To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

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

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

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

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

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

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

User's Manual

32

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.

SH7619 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer SuperHTM RISC engine Family / SH7619 Series

SH7619 R4S76190
R4S76191

Renesas Electronics
www.renesas.com

Rev.6.00 2009.07

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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 SH7619 Group RISC (Reduced Instruction Set Computer) microcomputers include 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 the SH7619 in the design of application systems. Target users 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 the SH7619 to the target users.
Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized 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-1/SH-2/SH-DSP Software Manual.
- In order to understand the details of a register when its name is known
  The addresses, bits, and initial values of the registers are summarized in section 24, List of Registers.

Examples:		
	Register name:	The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or 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 is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

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/

SH7619 Group manuals:

Document Title	Document No.
SH7619 Group Hardware Manual	This manual
SH-1/SH-2/SH-DSP Software Manual	REJ09B0171

User's manuals for development tools:

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

Application note:

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

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

Contents

Section 1 Overview 1
1.1 Features 2
1.2 Block Diagram 7
1.3 Pin Assignments 8
1.4 Pin Functions 9

Section 2 CPU 23
2.1 Features 23
2.2 Register Configuration 23
2.2.1 General Registers (Rn) 25
2.2.2 Control Registers 25
2.2.3 System Registers 26
2.2.4 Initial Values of Registers 27
2.3 Data Formats 28
2.3.1 Register Data Format 28
2.3.2 Memory Data Formats 28
2.3.3 Immediate Data Formats 29
2.4 Features of Instructions 29
2.4.1 RISC Type 29
2.4.2 Addressing Modes 32
2.4.3 Instruction Formats 35
2.5 Instruction Set 39
2.5.1 Instruction Set by Type 39
2.6 Processing States 51
2.6.1 State Transition 51

Section 3 Cache 53
3.1 Features 53
3.1.1 Cache Structure 53
3.1.2 Divided Areas and Cache 55
3.2 Register Descriptions 56
3.2.1 Cache Control Register 1 (CCR1) 56
3.3 Operation 57
3.3.1 Searching Cache 57
3.3.2 Read Access 58
3.3.3 Write Access 59

3.3.4 Write-Back Buffer 59
3.3.5 Coherency of Cache and External Memory 59
3.4 Memory-Mapped Cache 60
3.4.1 Address Array 60
3.4.2 Data Array 61
3.4.3 Usage Examples 63

Section 4 U Memory 65
4.1 Features 65
4.2 Usage Notes 65

Section 5 Exception Handling 67
5.1 Overview 67
5.1.1 Types of Exception Handling and Priority 67
5.1.2 Exception Handling Operations 68
5.1.3 Exception Handling Vector Table 69
5.2 Resets 71
5.2.1 Types of Resets 71
5.2.2 Power-On Reset 71
5.2.3 H-UDI Reset 72
5.3 Address Errors 73
5.3.1 Address Error Sources 73
5.3.2 Address Error Exception Source 73
5.4 Interrupts 74
5.4.1 Interrupt Sources 74
5.4.2 Interrupt Priority 75
5.4.3 Interrupt Exception Handling 75
5.5 Exceptions Triggered by Instructions 76
5.5.1 Types of Exceptions Triggered by Instructions 76
5.5.2 Trap Instructions 76
5.5.3 Illegal Slot Instructions 77
5.5.4 General Illegal Instructions 77
5.6 Cases when Exceptions are Accepted 78
5.7 Stack States after Exception Handling Ends 79
5.8 Usage Notes 81
5.8.1 Value of Stack Pointer (SP) 81
5.8.2 Value of Vector Base Register (VBR) 81
5.8.3 Address Errors Caused by Stacking for Address Error Exception Handling 81
5.8.4 Notes on Slot Illegal Instruction Exception Handling 81

Section 6 Interrupt Controller (INTC) 83
6.1 Features 83
6.2 Input/Output Pins 85
6.3 Register Descriptions 85
6.3.1 Interrupt Control Register 0 (ICR0) 86
6.3.2 IRQ Control Register (IRQCR) 87
6.3.3 IRQ Status register (IRQSR) 90
6.3.4 Interrupt Priority Registers A to G (IPRA to IPRG) 95
6.4 Interrupt Sources 97
6.4.1 External Interrupts 97
6.4.2 On-Chip Peripheral Module Interrupts 99
6.4.3 User Break Interrupt 99
6.4.4 H-UDI Interrupt 99
6.5 Interrupt Exception Handling Vector Table 100
6.6 Interrupt Operation 102
6.6.1 Interrupt Sequence 102
6.6.2 Stack after Interrupt Exception Handling 104
6.7 Interrupt Response Time 104

Section 7 Bus State Controller (BSC) 107
7.1 Features 107
7.2 Input/Output Pins 110
7.3 Area Overview 111
7.3.1 Area Division 111
7.3.2 Shadow Area 112
7.3.3 Address Map 112
7.3.4 Area 0 Memory Type and Memory Bus Width 114
7.3.5 Data Alignment 114
7.4 Register Descriptions 115
7.4.1 Common Control Register (CMNCR) 116
7.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5B, 6B) 117
7.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 3, 4, 5B, 6B) 122
7.4.4 SDRAM Control Register (SDCR) 138
7.4.5 Refresh Timer Control/Status Register (RTCSR) 139
7.4.6 Refresh Timer Counter (RTCNT) 141
7.4.7 Refresh Time Constant Register (RTCOR) 142
7.5 Operation 143
7.5.1 Endian/Access Size and Data Alignment 143
7.5.2 Normal Space Interface 149
7.5.3 Access Wait Control 154

7.5.4 Extension of Chip Select ($\overline{CSn}$) Assertion Period 156
7.5.5 SDRAM Interface 157
7.5.6 Byte-Selection SRAM Interface 186
7.5.7 PCMCIA Interface 191
7.5.8 Wait between Access Cycles 198
7.5.9 Others 198

Section 8 Clock Pulse Generator (CPG) 201
8.1 Features 201
8.2 Input/Output Pins 204
8.3 Clock Operating Modes 204
8.4 Register Descriptions 206
8.4.1 Frequency Control Register (FRQCR) 206
8.4.2 PHY Clock Frequency Control Register (MCLKCR) 208
8.4.3 Usage Notes 209
8.5 Changing Frequency 210
8.5.1 Changing Multiplication Ratio 210
8.5.2 Changing Division Ratio 211
8.5.3 Changing Clock Operating Mode 211
8.6 Notes on Board Design 213

Section 9 Watchdog Timer (WDT) 215
9.1 Features 215
9.2 Register Descriptions 217
9.2.1 Watchdog Timer Counter (WTCNT) 217
9.2.2 Watchdog Timer Control/Status Register (WTCSR) 217
9.2.3 Notes on Register Access 219
9.3 WDT Operation 220
9.3.1 Canceling Software Standbys 220
9.3.2 Changing Frequency 221
9.3.3 Using Watchdog Timer Mode 221
9.3.4 Using Interval Timer Mode 222
9.4 Usage Notes 222

Section 10 Power-Down Modes 223
10.1 Features 223
10.1.1 Types of Power-Down Modes 223
10.2 Input/Output Pins 224
10.3 Register Descriptions 224
10.3.1 Standby Control Register (STBCR) 225

10.3.2 Standby Control Register 2 (STBCR2) ... 226
10.3.3 Standby Control Register 3 (STBCR3) ... 227
10.3.4 Standby Control Register 4 (STBCR4) ... 228
10.4 Sleep Mode ... 229
10.4.1 Transition to Sleep Mode ... 229
10.4.2 Canceling Sleep Mode ... 229
10.5 Software Standby Mode ... 230
10.5.1 Transition to Software Standby Mode ... 230
10.5.2 Canceling Software Standby Mode ... 231
10.6 Module Standby Mode ... 232
10.6.1 Transition to Module Standby Mode ... 232
10.6.2 Canceling Module Standby Function ... 232

Section 11 Ethernet Controller (EtherC) ... 233
11.1 Features ... 233
11.2 Input/Output Pins ... 235
11.3 Register Description ... 237
11.3.1 EtherC Mode Register (ECMR) ... 238
11.3.2 EtherC Status Register (ECSR) ... 241
11.3.3 EtherC Interrupt Permission Register (ECSIPR) ... 243
11.3.4 PHY Interface Register (PIR) ... 244
11.3.5 MAC Address High Register (MAHR) ... 245
11.3.6 MAC Address Low Register (MALR) ... 245
11.3.7 Receive Frame Length Register (RFLR) ... 246
11.3.8 PHY Status Register (PSR) ... 247
11.3.9 Transmit Retry Over Counter Register (TROCR) ... 247
11.3.10 Delayed Collision Detect Counter Register (CDCR) ... 248
11.3.11 Lost Carrier Counter Register (LCCR) ... 248
11.3.12 Carrier Not Detect Counter Register (CNDCR) ... 248
11.3.13 CRC Error Frame Counter Register (CEFCR) ... 249
11.3.14 Frame Receive Error Counter Register (FRECR) ... 249
11.3.15 Too-Short Frame Receive Counter Register (TSFRCR) ... 249
11.3.16 Too-Long Frame Receive Counter Register (TLFRCR) ... 250
11.3.17 Residual-Bit Frame Counter Register (RFCR) ... 250
11.3.18 Multicast Address Frame Counter Register (MAFCR) ... 250
11.3.19 IPG Register (IPGR) ... 251
11.3.20 Automatic PAUSE Frame Set Register (APR) ... 251
11.3.21 Manual PAUSE Frame Set Register (MPR) ... 252
11.3.22 PAUSE Frame Retransfer Count Set Register (TPAUSER) ... 252
11.4 Operation ... 253

11.4.1 Transmission ... 253
11.4.2 Reception ... 255
11.4.3 MII Frame Timing ... 256
11.4.4 Accessing MII Registers ... 258
11.4.5 Magic Packet Detection ... 261
11.4.6 Operation by IPG Setting ... 262
11.4.7 Flow Control ... 262
11.5 Connection to PHY-LSI ... 263
11.6 Usage Notes ... 264

Section 12 Ethernet Controller Direct Memory Access Controller (E-DMAC) ... 267
12.1 Features ... 267
12.2 Register Descriptions ... 268
12.2.1 E-DMAC Mode Register (EDMR) ... 269
12.2.2 E-DMAC Transmit Request Register (EDTRR) ... 270
12.2.3 E-DMAC Receive Request Register (EDRRR) ... 271
12.2.4 Transmit Descriptor List Address Register (TDLAR) ... 272
12.2.5 Receive Descriptor List Address Register (RDLAR) ... 272
12.2.6 EtherC/E-DMAC Status Register (EESR) ... 273
12.2.7 EtherC/E-DMAC Status Interrupt Permission Register (EESIPR) ... 278
12.2.8 Transmit/Receive Status Copy Enable Register (TRSCER) ... 281
12.2.9 Receive Missed-Frame Counter Register (RMFCR) ... 283
12.2.10 Transmit FIFO Threshold Register (TFTR) ... 284
12.2.11 FIFO Depth Register (FDR) ... 285
12.2.12 Receiving Method Control Register (RMCR) ... 286
12.2.13 E-DMAC Operation Control Register (EDOCR) ... 287
12.2.14 Receiving-Buffer Write Address Register (RBWAR) ... 288
12.2.15 Receiving-Descriptor Fetch Address Register (RDFAR) ... 288
12.2.16 Transmission-Buffer Read Address Register (TBRAR) ... 288
12.2.17 Transmission-Descriptor Fetch Address Register (TDFAR) ... 289
12.2.18 Flow Control FIFO Threshold Register (FCFTR) ... 289
12.2.19 Transmit Interrupt Register (TRIMD) ... 290
12.3 Operation ... 291
12.3.1 Descriptor List and Data Buffers ... 291
12.3.2 Transmission ... 300
12.3.3 Reception ... 302
12.3.4 Multi-Buffer Frame Transmit/Receive Processing ... 304
12.4 Usage Notes ... 306
12.4.1 Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR) ... 306

12.4.2 Usage Notes on SH-Ether Transmit-FIFO Underflow 315

Section 13 Direct Memory Access Controller (DMAC) 325
13.1 Features 325
13.2 Input/Output Pins 327
13.3 Register Descriptions 328
13.3.1 DMA Source Address Registers 0 to 3 (SAR_0 to SAR_3) 329
13.3.2 DMA Destination Address Registers 0 to 3 (DAR_0 to DAR_3) 329
13.3.3 DMA Transfer Count Registers 0 to 3 (DMATCR_0 to DMATCR_3) 329
13.3.4 DMA Channel Control Registers 0 to 3 (CHCR_0 to CHCR_3)..... 330
13.3.5 DMA Operation Register (DMAOR) 336
13.3.6 DMA Extended Resource Selectors 0 and 1 (DMARS0 and DMARS1) 339
13.4 Operation 341
13.4.1 DMA Transfer Flow 341
13.4.2 DMA Transfer Requests 343
13.4.3 Channel Priority 345
13.4.4 DMA Transfer Types 348
13.4.5 Number of Bus Cycle States and DREQ Pin Sampling Timing 357
13.5 Usage Notes 361
13.5.1 Notes on DACK Pin Output 361
13.5.2 Notes On DREQ Sampling When DACK is Divided in External Access 362
13.5.3 Other Notes 365

Section 14 Compare Match Timer (CMT) 367
14.1 Features 367
14.2 Register Descriptions 368
14.2.1 Compare Match Timer Start Register (CMSTR) 368
14.2.2 Compare Match Timer Control/Status Register (CMCSR) 369
14.2.3 Compare Match Counter (CMCNT) 370
14.2.4 Compare Match Constant Register (CMCOR) 370
14.3 Operation 371
14.3.1 Interval Count Operation 371
14.3.2 CMCNT Count Timing 371
14.4 Interrupts 372
14.4.1 Interrupt Sources 372
14.4.2 Timing of Setting Compare Match Flag 372
14.4.3 Timing of Clearing Compare Match Flag 372
14.5 Usage Notes 373
14.5.1 Conflict between Write and Compare-Match Processes of CMCNT 373
14.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT 374

14.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT 375
14.5.4 Conflict between Write Processes to CMCNT with the Counting Stopped and CMCOR 375

Section 15 Serial Communication Interface with FIFO (SCIF) 377
15.1 Overview 377
15.1.1 Features 377
15.2 Pin Configuration 380
15.3 Register Description 381
15.3.1 Receive Shift Register (SCRSR) 382
15.3.2 Receive FIFO Data Register (SCFRDR) 382
15.3.3 Transmit Shift Register (SCTSR) 382
15.3.4 Transmit FIFO Data Register (SCFTDR) 383
15.3.5 Serial Mode Register (SCSMR) 383
15.3.6 Serial Control Register (SCSCR) 386
15.3.7 Serial Status Register (SCFSR) 390
15.3.8 Bit Rate Register (SCBRR) 398
15.3.9 FIFO Control Register (SCFCR) 405
15.3.10 FIFO Data Count Register (SCFDR) 408
15.3.11 Serial Port Register (SCSPTR) 409
15.3.12 Line Status Register (SCLSR) 413
15.4 Operation 414
15.4.1 Overview 414
15.4.2 Operation in Asynchronous Mode 416
15.4.3 Synchronous Mode 427
15.5 SCIF Interrupts 435
15.6 Serial Port Register (SCSPTR) and SCIF Pins 436
15.7 Usage Notes 440

Section 16 Serial I/O with FIFO (SIOF) 445
16.1 Features 445
16.2 Input/Output Pins 447
16.3 Register Descriptions 448
16.3.1 Mode Register (SIMDR) 449
16.3.2 Control Register (SICTR) 452
16.3.3 Transmit Data Register (SITDR) 455
16.3.4 Receive Data Register (SIRDR) 456
16.3.5 Transmit Control Data Register (SITCR) 457
16.3.6 Receive Control Data Register (SIRCR) 458
16.3.7 Status Register (SISTR) 459

16.3.8 Interrupt Enable Register (SIIER) ... 465
16.3.9 FIFO Control Register (SIFCTR) ... 467
16.3.10 Clock Select Register (SISCR) ... 469
16.3.11 Transmit Data Assign Register (SITDAR) ... 470
16.3.12 Receive Data Assign Register (SIRDAR) ... 472
16.3.13 Control Data Assign Register (SICDAR) ... 473
16.3.14 SPI Control Register (SPICR) ... 474
16.4 Operation ... 477
16.4.1 Serial Clocks ... 477
16.4.2 Serial Timing ... 478
16.4.3 Transfer Data Format ... 479
16.4.4 Register Allocation of Transfer Data ... 481
16.4.5 Control Data Interface ... 483
16.4.6 FIFO ... 485
16.4.7 Transmit and Receive Procedures ... 487
16.4.8 Interrupts ... 494
16.4.9 Transmit and Receive Timing ... 496
16.4.10 SPI Mode ... 502

Section 17 Host Interface (HIF) ... 511
17.1 Features ... 511
17.2 Input/Output Pins ... 513
17.3 Parallel Access ... 514
17.3.1 Operation ... 514
17.3.2 Connection Method ... 514
17.4 Register Descriptions ... 515
17.4.1 HIF Index Register (HIFIDX) ... 515
17.4.2 HIF General Status Register (HIFGSR) ... 518
17.4.3 HIF Status/Control Register (HIFSCR) ... 518
17.4.4 HIF Memory Control Register (HIFMCR) ... 521
17.4.5 HIF Internal Interrupt Control Register (HIFIICR) ... 523
17.4.6 HIF External Interrupt Control Register (HIFEICR) ... 523
17.4.7 HIF Address Register (HIFADR) ... 524
17.4.8 HIF Data Register (HIFDATA) ... 525
17.4.9 HIF Boot Control Register (HIFBCR) ... 525
17.4.10 HIFDREQ Trigger Register (HIFDTR) ... 526
17.4.11 HIF Bank Interrupt Control Register (HIFBICR) ... 527
17.5 Memory Map ... 529
17.6 Interface (Basic) ... 530
17.7 Interface (Details) ... 531

17.7.1 HIFIDX Write/HIFGSR Read 531
17.7.2 Reading/Writing of HIF Registers other than HIFIDX and HIFGSR 531
17.7.3 Consecutive Data Writing to HIFRAM by External Device 532
17.7.4 Consecutive Data Reading from HIFRAM to External Device 532
17.8 External DMAC Interface 533
17.9 Alignment Control 539
17.10 Interface When External Device Power is Cut Off 540

Section 18 Pin Function Controller (PFC) 543
18.1 Register Descriptions 553
18.1.1 Port A IO Register H (PAIORH) 554
18.1.2 Port A Control Register H1 and H2 (PACRH1 and PACRH2) 554
18.1.3 Port B IO Register L (PBIORL) 557
18.1.4 Port B Control Register L1 and L2 (PBCRL1 and PBCRL2) 557
18.1.5 Port C IO Register H and L (PCIORH and PCIORL) 561
18.1.6 Port C Control Register H2, L1, and L2 (PCCRH2, PCCRL1, and PCCRL2) 561
18.1.7 Port D IO Register L (PDIORL) 566
18.1.8 Port D Control Register L2 (PDCRL2) 567
18.1.9 Port E IO Register H and L (PEIORH and PEIORL) 569
18.1.10 Port E Control Register H1, H2, L1, and L2 (PECRH1, PECRH2, PECRL1, and PECRL2) 569
18.2 Notes on Usage 577
18.2.1 Restriction in Using 577
18.2.2 Details of Restriction 577

