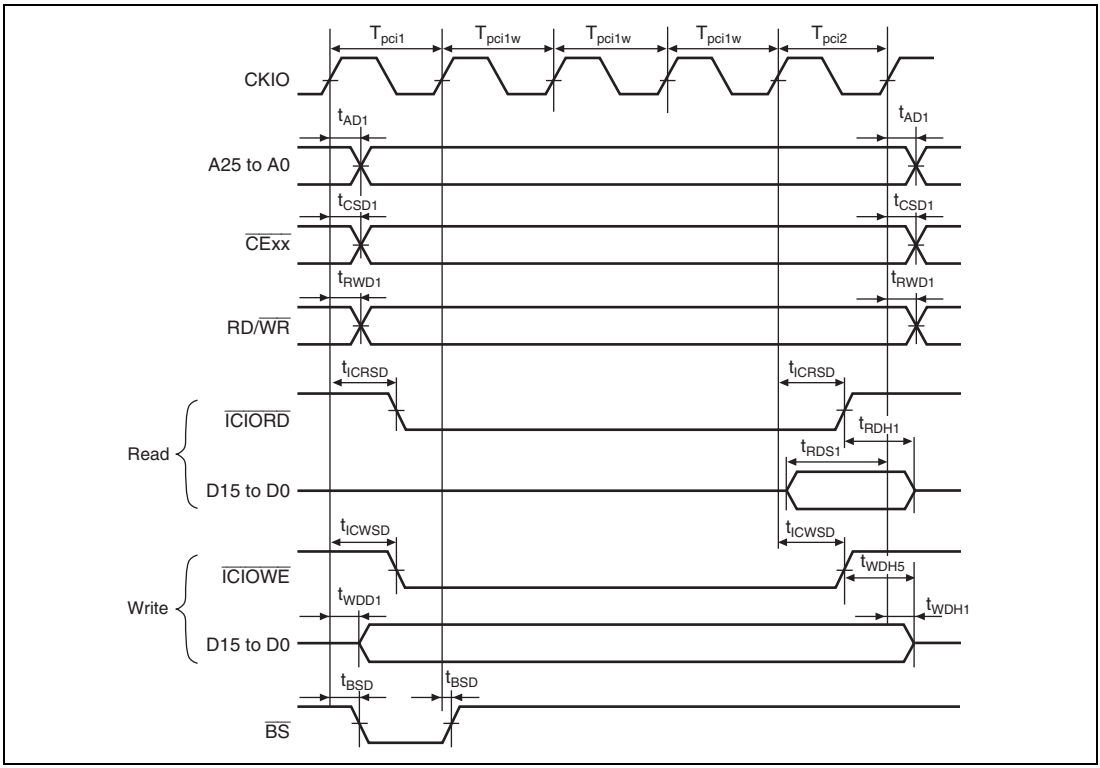


Figure 25.41 PCMCIA I/O Card Interface Bus Timing

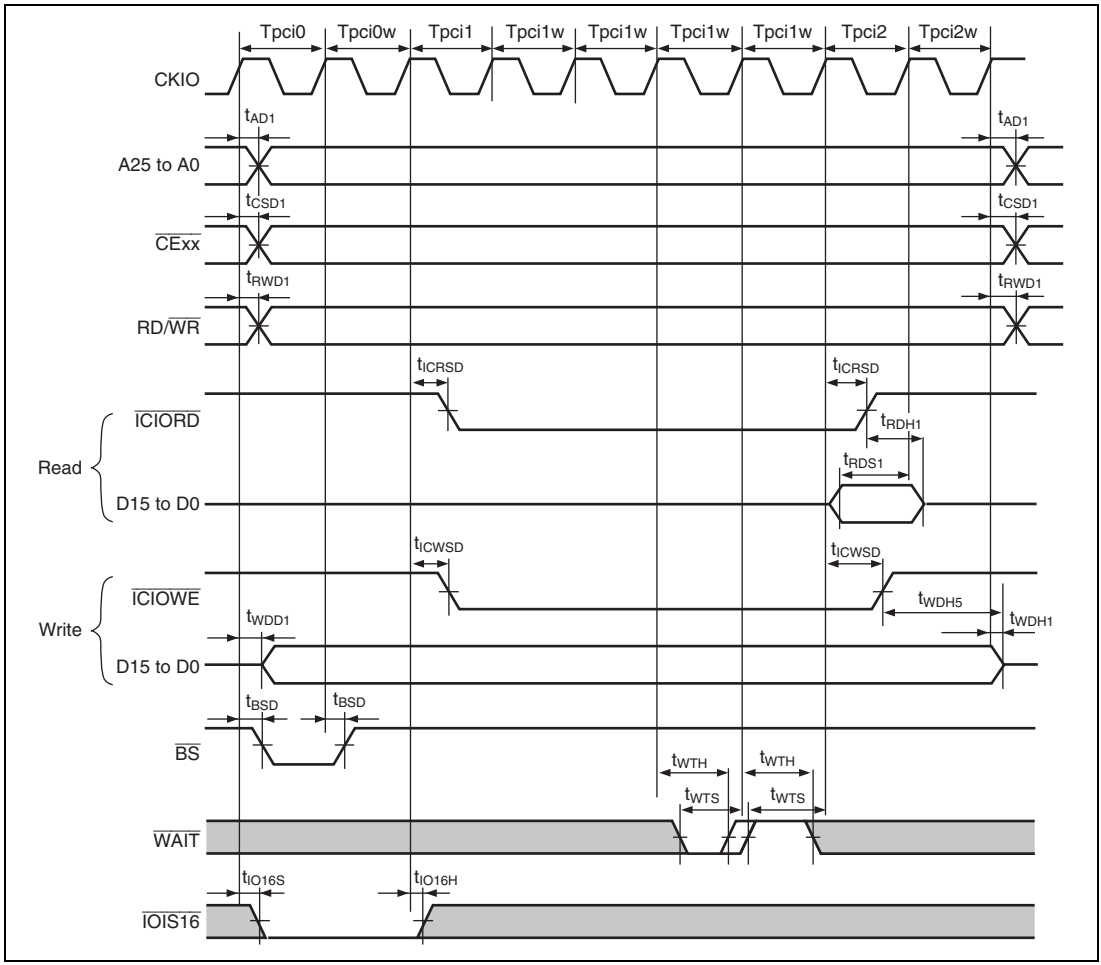


Figure 25.42 PCMCIA I/O Card Interface Bus Timing
(TED[3:0] = B'0010, TEH[3:0] = B'0001, One Software Wait, One Hardware Wait)

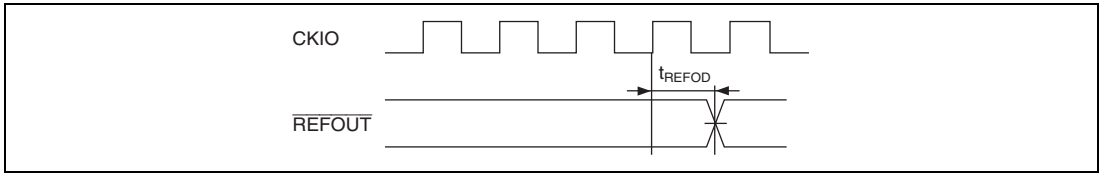
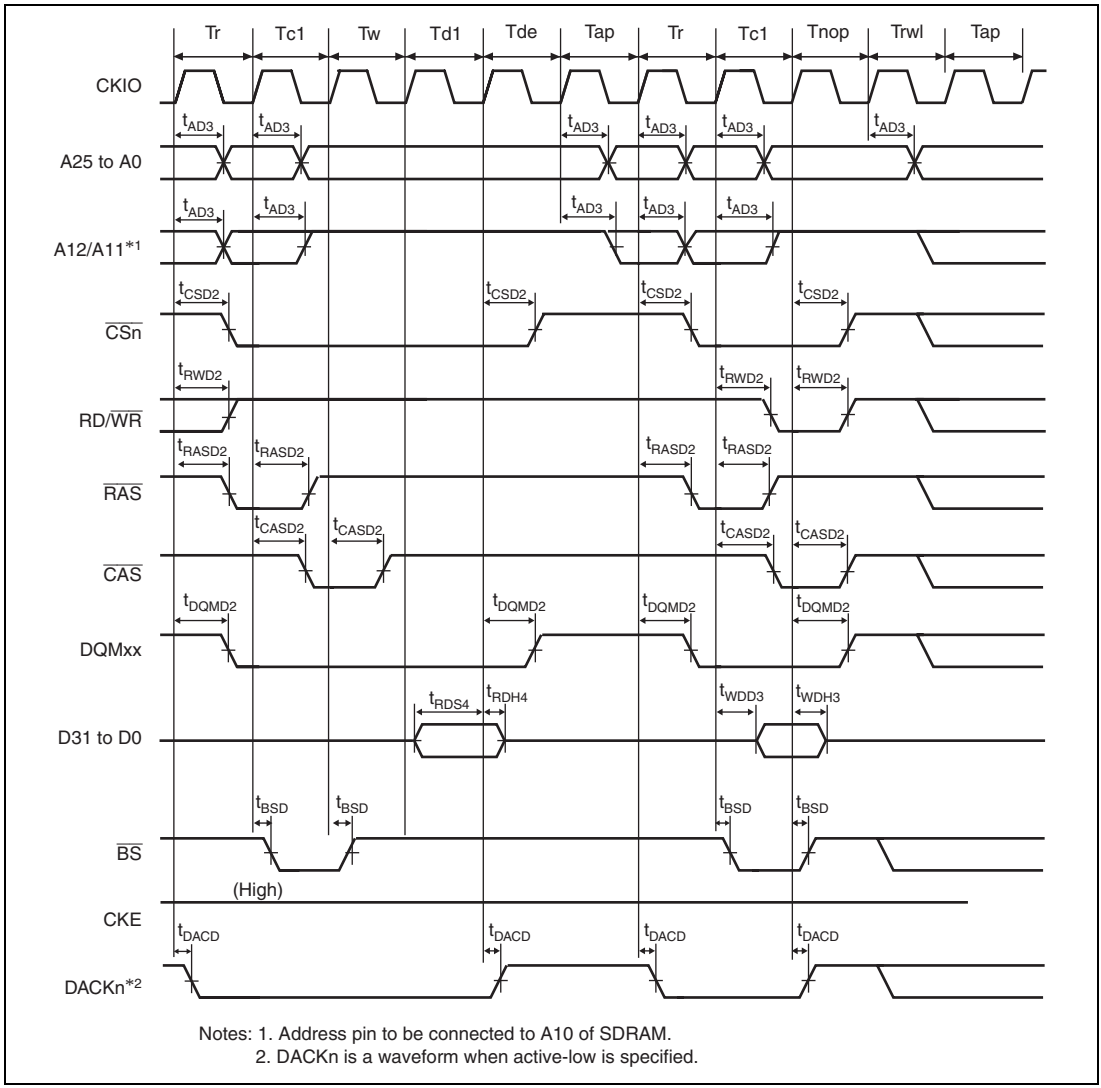


Figure 25.43 REFOUT Delay Time

Table 25.8 Bus Timing (2)

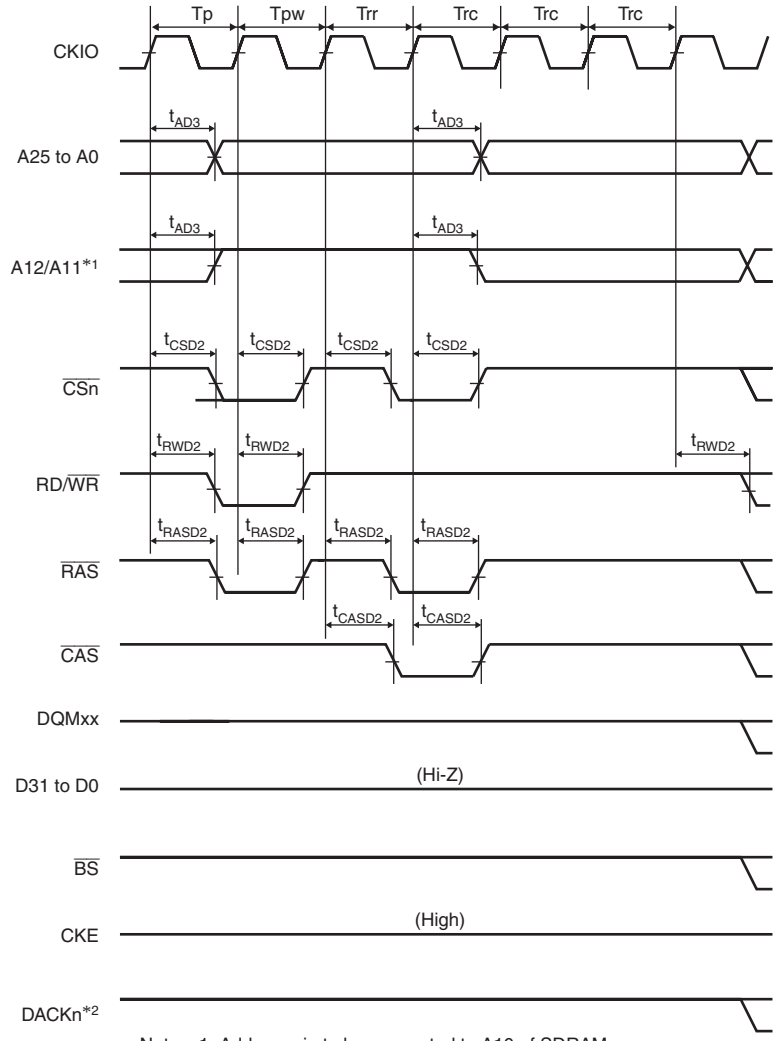
(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C, clock mode 0/1/2/4/5/6/7)

Item	Symbol	Min.	Max.	Unit	Figure
Address delay time 3	t_{AD3}	1/2 t_{cyc}	1/2 t_{cyc}+12	ns	25.44 to 25.47
$\overline{CS}$ delay time 2	t_{CSD2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.44 to 25.47
Read/write delay time 2	t_{RWD2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.44 to 25.47
Read data setup time 4	t_{RDS4}	1/2 t_{cyc}+6	—		25.44
Read data hold time 4	t_{RDH4}	0	—		25.44
Write data delay time 3	t_{WDD3}	—	1/2 t_{cyc}+12		25.44
Write data hold time 3	t_{WDH3}	1/2 t_{cyc}	—		25.44
$\overline{RAS}$ delay time 2	t_{RASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.44 to 25.47
$\overline{CAS}$ delay time 2	t_{CASD2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.44 to 25.47
DQM delay time 2	t_{DQMD2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.44
CKE delay time 2	t_{CKED2}	1/2 t_{cyc}	1/2 t_{cyc}+10		25.46