Section 19 I/O Ports 579
19.1 Port A 579
19.1.1 Register Description 579
19.1.2 Port A Data Register H (PADRH) 579
19.2 Port B 581
19.2.1 Register Description 581
19.2.2 Port B Data Register L (PBDRL) 581
19.3 Port C 583
19.3.1 Register Description 584
19.3.2 Port C Data Registers H and L (PCDRH and PCDRL) 584
19.4 Port D 586
19.4.1 Register Description 586
19.4.2 Port D Data Register L (PDDRL) 586
19.5 Port E 588

19.5.1 Register Description 589
19.5.2 Port E Data Registers H and L (PEDRH and PEDRL) 589
19.6 Usage Notes 591

Section 20 User Break Controller (UBC) 593
20.1 Features 593
20.2 Register Descriptions 595
20.2.1 Break Address Register A (BARA) 595
20.2.2 Break Address Mask Register A (BAMRA) 596
20.2.3 Break Bus Cycle Register A (BBRA) 596
20.2.4 Break Address Register B (BARB) 597
20.2.5 Break Address Mask Register B (BAMRB) 598
20.2.6 Break Data Register B (BDRB) 598
20.2.7 Break Data Mask Register B (BDMRB) 599
20.2.8 Break Bus Cycle Register B (BBRB) 599
20.2.9 Break Control Register (BRCR) 601
20.2.10 Execution Times Break Register (BETR) 604
20.2.11 Branch Source Register (BRSR) 604
20.2.12 Branch Destination Register (BRDR) 605
20.3 Operation 606
20.3.1 Flow of User Break Operation 606
20.3.2 Break on Instruction Fetch Cycle 607
20.3.3 Break on Data Access Cycle 607
20.3.4 Sequential Break 608
20.3.5 Value of Saved Program Counter (PC) 608
20.3.6 PC Trace 609
20.3.7 Usage Examples 610
20.3.8 Notes 614

Section 21 User Debugging Interface (H-UDI) 615
21.1 Features 615
21.2 Input/Output Pins 616
21.3 Register Descriptions 617
21.3.1 Bypass Register (SDBPR) 617
21.3.2 Instruction Register (SDIR) 617
21.3.3 Boundary Scan Register (SDBSR) 618
21.3.4 ID Register (SDID) 625
21.4 Operation 626
21.4.1 TAP Controller 626
21.4.2 Reset Configuration 627

21.4.3 TDO Output Timing 627
21.4.4 H-UDI Reset 628
21.4.5 H-UDI Interrupt 628
21.5 Boundary Scan 629
21.5.1 Supported Instructions 629
21.5.2 Points for Attention 630
21.6 Usage Notes 630

Section 22 Ethernet Physical Layer Transceiver (PHY) 631
22.1 Features 631
22.2 Pin Configuration 633
22.3 Top Level Functional Architecture 634
22.4 PHY Management Control 635
22.4.1 Serial Management Interface (SMI) 635
22.4.2 SMI Register Mapping 642
22.5 100Base-TX Transmit 648
22.6 100Base-TX Receive 651
22.7 10Base-T Transmit 654
22.8 10Base-T Receive 656
22.9 MAC Interface 657
22.10 Miscellaneous Functions 661
22.11 Internal I/O Signals 665
22.12 Signals Relevant to PHY-IF 667
22.13 Usage Notes 668
22.14 Guidelines for Layout 678
22.14.1 General Guidelines 678
22.14.2 Guidelines for Layout 679

Section 23 PHY Interface (PHY-IF) 683
23.1 Features 683
23.2 Register Descriptions 685
23.2.1 PHY-IF Control Register (PHYIFCR) 685
23.2.2 PHY-IF SMI Register 2 (PHYIFSMIR2) 686
23.2.3 PHY-IF SMI Register 3 (PHYIFSMIR3) 687
23.2.4 PHY-IF Address Register (PHYIFADDRR) 687
23.2.5 PHY-IF status Register (PHYIFSR) 688
23.3 PHY-IF Operation 689
23.3.1 The Procedures of Setting Up the On-Chip PHY 689
23.3.2 The Procedures of Set Up the External PHY LSI 690

Section 24 List of Registers 691
24.1 Register Addresses (Address Order) 692
24.2 Register Bits 700
24.3 Register States in Each Processing State 722

Section 25 Electrical Characteristics 729
25.1 Absolute Maximum Ratings 729
25.2 Power-On and Power-Off Order 730
25.3 DC Characteristics 732
25.4 AC Characteristics 734
25.4.1 Clock Timing 735
25.4.2 Control Signal Timing 739
25.4.3 AC Bus Timing 741
25.4.4 Basic Timing 743
25.4.5 Synchronous DRAM Timing 749
25.4.6 PCMCIA Timing 766
25.4.7 DMAC Signal Timing 770
25.4.8 SCIF Timing 771
25.4.9 SIOF Module Signal Timing 772
25.4.10 Port Timing 776
25.4.11 HIF Timing 777
25.4.12 EtherC Timing 780
25.4.13 H-UDI Related Pin Timing 783
25.4.14 AC Characteristic Test Conditions 785
25.5 Physical Layer Ttransceiver (PHY) Characteristics (Reference Values) 786

Appendix 787
A. Port States in Each Pin State 787
B. Product Code Lineup 792
C. Package Dimensions 793

Main Revisions and Additions in this Edition 795

Index 809

Figures

Section 1 Overview
Figure 1.1 Block Diagram 7
Figure 1.2 Pin Assignments 8

Section 2 CPU
Figure 2.1 CPU Internal Register Configuration 24
Figure 2.2 Register Data Format 28
Figure 2.3 Memory Data Format 28
Figure 2.4 CPU State Transition 51

Section 3 Cache
Figure 3.1 Cache Structure 53
Figure 3.2 Cache Search Scheme 58
Figure 3.3 Write-Back Buffer Configuration 59
Figure 3.4 Specifying Address and Data for Memory-Mapped Cache Access 62

Section 6 Interrupt Controller (INTC)
Figure 6.1 INTC Block Diagram 84
Figure 6.2 Block Diagram of IRQ7 to IRQ0 Interrupts Control 98
Figure 6.3 Interrupt Sequence Flowchart 103
Figure 6.4 Stack after Interrupt Exception Handling 104

Section 7 Bus State Controller (BSC)
Figure 7.1 Block Diagram of BSC 109
Figure 7.2 Address Space 112
Figure 7.3 Normal Space Basic Access Timing (No-Wait Access) 149
Figure 7.4 Consecutive Access to Normal Space (1): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 0 (Access Wait = 0, Cycle Wait = 0) 150
Figure 7.5 Consecutive Access to Normal Space (2): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 1 (Access Wait = 0, Cycle Wait = 0) 151
Figure 7.6 Example of 32-Bit Data-Width SRAM Connection 152
Figure 7.7 Example of 16-Bit Data-Width SRAM Connection 153
Figure 7.8 Example of 8-Bit Data-Width SRAM Connection 153
Figure 7.9 Wait Timing for Normal Space Access (Software Wait Only) 154
Figure 7.10 Wait Cycle Timing for Normal Space Access (Wait cycle Insertion using $\overline{\text{WAIT}}$) 155
Figure 7.11 Example of Timing when $\overline{\text{CSn}}$ Assertion Period is Extended 156
Figure 7.12 Example of 32-Bit Data-Width SDRAM Connection 158
Figure 7.13 Example of 16-Bit Data-Width SDRAM Connection 159

Figure 7.14 Burst Read Basic Timing (Auto Precharge) 170
Figure 7.15 Burst Read Wait Specification Timing (Auto Precharge) 171
Figure 7.16 Basic Timing for Single Read (Auto Precharge) 172
Figure 7.17 Basic Timing for Burst Write (Auto Precharge) 173
Figure 7.18 Basic Timing for Single Write (Auto-Precharge) 174
Figure 7.19 Burst Read Timing (No Auto Precharge) 176
Figure 7.20 Burst Read Timing (Bank Active, Same Row Address) 177
Figure 7.21 Burst Read Timing (Bank Active, Different Row Addresses) 178
Figure 7.22 Single Write Timing (No Auto Precharge) 179
Figure 7.23 Single Write Timing (Bank Active, Same Row Address) 180
Figure 7.24 Single Write Timing (Bank Active, Different Row Addresses) 181
Figure 7.25 Auto-Refreshing Timing 182
Figure 7.26 Self-Refreshing Timing 184
Figure 7.27 Write Timing for SDRAM Mode Register (Based on JEDEC) 186
Figure 7.28 Basic Access Timing for Byte-Selection SRAM (BAS = 0) 187
Figure 7.29 Basic Access Timing for Byte-Selection SRAM (BAS = 1) 188
Figure 7.30 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only) 189
Figure 7.31 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM 190
Figure 7.32 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM 190
Figure 7.33 Example of PCMCIA Interface Connection 192
Figure 7.34 Basic Access Timing for PCMCIA Memory Card Interface 193
Figure 7.35 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1) 194
Figure 7.36 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B'10, CS6BWCR.SA[1:0] = B'10) 195
Figure 7.37 Basic Timing for PCMCIA I/O Card Interface 196
Figure 7.38 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1) 197
Figure 7.39 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3) 197

Section 8 Clock Pulse Generator (CPG)

Figure 8.1 Block Diagram of CPG 202
Figure 8.2 Note on Using a Crystal Resonator 213
Figure 8.3 Note on Using a PLL Oscillator Circuit 214

Section 9 Watchdog Timer (WDT)

Figure 9.1 Block Diagram of WDT 216
Figure 9.2 Writing to WTCNT and WTCSR 220

Section 10 Power-Down Modes

Figure 10.1 Canceling Standby Mode with STBY Bit in STBCR 232

Section 11 Ethernet Controller (EtherC)

Figure 11.1 Configuration of EtherC 234
Figure 11.2 EtherC Transmitter State Transitions 254
Figure 11.3 EtherC Receiver State Transmissions 255
Figure 11.4 (1) MII Frame Transmit Timing (Normal Transmission) 256
Figure 11.4 (2) MII Frame Transmit Timing (Collision) 256
Figure 11.4 (3) MII Frame Transmit Timing (Transmit Error) 257
Figure 11.4 (4) MII Frame Receive Timing (Normal Reception) 257
Figure 11.4 (5) MII Frame Receive Timing (Reception Error (1)) 257
Figure 11.4 (6) MII Fame Receive Timing (Reception Error (2)) 257
Figure 11.5 MII Management Frame Format 258
Figure 11.6 (1) 1-Bit Data Write Flowchart 259
Figure 11.6 (2) Bus Release Flowchart (TA in Read in Figure 11.5) 260
Figure 11.6 (3) 1-Bit Data Read Flowchart 260
Figure 11.6 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 11.5) 261
Figure 11.7 Changing IPG and Transmission Efficiency 262
Figure 11.8 Example of Connection to DP83846AVHG 263

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

Figure 12.1 Configuration of E-DMAC, and Descriptors and Buffers 267
Figure 12.2 Relationship between Transmit Descriptor and Transmit Buffer 292
Figure 12.3 Relationship between Receive Descriptor and Receive Buffer 296
Figure 12.4 Sample Transmission Flowchart 301
Figure 12.5 Sample Reception Flowchart 303
Figure 12.6 E-DMAC Operation after Transmit Error 304
Figure 12.7 E-DMAC Operation after Receive Error 305
Figure 12.8 Timing of the Case where Setting of the Interrupt Source Bit in EESR by the E-DMAC Fails 306
Figure 12.9 Countermeasure by Monitoring the Transmit Descriptor in Processing of Interrupts Other than the Frame Transmit Complete (TC) Interrupt 312
Figure 12.10 Method of Adding Timeout Processing 314
Figure 12.11 Operation when E-DMAC Stops and the Transmit FIFO 316
Figure 12.12 Processing Transmission without Handling of the TC Interrupt 319
Figure 12.13 Countermeasure for the Case with TC Interrupt-Driven Software: Addition of Timeout Processing within the Limit Imposed by the Maximum Specified Time 322

Section 13 Direct Memory Access Controller (DMAC)

Figure 13.1 Block Diagram of DMAC 326
Figure 13.2 DMA Transfer Flowchart 342
Figure 13.3 Round-Robin Mode 346

Figure 13.4 Changes in Channel Priority in Round-Robin Mode 347
Figure 13.5 Data Flow of Dual Address Mode 349
Figure 13.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory) 350
Figure 13.7 Data Flow in Single Address Mode 351
Figure 13.8 Example of DMA Transfer Timing in Single Address Mode 352
Figure 13.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection) 353
Figure 13.10 Example of DMA Transfer in Cycle Steal Intermittent Mode (Dual Address, DREQ Low Level Detection) 354
Figure 13.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection) 354
Figure 13.12 Bus State when Multiple Channels are Operating 356
Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection 357
Figure 13.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection 357
Figure 13.15 Example of DREQ Input Detection in Burst Mode Edge Detection 358
Figure 13.16 Example of DREQ Input Detection in Burst Mode Level Detection 358
Figure 13.17 Example of DMA Transfer End in Cycle Steal Mode Level Detection 359
Figure 13.18 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device) 360
Figure 13.19 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 4 by Idle Cycles 362
Figure 13.20 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 2 by Idle Cycles 363
Figure 13.21 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 4 by Idle Cycles 363
Figure 13.22 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 2 by Idle Cycles 364

Section 14 Compare Match Timer (CMT)

Figure 14.1 Block Diagram of Compare Match Timer 367
Figure 14.2 Counter Operation 371
Figure 14.3 Count Timing 371
Figure 14.4 Timing of CMF Setting 372
Figure 14.5 Conflict between Write and Compare-Match Processes of CMCNT 373
Figure 14.6 Conflict between Word-Write and Count-Up Processes of CMCNT 374
Figure 14.7 Conflict between Byte-Write and Count-Up Processes of CMCNT 375

Section 15 Serial Communication Interface with FIFO (SCIF)

Figure 15.1 Block Diagram of SCIF 379
Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits) 416
Figure 15.3 Sample Flowchart for SCIF Initialization 419
Figure 15.4 Sample Flowchart for Transmitting Serial Data 420
Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, One Stop Bit) 422
Figure 15.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$) 422
Figure 15.7 Sample Flowchart for Receiving Serial Data (1) 423
Figure 15.8 Sample Flowchart for Receiving Serial Data (2) 424
Figure 15.9 Example of SCIF Receive Operation (8-Bit Data, Parity, One Stop Bit) 426
Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$) 426
Figure 15.11 Data Format in Synchronous Communication 427
Figure 15.12 Sample Flowchart for SCIF Initialization 429
Figure 15.13 Sample Flowchart for Transmitting Serial Data 430
Figure 15.14 Example of SCIF Transmit Operation 431
Figure 15.15 Sample Flowchart for Receiving Serial Data (1) 432
Figure 15.16 Sample Flowchart for Receiving Serial Data (2) 432
Figure 15.17 Example of SCIF Receive Operation 433
Figure 15.18 Sample Flowchart for Transmitting/Receiving Serial Data 434
Figure 15.19 RTSIO Bit, RTSDT Bit, and $\overline{\text{RTS}}$ Pin 436
Figure 15.20 CTSIO Bit, CTSDT Bit, and $\overline{\text{CTS}}$ Pin 437
Figure 15.21 SCKIO Bit, SCKDT Bit, and SCK Pin 438
Figure 15.22 SPBIO Bit, SPBDT Bit, and TxD Pin 438
Figure 15.23 SPBDT Bit and RxD Pin 439
Figure 15.24 Receive Data Sampling Timing in Asynchronous Mode 441

Section 16 Serial I/O with FIFO (SIOF)

Figure 16.1 Block Diagram of SIOF 446
Figure 16.2 Serial Clock Supply 477
Figure 16.3 Serial Data Synchronization Timing 478
Figure 16.4 SIOF Transmit/Receive Timing 479
Figure 16.5 Transmit/Receive Data Bit Alignment 481
Figure 16.6 Control Data Bit Alignment 482
Figure 16.7 Control Data Interface (Slot Position) 483
Figure 16.8 Control Data Interface (Secondary FS) 484
Figure 16.9 (1) Transmission/Reception Operation in Master Mode (Example of Reception and Full-Duplex Transmission by the CPU with TDMAE=0) 488
Figure 16.9 (2) Transmission Operation in Master Mode (Example of Half-Duplex Transmission by the CPU with TDMAE=0) 489

Figure 16.10 Example of Receive Operation in Master Mode 490
Figure 16.11 Example of Transmit Operation in Slave Mode 491
Figure 16.12 Example of Receive Operation in Slave Mode 492
Figure 16.13 Transmit and Receive Timing (8-Bit Monaural Data (1)) 496
Figure 16.14 Transmit and Receive Timing (8-Bit Monaural Data (2)) 496
Figure 16.15 Transmit and Receive Timing (16-Bit Monaural Data (1)) 497
Figure 16.16 Transmit and Receive Timing (16-Bit Stereo Data (1)) 497
Figure 16.17 Transmit and Receive Timing (16-Bit Stereo Data (2)) 498
Figure 16.18 Transmit and Receive Timing (16-Bit Stereo Data (3)) 498
Figure 16.19 Transmit and Receive Timing (16-Bit Stereo Data (4)) 499
Figure 16.20 Transmit and Receive Timing (16-Bit Stereo Data) 499
Figure 16.21 Example of Configuration in SPI Mode 502
Figure 16.22 SPI Data/Clock Timing 1 (CPHA = 0) 504
Figure 16.23 SPI Data/Clock Timing 2 (CPHA = 1) 505
Figure 16.24 SPI Transmission/Reception Operation (Example of Full-Duplex Transmission/Reception by the CPU with TDMAE = 0) 506
Figure 16.25 SPI Transmission Operation (Example of Half-Duplex Transmission by the CPU with TDMAE = 0) 507
Figure 16.26 SPI Transmission Operation (Example of Half-Duplex Transmission by DMA with TDMAE = 1) 508
Figure 16.27 SPI Reception Operation (Example of Half-Duplex Reception by DMA with RDMAE = 1) 509

Section 17 Host Interface (HIF)

Figure 17.1 Block Diagram of HIF 512
Figure 17.2 HIF Connection Example 514
Figure 17.3 Basic Timing for HIF Interface 530
Figure 17.4 HIFIDX Write and HIFGSR Read 531
Figure 17.5 HIF Register Settings 531
Figure 17.6 Consecutive Data Writing to HIFRAM 532
Figure 17.7 Consecutive Data Reading from HIFRAM 533
Figure 17.8 HIFDREQ Timing (When DMD = 0 and DPOL = 0) 534
Figure 17.9 HIFDREQ Timing (When DMD = 0 and DPOL = 1) 534
Figure 17.10 HIFDREQ Timing (When DMD = 1 and DPOL = 0) 535
Figure 17.11 HIFDREQ Timing (When DMD = 1 and DPOL = 1) 535
Figure 17.12 Image of High-Impedance Control of HIF Pins by HIFEBL Pin 540

Section 19 I/O Ports

Figure 19.1 Port A 579
Figure 19.2 Port B 581
Figure 19.3 Port C 583

Figure 19.4 Port D 586
Figure 19.5 Port E 588

Section 20 User Break Controller (UBC)
Figure 20.1 Block Diagram of UBC 594

Section 21 User Debugging Interface (H-UDI)
Figure 21.1 Block Diagram of H-UDI 615
Figure 21.2 TAP Controller State Transitions 626
Figure 21.3 H-UDI Data Transfer Timing 628
Figure 21.4 H-UDI Reset 628