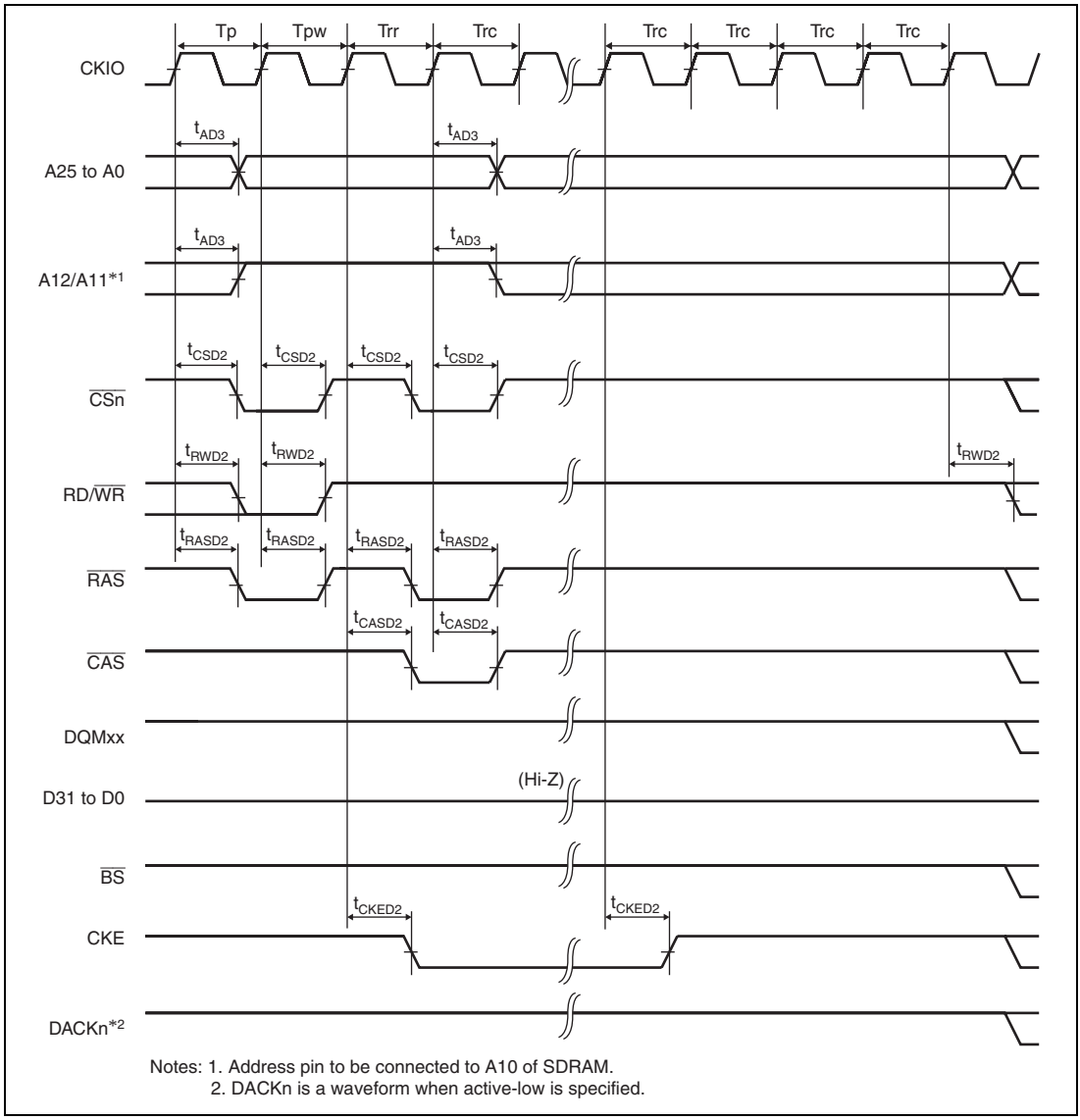
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.44 Access Timing in Low-Frequency Mode (Auto Precharge)



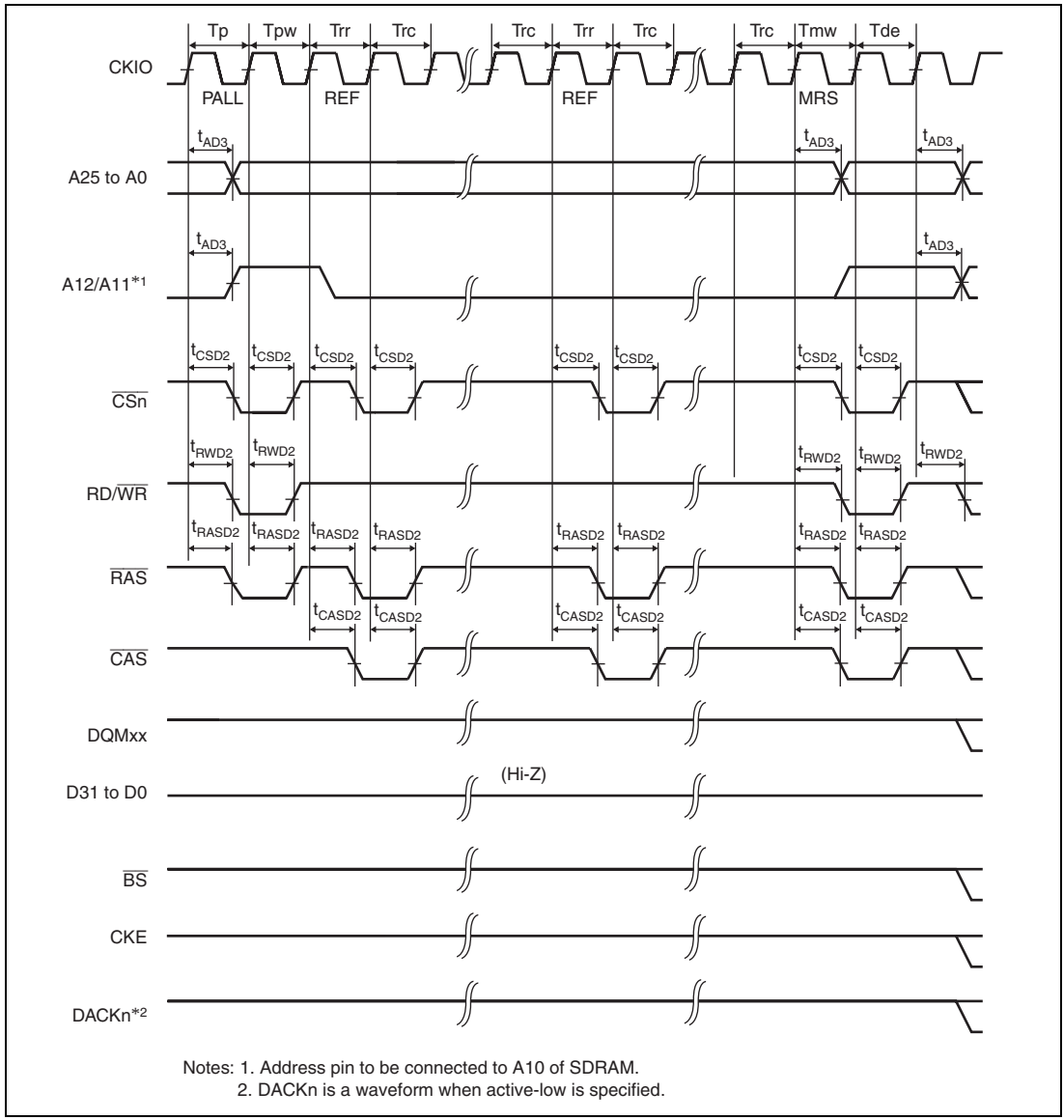
Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.45 Synchronous DRAM Auto-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.46 Synchronous DRAM Self-Refresh Timing (TRP = 2 Cycle, Low-Frequency Mode)



Notes: 1. Address pin to be connected to A10 of SDRAM.
2. DACKn is a waveform when active-low is specified.

Figure 25.47 Synchronous DRAM Mode Register Write Timing (TRP = 2 Cycle, Low-Frequency Mode)

25.3.7 DMAC Signal Timing

Table 25.9 DMAC Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Module	Item	Symbol	Min.	Max.	Unit	Figure
DMAC	DREQn setup time	t_{DRQS}	10	—	ns	25.48
	DREQn hold time	t_{DRQH}	3	—		
	TENDn, DACKn delay time	t_{DACD}	—	10		25.49

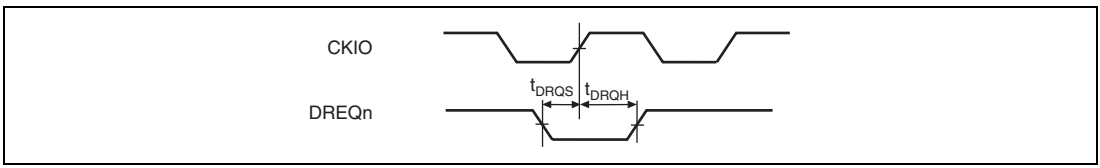


Figure 25.48 DREQn Input Timing

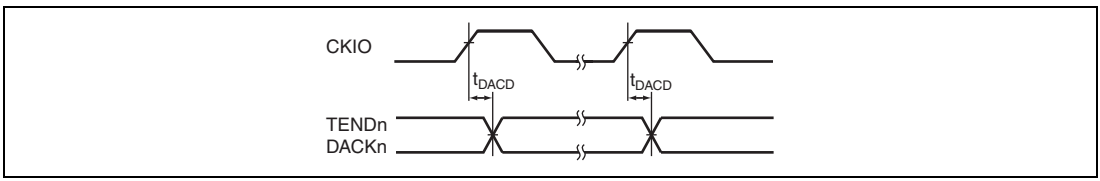


Figure 25.49 TENDn, DACKn Output Timing

25.3.8 RTC Signal Timing

Table 25.10 RTC Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Module	Item	Symbol	Min.	Max.	Unit	Figure
RTC	Oscillation settling time	t_{ROSC}	3	—	s	25.50

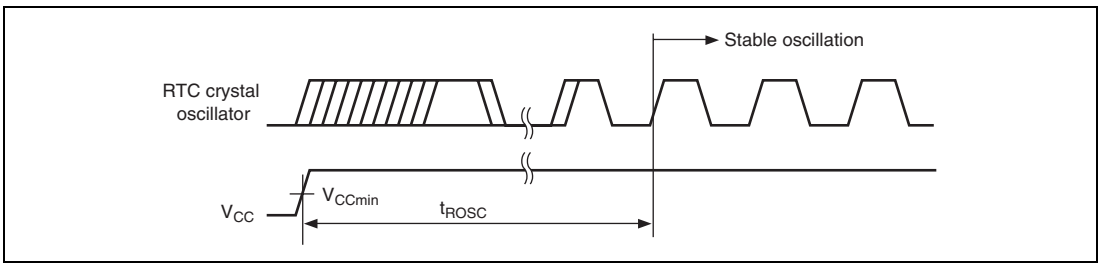


Figure 25.50 Oscillation Settling Time when RTC Crystal Oscillator is Turned On

25.3.9 SCIF Module Signal Timing

Table 25.11 SCIF Module Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Module	Item		Symbol	Min.	Max.	Unit	Figure
SCIF0, SCIF1	Input clock cycle	Clock synchronization	t_{Scyc}	12	—	t_{Pcyc}	25.51 25.52
		Asynchronization		4	—		
	Input clock rise time		t_{SCKR}	—	1.5		25.51
	Input clock fall time		t_{SCKF}	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Transmission data delay time		t_{TXD}	—	3 t_{Pcyc}* + 50	ns	25.52
	Receive data setup time (clock synchronization)		t_{RXS}	2 t_{Pcyc}*	—		
	Receive data hold time (clock synchronization)		t_{RXH}	2 t_{Pcyc}*	—		
	$\overline{RTS}$ delay time		t_{RTSD}	—	100		
	$\overline{CTS}$ setup time (clock synchronization)		t_{CTSS}	100	—		
	$\overline{CTS}$ hold time (clock synchronization)		t_{CTSH}	100	—		

Note: * t_{Pcyc} indicates a peripheral clock (Pϕ) cycle.

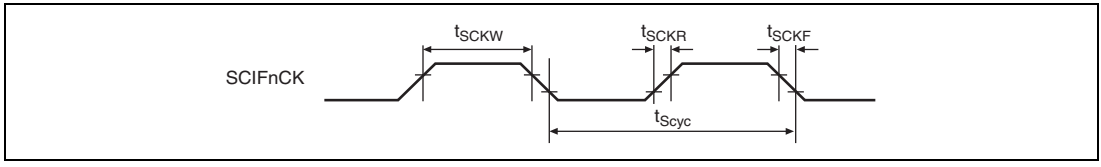


Figure 25.51 SCIFnCK Input Clock Timing

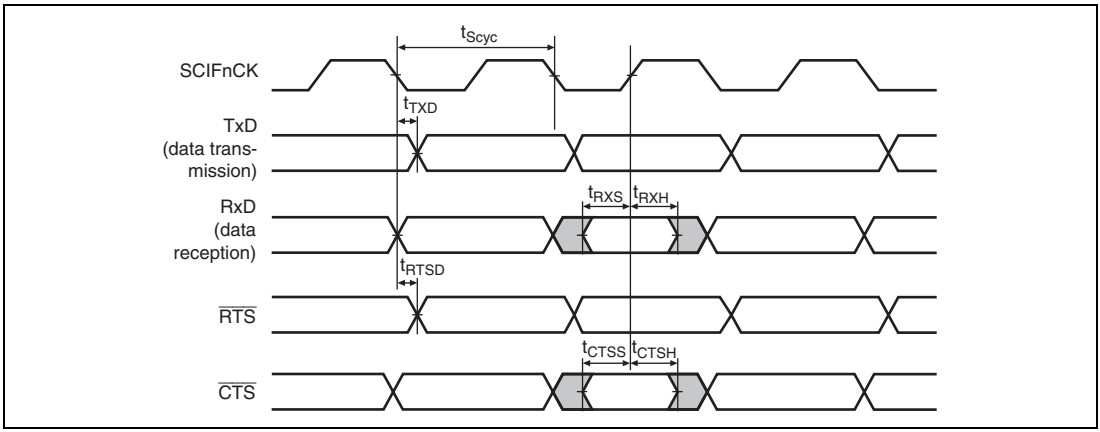


Figure 25.52 SCIF Input/Output Timing in Clock Synchronous Mode

25.3.10 SIOF Module Signal Timing

Table 25.12 SIOF Module Signal Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Item	Symbol	Min.	Max.	Unit	Figure
SIOMCLK clock input cycle time	t_{Mcyc}	30	—	ns	25.53
SIOMCLK input high-level width	t_{MWH}	$0.4 \times t_{Mcyc}$	—		
SIOMCLK input low-level width	t_{MWL}	$0.4 \times t_{Mcyc}$	—		
SCK_SIO clock cycle time	t_{SIcyc}	$2 \times t_{Pcyc}$	—		25.54 to 25.58
SCK_SIO output high-level width	t_{SWHO}	$0.4 \times t_{SIcyc}$	—		25.54 to 25.57
SCK_SIO output low-level width	t_{SWLO}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC output delay time	t_{FSD}	—	20		
SCK_SIO input high-level width	t_{SWHI}	$0.4 \times t_{SIcyc}$	—		25.58
SCK_SIO input low-level width	t_{SWLI}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC input setup time	t_{FSS}	20	—		
SIOFSYNC input hold time	t_{FSH}	20	—		
TXD_SIO output delay time	t_{STDD}	—	20		25.54 to 25.58
RXD_SIO input setup time	t_{SRDS}	20	—		
RXD_SIO input hold time	t_{SRDH}	20	—		

Note: t_{Pcyc} is the cycle time (ns) of the peripheral clock (Pϕ).