Section 22 Ethernet Physical Layer Transceiver (PHY)
Figure 22.1 The Block Diagram around PHY Module 632
Figure 22.2 Architectural Overview 634
Figure 22.3 How to Derive MDIO Signal from Core Signals 635
Figure 22.4 MDIO Timing and Frame Structure (READ Cycle) 636
Figure 22.5 MDIO Timing and Frame Structure (WRITE Cycle) 636
Figure 22.6 100Base-TX Data Path 648
Figure 22.7 Receive Data Path 651
Figure 22.8 Relationship between Received Data and Some MII Signals 653
Figure 22.9 Manchester Encoded Output 655
Figure 22.10 Example of Connection with a Pulse Transformer (RJ45) 679

Section 23 PHY Interface (PHY-IF)
Figure 23.1 Block Diagram of PHY-IF 684

Section 25 Electrical Characteristics
Figure 25.1 External Clock Input Timing 736
Figure 25.2 CKIO Clock Output Timing and CK_PHY Clock Input Timing 736
Figure 25.3 Oscillation Settling Timing after Power-On 737
Figure 25.4 Oscillation Settling Timing after Standby Mode (By Reset) 737
Figure 25.5 Oscillation Settling Timing after Standby Mode (By NMI or IRQ) 737
Figure 25.6 PLL Synchronize Settling Timing By Reset or NMI 738
Figure 25.7 Reset Input Timing 739
Figure 25.8 Interrupt Input Timing 740
Figure 25.9 Pin Drive Timing in Standby Mode 740
Figure 25.10 Basic Bus Timing: No Wait Cycle 743
Figure 25.11 Basic Bus Timing: One Software Wait Cycle 744
Figure 25.12 Basic Bus Timing: One External Wait Cycle 745
Figure 25.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle 746

Figure 25.14 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 0 (UB-/LB-Controlled Write Cycle) 747
Figure 25.15 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 1 (WE-Controlled Write Cycle) 748
Figure 25.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle) 749
Figure 25.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle) 750
Figure 25.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle) 751
Figure 25.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle) 752
Figure 25.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle) 753
Figure 25.21 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle) 754
Figure 25.22 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle) 755
Figure 25.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle) 756
Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0 Cycle) 757
Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: READ Command, Same Row Address, CAS Latency = 2, WTRCD = 0 Cycle) 758
Figure 25.26 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency = 2, WTRCD = 0 Cycle) 759
Figure 25.27 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle) 760
Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle) 761
Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle) 762
Figure 25.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles) 763
Figure 25.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle) 764

Figure 25.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle) 765
Figure 25.33 PCMCIA Memory Card Interface Bus Timing 766
Figure 25.34 PCMCIA Memory Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle) 767
Figure 25.35 PCMCIA I/O Card Interface Bus Timing 768
Figure 25.36 PCMCIA I/O Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle) 769
Figure 25.37 DREQ Input Timing 770
Figure 25.38 TENDn, DACKn Output Timing 770
Figure 25.39 SCK Input Clock Timing 771
Figure 25.40 SCI Input/Output Timing in Clocked Synchronous Mode 772
Figure 25.41 SIOMCLK Input Timing 773
Figure 25.42 SIOF Transmit/Receive Timing (Master Mode 1/Falling Edge Sampling) 773
Figure 25.43 SIOF Transmit/Receive Timing (Master Mode 1/Rising Edge Sampling) 774
Figure 25.44 SIOF Transmit/Receive Timing (Master Mode 2/Falling Edge Sampling) 774
Figure 25.45 SIOF Transmit/Receive Timing (Master Mode 2/Rising Edge Sampling) 775
Figure 25.46 SIOF Transmit/Receive Timing (Slave Mode 1/ Slave Mode 2) 775
Figure 25.47 I/O Port Timing 776
Figure 25.48 HIF Access Timing 778
Figure 25.49 $\overline{\text{HIFINT}}$ and HIFDREQ Timing 778
Figure 25.50 HIFRDY and HIF Pin Enable/Disable Timing 779
Figure 25.51 MII Transmission Timing (Normal Operation) 781
Figure 25.52 MII Transmission Timing (Collision Occurred) 781
Figure 25.53 MII Reception Timing (Normal Operation) 782
Figure 25.54 MII Reception Timing (Error Occurred) 782
Figure 25.55 MDIO Input Timing 782
Figure 25.56 MDIO Output Timing 782
Figure 25.57 WOL Output Timing 783
Figure 25.58 EXOUT Output Timing 783
Figure 25.59 TCK Input Timing 784
Figure 25.60 TCK Input Timing in Reset Hold State 784
Figure 25.61 H-UDI Data Transmission Timing 784
Figure 25.62 Output Load Circuit 785

Appendix

Figure C.1 Package Dimensions (BP-176) 793

Tables

Section 1 Overview
Table 1.1 Features of SH7619 2
Table 1.2 Pin Functions 9
Table 1.3 Pin Features 17

Section 2 CPU
Table 2.1 Initial Values of Registers 27
Table 2.2 Word Data Sign Extension 29
Table 2.3 Delayed Branch Instructions 30
Table 2.4 T Bit 30
Table 2.5 Access to Immediate Data 31
Table 2.6 Access to Absolute Address 31
Table 2.7 Access with Displacement 32
Table 2.8 Addressing Modes and Effective Addresses 32
Table 2.9 Instruction Formats 36
Table 2.10 Instruction Types 39

Section 3 Cache
Table 3.1 LRU and Way to be Replaced 54
Table 3.2 Correspondence between Divided Areas and Cache 55

Section 5 Exception Handling
Table 5.1 Types of Exceptions and Priority 67
Table 5.2 Timing for Exception Detection and Start of Exception Handling 68
Table 5.3 Vector Numbers and Vector Table Address Offsets 69
Table 5.4 Calculating Exception Handling Vector Table Addresses 70
Table 5.5 Reset Status 71
Table 5.6 Bus Cycles and Address Errors 73
Table 5.7 Interrupt Sources 74
Table 5.8 Interrupt Priority 75
Table 5.9 Types of Exceptions Triggered by Instructions 76
Table 5.10 Delay Slot Instructions, Interrupt Disabled Instructions, and Exceptions 78
Table 5.11 Stack Status after Exception Handling Ends 79

Section 6 Interrupt Controller (INTC)
Table 6.1 Pin Configuration 85
Table 6.2 Interrupt Exception Handling Vectors and Priorities 100
Table 6.3 Interrupt Response Time 105

Section 7 Bus State Controller (BSC)

Table 7.1 Pin Configuration 110
Table 7.2 Address Map 1 (CMNCR.MAP = 0) 113
Table 7.3 Address Map 2 (CMNCR.MAP = 1) 113
Table 7.4 Correspondence between External Pin (MD3), Memory Type, and Bus Width for CS0 114
Table 7.5 Correspondence between External Pin (MD5) and Endians 114
Table 7.6 32-Bit External Device/Big Endian Access and Data Alignment 143
Table 7.7 16-Bit External Device/Big Endian Access and Data Alignment 144
Table 7.8 8-Bit External Device/Big Endian Access and Data Alignment 145
Table 7.9 32-Bit External Device/Big Endian Access and Data Alignment 146
Table 7.10 16-Bit External Device/Little Endian Access and Data Alignment 147
Table 7.11 8-Bit External Device/Little Endian Access and Data Alignment 148
Table 7.12 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (1) 160
Table 7.13 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (2) 161
Table 7.14 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (3) 163
Table 7.15 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (4) 164
Table 7.16 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (5) 165
Table 7.17 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (6) 167
Table 7.18 Relationship between Access Size and Number of Bursts 169
Table 7.19 Access Address for SDRAM Mode Register Write 185

Section 8 Clock Pulse Generator (CPG)

Table 8.1 Pin Configuration 204
Table 8.2 Mode Control Pins and Clock Operating Modes 204
Table 8.3 Possible Combination of Clock Modes and FRQCR Values 205

Section 10 Power-Down Modes

Table 10.1 States of Power-Down Modes 223
Table 10.2 Pin Configuration 224
Table 10.3 Register States in Software Standby Mode 230

Section 11 Ethernet Controller (EtherC)

Table 11.1 Pin Configuration 235

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

Table 12.1 EESR Bits for which This Problem can Occur and Reflection of Interrupt Sources in the Descriptor 307
Table 12.2 Reference Values for Maximum Specified Time 323

Section 13 Direct Memory Access Controller (DMAC)

Table 13.1 Pin Configuration 327
Table 13.2 Transfer Request Sources 340
Table 13.3 Selecting External Request Modes with RS Bits 343
Table 13.4 Selecting External Request Detection with DL, DS Bits 344
Table 13.5 Selecting External Request Detection with DO Bit 344
Table 13.6 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits 345
Table 13.7 Supported DMA Transfers 348
Table 13.8 Relationship between Request Modes and Bus Modes by DMA Transfer Category 355

Section 15 Serial Communication Interface with FIFO (SCIF)

Table 15.1 SCIF Pins 380
Table 15.2 SCSMR Settings 399
Table 15.3 Bit Rates and SCBRR Settings in Asynchronous Mode 399
Table 15.4 Bit Rates and SCBRR Settings in Synchronous Mode 402
Table 15.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode) 403
Table 15.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode) 404
Table 15.7 Maximum Bit Rates with External Clock Input (Synchronous Mode) 404
Table 15.8 SCSMR Settings and SCIF Communication Formats 415
Table 15.9 SCSMR and SCSCR Settings and SCIF Clock Source Selection 415
Table 15.10 Serial Communication Formats (Asynchronous Mode) 417
Table 15.11 SCIF Interrupt Sources 435

Section 16 Serial I/O with FIFO (SIOF)

Table 16.1 Pin Configuration 447
Table 16.2 Operation in Each Transfer Mode 451
Table 16.3 SIOF Serial Clock Frequency 477
Table 16.4 Serial Transfer Modes 479
Table 16.5 Frame Length 480
Table 16.6 Audio Mode Specification for Transmit Data 482
Table 16.7 Audio Mode Specification for Receive Data 482
Table 16.8 Setting Number of Channels in Control Data 483
Table 16.9 Conditions to Issue Transmit Request 485
Table 16.10 Conditions to Issue Receive Request 485

Table 16.11 Transmit and Receive Reset 493
Table 16.12 SIOF Interrupt Sources 494
Table 16.13 States of Transmit and Receive Operations in SPI Mode 503

Section 17 Host Interface (HIF)
Table 17.1 Pin Configuration 513
Table 17.2 HIF Operations 514
Table 17.3 Memory Map 529
Table 17.4 Consecutive Write Procedure to HIFRAM by External DMAC 536
Table 17.5 Consecutive Read Procedure from HIFRAM by External DMAC 537
Table 17.6 HIFDATA Register Alignment for Access by an External Device 539
Table 17.7 HIF Registers (other than HIFDATA) Alignment for Access by an External Device 539
Table 17.8 Input/Output Control for HIF Pins 541

Section 18 Pin Function Controller (PFC)
Table 18.1 List of Multiplexed Pins (Port A) 543
Table 18.2 List of Multiplexed Pins (Port B) 543
Table 18.3 List of Multiplexed Pins (Port C) 545
Table 18.4 List of Multiplexed Pins (Port D) 546
Table 18.5 List of Multiplexed Pins (Port E) 546
Table 18.6 Pin Functions in Each Operating Mode 548
Table 18.7 Pins restricted in using in SH7619 577

Section 19 I/O Ports
Table 19.1 Port A Data Register H (PADRH) Read/Write Operation 580
Table 19.2 Port B Data Register L (PBDRL) Read/Write Operation 582
Table 19.3 Port C Data Registers H and L (PCDRH and PCDRL) Read/Write Operation 585
Table 19.4 Port D Data Register L (PDDRL) Read/Write Operation 587
Table 19.5 Port E Data Registers H, L (PEDRH, PEDRL) Read/Write Operation 590

Section 20 User Break Controller (UBC)
Table 20.1 Data Access Cycle Addresses and Operand Size Comparison Conditions 607

Section 21 User Debugging Interface (H-UDI)
Table 21.1 Pin Configuration 616
Table 21.2 H-UDI Commands 618
Table 21.3 External pins and Boundary Scan Register Bits 619
Table 21.4 Reset Configuration 627

Section 22 Ethernet Physical Layer Transceiver (PHY)
Table 22.1 Pin Configuration 633
Table 22.2 4B/5B Code Table 649

Section 25 Electrical Characteristics

Table 25.1 Absolute Maximum Ratings 729
Table 25.2 Recommended Timing at Power-On 730
Table 25.3 Recommended Timing in Power-Off 731
Table 25.4 DC Characteristics (1) 732
Table 25.4 DC Characteristics (2) 733
Table 25.5 Permissible Output Currents 734
Table 25.6 Maximum Operating Frequency 734
Table 25.7 Clock Timing 735
Table 25.8 Control Signal Timing 739
Table 25.9 Bus Timing 741
Table 25.10 DMAC Signal Timing 770
Table 25.11 SCIF Timing 771
Table 25.12 SCIF Timing 772
Table 25.13 Port Timing 776
Table 25.14 HIF Timing 777
Table 25.15 EtherC Timing 780
Table 25.16 H-UDI Related Pin Timing 783
Table 25.17 PHY Characteristics 786

Appendix

Table A.1 Port States in Each Pin State 787

Section 1 Overview

This LSI is a CMOS single-chip microcontroller that integrates a Renesas Technology original RISC (Reduced Instruction Set Computer) CPU core with peripheral functions required for an Ethernet system.

The CPU of this LSI has a RISC (Reduced Instruction Set Computer) type instruction set. The CPU basically operates at a rate of one instruction per cycle, offering a great improvement in instruction execution speed. In addition, the 32-bit internal architecture provides improved data processing power. With this CPU, it has become possible to assemble low-cost and high-performance/high-functionality systems even for applications such as realtime control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

This LSI is equipped with an Ethernet controller that includes a media access controller (MAC) conforming to the IEEE802.3u standard and a physical layer transceiver (PHY), enabling 10/100 Mbps LAN connection. As the equipped Ethernet controller also includes a media independent interface (MII) standard unit, a PHY LSI can be externally connected.

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

The external memory access support function of this LSI enables direct connection to various types of memory, such as standard memory, SDRAM, and PCMCIA. This greatly reduces system cost.

1.1 Features

The features of this LSI are listed in table 1.1.

Table 1.1 Features of SH7619

Items	Specification
CPU	• Central processing unit with an internal 32-bit RISC (Reduced Instruction Set Computer) architecture • Instruction length: 16-bit fixed length for improved code efficiency • Load-store architecture (basic operations are executed between registers) • Sixteen 32-bit general registers • Five-stage pipeline • On-chip multiplier: Multiplication operations (32 bits × 32 bits → 64 bits) executed in two to five cycles • C language-oriented 62 basic instructions Note: Some specifications on the slot illegal instruction differ from the conventional SH2 core. For details, see section 5.8, Usage Notes in section 5, Exception Handling.
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
U memory	• 16 kbytes
Cache memory	• Unified cache, mixture of instructions and data • 4-way set associative type • Selection of write-back or write-through mode • 16 kbytes

Items	Specification
Bus state controller (BSC)	• Address space is divided into five areas: three areas 0, 3, and 4; each a maximum of 64 Mbytes, and two areas 5B and 6B; each a maximum of 32 Mbytes (address map 1 mode). • Address space is divided into five areas, 0, 3, 4, 5, and 6; each a maximum of 64 Mbytes (address map 2 mode). • 32-bit external bus (max.) • The following features are settable for each area. — Bus size (8, 16, or 32 bits) (Area 0 does not support the bus size of 32 bits.) — Number of access wait cycles — Setting of idle wait cycles — Specifying the memory to be connected to each area enables direct connection to SRAM, SDRAM, and PCMCIA. — Outputs chip select signals (CS0, CS3, CS4, CS5B, and CS6B) for corresponding area • SDRAM refresh function — Supports auto-refresh and self-refresh modes • SDRAM burst access function • PCMCIA access function — Conforms to the JEIDA Ver. 4.2 standard, two slots • Selection of big or little endian mode (The mode of all the areas is switched collectively by a mode pin.)
Direct memory access controller (DMAC)	• Four channels; external request available for two of them • Burst mode and cycle steal mode • Outputs a transfer end signal of the channel handling an external request • Intermittent mode available (16 and 64 cycles supported)
Interrupt controller (INTC)	• Supports nine external interrupt pins (NMI, IRQ7 to IRQ0) • On-chip peripheral interrupt: Priority level is independently selected for each module • Vector address: Specified vector address for each interrupt source
User debugging interface (H-UDI)	• Supports the JTAG interface emulator • JTAG standard pins arranged

Items	Specification
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input or crystal resonator • Three types of clocks generated: — CPU clock: 125 MHz (max.) — Bus clock: 62.5 MHz (max.) — Peripheral clock: 31.25 MHz (max.) • Supports power-down modes: — Sleep mode — Software standby mode • Selection of four types of clock modes (PLL2 ×2/×4 and clock/crystal resonator are selectable)
Ethernet controller (EtherC)	• MAC (Media Access Control) function — Data frame assembly/disassembly (frame format conforming to IEEE802.3) — CSMA/CD link management (collision prevention and collision processing) — CRC processing — 512 bytes each for transmit/receive FIFO — Full-duplex transmit/receive support — Short frame/long frame detectable • Conforms to the MII (Media Independent Interface) standard — Conversion from 8-bit stream data in MAC layer to MII nibble (4-bit) stream — Station management (STA function) — 18 TTL-level signals — 10/100 Mbps transfer rate adjustable • Magic Packet™* (WOL (Wake-On-LAN) output)
Ethernet controller DMAC (EDMAC)	• CPU load reduced with the descriptor management method • For transferring from EtherC receive FIFO to receive buffer × 1 channel • For transferring from transmit buffer to EtherC transmit FIFO × 1 channel • 16-byte burst transfer improves the efficiency of system bus • Supports single frame and multiple buffer

Items	Specification
Ethernet physical layer transceiver (PHY)	• Conforms to the IEEE802.3u standard. 10Base-T and 100Base-TX supported • Supports Auto-negotiation and manual-negotiation modes • Supports power-down modes • Outputs the status of Link, Activity, Duplex, and Speed • Selection of either on-chip clock oscillator output or dedicated clock externally input
Host interface (HIF)	• 1 kbyte × 2 banks: in total 2-kbyte buffer RAM • The buffer RAM and the external device are connected in parallel via 16 data pins • The buffer RAM and the CPU of this LSI are connected in parallel via internal bus • The external device can access the desired register after the register index has been specified. (However, when the buffer RAM is accessed successively, the address is updated automatically.) • Selection of endian mode • Interrupt requested to the external device • Internal interrupt requested to the CPU of this LSI • Booting from the buffer RAM is enabled if the external device has stored the instruction code in the buffer RAM
Compare match timer (CMT)	• 16-bit counter • Generates compare match interrupts • Two channels
Serial communication interface with FIFO (SCIF)	• Synchronous and asynchronous modes • 16 bytes each for transmit/receive FIFO • High-speed UART • The UART supports FIFO stop and FIFO trigger • Flow control enabled (channel 0 and channel 1 only) • Three channels

Items	Specification
Serial IO with FIFO (SIOF)	• 64 bytes each for transmit/receive FIFO • Supports 8-/16-/16-bit stereo sound input/output • Can operate together with the DMAC • Supports frame synchronous signals • One channel
I/O ports	• 78 general input/output pins • Input or output can be set per bit within the input/output common port
Package	• BP1313-176 (0.8 pitch)
Power supply voltage	• I/O: 3.0 to 3.6 V Internal: 1.8±0.09 V (Two power sources are externally provided.)

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

1.2 Block Diagram

Figure 1.1 is a block diagram of this LSI.

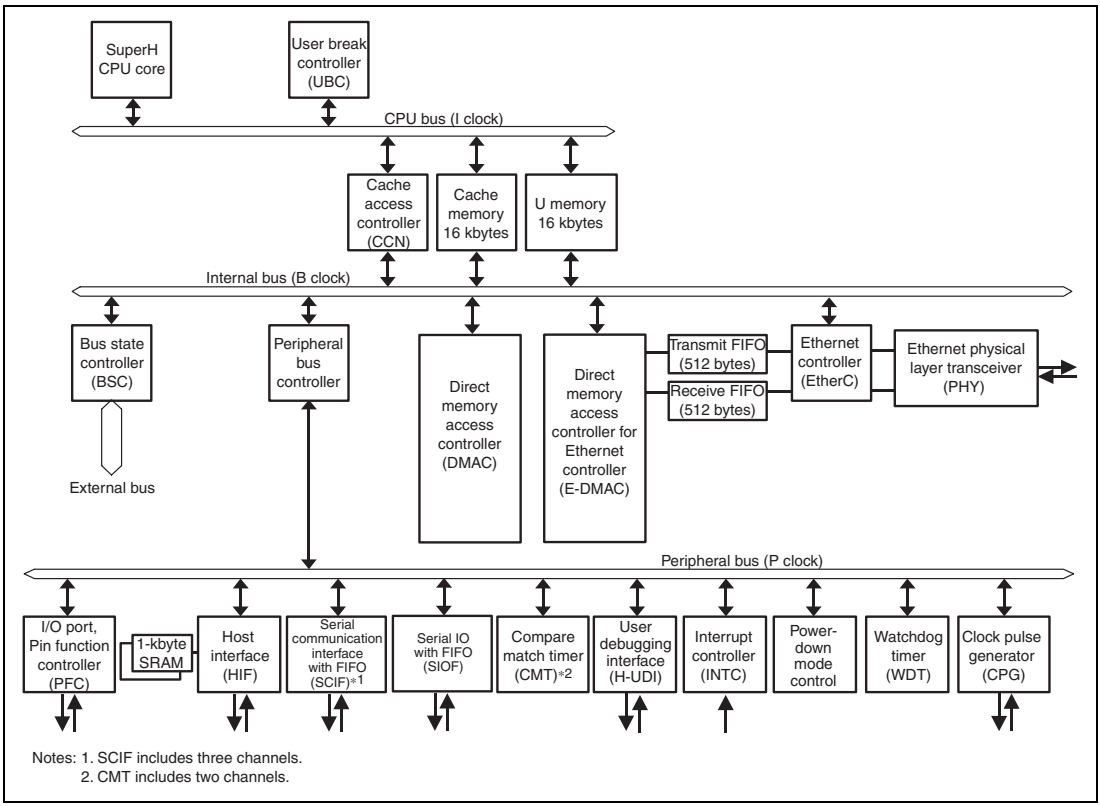


Figure 1.1 Block Diagram

1.3 Pin Assignments

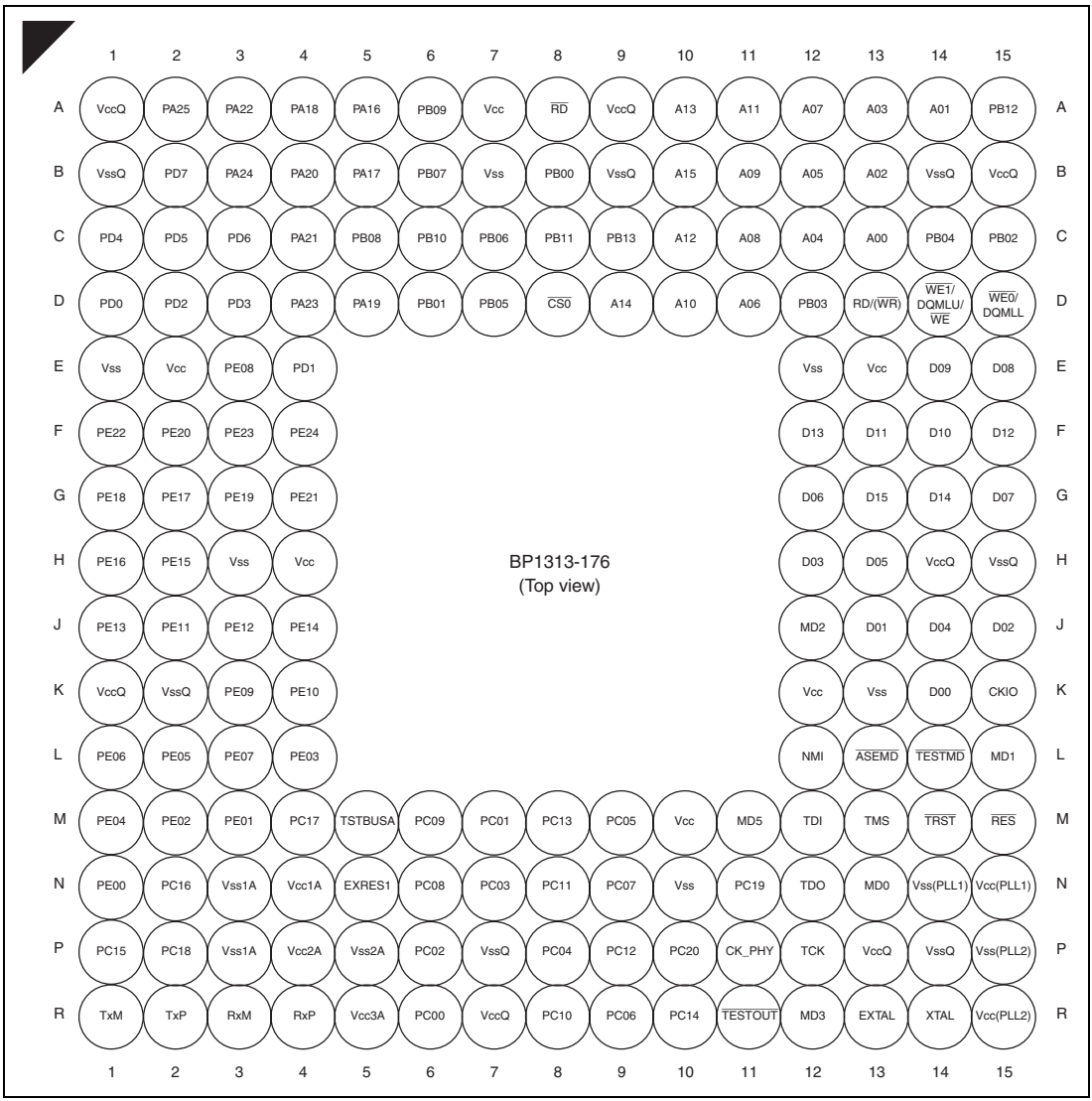


Figure 1.2 Pin Assignments

1.4 Pin Functions

Table 1.2 Pin Functions

Classification	Abbr.	I/O	Pin Name	Description
Power supply	Vcc	Input	Power Supply	Power supply for the internal logic of this LSI. All the Vcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	Vss	Input	Ground	Ground pins. All the Vss pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	VccQ	Input	Power Supply	Power supply for input/output pins. All the VccQ pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	VssQ	Input	Ground	Ground pins. All the VssQ pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
Clock	Vcc (PLL1)	Input	Power Supply for PLL1	Power supply pin for the on-chip PLL1 oscillator
	Vss (PLL1)	Input	Ground for PLL1	Ground pin for the on-chip PLL1 oscillator
	Vcc (PLL2)	Input	Power Supply for PLL2	Power supply pin for the on-chip PLL2 oscillator
	Vss (PLL2)	Input	Ground for PLL2	Ground pin for the on-chip PLL2 oscillator
	EXTAL	Input	External Clock	Connects to a crystal resonator. An external clock is also input on this pin. For details on connection of an external clock, see section 8, Clock Pulse Generator (CPG).
	XTAL	Output	Crystal	Connects to a crystal resonator.
	CKIO	Output	System Clock	Supplies the system clock to external devices.
Operating mode control	MD5, MD3 to MD0	Input	Mode Setting	These pins set operating mode. The signal levels of these pins must not be changed during operation. Pins MD2 to MD0 are used for setting clock mode, pin MD3 is for setting bus width mode for area 0, and pin MD5 is for setting endian.
System control	$\overline{\text{RES}}$	Input	Power-On Reset	This LSI enters the power-on reset state when this signal goes low.

Classification	Abbr.	I/O	Pin Name	Description
Interrupt	NMI	Input	Non-Maskable Interrupt	Non-maskable interrupt request pin. This pin must be fixed high when not in use.
	IRQ7 to IRQ0	Input	Interrupt Request 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising or falling edge can also be selected.
Address bus	A25 to A0	Output	Address Bus	These pins output addresses.
Data bus	D31 to D0	Input/ output	Data Bus	32-bit bidirectional bus
Bus control	$\overline{CS0}$, $\overline{CS3}$, $\overline{CS4}$, $\overline{CS5B}$, $\overline{CS6B}$	Output	Chip Select 0, 3, 4, 5B, 6B	Chip select signals for external memory and devices.
	$\overline{RD}$	Output	Read	Indicates that data is read from an external device.
	RD/$\overline{WR}$	Output	Read/Write	Read/write signal
	$\overline{BS}$	Output	Bus Cycle Start	Indicates start of a bus cycle.
	$\overline{WE3}$	Output	Most Significant Byte Write	Indicates that bits 31 to 24 of data of external memory or devices are written to.
	$\overline{WE2}$	Output	Second Byte Write	Indicates that bits 23 to 16 of data of external memory or devices are written to.
	$\overline{WE1}$	Output	Third Byte Write	Indicates that bits 15 to 8 of data of external memory or devices are written to.
	$\overline{WE0}$	Output	Least Significant Byte Write	Indicates that bits 7 to 0 of data of external memory or devices are written to.
	$\overline{WAIT}$	Input	Wait	Input pin used to insert wait cycles into the bus cycle when accessing the external space
	$\overline{RAS}$	Output	RAS	Connects to the $\overline{RAS}$ pin of SDRAM.
	$\overline{CAS}$	Output	CAS	Connects to the $\overline{CAS}$ pin of SDRAM.
	CKE	Output	Clock Enable	Connects to the CKE pin of SDRAM.
	DQMUU	Output	Most Significant Byte Select	Selects bits 31 to 24 of SDRAM data bus.

Classification	Abbr.	I/O	Pin Name	Description
Bus control	DQMUL	Output	Second Byte Select	Selects bits 23 to 16 of SDRAM data bus.
	DQMLU	Output	Third Byte Select	Selects bits 15 to 8 of SDRAM data bus.
	DQMLL	Output	Least Significant Byte Select	Selects bits 7 to 0 of SDRAM data bus.
	$\overline{\text{CE1A}}$	Output	PCMCIA Card Select Lower Side	Chip enable for PCMCIA allocated to area 5
	$\overline{\text{CE1B}}$	Output	PCMCIA Card Select Lower Side	Chip enable for PCMCIA allocated to area 6
	$\overline{\text{CE2A}}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 5
	$\overline{\text{CE2B}}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 6
	$\overline{\text{ICIOWR}}$	Output	PCMCIA I/O Write Strobe	Connects to the PCMCIA I/O write strobe pin.
	$\overline{\text{ICIORD}}$	Output	PCMCIA I/O Read Strobe	Connects to the PCMCIA I/O read strobe pin.
	$\overline{\text{WE}}$	Output	PCMCIA Memory Write Strobe	Connects to the PCMCIA memory write strobe.
	$\overline{\text{IOIS16}}$	Input	PCMCIA Dynamic Bus Sizing	In little endian mode, this signal indicates 16-bit bus width of PCMCIA. In big endian mode, fix this pin low.
Ethernet controller	CRS	Input	Carrier Sense	Carrier sense pin
	COL	Input	Collision	Collision detect pin
	MII_TXD3 to MII_TXD0	Output	Transmit Data	4-bit transmit data pins
	TX_EN	Output	Transmit Enable	Indicates that transmit data is on pins MII_TXD3 to MII_TXD0.

Classification	Abbr.	I/O	Pin Name	Description
Ethernet controller	TX_CLK	Input	Transmit Clock	Timing reference input for the TX_EN, TX_ER, and MII_TXD3 to MII_TXD0 pins
	TX_ER	Output	Transmit Error	Informs PHY LSI of an error during transmission.
	MII_RXD3 to MII_RXD0	Input	Receive Data	4-bit receive data pins
	RX_DV	Input	Receive Data Valid	Indicates that valid receive data is on pins MII_RXD3 to MII_RXD0.
	RX_CLK	Input	Receive Clock	Timing reference input for the RX_DV, RX_ER, and MII_RXD3 to MII_RXD0 pins
	RX_ER	Input	Receive Error	Pin for detection of an error during reception
	MDC	Output	Management Clock	Timing reference input for transfer information on the MDIO pin
	MDIO	Input/ output	Management Data I/O	Bidirectional pin for management information transfer
	WOL	Output	MAGIC Packet Receive	Indicates that a Magic Packet™* has received.
	LNKSTA	Input	Link Status	Input pin for a link state from a PHY LSI.
	EXOUT	Output	General Output	Output pin to external devices
Direct memory access controller	DREQ1, DREQ0	Input	DMA transfer request	Input pins for external DMA transfer request
	DACK1, DACK0	Output	DMA transfer request receive	Request receive output pins for external DMA transfer request
	TEND1, TEND0	Output	DMA transfer end	Output pins for DMA transfer end signal
Serial communication interface with FIFO	TXD2 to TXD0	Output	Transmit Data	Transmit data pins
	RXD2 to RXD0	Input	Receive Data	Receive data pins
	SCK2 to SCK0	Input/ output	Serial Clock	Clock input pins
	$\overline{\text{RTS1}}$ and $\overline{\text{RTS0}}$	Output	Transmit Request	Modem control pins. Supported only by SCIF0 and SCIF1.

Classification	Abbr.	I/O	Pin Name	Description
Serial Communication interface with FIFO	$\overline{\text{CTS1}}$ and $\overline{\text{CTS0}}$	Input	Transmit Enable	Modem control pins. Supported only by SCIF0 and SCIF1.
Serial I/O with FIFO	SIOMCLK0	Input	SIOF0 clock input	Master clock input pin
	SCK_SIO0	Input/ output	SIOF0 communication clock	Input/output pin for communication clock common to transmit/receive
	SIOFSYNC0	Input/ output	SIOF0 frame sync	Input/output pin for frame synchronization signal common to transmit/receive
	TXD_SIO0	Output	SIOF0 transmit data	Transmit data
	RXD_SIO0	Input	SIOF0 receive data	Receive data
Host interface	HIFD15 to HIFD00	Input/ output	HIF Data Bus	Address, data, and command input/output pins for the HIF.
	$\overline{\text{HIFCS}}$	Input	HIF Chip Select	Chip select input for the HIF.
	HIFRS	Input	HIF Register Select	Controls the access type switching for the HIF.
	$\overline{\text{HIFWR}}$	Input	HIF Write	Write strobe signal
	$\overline{\text{HIFRD}}$	Input	HIF Read	Read strobe signal
	$\overline{\text{HIFINT}}$	Output	HIF Interrupt	Interrupt request to external devices by the HIF.
	HIFMD	Input	HIF Mode	Specifies HIF boot mode.
	HIFDREQ	Output	HIF DMAC Transfer Request	Requests DMAC transfer for the HIFRAM to external devices.
	HIFRDY	Output	HIF Boot Ready	Indicates that a reset of the HIF has been cleared in this LSI and the HIF is ready for accesses to it.
	HIFEBL	Input	HIF Pin Enable	HIF pins other than this pin are enabled by driving this pin high.

Classification	Abbr.	I/O	Pin Name	Description
User debugging interface (H-UDI)	TCK	Input	Test Clock	Test clock input pin
	TMS	Input	Test Mode Select	Input pin for test mode select signal
	TDI	Input	Test Data Input	Serial input pin for an instruction and data
	TDO	Output	Test Data Output	Serial output pin for an instruction and data
	$\overline{\text{TRST}}$	Input	Test Reset	Input pin for initialization
I/O ports	PA25 to PA16	Input/ output	General Port	Pins for 10-bit general input/output port
	PB13 to PB00	Input/ output	General Port	Pins for 14-bit general input/output port
	PC20 to PC00	Input/ output	General Port	Pins for 21-bit general input/output port
	PD07 to PD00	Input/ output	General Port	Pins for 8-bit general input/output port
	PE24 to PE00	Input/ output	General Port	Pins for 25-bit general input/output port
Emulator interface	$\overline{\text{ASEMD}}$	Input	ASE Mode	Specifies ASE mode. This LSI enters ASE mode when this signal goes low and normal mode when this pin goes high. In ASE mode, functions for the emulator are available.
Test mode	$\overline{\text{TESTMD}}$	Input	Test Mode	Specifies test mode. This LSI enters test mode when this signal goes low. Fix this signal high.
	$\overline{\text{TESTOUT}}$	Output	Test Output	Output pin for testing. This pin should be open.
Physical layer trans-ceiver (PHY)	Vcc1A	Input	Analog Power Supply 1 for PHY	Analog power supply pin for the PHY
	Vcc2A	Input	Analog Power Supply 2 for PHY	Analog power supply pin for the PHY

Classification	Abbr.	I/O	Pin Name	Description
Physical layer transceiver (PHY)	Vcc3A	Input	Analog Power Supply 3 for PHY	Analog power supply pin for the PHY
	Vss1A	Input	Analog Ground 1 for PHY	Analog ground pin for the PHY
	Vss2A	Input	Analog Ground 2 for PHY	Analog ground pin for the PHY
	CK_PHY	Input	PHY Clock	This pins is used to externally supply clocks to the PHY. When clocks are supplied to the on-chip PHY from the on-chip clock pulse generator (CPG), this pins should be pulled up to VccQ or pulled down to VssQ.
	TxP	Output	Differential Transmit Data (+)	Differential transmit output (+) for the Ethernet circuit by the PHY.
	TxM	Output	Differential Transmit Data (-)	Differential transmit output (-) for the Ethernet circuit by the PHY.
	RxP	Input	Differential Receive Data (+)	Differential receive input (+) for the PHY by the Ethernet circuit.
	RxM	Input	Differential Receive Data (-)	Differential receive input (-) for the PHY by the Ethernet circuit.

Classification	Abbr.	I/O	Pin Name	Description
Physical layer transceiver (PHY)	$\overline{\text{SPEED100}}$	Output	SPEED100 signal	Monitor output pins indicating communication status
	$\overline{\text{LINK}}$	Output	LINK signal	
	$\overline{\text{CRS}}$	Output	CRS signal	
	$\overline{\text{DUPLEX}}$	Output	DUPLEX signal	
	EXRES1	Input	Reference resistor	Connect to the PHY analog ground through a 12.4-kΩ (accuracy: 1%) resistor.
	TSTBUSA	Input/output	Test I/O	Input/output pin for testing the on-chip PHY. This pin should be open.