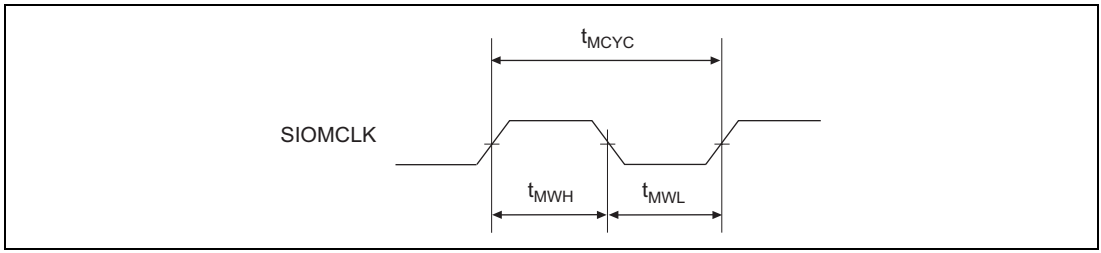


Figure 25.53 SIOMCLK Input Timing

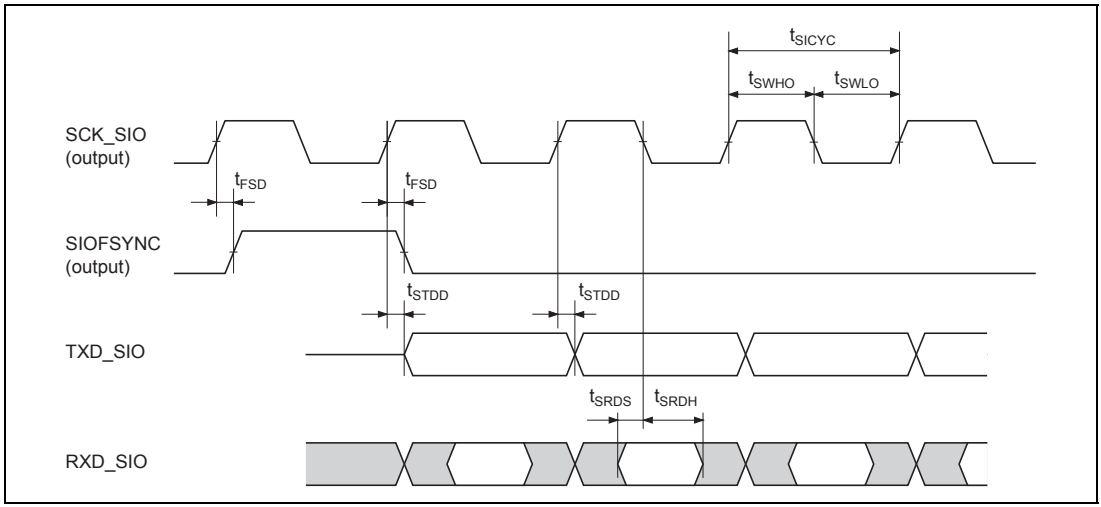


Figure 25.54 SIOF Transmit/Receive Timing (Master Mode 1: Fall Sampling Time)

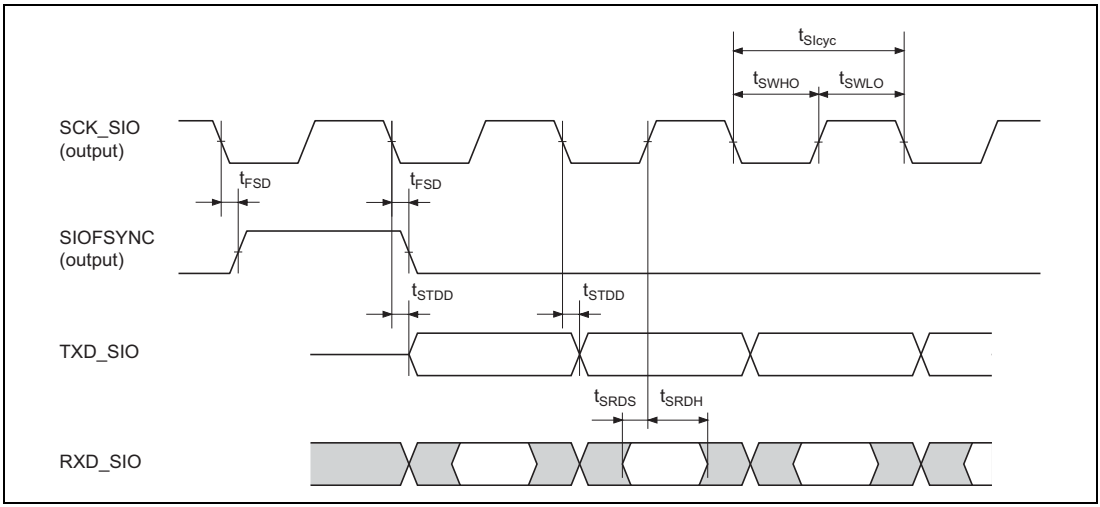


Figure 25.55 SIOF Transmit/Receive Timing (Master Mode 1: Rise Sampling Time)

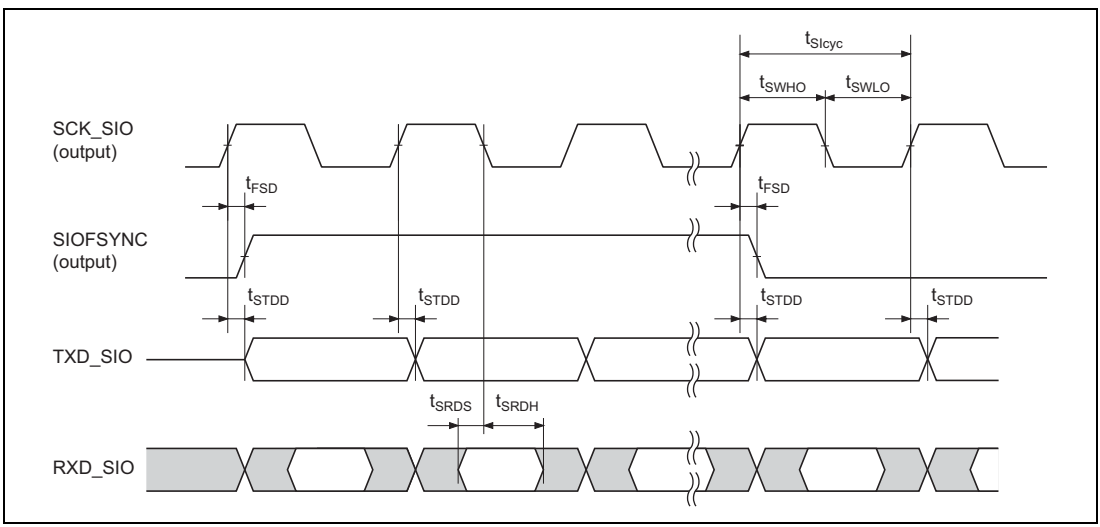


Figure 25.56 SIOF Transmit/Receive Timing (Master Mode 2: Fall Sampling Time)

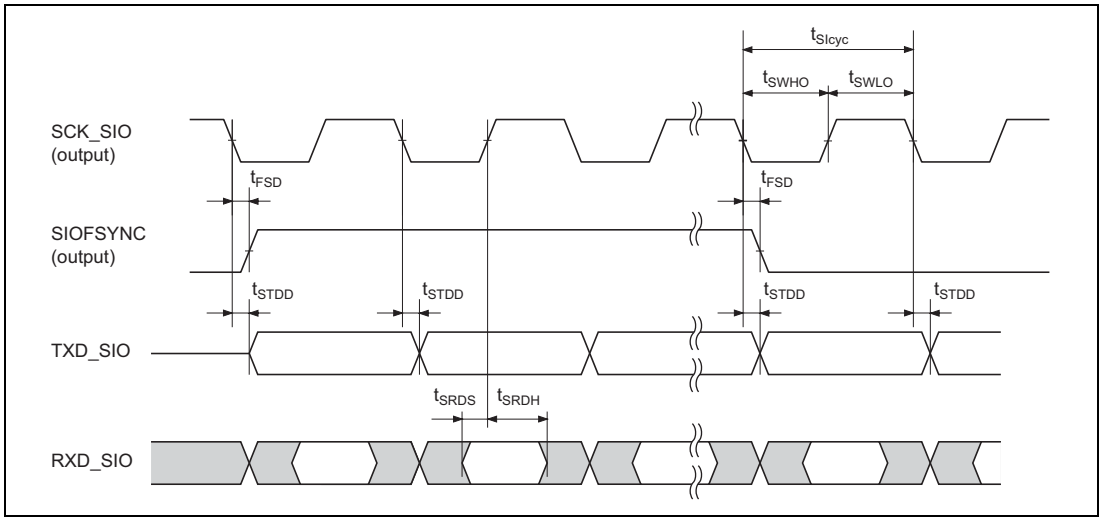


Figure 25.57 SIOF Transmit/Receive Timing (Master Mode 2: Rise Sampling Time)

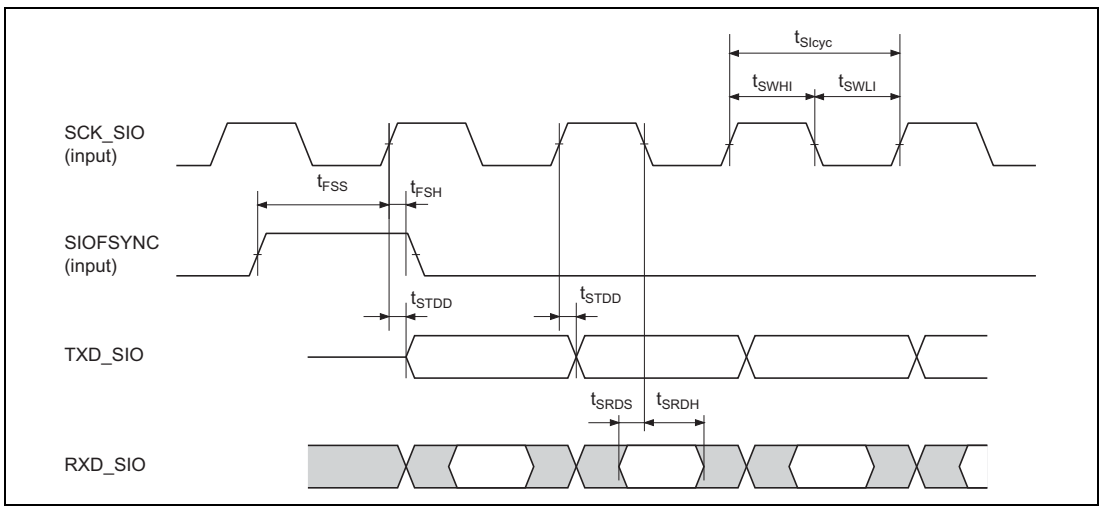


Figure 25.58 SIOF Transmit/Receive Timing (Slave Mode 1 and Slave Mode 2)

25.3.11 Ethernet Controller Timing

Table 25.13 Ethernet Controller Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
TX-CLK cycle time	t_{Tcyc}	40	—	—	ns	25.59
TX-EN output delay time	t_{TEND}	3	—	20		
ETXD[3:0] output delay time	t_{ETDD}	3	—	20		
CRS setup time	t_{CRSS}	10	—	—		
CRS hold time	t_{CRSH}	10	—	—		
COL setup time	t_{COLS}	10	—	—		25.60
COL hold time	t_{COLH}	10	—	—		
RX-CLK cycle time	t_{Rcyc}	40	—	—		25.61
RX-DV setup time	t_{RDVS}	10	—	—		
RX-DV hold time	t_{RDVH}	3	—	—		
ERXD[3:0] setup time	t_{ERDS}	10	—	—		
ERXD[3:0] hold time	t_{ERDH}	3	—	—		
RX-ER setup time	t_{RERS}	10	—	—		25.62
RX-ER hold time	t_{RERH}	3	—	—		
MDIO setup time	t_{MDIOS}	10	—	—		25.63
MDIO hold time	t_{MDIOH}	10	—	—		
MDIO output data hold time*	t_{MDIODH}	5	—	18		25.64
WOL output delay time	t_{WOLD}	1	—	18		25.65
EXOUT output delay time	t_{EXOUTD}	1	—	28		25.66
CAMSEN setup time	t_{CAMS}	10	—	—		25.67
CAMSEN hold time	t_{CAMH}	3	—	—		
$\overline{\text{ARBUSY}}$ output delay time	t_{ARBYD}	—	—	1/2 t_{cyc}+12		25.68

Note: * The user must ensure that the code satisfies this condition.