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

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

Table 1.3 Pin Features

Pin No.	Pin Name	I/O Features
A1	VccQ	Power
A2	PA25/A25/SIOFSYNC0	IO/O/IO
A3	PA22/A22/SIOMCLK0	IO/O/I
A4	PA18/A18	IO/O
A5	PA16/A16	IO/O
A6	PB09/$\overline{\text{CE2A}}$	IO/O
A7	Vcc	Power
A8	$\overline{\text{RD}}$	O
A9	VccQ	Power
A10	A13	O
A11	A11	O
A12	A07	O
A13	A03	O
A14	A01	O
A15	PB12/$\overline{\text{CS3}}$	IO/O
B1	VssQ	Power
B2	PD7/IRQ7/SCK2	IO/I/IO
B3	PA24/A24/TXD_SIO0	IO/O/O
B4	PA20/A20	IO/O
B5	PA17/A17	IO/O
B6	PB07/CE2B	IO/O
B7	Vss	Power
B8	PB00/$\overline{\text{WAIT}}$	IO/I
B9	VssQ	Power
B10	A15	O
B11	A09	O
B12	A05	O
B13	A02	O
B14	VssQ	Power
B15	VccQ	Power

Pin No.	Pin Name	I/O Features
C1	PD4/IRQ4/SCK1	IO/I/IO
C2	PD5/IRQ5/TxD2/DREQ1	IO/I/O/I
C3	PD6/IRQ6/RxD2/DACK1	IO/I/I/O
C4	PA21/A21/SCK_SIO0	IO/O/IO
C5	PB08/$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$	IO/O/O
C6	PB10/$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$	IO/O/O
C7	PB06/$\overline{\text{WE3}}$ ($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$	IO/O/O/O
C8	PB11/$\overline{\text{CS4}}$	IO/O
C9	PB13/$\overline{\text{BS}}$	IO/O
C10	A12	O
C11	A08	O
C12	A04	O
C13	A00	O
C14	PB04/$\overline{\text{RAS}}$	IO/O
C15	PB02/CKE	IO/O
D1	PD0/IRQ0/-/TEND0	IO/I/-/O
D2	PD2/IRQ2/TxD1/DREQ0	IO/I/O/I
D3	PD3/IRQ3/RxD1/DACK0	IO/I/I/O
D4	PA23/A23/RXD_SIO0	IO/O/I
D5	PA19/A19	IO/O
D6	PB01/$\overline{\text{IOIS16}}$	IO/I
D7	PB05/$\overline{\text{WE2}}$ ($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$	IO/O/O/O
D8	CS0	O
D9	A14	O
D10	A10	O
D11	A06	O
D12	PB03/$\overline{\text{CAS}}$	IO/O
D13	RD/$\overline{\text{WR}}$	O
D14	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	O/O/O
D15	$\overline{\text{WE0}}$/DQMLL	O/O
E1	Vss	Power
E2	Vcc	Power

Pin No.	Pin Name	I/O Features
E3	PE08/$\overline{\text{HIFCS}}$	IO/I
E4	PD1/IRQ1/-/TEND1	IO/I/-/O
E12	Vss	Power
E13	Vcc	Power
E14	D09	IO
E15	D08	IO
F1	PE22/HIFD13/CTS0/D29	IO/IO/I/IO
F2	PE20/HIFD11/SCK1/D27	IO/IO/IO/IO
F3	PE23/HIFD14/RTS1/D30	IO/IO/O/IO
F4	PE24/HIFD15/CTS1/D31	IO/IO/I/IO
F12	D13	IO
F13	D11	IO
F14	D10	IO
F15	D12	IO
G1	PE18/HIFD09/TxD1/D25	IO/IO/O/IO
G2	PE17/HIFD08/SCK0/D24	IO/IO/IO/IO
G3	PE19/HIFD10/RxD1/D26	IO/IO/I/IO
G4	PE21/HIFD12/RTS0/D28	IO/IO/O/IO
G12	D06	IO
G13	D15	IO
G14	D14	IO
G15	D07	IO
H1	PE16/HIFD07/RxD0/D23	IO/IO/I/IO
H2	PE15/HIFD06/TxD0/D22	IO/IO/O/IO
H3	Vss	Power
H4	Vcc	Power
H12	D03	IO
H13	D05	IO
H14	VccQ	Power
H15	VssQ	Power
J1	PE13/HIFD04/-/D20	IO/IO/-/IO
J2	PE11/HIFD02/-/D18	IO/IO/-/IO

Pin No.	Pin Name	I/O Features
J3	PE12/HIFD03/-/D19	IO/IO/-/IO
J4	PE14/HIFD05/-/D21	IO/IO/-/IO
J12	MD2	I
J13	D01	IO
J14	D04	IO
J15	D02	IO
K1	VccQ	Power
K2	VssQ	Power
K3	PE09/HIFD00/-/D16	IO/IO/-/IO
K4	PE10/HIFD01/-/D17	IO/IO/-/IO
K12	Vcc	Power
K13	Vss	Power
K14	D00	IO
K15	CKIO	O
L1	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0	IO/I/IO
L2	PE05/$\overline{\text{HIFRD}}$	IO/I
L3	PE07/HIFRS	IO/I
L4	PE03/HIFMD	IO/I
L12	NMI	I
L13	$\overline{\text{ASEMD}}$	I
L14	$\overline{\text{TESTMD}}$	I
L15	MD1	I
M1	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0	IO/O/O
M2	PE02/HIFDREQ/RXD_SIO0	IO/O/I
M3	PE01/HIFRDY/SIOMCLK0	IO/O/I
M4	PC17/MDC	IO/O
M5	TSTBUSA	IO
M6	PC09/RX_ER	IO/I
M7	PC01/MII_RXD1	IO/I
M8	PC13/TX_CLK	IO/I
M9	PC05/MII_TXD1/-/$\overline{\text{LINK}}$	IO/O/-/O
M10	Vcc	Power

Pin No.	Pin Name	I/O Features
M11	MD5	I
M12	TDI	I
M13	TMS	I
M14	$\overline{\text{TRST}}$	I
M15	$\overline{\text{RES}}$	I
N1	PE00/HIFEBL/SCK_SIO0	IO/I/IO
N2	PC16/MDIO	IO/IO
N3	Vss1A	Power
N4	Vcc1A	Power
N5	EXRES1	I
N6	PC08/RX_DV	IO/I
N7	PC03/MII_RXD3	IO/I
N8	PC11/TX_ER	IO/O
N9	PC07/MII_TXD3/-/DUPLEX	IO/O/-/O
N10	Vss	Power
N11	PC19/EXOUT	IO/O
N12	TDO	O
N13	MD0	I
N14	Vss (PLL1)	Power
N15	Vcc (PLL1)	Power
P1	PC15/CRS	IO/I
P2	PC18/LNKSTA	IO/I
P3	Vss1A	Power
P4	Vcc2A	Power
P5	Vss2A	Power
P6	PC02/MII_RXD2	IO/I
P7	VssQ	Power
P8	PC04/MII_TXD0/-/$\overline{\text{SPEED100}}$	IO/O/-/O
P9	PC12/TX_EN	IO/O
P10	PC20/WOL	IO/O
P11	CK_PHY	I
P12	TCK	I

Pin No.	Pin Name	I/O Features
P13	VccQ	Power
P14	VssQ	Power
P15	Vss (PLL2)	Power
R1	TxM	O
R2	TxP	O
R3	RxM	I
R4	RxP	I
R5	Vcc3A	Power
R6	PC00/MII_RXD0	IO/I
R7	VccQ	Power
R8	PC10/RX_CLK	IO/I
R9	PC06/MII_TXD2/-/$\overline{\text{CRS}}$	IO/O/-/O
R10	PC14/COL	IO/I
R11	$\overline{\text{TESTOUT}}$	O
R12	MD3	I
R13	EXTAL	I
R14	XTAL	O
R15	Vcc (PLL2)	Power

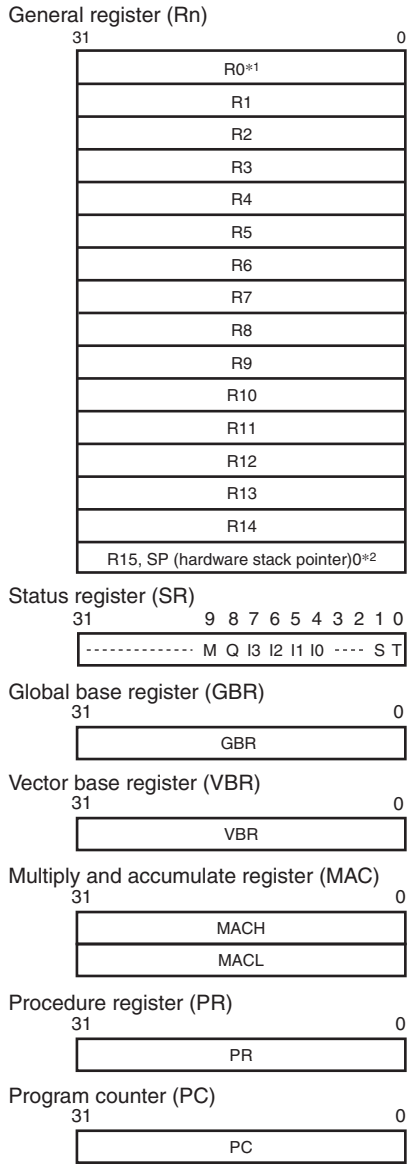
Section 2 CPU

2.1 Features

- General registers: 32-bit register × 16
- Basic instructions: 62
- Addressing modes: 11
  Register direct (Rn)
  Register indirect (@Rn)
  Post-increment register indirect (@Rn+)
  Pre-decrement register indirect (@-Rn)
  Register indirect with displacement (@disp:4, Rn)
  Index register indirect (@R0, Rn)
  GBR indirect with displacement (@disp:8, GBR)
  Index GBR indirect (@R0, GBR)
  PC relative with displacement (@disp:8, PC)
  PC relative (disp:8/disp:12/Rn)
  Immediate (#imm:8)

2.2 Register Configuration

There are three types of registers: general registers (32-bit × 16), control registers (32-bit × 3), and system registers (32-bit × 4).



Notes: 1. R0 can be used as an index register in index register indirect or index GBR indirect addressing mode. For some instructions, only R0 is used as the source or destination register.
2. R15 is used as a hardware stack pointer during exception handling.

Figure 2.1 CPU Internal Register Configuration

2.2.1 General Registers (Rn)

There are sixteen 32-bit general registers (Rn), designated R0 to R15. The general registers are used for data processing and address calculation. R0 is also used as an index register. With a number of instructions, R0 is the only register that can be used. R15 is used as a hardware stack pointer (SP). In exception handling, R15 is used for accessing the stack to save or restore the status register (SR) and program counter (PC) values.

2.2.2 Control Registers

There are three 32-bit control registers, designated status register (SR), global base register (GBR), and vector base register (VBR). SR indicates a processing state. GBR is used as a base address in GBR indirect addressing mode for data transfer of on-chip peripheral module registers. VBR is used as a base address of the exception handling (including interrupts) vector table.

- Status register (SR)

Bit	Bit name	Default	Read/ Write	Description
31 to 10	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
9	M	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
8	Q	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
7	I3	1	R/W	Interrupt Mask
6	I2	1	R/W	
5	I1	1	R/W	
4	I0	1	R/W	
3, 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1	S	Undefined	R/W	S Used by the multiply and accumulate instruction.

Bit	Bit name	Default	Read/ Write	Description
0	T	Undefined	R/W	T Indicates true (1) or false (0) in the following instructions: MOVT, CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SETT, CLRT Indicates carry, borrow, overflow, or underflow in the following instructions: ADDV, ADDC, SUBV, SUBC, NEGC, DIV0U, DIV0S, DIV1, SHAR, SHAL, SHLR, SHLL, ROTR, ROTL, ROTCR, ROTCL

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR indirect addressing mode is used for data transfer of the on-chip peripheral module registers and logic operations.
- Vector-base register (VBR)

  This register indicates the base address of the exception handling vector table.

2.2.3 System Registers

There are four 32-bit system registers, designated two multiply and accumulate registers (MACH and MACL), a procedure register (PR), and program counter (PC).

- Multiply and accumulate registers (MAC)

  This register stores the results of multiplication and multiply-and-accumulate operation.
- Procedure register (PR)

  This register stores the return-destination address from subroutine procedures.
- Program counter (PC)

  The PC indicates the point which is four bytes (two instructions) after the current execution instruction.

2.2.4 Initial Values of Registers

Table 2.1 lists the initial values of registers after a reset.

Table 2.1 Initial Values of Registers

Type of register	Register	Default
General register	R0 to R14	Undefined
	R15 (SP)	SP value set in the exception handling vector table
Control register	SR	I3 to I0: 1111 (H'F) Reserved bits: 0 Other bits: Undefined
	GBR	Undefined
	VBR	H'00000000
System register	MACH, MACL, PR	Undefined
	PC	PC value set in the exception handling vector table

2.3 Data Formats

2.3.1 Register Data Format

The size of register operands is always longwords (32 bits). When loading byte (8 bits) or word (16 bits) data in memory into a register, the data is sign-extended to longword and stored in the register.

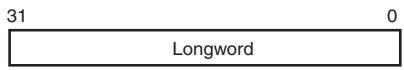


Figure 2.2 Register Data Format

2.3.2 Memory Data Formats

Memory data formats are classified into byte, word, and longword.

Byte data can be accessed from any address. If word data starting from boundary other than 2n or longword data starting from a boundary other than 4n is accessed, an address error will occur. In such cases, the data accessed cannot be guaranteed. See figure 2.3.

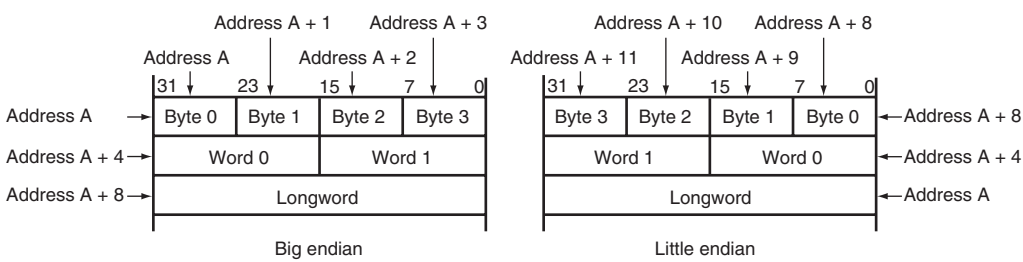


Figure 2.3 Memory Data Format

Either big endian and little endian formats can be selected according to the mode pin setting at a reset. For details on mode pin settings, see section 7, Bus State Controller (BSC).

2.3.3 Immediate Data Formats

Immediate data of eight bits is placed in the instruction code.

For the MOV, ADD, and CMP/EQ instructions, the immediate data is sign-extended to longword and then calculated. For the TST, AND, OR, and XOR instructions, the immediate data is zero-extended to longword and then calculated. Thus, if the immediate data is used for the AND instruction, the upper 24 bits in the destination register are always cleared.

The immediate data of word or longword is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed by the MOV immediate data instruction in PC relative addressing mode with displacement.

2.4 Features of Instructions

2.4.1 RISC Type

The instructions are RISC-type instructions with the following features:

Fixed 16-Bit Length: All instructions have a fixed length of 16 bits. This improves program code efficiency.

One Instruction per Cycle: Since pipelining is used, basic instructions can be executed in one cycle. One cycle is 25ns with 40 MHz operation.

Data Size: The basic data size for operations is longword. Byte, word, or longword can be selected as the memory access size. Byte or word data in memory is sign-extended to longword and then calculated. Immediate data is sign-extended to longword for arithmetic operations or zero-extended to longword size for logical operations.

Table 2.2 Word Data Sign Extension

CPU in this LSI		Description	Example of Other CPUs
MOV.W	@(disp,PC),R1	Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

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

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 in memory.

Delayed Branching: Unconditional branch instructions mean the delayed branch instructions. With a delayed branch instruction, the branch is made after execution of the instruction immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made. The conditional branch instructions have two types of instructions: conditional branch instructions and delayed branch instructions.

Table 2.3 Delayed Branch Instructions

CPU in this LSI		Description	Example of Other CPUs	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: A 16 × 16 → 32 multiply operation is executed in one to two cycles, and a 16 × 16 + 64 → 64 multiply-and-accumulate operation in two to three cycles. A 32 × 32 → 64 multiply operation and a 32 × 32 + 64 → 64 multiply-and-accumulate operation are each executed in two to four cycles.

T Bit: The result of a comparison is indicated by the T bit in SR, and a conditional 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.

Table 2.4 T Bit

CPU in this LSI		Description	Example of Other CPUs	
CMP/GE	R1,R0	When R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	When R0 ≥ R1, a branch is made to TRGET0.	BGE	TRGET0
BF	TRGET1	When R0 < R1, a branch is made to TRGET1.	BLT	TRGET1
ADD	#–1,R0	The T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	When R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made when R0 = 0.		

Immediate Data: 8-bit immediate data is placed in the instruction code. Word and longword immediate data is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed with the MOV immediate data instruction using PC relative addressing mode with displacement.

Table 2.5 Access to Immediate Data

Type	This LSI's CPU	Example of Other CPU
8-bit immediate	MOV #H'12,R0	MOV.B #H'12,R0
16-bit immediate	MOV.W @(disp,PC),R0 ..... .DATA.W H'1234	MOV.W #H'1234,R0
32-bit immediate	MOV.L @(disp,PC),R0 ..... .DATA.L H'12345678	MOV.L #H'12345678,R0

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

Absolute Addresses: When data is accessed by absolute address, place the absolute address value in a table in memory beforehand. The absolute address value is transferred to a register using the method whereby immediate data is loaded when an instruction is executed, and the data is accessed using the register indirect addressing mode.

Table 2.6 Access to Absolute Address

Type	CPU in this LSI	Example of Other CPUs
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 ..... .DATA.L H'12345678	MOV.B @H'12345678,R0

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

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

Table 2.7 Access with Displacement

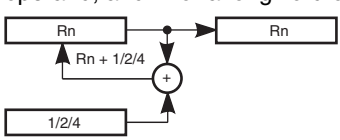
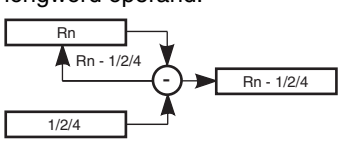
Type	CPU in this LSI	Example of Other CPUs
16-bit displacement	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 ..... .DATA.W H'1234	MOV.W @(H'1234,R1),R2

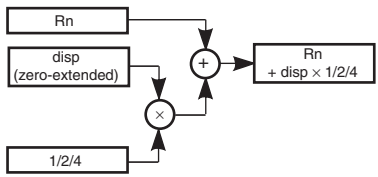
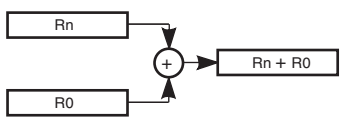
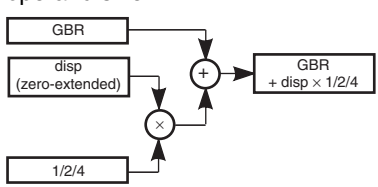
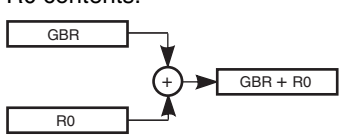
Note: * Immediate data is referenced by @(disp,PC).

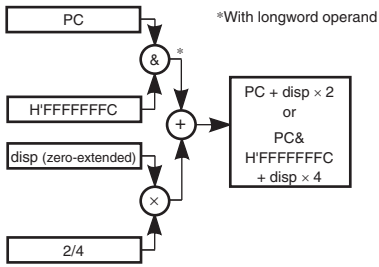
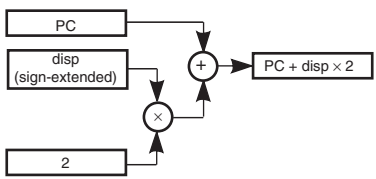
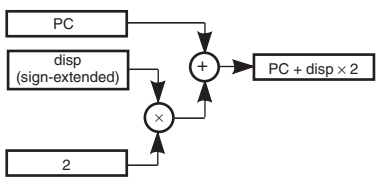
2.4.2 Addressing Modes

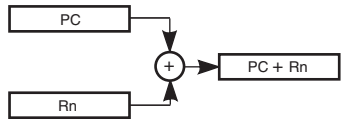
Table 2.8 lists addressing modes and effective address calculation methods.