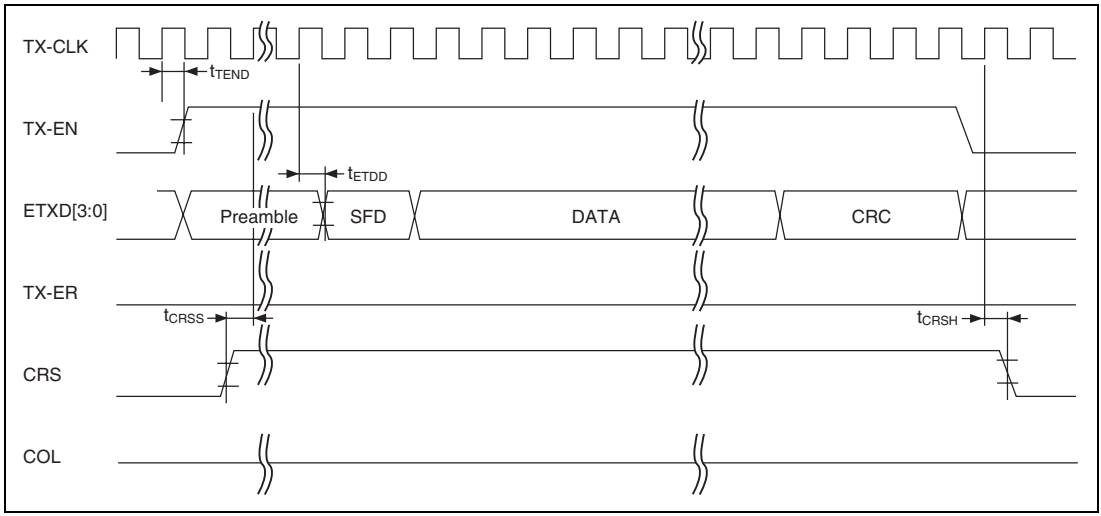


Figure 25.59 MII Transmit Timing (Normal Operation)

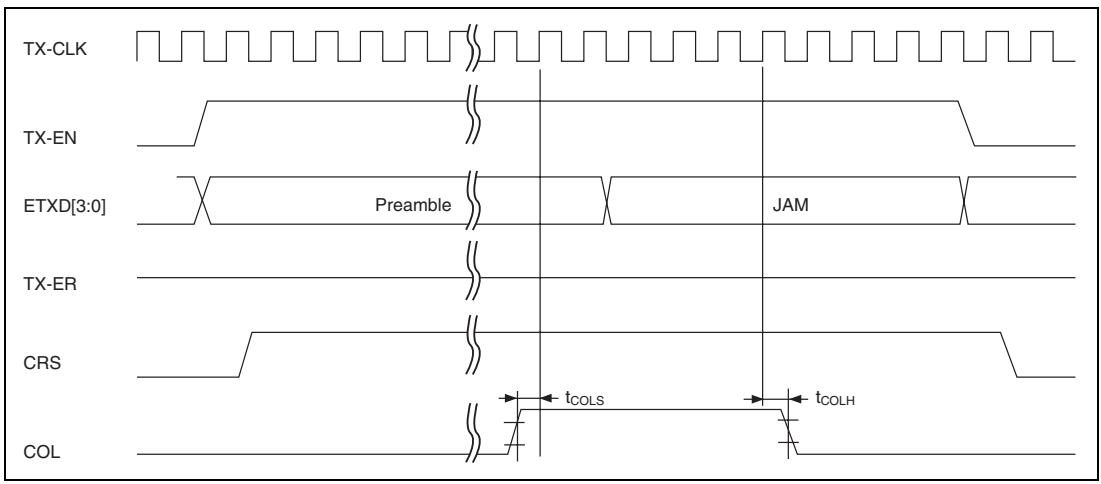


Figure 25.60 MII Transmit Timing (Case of Conflict)

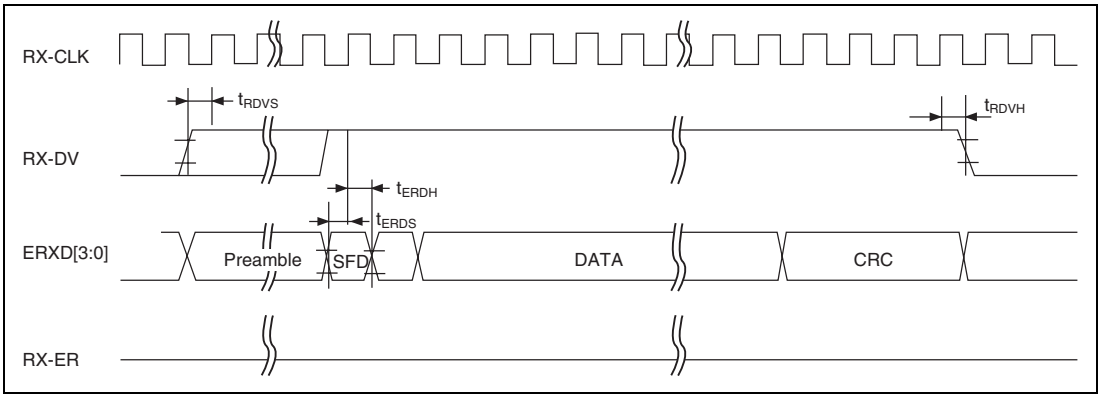


Figure 25.61 MII Receive Timing (Normal Operation)

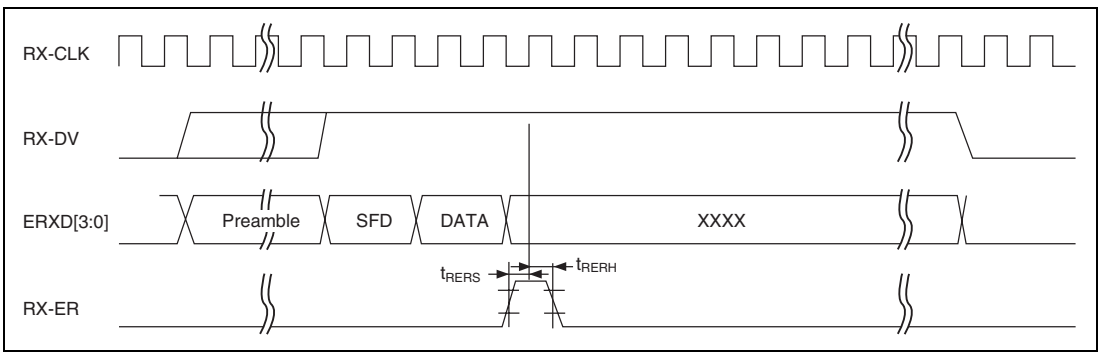


Figure 25.62 MII Receive Timing (Case of Error)

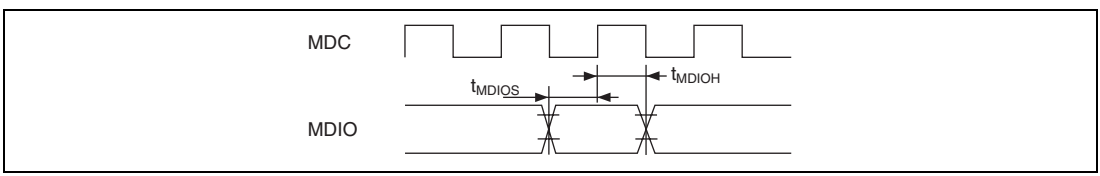


Figure 25.63 MDIO Input Timing

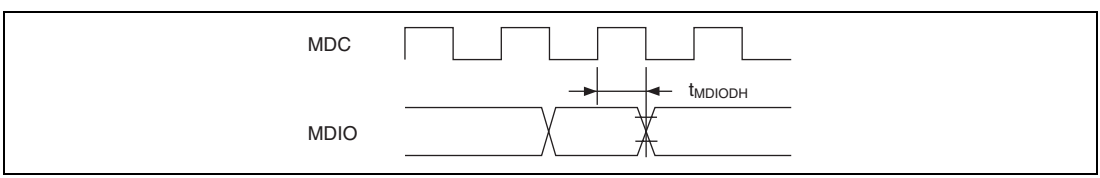


Figure 25.64 MDIO Output Timing

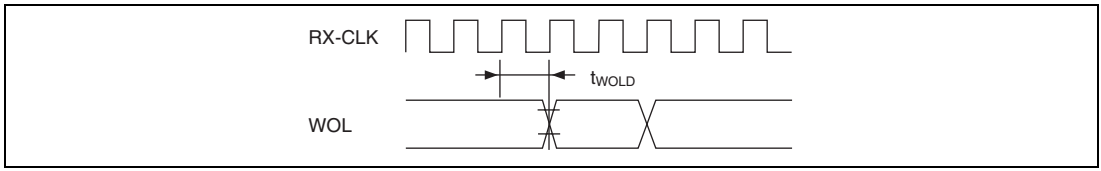


Figure 25.65 WOL Output Timing

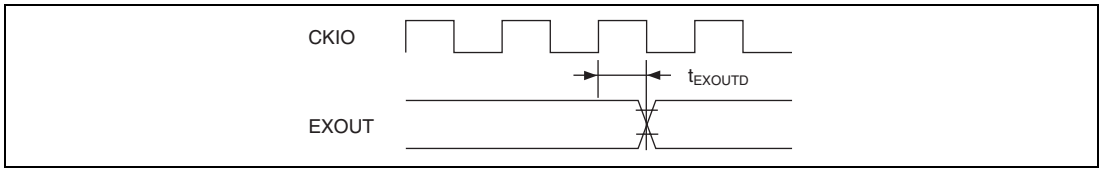


Figure 25.66 EXOUT Output Timing

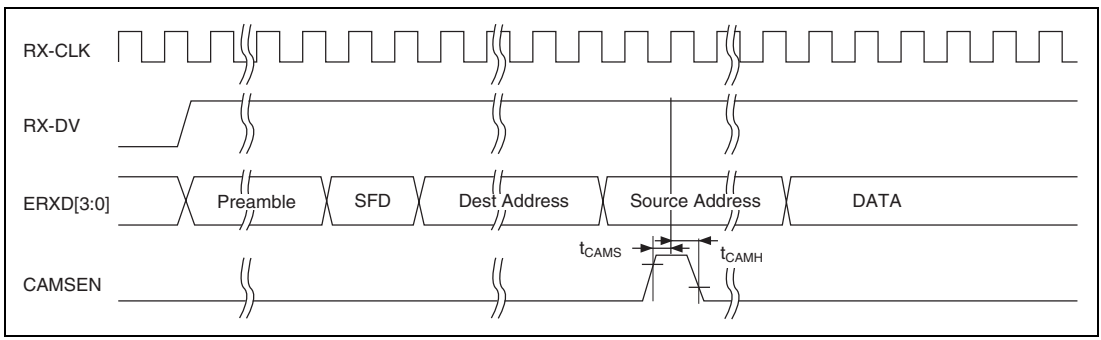


Figure 25.67 CAMSEN Input Timing

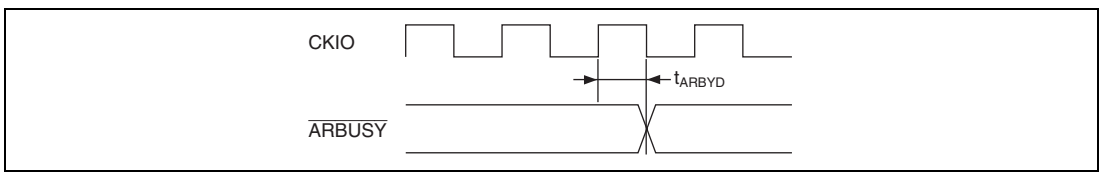


Figure 25.68 $\overline{\text{ARBUBY}}$ Output Timing

25.3.12 Port Input/Output Timing

Table 25.14 Port Input/Output Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Module	Item		Symbol	Min.	Max.	Unit	Figure
Port	Output data delay time		t_{PORTD}	—	17	ns	25.69
	Input data setup time	B:P clock ratio = 1:1	t_{PORTS}	15	—		
		B:P clock ratio = 2:1		t_{cyc}+15	—		
		B:P clock ratio = 4:1		$3 \times t_{cyc}$+15	—		
	Input data hold time		t_{PORTH}	8	—		

Note: t_{cyc} is the output cycle time of the CKIO clock.

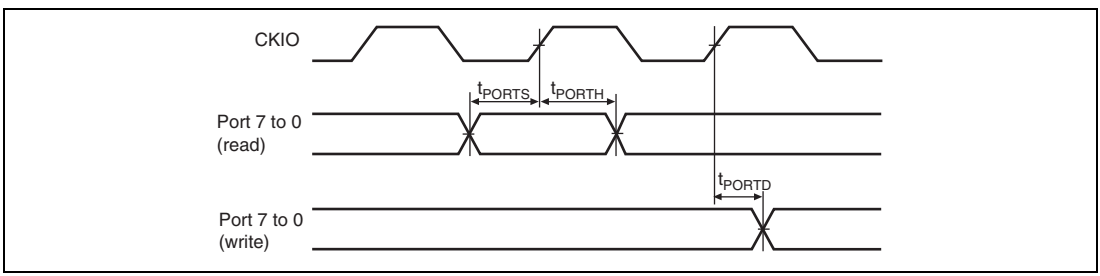


Figure 25.69 I/O Port Timing

25.3.13 H-UDI Related Pin Timing

Table 25.15 H-UDI Related Pin Timing

(Conditions: $V_{CC}Q = V_{CC}Q\text{-RTC} = 3.0$ to 3.6 V, $V_{CC} = V_{CC}\text{-PLL1} = V_{CC}\text{-PLL2} = 1.4$ to 1.6 V, $V_{SS}Q = V_{SS} = V_{SS}Q\text{-RTC} = V_{SS}\text{-PLL1} = V_{SS}\text{-PLL2} = 0$ V, $T_a = -20$ to 75°C)

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50	—	ns	25.70
TCK high-pulse width	t_{TCKH}	12	—	ns	
TCK low-pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall time	t_{TCKR}/t_{TCKF}	—	4	ns	
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	12	—	ns	25.71
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	25.72
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	15	ns	
$\overline{\text{ASEMD0}}$ setup time	$t_{ASEMD0S}$	12	—	ns	25.73
$\overline{\text{ASEMD0}}$ hold time	$t_{ASEMD0H}$	12	—	ns	
$\overline{\text{ASEBRKAK}}$ delay time	$t_{ASBRAKD}$	—	15	ns	25.74

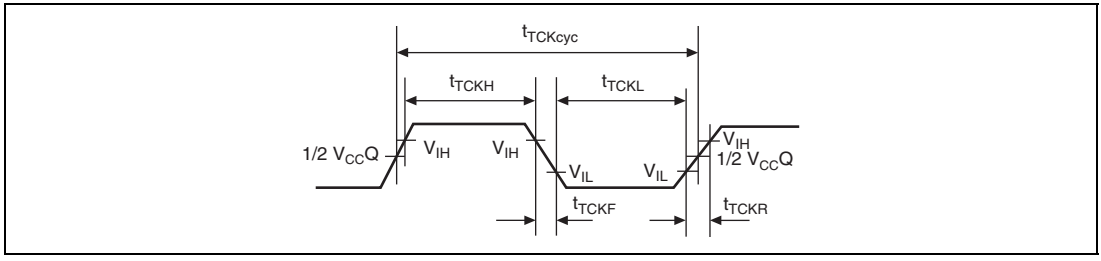


Figure 25.70 TCK Input Timing

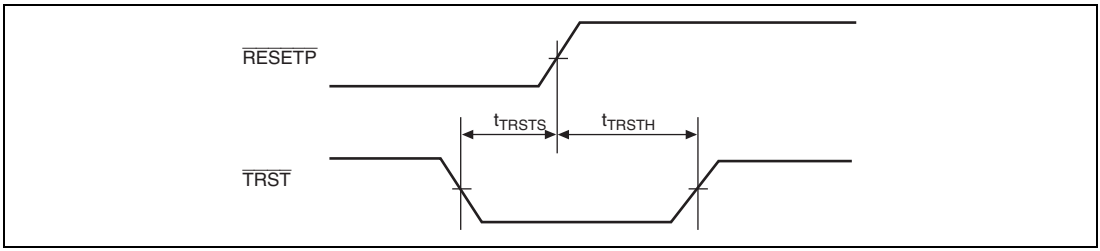


Figure 25.71 $\overline{\text{TRST}}$ Input Timing (Reset Hold)

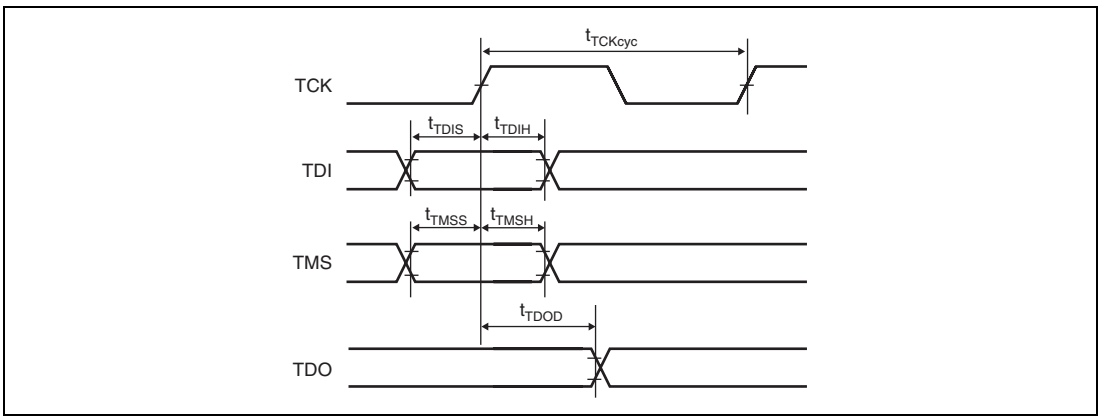


Figure 25.72 H-UDI Data Transfer Timing

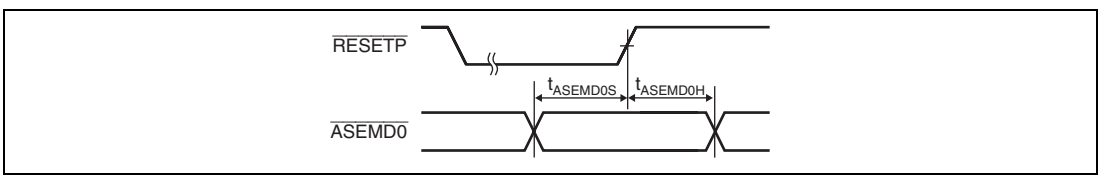


Figure 25.73 $\overline{\text{ASEMD0}}$ Input Timing

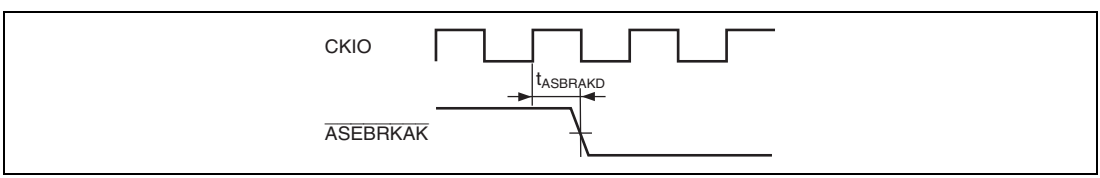
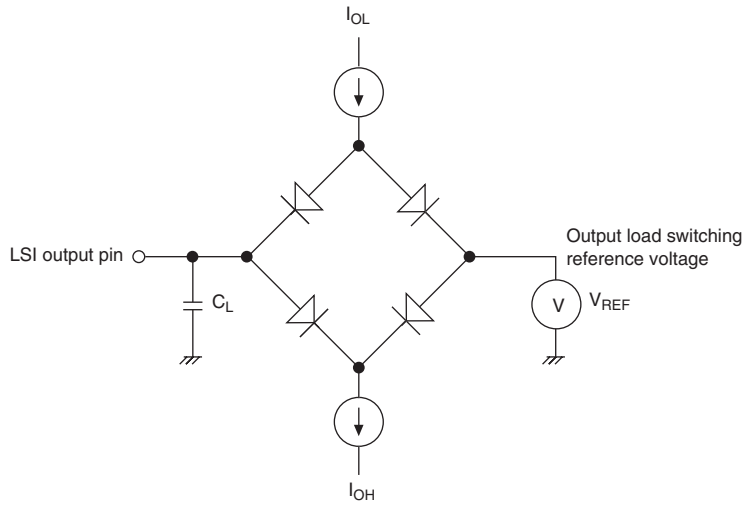


Figure 25.74 $\overline{\text{ASEBRKAK}}$ Delay Time

25.3.14 AC Characteristics Measurement Conditions

- I/O signal reference level: $V_{CC}Q/2$ ($V_{CC}Q$ = 3.0 to 3.6 V, V_{CC} = 1.4 to 1.6 V)
- Input pulse level: $V_{SS}Q$ to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{ASEMD0}}$, NMI, IRQ5 to IRQ0, CKIO, and MD5 to MD0 are within $V_{SS}Q$ to $V_{CC}Q$)
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30 pF: CKIO, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS6B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
2. I_{OL} = 2 mA, I_{OH} = –2 mA

Figure 25.75 Output Load Circuit

25.4 Delay Time Variation Due to Load Capacitance

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated (30 pF) is connected to this LSI's pins is shown below. The graph shown in figure 25.76 should be taken into consideration in the design process if the stipulated capacitance is exceeded in connecting an external device.

If the connected load capacitance exceeds the range shown in figure 25.76, the graph will not be a straight line.

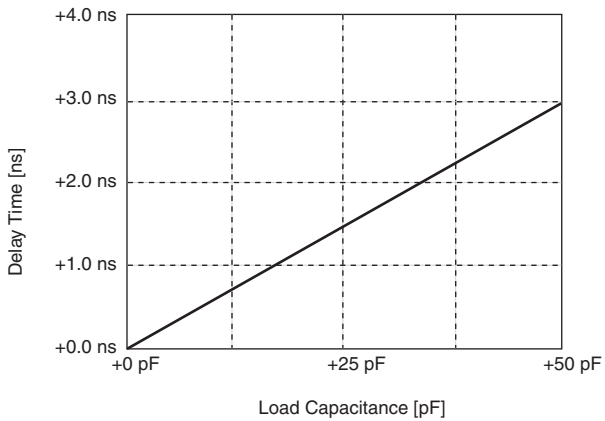


Figure 25.76 Load Capacitance vs. Delay Time

Appendix

A. Pin States and States of Unused Pins

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
$\overline{\text{REFOUT}}$/$\overline{\text{IRQOUT}}$/ $\overline{\text{ARBUSY}}$	O/O/O	H	H	Z	H	H	Open
$\overline{\text{BREQ}}$	I	I	i	Z	I	I	Pull-up
$\overline{\text{BACK}}$	O	O	O	Z	O	L	Open
$\overline{\text{CS0}}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{\text{CS4}}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{\text{CS5A}}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{\text{CS6A}}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{\text{WAIT}}$	I	I	I	Z	I	Z	Pull-up
$\overline{\text{RD}}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{\text{BS}}$	O	H	H	HZ*[4]	O	Z	Open
D0	IO	Z	Z	Z	IO	Z	Pull-up
D1	IO	Z	Z	Z	IO	Z	Pull-up
D2	IO	Z	Z	Z	IO	Z	Pull-up
D3	IO	Z	Z	Z	IO	Z	Pull-up
D4	IO	Z	Z	Z	IO	Z	Pull-up
D5	IO	Z	Z	Z	IO	Z	Pull-up
D6	IO	Z	Z	Z	IO	Z	Pull-up
D7	IO	Z	Z	Z	IO	Z	Pull-up
D8	IO	Z	Z	Z	IO	Z	Pull-up
D9	IO	Z	Z	Z	IO	Z	Pull-up
D10	IO	Z	Z	Z	IO	Z	Pull-up
D11	IO	Z	Z	Z	IO	Z	Pull-up
D12	IO	Z	Z	Z	IO	Z	Pull-up
D13	IO	Z	Z	Z	IO	Z	Pull-up

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
D14	IO	Z	Z	Z	IO	Z	Pull-up
D15	IO	Z	Z	Z	IO	Z	Pull-up
$\overline{WE0}$ ($\overline{BE0}$)/DQMLL	O/O	H	H	HZ*[4]	O	Z	Open
$\overline{WE1}$($\overline{BE1}$)/DQMLU /$\overline{WE}$	O/O/O	H	H	HZ*[4]	O	Z	Open
RD/$\overline{WR}$	O	H	H	HZ*[4]	O	Z	Open
CKIO	IO	IO*[1]	Z*[4]IO*[1]	Z*[4]IO*[1]	IO*[1]	Z*[4]IO*[1]	Open
$\overline{CAS}$	O	H	H	HZ*[4]	O	HZ*[4]	Open
CKE	O	H	O	HZ*[4]	O	OZ*[5]	Open
$\overline{RAS}$	O	H	H	HZ*[4]	O	HZ*[4]	Open
$\overline{CS2}$	O	H	H	HZ*[4]	O	Z	Open
$\overline{CS3}$	O	H	H	HZ*[4]	O	Z	Open
A0	O	O	O	OZ*[5]	O	Z	Open
A1	O	O	O	OZ*[5]	O	Z	Open
A2	O	O	O	OZ*[5]	O	Z	Open
A3	O	O	O	OZ*[5]	O	Z	Open
A4	O	O	O	OZ*[5]	O	Z	Open
A5	O	O	O	OZ*[5]	O	Z	Open
A6	O	O	O	OZ*[5]	O	Z	Open
A7	O	O	O	OZ*[5]	O	Z	Open
A8	O	O	O	OZ*[5]	O	Z	Open
A9	O	O	O	OZ*[5]	O	Z	Open
A10	O	O	O	OZ*[5]	O	Z	Open
A11	O	O	O	OZ*[5]	O	Z	Open
A12	O	O	O	OZ*[5]	O	Z	Open
A13	O	O	O	OZ*[5]	O	Z	Open
A14	O	O	O	OZ*[5]	O	Z	Open
A15	O	O	O	OZ*[5]	O	Z	Open
A16	O	O	O	OZ*[5]	O	Z	Open
A17	O	O	O	OZ*[5]	O	Z	Open

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
$\overline{WE2}$ ($\overline{BE2}$)/ DQMUL/$\overline{ICIORD}$	O/O/O	H	H	HZ*[4]	O	Z	Open
$\overline{WE3}$ ($\overline{BE3}$)/ DQMUU/$\overline{ICIOWR}$	O/O/O	H	H	HZ*[4]	O	Z	Open
D16	IO	Z	Z	Z	IO	Z	Pull-up
D17	IO	Z	Z	Z	IO	Z	Pull-up
D18	IO	Z	Z	Z	IO	Z	Pull-up
D19	IO	Z	Z	Z	IO	Z	Pull-up
D20	IO	Z	Z	Z	IO	Z	Pull-up
D21	IO	Z	Z	Z	IO	Z	Pull-up
D22	IO	Z	Z	Z	IO	Z	Pull-up
D23	IO	Z	Z	Z	IO	Z	Pull-up
D24	IO	Z	Z	Z	IO	Z	Pull-up
D25	IO	Z	Z	Z	IO	Z	Pull-up
D26	IO	Z	Z	Z	IO	Z	Pull-up
D27	IO	Z	Z	Z	IO	Z	Pull-up
D28	IO	Z	Z	Z	IO	Z	Pull-up
D29	IO	Z	Z	Z	IO	Z	Pull-up
D30	IO	Z	Z	Z	IO	Z	Pull-up
D31	IO	Z	Z	Z	IO	Z	Pull-up
A18	O	O	O	OZ*[5]	O	Z	Open
A19	O	O	O	OZ*[5]	O	Z	Open
A20	O	O	O	OZ*[5]	O	Z	Open
A21	O	O	O	OZ*[5]	O	Z	Open
A22	O	O	O	OZ*[5]	O	Z	Open
A23	O	O	O	OZ*[5]	O	Z	Open
A24	O	O	O	OZ*[5]	O	Z	Open
A25	O	O	O	OZ*[5]	O	Z	Open
PTB0	IO	V	P*[2]Z	K*[3]Z	P*[2]	P*[2]Z	Open
PTB1/$\overline{CTS1}$	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
PTB2/$\overline{\text{RTS1}}$	IO/O	V	P*[2]O	K*[3]Z	P*[2]O	P*[2]O	Open
PTB3/RXD1	IO/I	V	P*[2]Z	K*[3]Z	P*[2]I	P*[2]Z	Open
PTB4/TXD1	IO/O	V	P*[2]Z	K*[3]Z	P*[2]O	P*[2]Z	Open
PTB5/SCIF1CK	IO/IO	V	P*[2]Z	K*[3]Z	P	P*[2]Z	Open
PTB6/$\overline{\text{CTS0}}$	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTB7/$\overline{\text{RTS0}}$	IO/O	V	P*[2]O	K*[3]Z	P*[2]O	P*[2]O	Open
PTA0/RXD0	IO/I	V	P*[2]Z	K*[3]Z	P*[2]I	P*[2]Z	Open
PTA1/TXD0	IO/O	V	P*[2]Z	K*[3]Z	P*[2]O	P*[2]Z	Open
PTA2/SCIF0CK	IO/IO	V	P*[2]Z	K*[3]Z	P	P*[2]Z	Open
PTA3/SCK_SIO0	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
PTA4/SIOMCLK0	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTA5/RXD_SIO0	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTA6/TXD_SIO0	IO/O	V	P*[2]O	K*[3]Z	P*[2]O	P*[2]O	Open
PTA7/SIOFSYNC0	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
CRS1	I	I	I	Z	I	I	Pull-down
COL1	I	I	I	Z	I	I	Pull-down
ETXD13	O	O	O	O	O	O	Open
ETXD12	O	O	O	O	O	O	Open
ETXD11	O	O	O	O	O	O	Open
ETXD10	O	O	O	O	O	O	Open
TX-EN1	O	O	O	O	O	O	Open
TX-CLK1	I	I	I	Z	I	I	Pull-down
TX-ER1	O	O	O	O	O	O	Open
RX-ER1	I	I	I	Z	I	I	Pull-down
RX-CLK1	I	I	I	Z	I	I	Pull-down
RX-DV1	I	I	I	Z	I	I	Pull-down
ERXD10	I	I	I	Z	I	I	Pull-down
ERXD11	I	I	I	Z	I	I	Pull-down
ERXD12	I	I	I	Z	I	I	Pull-down
ERXD13	I	I	I	Z	I	I	Pull-down