Table 2.8 Addressing Modes and Effective Addresses

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

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

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
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.	—

2.4.3 Instruction Formats

This section describes the instruction formats, and the meaning of the source and destination operands. 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 2.9 Instruction Formats

Instruction Format	Source Operand	Destination Operand	Sample Instruction
0 type 15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n type 15 0 xxxx nnnn xxxx xxxx	—	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 15 0 xxxx mmmm xxxx xxxx	mmmm: register direct	Control register or system register	LDC Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L @Rm+,SR
	mmmm: register indirect	—	JMP @Rm
	PC relative using Rm	—	BRAF Rm

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type 15 0 xxxx nnnn mmmm xxxx	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register direct	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+
	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: index register indirect	MOV.L Rm,@(R0,Rn)
md type 15 0 xxxx xxxx mmmm dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 15 0 xxxx xxxx nnnn dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 15 0 xxxx nnnn mmmm dddd	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

Instruction Format	Source Operand	Destination Operand	Sample Instruction
d type 15 0 xxxx xxxx dddd dddd	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 15 0 xxxx dddd dddd dddd	—	ddddddddddddd: PC relative	BRA label (label=disp+PC)
nd8 type 15 0 xxxx nnnn dddd dddd	dddddddd: PC relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type 15 0 xxxx xxxx iiii iiii	iiiiiiii: immediate	Index GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 15 0 xxxx nnnn iiii iiii	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

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

2.5 Instruction Set

2.5.1 Instruction Set by Type

Table 2.10 lists the instructions classified by type.

Table 2.10 Instruction Types

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

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	MULS	Signed multiplication	33
		MULU	Unsigned multiplication	
		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	T bit setting for logical AND	
		XOR	Exclusive logical OR	
Shift instructions	10	ROTL	1-bit left shift	14
		ROTR	1-bit right shift	
		ROTCL	1-bit left shift with T bit	
		ROTCR	1-bit right shift with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	

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	11	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load into control register	
		LDS	Load into system register	
		NOP	No operation	
		RTE	Return from exception handling	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	62			142

The instruction code, operation, and execution cycles of the instructions are listed in the following tables, classified by type.

Instruction	Instruction Code	Summary of Operation	Execution Cycles	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait cycles are inserted*1	Value of T bit after instruction is executed Explanation of Symbols —: No change
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*2	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement	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		

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:
- When there is contention between an instruction fetch and a data access
- When the destination register of a load instruction (memory → register) is also used by the following instruction

2. Scaled (×1, ×2, or ×4) according to the instruction operand size, etc.
For details, see SH-1/SH-2/SH-DSP Software Manual.

- Data Transfer Instructions

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

Instruction	Operation	Code	Execution Cycles	T Bit
MOV.L Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	1	—
MOV.B @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	1	—
MOV.W @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	1	—
MOV.L @(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	1	—
MOV.B R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	1	—
MOV.W R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	1	—
MOV.L R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	1	—
MOV.B @(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	1	—
MOV.W @(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	1	—
MOV.L @(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	1	—
MOVA @(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	1	—
MOVT Rn	T → Rn	0000nnnn00101001	1	—
SWAP.B Rm,Rn	Rm → Swap lowest two bytes → Rn	0110nnnnmmmm1000	1	—
SWAP.W Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	1	—
XTRCT Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	1	—

- Arithmetic Operation Instructions

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

Instruction	Operation	Code	Execution Cycles	T Bit
DMULU.L Rm,Rn	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm0101	2 to 5*	—
DT Rn	Rn - 1 → Rn, if Rn = 0, 1 → T, else 0 → T	0100nnnn00010000	1	Comparison result
EXTS.B Rm,Rn	A byte in Rm is sign-extended → Rn	0110nnnnmmmm1110	1	—
EXTS.W Rm,Rn	A word in Rm is sign-extended → Rn	0110nnnnmmmm1111	1	—
EXTU.B Rm,Rn	A byte in Rm is zero-extended → Rn	0110nnnnmmmm1100	1	—
EXTU.W Rm,Rn	A word in Rm is zero-extended → Rn	0110nnnnmmmm1101	1	—
MAC.L @Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	2 to 5*	—
MAC.W @Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	2 to 4*	—
MUL.L Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	2 to 5*	—
MULS.W Rm,Rn	Signed operation of Rn × Rm → MAC 16 × 16 → 32 bits	0010nnnnmmmm1111	1 (3)*	—
MULU.W Rm,Rn	Unsigned operation of Rn × Rm → MAC 16 × 16 → 32 bits	0010nnnnmmmm1110	1 (3)*	—
NEG Rm,Rn	0-Rm → Rn	0110nnnnmmmm1011	1	—
NEGC Rm,Rn	0-Rm-T → Rn, Borrow → T	0110nnnnmmmm1010	1	Borrow
SUB Rm,Rn	Rn-Rm → Rn	0011nnnnmmmm1000	1	—
SUBC Rm,Rn	Rn-Rm–T → Rn, Borrow → T	0011nnnnmmmm1010	1	Borrow

Instruction	Operation	Code	Execution Cycles	T Bit
SUBV Rm,Rn	Rn-Rm → Rn, Underflow → T	0011nnnnmmmm1011	1	Overflow

Note: * Indicates the number of execution cycles for normal operation. The values in parentheses indicate the number of execution cycles when conflicts occur with the previous or next instruction.

- Logic Operation Instructions

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

- Shift Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
ROTL Rn	T ← Rn ← MSB	0100nnnn00000100	1	MSB
ROTR Rn	LSB → Rn → T	0100nnnn00000101	1	LSB
ROTCL Rn	T ← Rn ← T	0100nnnn00100100	1	MSB
ROTCR Rn	T → Rn → T	0100nnnn00100101	1	LSB
SHAL Rn	T ← Rn ← 0	0100nnnn00100000	1	MSB
SHAR Rn	MSB → Rn → T	0100nnnn00100001	1	LSB
SHLL Rn	T ← Rn ← 0	0100nnnn00000000	1	MSB
SHLR Rn	0 → Rn → T	0100nnnn00000001	1	LSB
SHLL2 Rn	Rn << 2 → Rn	0100nnnn00001000	1	—
SHLR2 Rn	Rn >> 2 → Rn	0100nnnn00001001	1	—
SHLL8 Rn	Rn << 8 → Rn	0100nnnn00011000	1	—
SHLR8 Rn	Rn >> 8 → Rn	0100nnnn00011001	1	—
SHLL16 Rn	Rn << 16 → Rn	0100nnnn00101000	1	—
SHLR16 Rn	Rn >> 16 → Rn	0100nnnn00101001	1	—

- Branch Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
BF label	If T = 0, disp × 2 + PC → PC; if T = 1, nop	10001011dddddddd	3/1*	—
BF/S label	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	10001111dddddddd	2/1*	—
BT label	If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	3/1*	—
BT/S label	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	2/1*	—

Instruction	Operation	Code	Execution Cycles	T Bit
BRA label	Delayed branch, disp × 2 + PC → PC	1010ddddddddddddd	2	—
BRAF Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	2	—
BSR label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011ddddddddddddd	2	—
BSRF Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	2	—
JMP @Rm	Delayed branch, Rm → PC	0100mmmm00101011	2	—
JSR @Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	2	—
RTS	Delayed branch, PR → PC	0000000000001011	2	—

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

- System Control Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
CLRT	0 → T	0000000000001000	1	0
CLRMAC	0 → MACH, MACL	0000000000101000	1	—
LDC Rm,SR	Rm → SR	0100mmmm00001110	6	LSB
LDC Rm,GBR	Rm → GBR	0100mmmm00011110	4	—
LDC Rm,VBR	Rm → VBR	0100mmmm00101110	4	—
LDC.L @Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	8	LSB
LDC.L @Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	4	—
LDC.L @Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	4	—
LDS Rm,MACH	Rm → MACH	0100mmmm00001010	1	—
LDS Rm,MACL	Rm → MACL	0100mmmm00011010	1	—
LDS Rm,PR	Rm → PR	0100mmmm00101010	1	—
LDS.L @Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	1	—
LDS.L @Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	1	—

Instruction	Operation	Code	Execution Cycles	T Bit
LDS.L @Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	1	—
NOP	No operation	0000000000001001	1	—
RTE	Delayed branch, Stack area → PC/SR	0000000000101011	5	—
SETT	1 → T	0000000000011000	1	1
SLEEP	Sleep	0000000000011011	4*	—
STC SR,Rn	SR → Rn	0000nnnn00000010	1	—
STC GBR,Rn	GBR → Rn	0000nnnn00010010	1	—
STC VBR,Rn	VBR → Rn	0000nnnn00100010	1	—
STC.L SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	1	—
STC.L GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	1	—
STC.L VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	1	—
STS MACH,Rn	MACH → Rn	0000nnnn00001010	1	—
STS MACL,Rn	MACL → Rn	0000nnnn00011010	1	—
STS PR,Rn	PR → Rn	0000nnnn00101010	1	—
STS.L MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	1	—
STS.L MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	1	—
STS.L PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	1	—
TRAPA #imm	PC/SR → Stack area, (imm × 4 + VBR) → PC	11000011iiiiiiii	8	—

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

About the number of execution cycles:

The table lists the minimum number of execution cycles. In practice, the number of execution cycles will be increased depending on the conditions such as:

- When there is a conflict between instruction fetch and data access
- When the destination register of a load instruction (memory → register) is also used by the instruction immediately after the load instruction.

2.6 Processing States

2.6.1 State Transition

The CPU has the four processing states: reset, exception handling, program execution, and power-down. Figure 2.4 shows the CPU state transition. Note that some products do not support the manual reset function and the $\overline{\text{MRES}}$ pin.

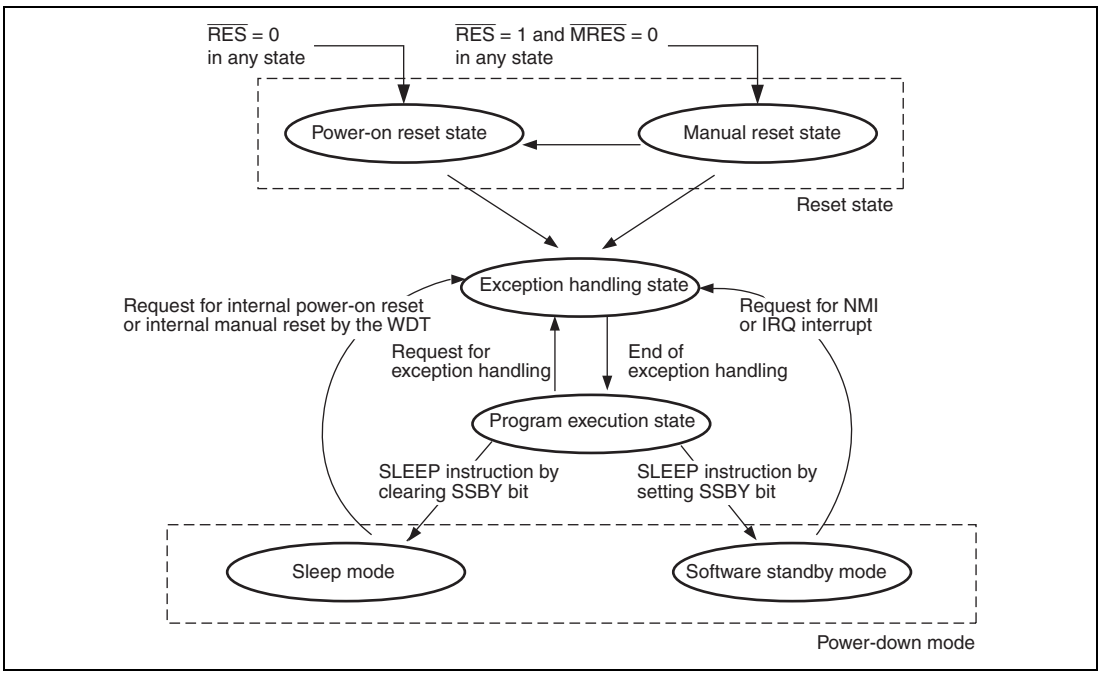


Figure 2.4 CPU State Transition

- Reset state

  The CPU is reset. When the $\overline{\text{RES}}$ pin is driven low, the CPU enters the power-on reset state. When the $\overline{\text{RES}}$ pin is high and $\overline{\text{MRES}}$ pin is low, the CPU enters the manual reset state.

- Exception handling state

  This state is a transitional state in which the CPU processing state changes due to a request for exception handling such as a reset or an interrupt.

  When a reset occurs, the execution start address as the initial value of the program counter (PC) and the initial value of the stack pointer (SP) are fetched from the exception handling vector table. Then, a branch is made for the start address to execute a program.

  When an interrupt occurs, the PC and status register (SR) are saved in the stack area pointed to by SP. The start address of an exception handling routine is fetched from the exception handling vector table and a branch to the address is made to execute a program.

  Then the processing state enters the program execution state.

- Program execution state

  The CPU executes programs sequentially.

- Power-down state

  The CPU stops to reduce power consumption. The SLEEP instruction makes the CPU enter sleep mode or software standby mode.

Section 3 Cache

3.1 Features

- Capacity: 16 kbytes
- Structure: Instructions/data unified, 4-way set associative
- Line size: 16 bytes
- Number of entries: 256 entries/way in 16-kbyte mode
- Write method: Write-back/write-through is selectable
- Replacement method: Least-recently-used (LRU) algorithm

3.1.1 Cache Structure

The cache holds both instructions and data and employs a 4-way set associative system. 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 256 entries. The data of an entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries), with a total of 16 kbytes in the cache (4 ways).

Figure 3.1 shows the cache structure.

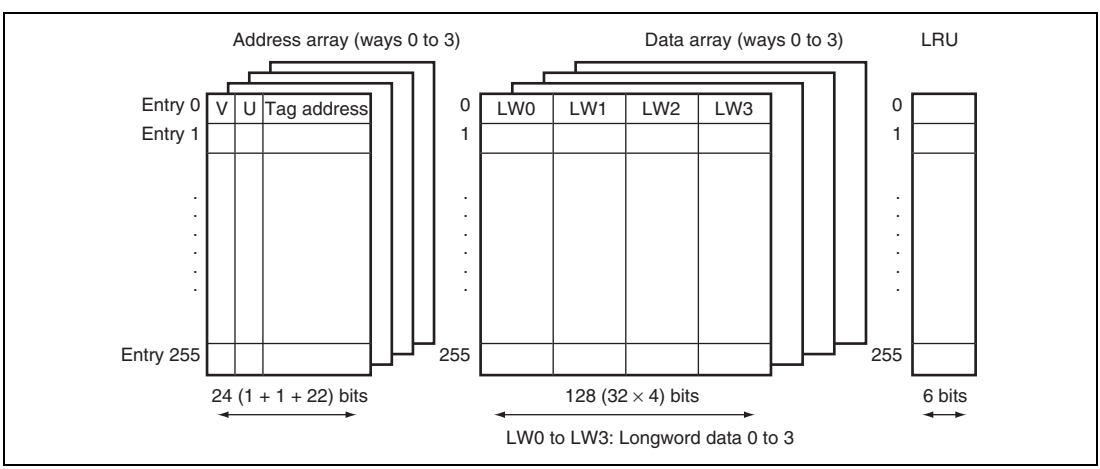


Figure 3.1 Cache Structure

Address Array: The V bit indicates whether or not 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 or not 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 is composed of 22 bits (address bits 31 to 10) used for comparison during cache searches.

In this LSI, the upper three bits of 32 address bits are used as shadow bits (see section 7, Bus State Controller (BSC)), therefore, the upper three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset. The tag address is not initialized by a power-on reset.

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

LRU: With the 4-way set associative system, up to four instructions or data with the same entry address can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is registered in. There are six LRU bits, controlled by hardware. The least-recently-used (LRU) algorithm is used to select the way.

When a cache miss occurs, six LRU bits indicate the way to be replaced. If a bit pattern other than those listed in table 3.1 is set in the LRU bits by software, the cache will not function correctly. When changing the LRU bits by software, set one of the patterns listed in table 3.1.

The LRU bits are initialized to 000000 by a power-on reset.

Table 3.1 LRU and Way to be Replaced

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

3.1.2 Divided Areas and Cache

A 4-G byte address space is divided into five areas with the architecture of this LSI. The cache access methods can be specified for each area. Table 3.2 lists the correspondence between the divided areas and cache.

Table 3.2 Correspondence between Divided Areas and Cache

Address	Area	Cacheable	Cache Operating Control
H'00000000 to H'7FFFFFFF	P0	Cacheable	WT bit in CCR1
H'80000000 to H'9FFFFFFF	P1	Cacheable	CB bit in CCR1
H'A0000000 to H'BFFFFFFF	P2	Non cacheable	—
H'C0000000 to H'DFFFFFFF	P3	Cacheable	WT bit in CCR1
H'E0000000 to H'FFFFFFFF	P4	Non cacheable (internal I/O)	—

3.2 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)

3.2.1 Cache Control Register 1 (CCR1)

The cache is enabled or disabled by the CE bit in CCR1. CCR1 also has the CF bit (which invalidates all cache entries), and the WT and CB bits (which select either write-through mode or write-back mode). Programs that change the contents of CCR1 should be placed in the 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 meaning that it 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 operating mode for H'80000000 to H'9FFFFFFF. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write-Through Indicates the cache operating mode for H'00000000 to H'7FFFFFFF and H'C0000000 to H'DFFFFFFF. 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 or not the cache function is used. 0: Cache function is not used. 1: Cache function is used.

3.3 Operation

3.3.1 Searching Cache

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

Entries are selected using bits 11 to 4 of the memory access address and the tag address of that entry is read. The address comparison is performed on 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 3.2 shows a hit on way 1.

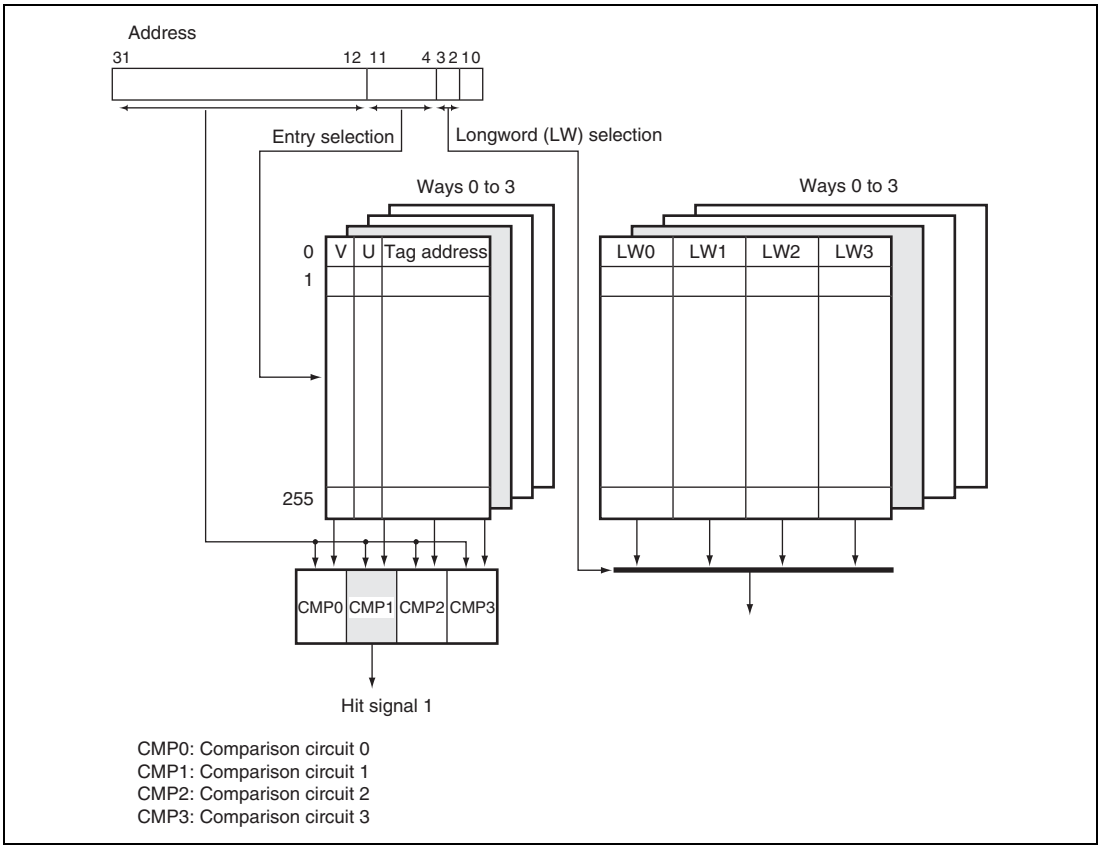


Figure 3.2 Cache Search Scheme

3.3.2 Read Access

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

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

3.3.3 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 generated. The U bit of the entry that has been written to is set to 1, and the LRU bits are 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 generated. The U bit of the entry that has been written to is not updated, and the LRU bits are 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 3.1. 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 bits are 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.

3.3.4 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry must be 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 3.3 shows the configuration of the write-back buffer.

PA (31 to 4)	Longword 0	Longword 1	Longword 2	Longword 3

PA (31 to 4): Physical address to be written to external memory
Longword 0 to 3: One line of cache data to be written to external memory

Figure 3.3 Write-Back Buffer Configuration

3.3.5 Coherency of Cache and External Memory

Coherency between the cache and the external memory must be ensured by software. When memory shared by this LSI and another device is allocated to a cacheable address space, invalidate and write back the cache by accessing the memory-mapped cache, as required. Memory that is shared by the CPU, DMAC, and E-DMAC of this LSI should also be handled in this way.

3.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read from or written to by the MOV instructions. The address array is allocated to addresses H'F0000000 to H'F0FFFFFF, and the data array to addresses H'F1000000 to H'F1FFFFFF. The address array and data array must be accessed in longwords, and instruction fetches cannot be performed.

3.4.1 Address Array

The address array is allocated to 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. Always clear the upper three bits (bits 31 to 29) of the tag address to 0. Figure 3.4 shows 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 field set to 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. 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 has 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 had a hit is 1 at this time, 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.

3.4.2 Data Array

The data array is allocated to 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.

Figure 3.4 shows 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 field, from the entry that corresponds to the entry address and the way that is specified by the address field.

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

(1) Address array access

(a) Address specification

Read access

31 – 24	23 – 14	13	12 – 11	4	3	2	1	0
1111 0000	*--------*	W	Entry address		0	*	0	0

Write access

31 – 24	23 – 14	13	12 – 11	4	3	2	1	0
1111 0000	*--------*	W	Entry address		A	*	0	0

(b) Data specification (both read and write accesses)

31	30	29	28 – 10	9 – 4	3	2	1	0
0	0	0	Tag address (28 to 10)	LRU	X	X	U	V

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

(a) Address specification

31 – 24	23 – 14	13	12 – 11	4	3	2	1	0
1111 0001	*--------*	W	Entry address		L		0	0