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
MDC1	O	O	O	O	O	O	Open
MDIO1	IO	I	I	Z	I	I	Pull-down
WOL1	O	O	O	O	O	O	Open
LNKSTA1	I	I	I	Z	I	I	Pull-down
EXOUT1/TEND1	O/O	O	O	O	O	O	Open
CAMSEN1/IRQ5	I/I	I	I	ZI*[6]	I	I	Pull-down
CRS0	I	I	I	Z	I	I	Pull-down
COL0	I	I	I	Z	I	I	Pull-down
ETXD03	O	O	O	O	O	O	Open
ETXD02	O	O	O	O	O	O	Open
ETXD01	O	O	O	O	O	O	Open
ETXD00	O	O	O	O	O	O	Open
TX-EN0	O	O	O	O	O	O	Open
TX-CLK0	I	I	I	Z	I	I	Pull-down
TX-ER0	O	O	O	O	O	O	Open
RX-ER0	I	I	I	Z	I	I	Pull-down
RX-CLK0	I	I	I	Z	I	I	Pull-down
RX-DV0	I	I	I	Z	I	I	Pull-down
ERXD00	I	I	I	Z	I	I	Pull-down
ERXD01	I	I	I	Z	I	I	Pull-down
ERXD02	I	I	I	Z	I	I	Pull-down
ERXD03	I	I	I	Z	I	I	Pull-down
MDC0	O	O	O	O	O	O	Open
MDIO0	IO	I	I	Z	I	I	Pull-down
WOL0	O	O	O	O	O	O	Open
LNKSTA0	I	I	I	Z	I	I	Pull-down
EXOUT0/TEND0	O/O	O	O	O	O	O	Open
CAMSEN0/IRQ4	I/I	I	I	ZI*[6]	I	I	Pull-down
MD4	I	I	i	Z	I	i	Must be used

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
MD5	I	I	i	I	I	i	Must be used
XTAL2	O	O	O	O	O	O	Open
EXTAL2	I	I	I	I	I	I	Pull-up
$\overline{\text{ASEMD0}}$	I	M	V	Z	V	V	Must be used
TDI	I	M	M	V	M	M	Open
TMS	I	M	M	V	M	M	Open
TDO	O	Z	Z	Z	O	Z	Open
$\overline{\text{TRST}}$	I	M	M	V	M	M	Must be used
TCK	I	M	M	V	M	M	Open
$\overline{\text{ASEBRKAK}}$	O	V	O	O	O	O	Open
$\overline{\text{AUDSYNC}}$	O	Z	O	O	O	O	Open
AUDCK	O	O	O	O	O	O	Open
AUDATA3	O	Z	O	O	O	O	Open
AUDATA2	O	Z	O	O	O	O	Open
AUDATA1	O	Z	O	O	O	O	Open
AUDATA0	O	Z	O	O	O	O	Open
$\overline{\text{RESETM}}$	I	I	I	I	I	I	Pull-up
$\overline{\text{RESETP}}$	I	I	I	I	I	I	Must be used
NMI	I	I	I	I	I	I	Pull-up
IRQ0/$\overline{\text{IRL0}}$	I	Z	I	I	I	I	Pull-up
IRQ1/$\overline{\text{IRL1}}$	I	Z	I	I	I	I	Pull-up
IRQ2/$\overline{\text{IRL2}}$	I	Z	I	I	I	I	Pull-up
IRQ3/$\overline{\text{IRL3}}$	I	Z	I	I	I	I	Pull-up
STATUS0	O	H	H	H	L	L	Open
STATUS1	O	H	H	L	H	L	Open
CKIO2	O	O	O	OZ*[5]	O	OZ*[5]	Open
DACK0	O	Z	O	Z	O	O	Open

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
DACK1	O	Z	O	Z	O	O	Open
DREQ0	I	I	Z	Z	I	I	Pull-up
DREQ1	I	I	Z	Z	I	I	Pull-up
PTC0/SCK_SIO1	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
PTC1/SIOMCLK1	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTC2/RXD_SIO1	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
PTC3/TXD_SIO1	IO/O	V	P*[2]O	K*[3]Z	P*[2]O	P*[2]O	Open
PTC4/SIOFSYNC1	IO/IO	V	P*[2]I	K*[3]Z	P	P*[2]I	Open
PTC5/$\overline{\text{CE2A}}$	IO/O	V	P*[2]O	K*[3]H	P*[2]O	P*[2]Z	Open
PTC6/$\overline{\text{CE2B}}$	IO/O	V	P*[2]O	K*[3]H	P*[2]O	P*[2]Z	Open
PTC7/$\overline{\text{IOIS16}}$	IO/I	V	P*[2]I	K*[3]Z	P*[2]I	P*[2]I	Open
CS5B/$\overline{\text{CE1A}}$	O/O	H	H	HZ*[4]	O	Z	Open
CS6B/$\overline{\text{CE1B}}$	O/O	H	H	HZ*[4]	O	Z	Open
MD0	I	I	i	I	I	i	Must be used
MD1	I	I	i	I	I	i	Must be used
MD2	I	I	i	I	I	i	Must be used
MD3	I	I	i	Z	I	i	Must be used
XTAL	O	O	O	O	O	O	Open
EXTAL	I	I	I	I	I	I	Pull-up
VccQ		—	—	—	—	—	VccQ
VssQ		—	—	—	—	—	VssQ
Vcc		—	—	—	—	—	Vcc
Vss		—	—	—	—	—	Vss
VccQ-RTC		—	—	—	—	—	VccQ
VssQ-RTC		—	—	—	—	—	VssQ
Vcc-PLL1		—	—	—	—	—	Vcc*[7]

Pin Name	I/O	Reset		Power-Down States		Release of Bus Mastership	Handling of Unused Pins
		Power-on Reset	Manual Reset	Software Standby	Sleep		
Vss-PLL1		—	—	—	—	—	Vss*[7]
Vcc-PLL2		—	—	—	—	—	Vcc*[7]
Vss-PLL2		—	—	—	—	—	Vss*[7]

[Legend]

I: Input state

i: Input state (however, input is fixed by the internal logic)

O: Output state (undefined although the level is high or low)

L: Low-level output

H: High-level output

Z: High impedance (input or output buffer off)

V: Input/output buffer off, pull-up on

M: Input buffer on, output buffer off, pull-up on

K: Output buffer on or input buffer off (pull-up on or off), depending on register settings

P: Input or output depending on register settings

Notes:

1. Depends on clock mode.
2. The state is P when the port function is used.
3. The state is K when the port function is used.
4. The state is Z or H depending on register settings.
5. The state is Z or O depending on register settings.
6. The state is Z when the Ethernet controller function is used.
7. To avoid the power friction, Vcc-PLL1, Vcc-PLL2, Vss-PLL1, Vss-PLL2, and other Vcc and Vss should be wired in three independent patterns from the board power-supply source.

B. Package Dimensions

Figure B.1 and Figure B.2 show the SH7710 package dimensions.

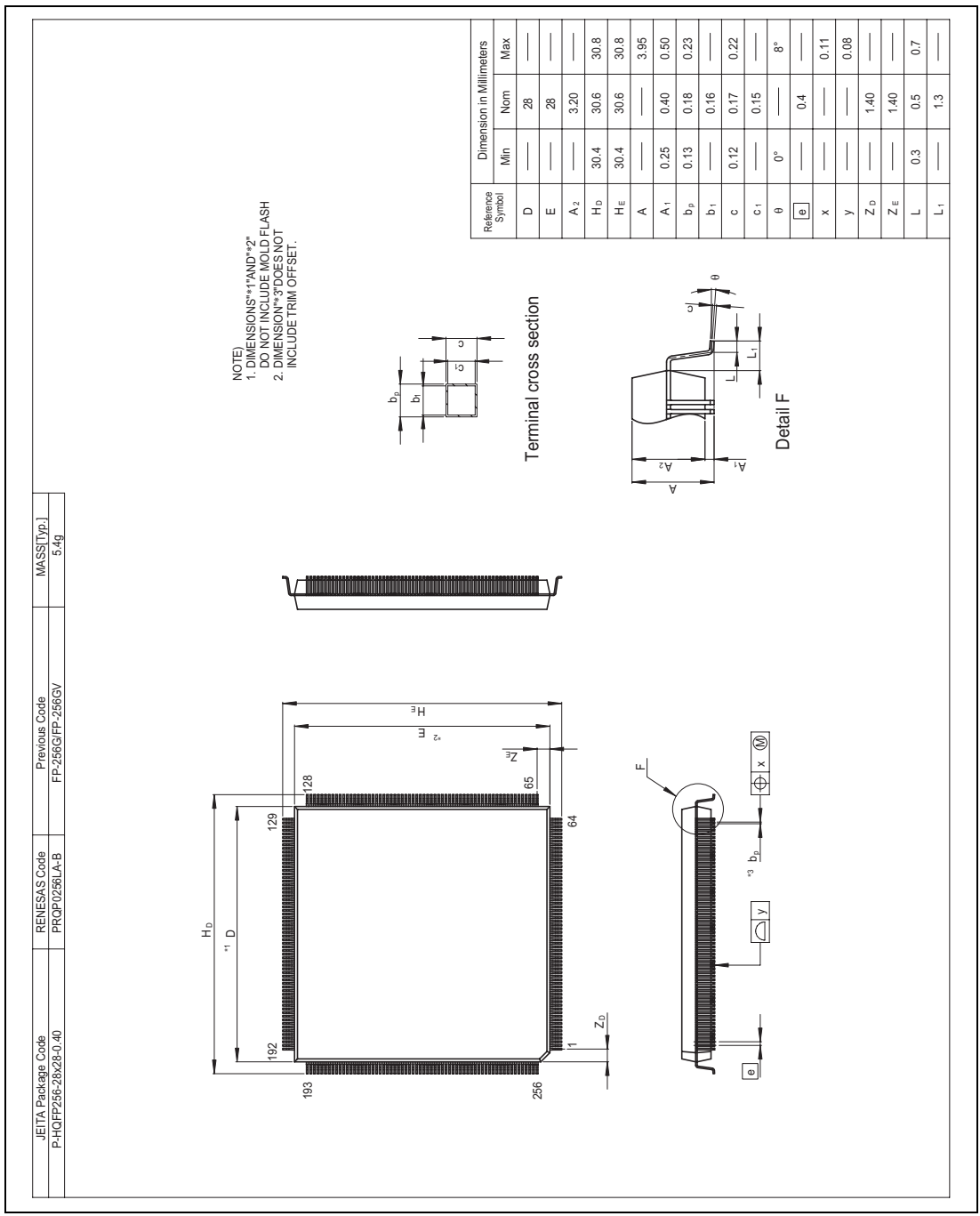
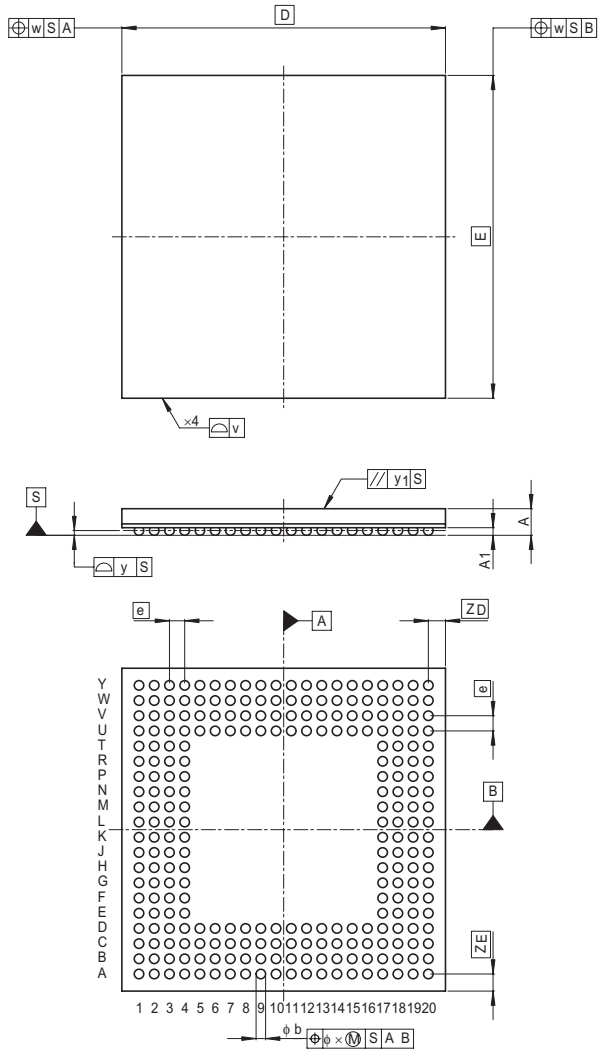


Figure B.1 Package Dimensions (HQFP2828-256 (FP-256G/GV))

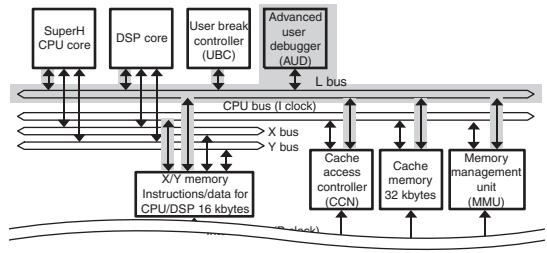
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA256-17x17-0.80	PLBG0256GA-A	BP-256H/BP-256HV	0.6g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	17.0	—
E	—	17.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.2
SD	—	—	—
SE	—	—	—
ZD	—	0.9	—
ZE	—	0.9	—

Figure B.2 Package Dimensions (P-LFBGA1717-256 (BP-256H/HV))

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
Section 1 Overview and Pin Function 1.1 Features	4	Realtime clock (RTC)*1
	5	• Detects short frames and long frames • Conforms to MII (Media Independent Interface) standard *2 : • WOL (Wake-On-LAN) signal output with Magic Packet*3 detection
	6	Notes: 1. As the power supply is connected, power should always be supplied to all power supplies even if only RTC operates. 2. +5 V I/O is not supported. 3. Magic Packet is the registered trademark of Advanced Micro Devices Inc.
		Product Lineup added
1.2 Block Diagram Figure 1.1 Block Diagram	7	
1.3 Pin Description 1.3.1 Pin Assignment	8, 9	HQFP2828-256 (FP-256G/GV) Figure 1.3 Pin Assignment (P-LFBGA1717-256 (BP-256H/HV)) added

Item	Page	Revision (See Manual for Details)
Table 1.1 Pin Assignment	10	

Pin No. (FP-256G/GV)	Pin No. (BP-256H/HV)	Pin Name
1	B2	$\overline{\text{REFOUT}}$/IRQOUT/ $\overline{\text{ARBUSY}}$
2	C2	$\overline{\text{BREQ}}$
3	D2	VccQ
4	B1	VssQ
5	E2	$\overline{\text{BACK}}$
:	:	:

Notes:
1. VccQ-RTC must be supplied even if the realtime clock (RTC) is not used.
2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal oscillator circuit for RTC operates with VccQ-RTC. The control circuit and the RTC counter operate with Vcc (common to the internal circuit). Therefore, all power supplies other than VccQ-RTC should always be turned on even if only RTC operates.
3. Vcc-PLL1/Vcc-PLL2 must be supplied even if the on-chip CPG is not used.
4. VccQ (3.3 V), Vcc (1.5 V), VssQ, and Vss must be connected to the system power supply (for uninterrupted supply).

Section 2 CPU 2.4.2 Memory Data Formats	42	Note: The CPU instruction codes of this LSI must be stored in word units. In big endian mode, the instruction code must be stored from upper byte to lower byte in this order from the word boundary of the memory.
Section 3 DSP Operating Unit 3.6.3 Single-Data Transfer Instructions Table 3.37 Single Data Transfer Instructions	143	

Instruction	Operation
MOVS.W @As + Ix, Ds	(As)→MSW of Ds, 0→LSW of Ds, As + Ix→As

Item | **Page** | **Revision (See Manual for Details)**

3.6.4 DSP Operation Instructions

Table 3.39 DSP Operation Instructions

Page: 145

Instruction	Operation
PSUB Sx,Sy,Du PMULS Se,Sf,Dg	Sx-Sy ->Du Se*Sf ->Dg (Signed)
PRND Sx,Dz	Sx + h'00008000 ->Dz h'0000 ->LSW of Dz
PRND Sy,Dz	Sy + h'00008000 ->Dz h'0000 ->LSW of Dz

3.6.5 Operation Code Map in DSP Mode

Table 3.40 Operation Code Map

Page: 151

The shade of blanks in table 3.40 deleted.

Section 4 Exception Handling

4.2.5 Exception Source Acceptance Timing and Priority

Table 4.1 Exception Event Vectors

Page: 163

Exception Event	Priority*[1]	Exception Order	Process at BL = 1	Vector Code	Vector Offset
H-UDI reset	1	1	Reset	H'000	—

4.3 Individual Exception Operations

4.3.1 Resets

H-UDI Reset

Page: 165

Added

Section 7 X/Y Memory

7.3.5 Address Error

Page: 234

Added

Section 11 On-Chip Oscillation Circuits

11.2.2 Input/Output Pins

Table 11.1 Pin Configuration

Page: 315

Note: To prevent device malfunction, the value of the mode control pin is sampled only by a power-on reset.

Item	Page	Revision (See Manual for Details)
11.9 Notes on Board Design Bypass Capacitors:	329	Pin assignments of HQFP2828-256 (FP-256G/GV) Vss/VssQ and Vcc/VccQ pair of digital circuitry : Vss/VssQ and Vcc/VccQ pair of the on-chip oscillator 193 and 196, 251 and 252, 253 and 254 Pin assignments of P-LFGA1717-256 (BP-256H/HV) Vss/VssQ and Vcc/VccQ pair of digital circuitry D2-B1, E1-F4, G2-G3, J2-J4, L3-L2, N3-N2, R4-P2, U2-W1, V4-Y4, Y6-U7, W8-V8, V10-W10, V11-W11, V13-W13, U15-W14, W17-Y19, U18-U20, P17-N19, N18-P20, L17-L20, J18-J19, E20-E17, D19-B20, C19-A20, D15-B14, C14-A15, B11-D11, C10-B10, C8-B8, D6-B7 Vss/VssQ and Vcc/VccQ pair of the on-chip oscillator B19-A19, C3-B5, and C5-C4
Section 12 Bus State Controller (BSC) 12.4 Register Descriptions	341	• Refresh timer control/status register (RTCSR)*1 • Refresh timer counter (RTCNT)*1 • Refresh time constant register (RTCOR)*1 • SDRAM mode register for area 2 (SDMR2)*2 • SDRAM mode register for area 3 (SDMR3)*2 Notes: 1. This register only accepts 32-bit writing to prevent incorrect writing. In this case, the upper 16 bits of the data must be H'A55A. Otherwise, writing cannot be performed. In reading, the upper 16 bits are read as H'0000. 2. The contents of this register are stored in SDRAM. When this register space is accessed, the corresponding register in SDRAM is written to. For details, see description of Power-on Sequence in section 12.5.5, SDRAM Interface.

Item	Page	Revision (See Manual for Details)
12.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 4, 5A, 5B, 6A, 6B) Normal Space, Byte-Selection SRAM: • CS2WCR, CS3WCR • CS4WCR • CS5AWCR • CS5BWCR • CS6AWCR	352 to 361	**Bit** / **Bit Name** / **Description** 10 WR3 / 9 WR2 / 8 WR1 / 7 WR0: Number of Access Wait Cycles Specify the number of wait cycles that are necessary for read/write access. 0000: 0 cycle :
Burst ROM (Clock Asynchronous): • CS0WCR • CS4WCR	363 to 365	**Bit** / **Bit Name** / **Description** 10 W3 / 9 W2 / 8 W1 / 7 W0: Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first read/write access cycle. :
SDRAM* • CS2WCR	367	**Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 31 to 11 / — / All 0 / R / Reserved These bits are always read as 0. The write value should always be 0. 10 / — / 1 / R / Reserved These bits are always read as 1. The write value should always be 1. 9 / — / 0 / R / Reserved These bits are always read as 0. The write value should always be 0. 8 / A2CL1 / 1 / R/W / CAS Latency for Area 2 7 / A2CL0 / 0 / R/W / Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

Item	Page	Revision (See Manual for Details)

- CS3WCR — 368

Bit	Bit Name	Description
14 13	TRP1 TRP0	Number of Wait Cycles Waiting Completion of Precharge Specify the number of minimum wait cycles to be inserted to wait the completion of precharge. The setting for areas 2 and 3 is common. (1) From starting auto-charge to issuing the ACTV command for the same bank (2) From issuing the PRE/PALL command to issuing the ACTV command for the same bank (3) To transiting to power-down mode/deep power-down mode (4) From issuing the PALL command at auto-refresh to issuing the REF command (5) From issuing the PALL command at self-refresh to issuing the SELF command 00: 0 cycle 01: 1 cycles 10: 2 cycles 11: 3 cycles
11 10	TRCD1 TRCD0	Number of Wait Cycles from ACTV Command to READ (A)/WRIT (A) Command Specify the number of minimum wait cycles from issuing the ACTV command to issuing the READ (A)/WRIT (A) command. The setting for areas 2 and 3 is common. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

Item	Page	Revision (See Manual for Details)
• CS3WCR	370	

Bit	Bit Name	Description
4	TRWL1	Number of Wait Cycles Waiting Start of Precharge Specify the number of minimum wait cycles to be inserted to wait the start of precharge. The setting for areas 2 and 3 is common. (1) This LSI is in non-bank active mode from the issue of the WRITA command to the start of auto-precharge in SDRAM, and issues the ACTV command for the same bank after issuing the WRITA command. Confirm how many cycles are required from the reception of the WRITA command to the start of auto-precharge in each SDRAM data sheet. Set this bit so that the number of cycles is not above the cycles specified by this bit. (2) This LSI is in bank active mode from issuing the WRIT command to issuing the PRE command, and the access to different row address in the same bank is performed. 00: 0 cycle (No wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles
3	TRWL0	
1	TRC1	Number of Idle Cycles from REF Command/Self-refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles between the commands in the following cases. The setting for areas 2 and 3 is common. (1) From issuing the REF command to issuing the ACTV/REF/MSR command (2) From releasing self-refresh to issuing the ACTV/REF/MSR command 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles
0	TRC0	

Item	Page	Revision (See Manual for Details)

Burst ROM (Clock Synchronous)

- CS0WCR

375

Bit	Bit Name	R/W	Description
10	W3	R/W	Number of Access Wait CyclesSpecify the number of wait cycles to be inserted in the first read/write access cycle.
9	W2	R/W	
8	W1	R/W	
7	W0	R/W	:

12.4.4 SDRAM Control Register (SDCR)

378

Bit	Bit Name	Description
9	PDOWN	Power-Down Mode Specify whether SDRAM is put in power-down mode or not after the access to memory other than SDRAM is completed. This bit, when set to 1, drives the CKE pin low and places SDRAM in power-down mode by using an access to a memory other than SDRAM as a trigger.

Table 12.12 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)

401

Output Pin of This LSI	Row Address Output	Column Address Output	Synchronous DRAM Pin	Function
A14	A22*2	A22*2	A12 (BA1)*3	Specifies bank
A13	A21*2	A21*2	A11 (BA0)	

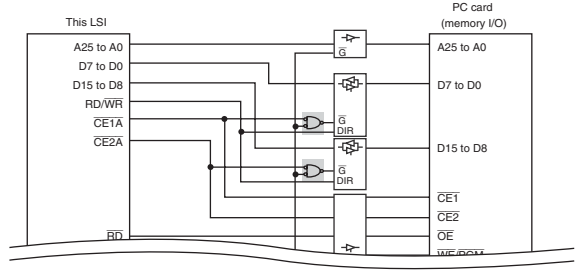
Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

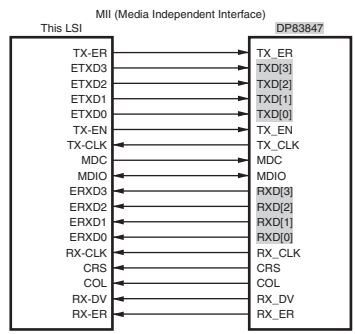
3. If the number of 16-Mbit pins (512 kwords × 16 bits × 2 banks: pin with 8-bit column) is two, the bank address specification is not required. Therefore, the bank address should be not used.

Item	Page	Revision (See Manual for Details)
Table 12.15 Relationship between A2/3BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1	408	

Output Pin of This LSI	Row Address Output	Column Address Output	Synchronous DRAM Pin	Function
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*2	A20*[2]	A11 (BA0)	Specifies bank

Burst Read:	414	In this LSI, wait cycles can be inserted by specifying each bit in CSnWCR to connect the SDRAM in variable frequencies. Figure 12.15 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READA command is output can be specified using the TRCD1 and TRCD0 bits in CS3WCR. If the TRCD1 and TRCD0 bits specify one cycle or more, a Trw cycle where the NOP command is issued is inserted between the Tr cycle and Tc1 cycle.
1. Auto-refreshing	428	A NOP cycle is inserted between the Tp cycle and Trr cycle when the setting value of the TRP[1:0] bits in CSnWCR is longer than or equal to 1 cycle.
Figure 12.38 Example of PCMCIA Interface Connection	447	

Item	Page	Revision (See Manual for Details)
Section 15 Realtime Clock (RTC) 15.2 Input/Output Pins Table 15.1 Pin Configuration	507	Note: 1. Pull up (VccQ-RTC) the EXTAL2 pin, and open (NC) the XTAL2 pin when the realtime clock (RTC) is not used. 2. RTC in this LSI does not operate even if VccQ-RTC is turned on. The crystal oscillator circuit for RTC operates with VccQ-RTC. The control circuit and the RTC counter operate with Vcc (common to the internal circuit). Therefore, all power supplies other than VccQ-RTC should always be turned on even if only RTC operates.
15.5.4 Usage Note about RTC Power Supply	530	Added
Section 17 Serial I/O with FIFO (SIOF) 17.4 Operation 17.4.1 Serial Clocks Table 17.2 SIOF Serial Clock Frequency	614	Sampling Rate Frame Length: 8 kHz: 96 kHz 32 bits: 256 kHz: 3.072 MHz
Figure 17.13 Transmission and Reception Timings (8-Bit Monaural Data (1))	631	Setting: TRMD = 00or10, REDG = 0, FL = 0000 (frame legth: 8 bits) TDLE = 1, TDLA3 to TDLA0 = 0000, TDRE = 0, TDRA3 to TDRA0 = 0000, RDLE = 1, RDLA3 to RDLA0 = 0000, RDRE = 0, RDRA3 to RDRA0 = 0000, CD0E = 0, CD0A3 to CD0A0 = 0000, CD1E = 0, CD1A3 to CD1A0 = 0000
Figure 17.15 Transmission and Reception Timings (16-Bit Monaural Data (1))	632	Setting: TRMD = 00 or 10, REDG = 0, FL = 1101 (frame length: 64 bits) TDLE = 1, TDLA3 to TDLA0 = 0000, TDRE = 0, TDRA3 to TDRA0 = 0000, RDLE = 1, RDLA3 to RDLA0 = 0000, RDRE = 0, RDRA3 to RDRA0 = 0000, CD0E = 0, CD0A3 to CD0A0 = 0000, CD1E = 0, CD1A3 to CD1A0 = 0000
Figure 17.17 Transmission and Reception Timings (16-Bit Stereo Data (2))	633	Setting: TRMD = 01, REDG = 1, FL = 1101 (frame length: 64 bits), TDLE = 1, TDLA3 to TDLA0 = 0000, TDRE = 1, TDRA3 to TDRA0 = 0010, RDLE = 1, RDLA3 to RDLA0 = 0001, RDRE = 1, RDRA3 to RDRA0 = 0011, CD0E = 0, CD0A3 to CD0A0 = 0000, CD1E = 0, CD1A3 to CD1A0 = 0000
Figure 17.18 Transmission and Reception Timings (16-Bit Stereo Data (3))	633	Setting: TRMD = 00 or 10, REDG = 0, FL = 1110 (frame length: 128 bits), TDLE = 1, TDLA3 to TDLA0 = 0000, TDRE = 1, TDRA3 to TDRA0 = 0010, RDLE = 1, RDLA3 to RDLA0 = 0000, RDRE = 1, RDRA3 to RDRA0 = 0010, CD0E = 1, CD0A3 to CD0A0 = 0001, CD1E = 1, CD1A3 to CD1A0 = 0011
Section 18 Ethernet Controller (EtherC) 18.3.2 EtherC Mode Register (ECMR)	645	Bit: 13 — Bit Name: MCT — R/W: R/W

Item	Page	Revision (See Manual for Details)
18.3.36 Added Qtag Value Set Register (Port 0 to 1) (TSU_ADQT0)	679	**Bit** / **Bit Name** / **R/W** 12 / — / R
18.3.37 Added Qtag Value Set Register (Port 1 to 0) (TSU_ADQT1)	680	**Bit** / **Bit Name** / **R/W** 12 / — / R
18.4.4 CAM Function	712	The internal CAM entry table has 32 entries and 32 POST tables, and the POST table can be specified in each entry. The external connection CAM logic configuration is based on pins because POST tables (total of 2) are allocated to the CAMSEN0 and CAMSEN1 pins.
18.5 Connection to LSI Figure 18.13 Example of Connection to DP83847	725	Figure 18.13 shows the example of connection to a DP83847 (National Semiconductor Corporation). 
Figure 18.14 Example of Connection to DP83846AVHG	726	Deleted
Section 19 Ethernet Controller Direct Memory Access Controller (E-DMAC) 19.1 Features	727	Deleted • Improves software handling capability by padding in receive data

Item	Page	Revision (See Manual for Details)
19.2 Register Descriptions	729	Deleted • Receive data padding insert register (RPADIR0) • Receive data padding insert register (RPADIR1)
19.2.2 E-DMAC Transmit Request Register (EDTRR)	731	If the TACT bit of a transmit descriptor is cleared to 0 (invalid), the E-DMAC clears the TR bit and stops transmit DMAC operation. For details of writing to the TR bit, see section 19.4.1, Using of EDTRR and EDRRR.
19.2.3 E-DMAC Receive Request Register (EDRRR)	732	If the RACT bit of the receive descriptor is cleared to 0 (invalid), the E-DMAC clears the RR bit and stops receive DMAC operation. For details of writing to the RR bit, see section 19.4.1, Using of EDTRR and EDRRR.