(b) Data specification

31 – 0
Longword

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

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

3.4.3 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 the V bit and U bit specified by the write data are written when a match is found. If no match is found, there is no operation. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1. 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 Data of Specific Entry: The data section of a specific entry can be read from by the memory-mapped cache access. The longword indicated in the data field of the data array in figure 3.4 is read into the register. In the example shown below, R0 specifies the address and R1 shows what is read.

```
; R0=H'F100004C; data array access, entry=B'00000100
; Way = 0, longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 4 U Memory

This LSI has on-chip U memory which can be used to store instructions and data.

4.1 Features

Features of the U Memory are shown below.

- Size
  16 kbytes
- Address
  H'E55F_C000 to H'E55F_FFFF
- Priority
  The U memory can be accessed from the I bus by the DMAC and E-DMAC and from the L bus by the CPU. In the event of simultaneous accesses from different buses, the accesses are processed according to the priority. The priority is: I bus > L bus.

4.2 Usage Notes

In sleep mode, the U memory cannot be accessed by the DMAC and E-DMAC.

Section 5 Exception Handling

5.1 Overview

5.1.1 Types of Exception Handling and Priority

Exception handling is started by four sources: resets, address errors, interrupts and instructions and have the priority, as shown in table 5.1. When several exceptions are detected at once, they are processed according to the priority.

Table 5.1 Types of Exceptions and Priority

Exception	Exception Source		Priority
Reset	Power-on reset		High
	H-UDI reset		↑
Interrupt	User break (break before instruction execution)		
Address error	CPU address error (instruction fetch)		
Instruction	General illegal instructions (undefined code)		
	Illegal slot instruction (undefined code placed immediately after a delayed branch instruction*[1] or instruction that changes the PC value*[2])		
	Trap instruction (TRAPA instruction)		
Address error	CPU address error (data access)		
Interrupt	User break (break after instruction execution or operand break)		
	NMI		
	H-UDI		
	IRQ		
	On-chip peripheral modules	Watchdog timer (WDT)	
		Ether controller (EtherC and E-DMAC)	
		Compare match timer 0 and 1 (CMT0 and CMT1)	
		Serial communication interface with FIFO (SCIF0, SCIF1, and SCIF2)	↓
		Host interface (HIF)	Low

Exception	Exception Source		Priority
Interrupt	On-chip peripheral modules	Direct memory access controller (DMAC0, DMAC1 DMAC2, and DMAC3)	High ↕
		Serial I/O with FIFO (SIOF)	Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.
2. Instructions that change the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR.

5.1.2 Exception Handling Operations

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

Table 5.2 Timing for Exception Detection and Start of Exception Handling

Exception		Timing of Source Detection and Start of Exception Handling
Reset	Power-on reset	Started when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows.
	H-UDI reset	Started when the reset assert command and the reset negate command are input to the H-UDI in this order.
Address error		Detected during the instruction decode stage and started after the execution of the current instruction is completed.
Interrupt		
Instruction	Trap instruction	Started by the execution of the TRAPA instruction.
	General illegal instructions	Started when an undefined code placed at other than a delay slot (immediately after a delayed branch instruction) is decoded.
	Illegal slot instructions	Started when an undefined code placed at a delay slot (immediately after a delayed branch instruction) or an instruction that changes the PC value is detected.

When exception handling starts, the CPU operates

Exception Handling Triggered by Reset: The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC from the address H'A0000000 and SP from the address H'A0000004). For details, see section 5.1.3, Exception Handling Vector Table. H'00000000 is then written to the vector base register (VBR), and H'F (B'1111) is written to the interrupt mask bits (I3 to I0) in the status register (SR). The program starts from the PC address fetched from the exception handling vector table.

Exception Handling Triggered by Address Error, Interrupt, and Instruction: SR and PC are saved to the stack indicated by R15. For interrupt exception handling, the interrupt priority level is written to the interrupt mask bits (I3 to I0) in SR. For address error and instruction exception handling, bits I3 to I0 are not affected. The start address is then fetched from the exception handling vector table and the program starts from that address.

5.1.3 Exception Handling Vector Table

Before exception handling starts, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception handling routines. (The reset exception handling table holds the initial values of PC and SP.)

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

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

Table 5.3 Vector Numbers and Vector Table Address Offsets

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

Exception Handling Source		Vector Number	Vector Table Address Offset
(Reserved by system)		14	H'00000038 to H'0000003B
		:	:
		31	H'0000007C to H'0000007F
Trap instruction (user vector)		32	H'00000080 to H'00000083
		:	:
		63	H'000000FC to H'000000FF
Interrupt	IRQ0	64	H'00000100 to H'00000103
	IRQ1	65	H'00000104 to H'00000107
	IRQ2	66	H'00000108 to H'0000010B
	IRQ3	67	H'0000010C to H'0000010F
	(Reserved by system)	68	H'00000110 to H'00000113
		:	:
		79	H'0000013C to H'0000013F
	IRQ4	80	H'00000140 to H'00000143
	IRQ5	81	H'00000144 to H'00000147
	IRQ6	82	H'00000148 to H'0000014B
	IRQ7	83	H'0000014C to H'0000014F
On-chip peripheral module*		84	H'00000120 to H'00000124
		:	:
		255	H'000003FC to H'000003FF

Note: * For details on the vector numbers and vector table address offsets of on-chip peripheral module interrupts, see table 6.2, Interrupt Exception Handling Vectors and Priorities in section 6, Interrupt Controller (INTC).

Table 5.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = H'A0000000 + (vector table address offset) = H'A0000000 + (vector number) × 4
Address errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. VBR: Vector base register
2. Vector table address offset: See table 5.3.
3. Vector number: See table 5.3.

5.2 Resets

5.2.1 Types of Resets

Resets have priority over any exception source. As table 5.5 shows, a power-on reset initializes all modules in this LSI.

Table 5.5 Reset Status

	Conditions for Transition to Reset State			Internal State		
Type	**$\overline{\text{RES}}$**	**WDT Overflow**	**H-UDI Command**	**CPU, INTC**	**On-Chip Peripheral Module**	**PFC, I/O Port**
Power-on reset	Low	—	—	Initialized	Initialized	Initialized
	High	Overflow	—	Initialized	Initialized	Initialized
H-UDI reset	High	Not overflowed	Reset assert command	Initialized	Initialized	Initialized

5.2.2 Power-On Reset

Power-On Reset by $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept low for at least the oscillation settling time when applying the power or when in standby mode (when the clock is halted) or at least 20 tcyc when the clock is operating. During the power-on reset state, CPU internal states and all registers of on-chip peripheral modules are initialized.

In the power-on reset state, power-on reset exception handling starts when driving the $\overline{\text{RES}}$ pin high after driving the pin low for the given time. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

Be certain to always perform power-on reset exception handling when turning the system power on.

Power-On Reset by WDT: When TCNT of the WDT overflows while a setting is made so that a power-on reset can be generated in watchdog timer mode of the WDT, this LSI enters the power-on reset state.

If a reset caused by the signal input on the $\overline{\text{RES}}$ pin and a reset caused by a WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When the power-on reset exception handling caused by the WDT is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in the PC and SP, then the program starts.

5.2.3 H-UDI Reset

The H-UDI reset is generated by issuing the H-UDI reset assert command. The CPU operation is described below. For details, see section 21, User Debugging Interface (H-UDI).

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3 to I0) in the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur when instructions are fetched or data is read from or written to, as shown in table 5.6.

Table 5.6 Bus Cycles and Address Errors

Bus Cycle			
Type	**Bus Master**	**Bus Cycle Description**	**Address Errors**
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
Data read/write	CPU	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs

5.3.2 Address Error Exception Source

When an address error exception is generated, the bus cycle which caused the address error ends, the current instruction finishes, and then the address error exception handling starts. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value to be saved is the start address of the instruction which caused an address error exception. When the instruction that caused the exception is placed in the delay slot, the address of the delayed branch instruction which is placed immediately before the delay slot.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the generated address error, and the program starts executing from that address. This branch is not a delayed branch.

5.4 Interrupts

5.4.1 Interrupt Sources

Table 5.7 shows the sources that start the interrupt exception handling. They are NMI, user break, H-UDI, IRQ, and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debug interface (H-UDI)	1
IRQ	IRQ0 to IRQ7 pins (external input)	8
On-chip peripheral module	Watchdog timer (WDT)	1
	Ether controller (EtherC and E-DMAC)	1
	Compare match timer (CMT0 and CMT1)	2
	Serial communication interface with FIFO (SCIF0, SCIF1, and SCIF2)	12
	Host interface (HIF)	2
	Direct memory access controller (DMAC0, DMAC1, DMAC2, and DMAC3)	4
	Serial I/O with FIFO (SIOF)	1

All interrupt sources are given different vector numbers and vector table address offsets. For details on vector numbers and vector table address offsets, see table 6.2, Interrupt Exception Handling Vectors and Priorities in section 6, Interrupt Controller (INTC).

5.4.2 Interrupt Priority

The interrupt priority is predetermined. When multiple interrupts occur simultaneously (overlapped interruptions), the interrupt controller (INTC) determines their relative priorities and starts the exception handling according to the results.

The priority of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The priority level of the user break interrupt and H-UDI is 15. IRQ interrupt and on-chip peripheral module interrupt priority levels can be set freely using the interrupt priority level setting registers A to G (IPRA to IPRG) of the INTC as shown in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. For details on IPRA to IPRG, see section 6.3.4, Interrupt Priority Registers A to G (IPRA to IPRG).

Table 5.8 Interrupt Priority

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level. Can be masked.
H-UDI	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority level setting registers A through G (IPRA to IPRG).
On-chip peripheral module		

5.4.3 Interrupt Exception Handling

When an interrupt occurs, the interrupt controller (INTC) ascertains its priority level. NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, exception handling begins. In interrupt exception handling, the CPU saves SR and the program counter (PC) to the stack. The priority level of the accepted interrupt is written to bits I3 to I0 in SR. Although the priority level of the NMI is 16, the value set in bits I3 to I0 is H'F (level 15). Next, the start address of the exception handling routine is fetched from the exception handling vector table for the accepted interrupt, and program execution branches to that address and the program starts. For details on the interrupt exception handling, see section 6.6, Interrupt Operation.

5.5 Exceptions Triggered by Instructions

5.5.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by the trap instruction, illegal slot instructions, and general illegal instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	—
Illegal slot instructions*	Undefined code placed immediately after a delayed branch instruction (delay slot) or instructions that changes the PC value	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that changes the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR
General illegal instructions*	Undefined code anywhere besides in a delay slot	—

Note: * The operation is not guaranteed when undefined instructions other than H'FC00 to H'FFFF are decoded.

5.5.2 Trap Instructions

When a TRAPA instruction is executed, the trap instruction exception handling starts. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
3. The CPU reads the start address of the exception handling routine from the exception handling vector table that corresponds to the vector number specified in the TRAPA instruction, program execution branches to that address, and then the program starts. This branch is not a delayed branch.

5.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is called “instruction placed in a delay slot”. When the instruction placed in the delay slot is an undefined code, illegal slot exception handling starts after the undefined code is decoded. Illegal slot exception handling also starts when an instruction that changes the program counter (PC) value is placed in a delay slot and the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the target address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the exception that occurred. Program execution branches to that address and the program starts. This branch is not a delayed branch.

5.5.4 General Illegal Instructions

When an undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception handling starts. The CPU handles the general illegal instructions in the same procedures as in the illegal slot instructions. Unlike processing of illegal slot instructions, however, the program counter value that is stacked is the start address of the undefined code.

5.6 Cases when Exceptions are Accepted

When an exception other than resets occurs during decoding the instruction placed in a delay slot or immediately after an interrupt disabled instruction, it may not be accepted and be held shown in table 5.10. In this case, when an instruction which accepts an interrupt request is decoded, the exception is accepted.

Table 5.10 Delay Slot Instructions, Interrupt Disabled Instructions, and Exceptions

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

[Legend]

√: Accepted

×: Not accepted

—: Does not occur

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

2. An exception is accepted before the execution of a delayed branch instruction. However, when an address error or a slot illegal instruction exception occurs in the delay slot of the RTE instruction, correct operation is not guaranteed.

3. An exception is accepted after a delayed branch (between instructions in the delay slot and the branch destination).

4. An exception is accepted after the execution of the next instruction of an interrupt disabled instruction (before the execution two instructions after an interrupt disabled instruction).

5.7 Stack States after Exception Handling Ends

The stack states after exception handling ends are shown in table 5.11.

Table 5.11 Stack Status after Exception Handling Ends

Types	Stack State
Address error (when the instruction that caused an exception is placed in the delay slot)	SP → Address of delayed branch instruction (32 bits) SR (32 bits)
Address error (other than above)	SP → Address of instruction that caused exception (32 bits) SR (32 bits)
Interrupt	SP → Address of instruction after executed instruction (32 bits) SR (32 bits)
Trap instruction	SP → Address of instruction after TRAPA instruction (32 bits) SR (32 bits)

Types	Stack State
Illegal slot instruction	SP → Address of delayed branch instruction (32 bits); SR (32 bits)
General illegal instruction	SP → Address of general illegal instruction (32 bits); SR (32 bits)

5.8 Usage Notes

5.8.1 Value of Stack Pointer (SP)

The SP value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

5.8.2 Value of Vector Base Register (VBR)

The VBR value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

5.8.3 Address Errors Caused by Stacking for Address Error Exception Handling

When the SP value is not a multiple of 4, an address error will occur when stacking for exception handling (interrupts, etc.) and address error exception handling will start after the first exception handling is ended. Address errors will also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be passed to the handling routine for address error exception and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. When stacking the SR and PC values, the SP values for both are subtracted by 4, therefore, the SP value is still not a multiple of 4 after the stacking. The address value output during stacking is the SP value whose lower two bits are cleared to 0. So the write data stacked is undefined.

5.8.4 Notes on Slot Illegal Instruction Exception Handling

Some specifications on slot illegal instruction exception handling in this LSI differ from those of the conventional SH2.

- Conventional SH2: Instructions LDC Rm,SR and LDC.L @Rm+,SR are not subject to the slot illegal instructions.
- This LSI: Instructions LDC Rm,SR and LDC.L @Rm+,SR are subject to the slot illegal instructions.

The supporting status on our software products regarding this note is as follows:

Compiler

This instruction is not allocated in the delay slot in the compiler V.4 and its subsequent versions.

Real-time OS for μITRON specifications

1. HI7000/4, HI-SH7

   This instruction does not exist in the delay slot within the OS.

2. HI7000

   This instruction is in part allocated to the delay slot within the OS, which may cause the slot illegal instruction exception handling in this LSI.

3. Others

   The slot illegal instruction exception handling may be generated in this LSI in a case where the instruction is described in assembler or when the middleware of the object is introduced.

   Note that a check-up program (checker) to pick up this instruction is available on our website. Download and utilize this checker as needed.

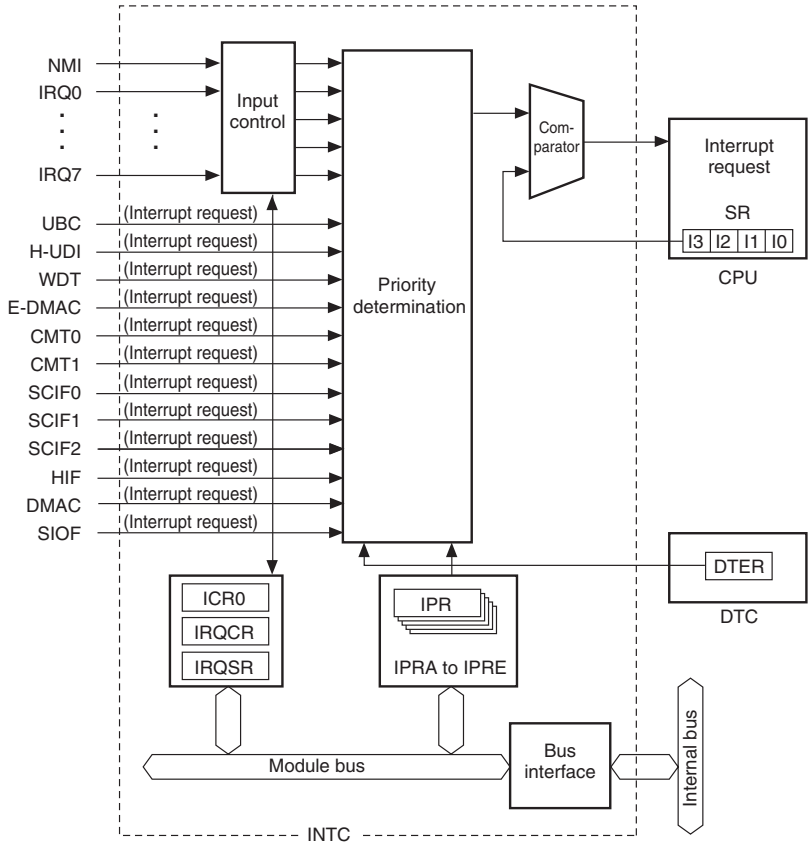
Section 6 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU.

6.1 Features

- 16 levels of interrupt priority

Figure 6.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller	HIF:	Host interface
H-UDI:	User debugging interface	DMAC:	Direct memory access controller
WDT:	Watchdog timer	ICR0:	Interrupt control register 0
E-DMAC:	DMAC for Ethernet controller	IRQCR:	IRQ control register
CMT:	Compare match timer	IRQSR:	IRQ status register
SCIF:	Serial communications interface with FIFO	IPRA to IPRG:	Interrupt priority registers A to G
		SR:	Status register

Figure 6.1 INTC Block Diagram

6.2 Input/Output Pins

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbr.	I/O	Function
Non-maskable interrupt input pin	NMI	Input	Input of non-maskable interrupt request signal
Interrupt request input pins	IRQ0 to IRQ7	Input	Input of maskable interrupt request signals

6.3 Register Descriptions

The interrupt controller has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Interrupt control register 0 (ICR0)
- IRQ control register (IRQCR)
- IRQ status register (IRQSR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)

6.3.1 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and indicates the input signal level on the NMI pin.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	1/0	R	NMI Input Level Indicates the state of the signal input to the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: State of the NMI input is low 1: State of the NMI input is high
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select 0: Interrupt request is detected on the falling edge of the NMI input 1: Interrupt request is detected on the rising edge of the NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3.2 IRQ Control Register (IRQCR)

IRQCR is a 16-bit register that sets the input signal detection mode of the external interrupt input pins IRQ0 to IRQ7.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ7 Sense Select
14	IRQ70S	0	R/W	Set the interrupt request detection mode for pin IRQ7. 00: Interrupt request is detected at the low level of pin IRQ7 01: Interrupt request is detected at the falling edge of pin IRQ7 10: Interrupt request is detected at the rising edge of pin IRQ7 11: Interrupt request is detected at both the falling and rising edges of pin IRQ7
13	IRQ61S	0	R/W	IRQ6 Sense Select
12	IRQ60S	0	R/W	Set the interrupt request detection mode for pin IRQ6. 00: Interrupt request is detected at the low level of pin IRQ6 01: Interrupt request is detected at the falling edge of pin IRQ6 10: Interrupt request is detected at the rising edge of pin IRQ6 11: Interrupt request is detected at both the falling and rising edges of pin IRQ6

Bit	Bit Name	Initial Value	R/W	Description
11	IRQ51S	0	R/W	IRQ5 Sense Select
10	IRQ50S	0	R/W	Set the interrupt request detection mode for pin IRQ5. 00: Interrupt request is detected at the low level of pin IRQ5 01: Interrupt request is detected at the falling edge of pin IRQ5 10: Interrupt request is detected at the rising edge of pin IRQ5 11: Interrupt request is detected at both the falling and rising edges of pin IRQ5
9	IRQ41S	0	R/W	IRQ4 Sense Select
8	IRQ40S	0	R/W	Set the interrupt request detection mode for pin IRQ4. 00: Interrupt request is detected at the low level of pin IRQ4 01: Interrupt request is detected at the falling edge of pin IRQ4 10: Interrupt request is detected at the rising edge of pin IRQ4 11: Interrupt request is detected at both the falling and rising edges of pin IRQ4
7	IRQ31S	0	R/W	IRQ3 Sense Select
6	IRQ30S	0	R/W	Set the interrupt request detection mode for pin IRQ3. 00: Interrupt request is detected at the low level of pin IRQ3 01: Interrupt request is detected at the falling edge of pin IRQ3 10: Interrupt request is detected at the rising edge of pin IRQ3 11: Interrupt request is detected at both the falling and rising edges of pin IRQ3

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ21S	0	R/W	IRQ2 Sense Select
4	IRQ20S	0	R/W	Set the interrupt request detection mode for pin IRQ2. 00: Interrupt request is detected at the low level of pin IRQ2 01: Interrupt request is detected at the falling edge of pin IRQ2 10: Interrupt request is detected at the rising edge of pin IRQ2 11: Interrupt request is detected at both the falling and rising edges of pin IRQ2
3	IRQ11S	0	R/W	IRQ1 Sense Select
2	IRQ10S	0	R/W	Set the interrupt request detection mode for pin IRQ1. 00: Interrupt request is detected at the low level of pin IRQ1 01: Interrupt request is detected at the falling edge of pin IRQ1 10: Interrupt request is detected at the rising edge of pin IRQ1 11: Interrupt request is detected at both the falling and rising edges of pin IRQ1
1	IRQ01S	0	R/W	IRQ0 Sense Select
0	IRQ00S	0	R/W	Set the interrupt request detection mode for pin IRQ0. 00: Interrupt request is detected at the low level of pin IRQ0 01: Interrupt request is detected at the falling edge of pin IRQ0 10: Interrupt request is detected at the rising edge of pin IRQ0 11: Interrupt request is detected at both the falling and rising edges of pin IRQ0

6.3.3 IRQ Status register (IRQSR)

IRQSR is a 16-bit register that indicates the states of the external interrupt input pins IRQ0 to IRQ7 and the status of interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7L	0/1	R	Indicates the state of pin IRQ7. 0: State of pin IRQ7 is low 1: State of pin IRQ7 is high
14	IRQ6L	0/1	R	Indicates the state of pin IRQ6. 0: State of pin IRQ6 is low 1: State of pin IRQ6 is high
13	IRQ5L	0/1	R	Indicates the state of pin IRQ5. 0: State of pin IRQ5 is low 1: State of pin IRQ5 is high
12	IRQ4L	0 or 1	R	Indicates the state of pin IRQ4. 0: State of pin IRQ4 is low 1: State of pin IRQ4 is high
11	IRQ3L	0 or 1	R	Indicates the state of pin IRQ3. 0: State of pin IRQ3 is low 1: State of pin IRQ3 is high
10	IRQ2L	0 or 1	R	Indicates the state of pin IRQ2. 0: State of pin IRQ2 is low 1: State of pin IRQ2 is high
9	IRQ1L	0 or 1	R	Indicates the state of pin IRQ1. 0: State of pin IRQ1 is low 1: State of pin IRQ1 is high
8	IRQ0L	0 or 1	R	Indicates the state of pin IRQ0. 0: State of pin IRQ0 is low 1: State of pin IRQ0 is high

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/W	Indicates the status of an IRQ7 interrupt request. • When level detection mode is selected 0: An IRQ7 interrupt has not been detected [Clearing condition] Driving pin IRQ7 high 1: An IRQ7 interrupt has been detected [Setting condition] Driving pin IRQ7 low • When edge detection mode is selected 0: An IRQ7 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ7F = 1 — Accepting an IRQ7 interrupt 1: An IRQ7 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ7
6	IRQ6F	0	R/W	Indicates the status of an IRQ6 interrupt request. • When level detection mode is selected 0: An IRQ6 interrupt has not been detected [Clearing condition] Driving pin IRQ6 high 1: An IRQ6 interrupt has been detected [Setting condition] Driving pin IRQ6 low • When edge detection mode is selected 0: An IRQ6 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ6F = 1 — Accepting an IRQ6 interrupt 1: An IRQ6 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ6

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ5F	0	R/W	Indicates the status of an IRQ5 interrupt request. • When level detection mode is selected 0: An IRQ5 interrupt has not been detected [Clearing condition] Driving pin IRQ5 high 1: An IRQ5 interrupt has been detected [Setting condition] Driving pin IRQ5 low • When edge detection mode is selected 0: An IRQ5 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ5F = 1 — Accepting an IRQ5 interrupt 1: An IRQ5 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ5
4	IRQ4F	0	R/W	Indicates the status of an IRQ4 interrupt request. • When level detection mode is selected 0: An IRQ4 interrupt has not been detected [Clearing condition] Driving pin IRQ4 high 1: An IRQ4 interrupt has been detected [Setting condition] Driving pin IRQ4 low • When edge detection mode is selected 0: An IRQ4 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ4F = 1 — Accepting an IRQ4 interrupt 1: An IRQ4 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ4

Bit	Bit Name	Initial Value	R/W	Description
3	IRQ3F	0	R/W	Indicates the status of an IRQ3 interrupt request. • When level detection mode is selected 0: An IRQ3 interrupt has not been detected [Clearing condition] Driving pin IRQ3 high 1: An IRQ3 interrupt has been detected [Setting condition] Driving pin IRQ3 low • When edge detection mode is selected 0: An IRQ3 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ3F = 1 — Accepting an IRQ3 interrupt 1: An IRQ3 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ3
2	IRQ2F	0	R/W	Indicates the status of an IRQ2 interrupt request. • When level detection mode is selected 0: An IRQ2 interrupt has not been detected [Clearing condition] Driving pin IRQ2 high 1: An IRQ2 interrupt has been detected [Setting condition] Driving pin IRQ2 low • When edge detection mode is selected 0: An IRQ2 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ2F = 1 — Accepting an IRQ2 interrupt 1: An IRQ2 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ2

Bit	Bit Name	Initial Value	R/W	Description
1	IRQ1F	0	R/W	Indicates the status of an IRQ1 interrupt request. • When level detection mode is selected 0: An IRQ1 interrupt has not been detected [Clearing condition] Driving pin IRQ1 high 1: An IRQ1 interrupt has been detected [Setting condition] Driving pin IRQ1 low • When edge detection mode is selected 0: An IRQ1 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ1F = 1 — Accepting an IRQ1 interrupt 1: An IRQ1 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ1
0	IRQ0F	0	R/W	Indicates the status of an IRQ0 interrupt request. • When level detection mode is selected 0: An IRQ0 interrupt has not been detected [Clearing condition] Driving pin IRQ0 high 1: An IRQ0 interrupt has been detected [Setting condition] Driving pin IRQ0 low • When edge detection mode is selected 0: An IRQ0 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ0F = 1 — Accepting an IRQ0 interrupt 1: An IRQ0 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ0

6.3.4 Interrupt Priority Registers A to G (IPRA to IPRG)

Interrupt priority registers are seven 16-bit readable/writable registers that set priority levels from 0 to 15 for interrupts except NMI. For the correspondence between interrupt request sources and IPR, refer to table 6.2, Interrupt Exception Handling Vectors and Priorities. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 to H'F in each of the four-bit groups 15 to 12, 11 to 8, 7 to 4 and 3 to 0. Reserved bits that are not assigned should be set H'0 (B'0000).

Bit	Bit Name	Initial Value	R/W	Description
15	IPR15	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
14	IPR14	0	R/W	
13	IPR13	0	R/W	
12	IPR12	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
11	IPR11	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
10	IPR10	0	R/W	
9	IPR9	0	R/W	
8	IPR8	0	R/W	
7	IPR7	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
6	IPR6	0	R/W	
5	IPR5	0	R/W	
4	IPR4	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	IPR3	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
2	IPR2	0	R/W	
1	IPR1	0	R/W	
0	IPR0	0	R/W	

Note: Name in the tables above is represented by a general name. Name in the list of register is, on the other hand, represented by a module name.

6.4 Interrupt Sources

6.4.1 External Interrupts

There are five types of interrupt sources: User break, NMI, H-UDI, IRQ, and on-chip peripheral modules. Individual interrupts are given priority levels (0 to 16, with 0 the lowest and 15 the highest). Giving an interrupt a priority level of 0 masks it.

NMI Interrupt: The NMI interrupt is given a priority level of 16 and is always accepted. An NMI interrupt is detected at the edge of the pins. Use the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) to select either the rising or falling edge. In the NMI interrupt exception handler, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 15.

IRQ7 to IRQ0 Interrupts: IRQ interrupts are requested by input from pins IRQ0 to IRQ7. Use the IRQ sense select bits (IRQ71S to IRQ 01S and IRQ70S to IRQ00S) in the IRQ control register (IRQCR) to select the detection mode from low level detection, falling edge detection, rising edge detection, and both edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority registers A and B (IPRA and IPRB).

In the case that the low level detection is selected, an interrupt request signal is sent to the INTC while the IRQ pin is driven low. The interrupt request signal stops to be sent to the INTC when the IRQ pin becomes high. It is possible to confirm that an interrupt is requested by reading the IRQ flags (IRQ7F to IRQ0F) in the IRQ status register (IRQSR).

In the case that the edge detection is selected, an interrupt request signal is sent to the INTC when the following change on the IRQ pin is detected: from high to low in falling edge detection mode, from low to high in rising edge detection mode, and from low to high or from high to low in both edge detection mode. The IRQ interrupt request by detecting the change on the pin is held until the interrupt request is accepted. It is possible to confirm that an IRQ interrupt request has been detected by reading the IRQ flags (IRQ7F to IRQ0F) in the IRQ status register (IRQSR). An IRQ interrupt request by detecting the change on the pin can be withdrawn by writing 0 to an IRQ flag after reading 1.

In the IRQ interrupt exception handling, the interrupt mask bits (I3 to I0) in the status register (SR) are set to the priority level value of the accepted IRQ interrupt. Figure 6.2 shows the block diagram of the IRQ7 to IRQ0 interrupts.

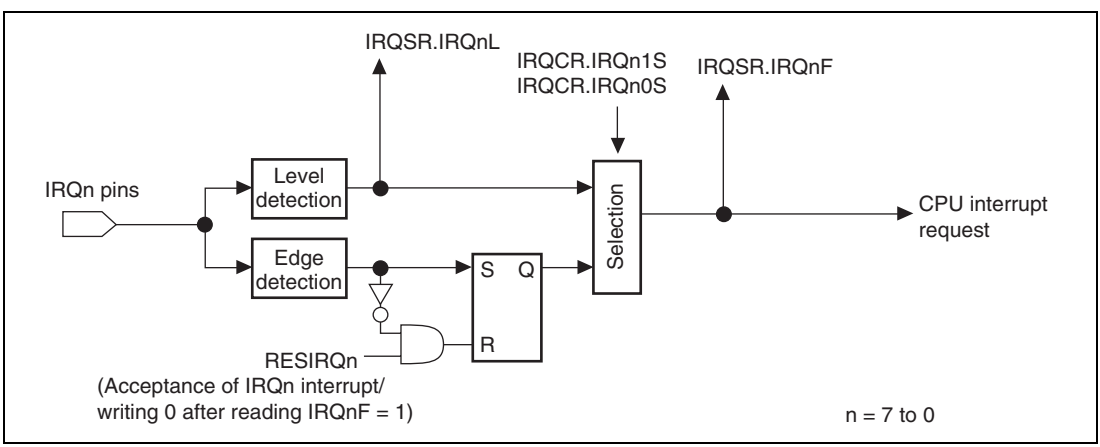


Figure 6.2 Block Diagram of IRQ7 to IRQ0 Interrupts Control

6.4.2 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are interrupts generated by the following on-chip peripheral modules.

Since a different interrupt vector is allocated to each interrupt source, the exception handling routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be allocated to individual on-chip peripheral modules in interrupt priority registers C to G (IPRC to IPRG). On-chip peripheral module interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

6.4.3 User Break Interrupt

A user break interrupt has a priority level of 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15. For more details on the user break interrupt, see section 20, User Break Controller (UBC).

6.4.4 H-UDI Interrupt

User debugging interface (H-UDI) interrupt has a priority level of 15, and occurs when an H-UDI interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and are held until accepted. H-UDI exception handling sets the interrupt mask level bits (I3-I0) in the status register (SR) to level 15. For more details on the H-UDI interrupt, see section 21, User Debugging Interface (H-UDI).

6.5 Interrupt Exception Handling Vector Table

Table 6.2 lists interrupt sources, their vector numbers, vector table address offsets, and interrupt priorities.

Individual interrupt sources are allocated to different vector numbers and vector table address offsets. Vector table addresses are calculated from the vector numbers and vector table address offsets. For interrupt exception handling, the start address of the exception handling routine is fetched from the vector table address in the vector table. For the details on calculation of vector table addresses, see table 5.4, Calculating Exception Handling Vector Table Addresses in section 5, Exception Handling.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A to G (IPRA to IPRG). However, when interrupt sources whose priority levels are allocated with the same IPR are requested, the interrupt of the smaller vector number has priority. This priority cannot be changed. Priority levels of IRQ interrupts and on-chip peripheral module interrupts are initialized to level 0 at a power-on reset. If the same priority level is allocated to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priority order shown in table 6.2.

Table 6.2 Interrupt Exception Handling Vectors and Priorities

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
User break		12	H'00000030	—	High
External pin	NMI	11	H'0000002C	—	↑
H-UDI		13	H'00000034	—	
External pin	IRQ0	64	H'00000100	IPRA15 to IPRA12	
	IRQ1	65	H'00000104	IPRA11 to IPRA8	
	IRQ2	66	H'00000108	IPRA7 to IPRA4	
	IRQ3	67	H'0000010C	IPRA3 to IPRA0	
	IRQ4	80	H'00000140	IPRB15 to IPRB12	
	IRQ5	81	H'00000144	IPRB11 to IPRB8	
	IRQ6	82	H'00000148	IPRB7 to IPRB4	↓
	IRQ7	83	H'0000014C	IPRB3 to IPRB0	Low

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
WDT	ITI	84	H'00000150	IPRC15 to IPRC12	Hight
E-DMAC	EINT0	85	H'00000154	IPRC11 to IPRC8	↑
CMT channel 0	CMI0	86	H'00000158	IPRC7 to IPRC4	
CMT channel 1	CMI1	87	H'0000015C	IPRC3 to IPRC0	
SCIF channel 0	ERI_0	88	H'00000160	IPRD15 to IPRD12	
	RXI_0	89	H'00000164		
	BRI_0	90	H'00000168		
	TXI_0	91	H'0000016C		
SCIF channel 1	ERI_1	92	H'00000170	IPRD11 to IPRD8	
	RXI_1	93	H'00000174		
	BRI_1	94	H'00000178		
	TXI_1	95	H'0000017C		
SCIF channel 2	ERI_2	96	H'00000180	IPRD7 to IPRD4	
	RXI_2	97	H'00000184		
	BRI_2	98	H'00000188		
	TXI_2	99	H'0000018C		
HIF	HIFI	100	H'00000190	IPRE15 to IPRE12	
	HIFBI	101	H'00000194	IPRE11 to IPRE8	
DMAC	DEI0	104	H'000001A0	IPRF15 to IPRF12	
	DEI1	105	H'000001A4	IPRF11 to IPRF8	
	DEI2	106	H'000001A8	IPRF7 to IPRF4	
	DEI3	107	H'000001AC	IPRF3 to IPRF0	↓
SIOF	SIOFI	108	H'000001B0	IPRG15 to IPRG12	Low