Item	Page	Revision (See Manual for Details)
19.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)	743	TRSCER indicates whether multicast address frame receive status information reported by bit 7 in EESR is reflected in the RFE bit in the corresponding register (for details of descriptor descriptions, see section 19.3.1, Descriptors and Descriptor List). The RMAFCE bit in this register corresponds to bit 7 in EER. When the RMAFCE bit is cleared to 0, the receive status (bit 7 in ESSR) is reflected in the REF bit in the receive descriptor. When this bit is set to 0, the status is not reflected in the descriptor even if the corresponding source occurs. The RMAFCE bit is cleared to 0 after a power-on reset and manual reset.

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	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

19.2.19 Receive Data Padding Insert Register (RPADIR)	753	Deleted

Item	Page	Revision (See Manual for Details)

19.3.1 Descriptors and Descriptor List

(2) Receive Descriptor

(a) Receive Descriptor 0 (RD0)

765

Bit	Bit Name	Initial Value	R/W	Description
27	RFE	0	R/W	Receive Frame Error Occurrence Indicates that an error occurred in the receive frame. The errors occurred in RFS8 (bit 8), or RFS3 to RFS0 (bits 3 to 0). TRSCER can specify whether the multicast address frame receive information is reflected in this bit or not.
26 to 0	RFS26 to RFS0	All 0	R/W	Receive Frame Status Indicate the status of the corresponding frame. A bit below, when set to 1, indicates the occurrence of the corresponding event. If the events of RFS8, or RFS4 to RFS0 occur, frames are incompletely received. RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (corresponding to the RFOF bit in EESR) RFS8: Receive abort detected Note: The RFS8 bit is set if any bit of RFS3 to RFS0 is set. RFS7: Multicast address frame received (corresponding to the RMAF bit in EESR) RFS6 and RFS5: Reserved (only 0 should be written to these bits) :

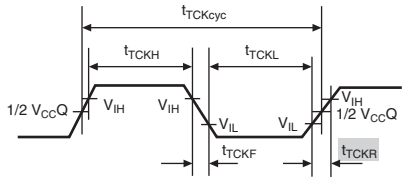
Item	Page	Revision (See Manual for Details)
19.3.5 Padding Insertion in Receive data	773	Deleted
19.3.5 Receive FIFO Overflow Alert Signal ($\overline{\text{ARBUSY}}$)	773	The $\overline{\text{ARBUSY}}$ signal synchronized with the bus clock (B clock) signal is also output to an external pin of this LSI. When the capacity of data received in receive FIFO or the number of receive frames reach the threshold (RFF2 to RFF0, or RFD2 to RFD0) specified in FCFTR in E-DMAC, $\overline{\text{ARBUSY}}$ is valid.
19.4 Usage Notes	776	Added
Section 24 List of Registers 24.1 Register Addresses (by functional module, in order of the corresponding section numbers)	816	RPADIR0 deleted RPADIR1 deleted
24.2 Register Bits	843	

Register Abbreviation	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0
TRSCERn (n = 0, 1)	—	—	—	—	—
	—	—	—	—	—
	—	—	—	—	—
	—	—	—	—	—

844	RPADIRn (n = 0, 1) deleted	

Register Abbreviation	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/1	Bit 24/ 16/8/0
EDOCRn (n = 0, 1)	—	—	—	—	—
	—	—	—	—	—
	—	—	—	—	—
	—	FEC	AEC	—	—

Item	Page	Revision (See Manual for Details)
24.3 Register States in Each Operating Mode	847	**Register Abbreviation** / **Manual Reset*[1]** CMNCR — Retained CS0BCR — Retained CS2BCR — Retained CS3BCR — Retained CS4BCR — Retained CS5ABCR — Retained CS5BBCR — Retained CS6ABCR — Retained CS6BBCR — Retained CS0WCR — Retained CS2WCR — Retained CS3WCR — Retained CS4WCR — Retained CS5AWCR — Retained CS5BWCR — Retained
24.3 Register States in Each Operating Mode	848	**Register Abbreviation** / **Manual Reset*[1]** CS6AWCR — Retained CS6BWCR — Retained SDCR — Retained RTCSR — Retained RTCNT — Retained RTCOR — Retained
	854	RPADIRn (n = 0,1) deleted
Section 25 Electrical Characteristics 25.3.2 Control Signal Timing Table 25.6 Control Signal Timing	865	**66.67 MHz** **Item** / **Symbol** / **Min.** / **Max.** $\overline{\text{IRQOUT}}$ delay time — t_{IRQOTD} — — 1/2 t_{cyc}+12

Item	Page	Revision (See Manual for Details)
Table 25.7 Bus Timing (1)	868	**66.67 MHz** **Item** / **Symbol** / **Min.** / **Max.**
Table 25.13 Ethernet Controller Timing	908	**Item** / **Symbol** / **Max.** $\overline{\text{ARBUSY}}$ output delay time / t_{ARBYD} / 1/2 t_{cyc}+12
25.3.13 H-UDI Related Pin Timing Table 25.15 H-UDI Related Pin Timing	913	**Item** / **Symbol** TCK rise/fall time / t_{TCKF}/t_{TCKR}
Figure 25.70 TCK Input Timing	913	
Appendix B. Package Dimensions	926, 927	Figure B.1 Package Dimensions (HQFP2828-256 (FP-256G/GV)) Figure B.2 Package Dimensions (P-LFBGA1717-256 (BP-256H/HV)) Added

Index

Numerics

16-bit/32-bit displacement 45

A

Absolute addresses 45
Acceptance priority and test priority 162
Address space identifier (ASID)..... 189
Address transition 188
Auto-refreshing..... 428
Auto-request mode 477

B

Baud rate generator (BRG)..... 613
Big endian mode..... 42

C

Control by slot position 620
Control registers 32

D

Delayed branching..... 44
Double data transfer instructions 94
DSP registers 77, 106

E

Exception handling state..... 27
Exception request of instruction synchronous type and instruction asynchronous type 161
Extension of status register (SR) 74
External request mode 477, 487

G

General registers 32
Global base register (GBR)..... 40

I

Instruction length 44
IPG settings..... 723

L

Literal constant..... 45
Little endian mode 43
Load/sStore architecture 44
Low-power consumption state 27

M

Magic Packet..... 722
MII registers..... 719, 720
Modulo register (MOD)..... 75
Multiplexed pins 781
Multiply and accumulate registers 36

O

On-chip peripheral module request..... 478

P

P0/U0 area..... 29
P1 area..... 29
P2 area..... 29
P3 area..... 29
P4 area..... 29
Physical address space 188
Procedure register 36

Program counter 32
Program execution state..... 27

Q

Qtags..... 723

R

Receive descriptor 760
receive FIFO overflow alert signal 773
Re-execution type and processing-completion type exceptions 161
Registers
ARSTR 644, 817, 851
BAMRA 267, 811, 847
BAMRB..... 270, 811, 847
BARA 266, 811, 847
BARB 269, 811, 847
BASRA..... 280, 811, 847
BASRB..... 281, 811, 847
BBRA 267, 811, 847
BBRB 272, 811, 847
BDMRB..... 271, 811, 847
BDRB 270, 811, 847
BETR..... 278, 811, 847
BRCR 274, 811, 847
BRDR 280, 811, 847
BRSR..... 279, 811, 847
CCR1 ..... 217, 810, 846
CCR2 ..... 218, 810, 846
CCR3 ..... 221, 810, 846
CDCR 654, 816, 851
CEFCR 655, 816, 851
CHCR 464, 812, 848
CMNCR..... 342, 811, 847
CNDCR 654, 816, 851
CSnBCR 345, 811, 847
CSnWCR 350, 812, 848
DAR..... 463, 812, 848
DMAOR 469, 813, 848
DMARS 471, 813, 848
DMATCR 464, 812, 848
ECMR 645, 815, 850
ECSIPR..... 649, 815, 850
ECSR 648, 815, 850
EDMR..... 730, 820, 853
EDOCR..... 749, 820, 853
EDRRR 732, 820, 853
EDTRR 731, 820, 853
EESIPR 740, 820, 853
EESR..... 734, 820, 853
EXPEVT 157, 810, 846
FCFTR 751, 820, 854
FDR..... 747, 820, 853
FRECR..... 655, 816, 851
FRQCR 320, 811, 847
FWALCR..... 702, 818, 852
FWNLCR..... 701, 818, 852
ICR0..... 248, 810, 846
ICR1..... 249, 810, 846
INTEVT..... 157, 810, 846
INTEVT2..... 158, 810, 846
IPGR 657, 816, 851
IPR 246, 810, 846
IRR0..... 251, 811, 846
IRR1..... 251, 811, 846
IRR2..... 253, 811, 846
IRR3..... 254, 811, 846
IRR4..... 255, 811, 846
IRR5..... 256, 811, 847
IRR7..... 257, 811, 847
IRR8..... 258, 811, 847
LCCR..... 654, 816, 851
MAFCR 657, 816, 851
MAHR 651, 816, 851
MALR..... 651, 816, 851
MMUCR 191, 810, 846
PACR..... 783, 821, 854
PADR..... 789, 821, 854

PBCR 784, 821, 854
PBDR 790, 821, 854
PCCR 785, 821, 854
PCDR 792, 821, 854
PETCR 786, 821, 854
PIR 650, 816, 851
PSR 653, 816, 851
PTEH 190, 810, 846
PTEL 191, 810, 846
R64CNT 508, 813, 849
RBWAR 750, 820, 853
RCR1 ..... 520, 814, 849
RCR2 ..... 522, 814, 849
RCR3 ..... 524, 814, 849
RDAYAR 517, 814, 849
RDAYCNT 512, 814, 849
RDFAR 750, 820, 853
RDLAR 734, 820, 853
RFCR 656, 816, 851
RFLR 652, 816, 851
RHRAR 515, 814, 849
RHRCNT 509, 813, 849
RMCR 748, 820, 853
RMFCR 744, 820, 853
RMINAR 514, 814, 849
RMINCNT 509, 813, 849
RMONAR 518, 814, 849
RMONCNT 512, 814, 849
RSECAR 514, 814, 849
RSECCNT 508, 813, 849
RTCNT 381, 812, 848
RTCOR 382, 812, 848
RTCSR 380, 812, 848
RWKAR 516, 814, 849
RWKCNT 510, 814, 849
RXALCR 701, 818, 852
RXNLCR 700, 818, 852
RYRAR 519, 814, 849
RYRCNT 513, 814, 849
SAR 463, 812, 848
SCBRR 553, 814, 849
SCFCR 554, 814, 849
SCFDR 556, 814, 850
SCFRDR 536, 814, 849
SCFSR 545, 814, 849
SCFTDR 537, 814, 849
SCLSR 558, 814, 850
SCRSR 536
SCSCR 541, 814, 849
SCSMR 537, 814, 849
SCTSR 537
SDBPR 795
SDBSR 796
SDCR 377, 812, 848
SDID 803, 821, 854
SDIR 795, 821, 854
SICDAR 596, 815, 850
SICTR 598, 815, 850
SIFCTR 601, 815, 850
SIIER 607, 815, 850
SIMDR 591, 815, 850
SIRCR 612, 815, 850
SIRDAR 595, 815, 850
SIRDR 610, 815, 850
SISCR 593, 815, 850
SISTR 603, 815, 850
SITCR 611, 815, 850
SITDAR 594, 815, 850
SITDR 609, 815, 850
STBCR 298, 811, 847
STBCR2 ..... 300, 811, 847
STBCR3 ..... 301, 811, 847
TBRAR 750, 821, 854
TCNT 499, 813, 848
TCOR 499, 813, 848
TCR 498, 813, 848
TDFAR 751, 821, 854
TDLAR 733, 820, 853
TEA 158, 810, 846
TFTR 745, 820, 853

TLFRCR 656, 816, 851
TRA 156, 810, 846
TRIMD 753, 820, 854
TROCR 653, 816, 851
TRSCER 743, 820, 853
TSFRCR 655, 816, 851
TSTR 497, 813, 848
TSU_ADQT0 ..... 679, 817, 852
TSU_ADQT1 ..... 680, 817, 852
TSU_ADRH 698, 818, 853
TSU_ADRL 699, 818, 853
TSU_ADSBSY 681, 817, 852
TSU_BSYSL0 ..... 661, 817, 852
TSU_BSYSL1 ..... 662, 817, 852
TSU_CTRST 658, 817, 851
TSU_FCM 660, 817, 852
TSU_FWEN0 ..... 658, 817, 851
TSU_FWEN1 ..... 659, 817, 851
TSU_FWINMK 675, 817, 852
TSU_FWSL0 ..... 666, 817, 852
TSU_FWSL1 ..... 667, 817, 852
TSU_FWSLC 669, 817, 852
TSU_FWSR 673, 817, 852
TSU_POST1 ..... 686, 817, 852
TSU_POST2 ..... 689, 817, 852
TSU_POST3 ..... 692, 817, 852
TSU_POST4 ..... 695, 817, 852
TSU_PRISL0 ..... 663, 817, 852
TSU_PRISL1 ..... 664, 817, 852
TSU_QTAGM0 ..... 671, 817, 852
TSU_QTAGM1 ..... 672, 817, 852
TSU_TEN 682, 817, 852
TTB 191, 810, 846
TXALCR 700, 817, 852
TXNLCR 699, 817, 852
WTCNT 324, 811, 847
WTCSR 324, 811, 847
Registers addresses 809
Registers bits 809
Registers states 809
Repeat end register (RE) 75
Repeat start register (RS) 75
Reset state 27
Round-robin mode 480

S

Save program counter (SPC) 40
Save status register (SSR) 40
Secondary FS 621
Self-refreshing 429
Single data transfer instructions 95
Single virtual memory mode and multiple virtual memory mode 189
Software standby mode 303
Status register (SR) 38
System control instructions 95
System registers 32

T

T bit 44
the RTC crystal oscillator circuit 528
Transmit descriptor 754

V

Vector base register (VBR) 40
Virtual address space 183

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7710 Group

Publication Date:	1st Edition, Jan. 21, 2004
	Rev.2.00, Dec. 07, 2005
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

© 2005. Renesas Technology Corp., All rights reserved. Printed in Japan.

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

SH7710 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan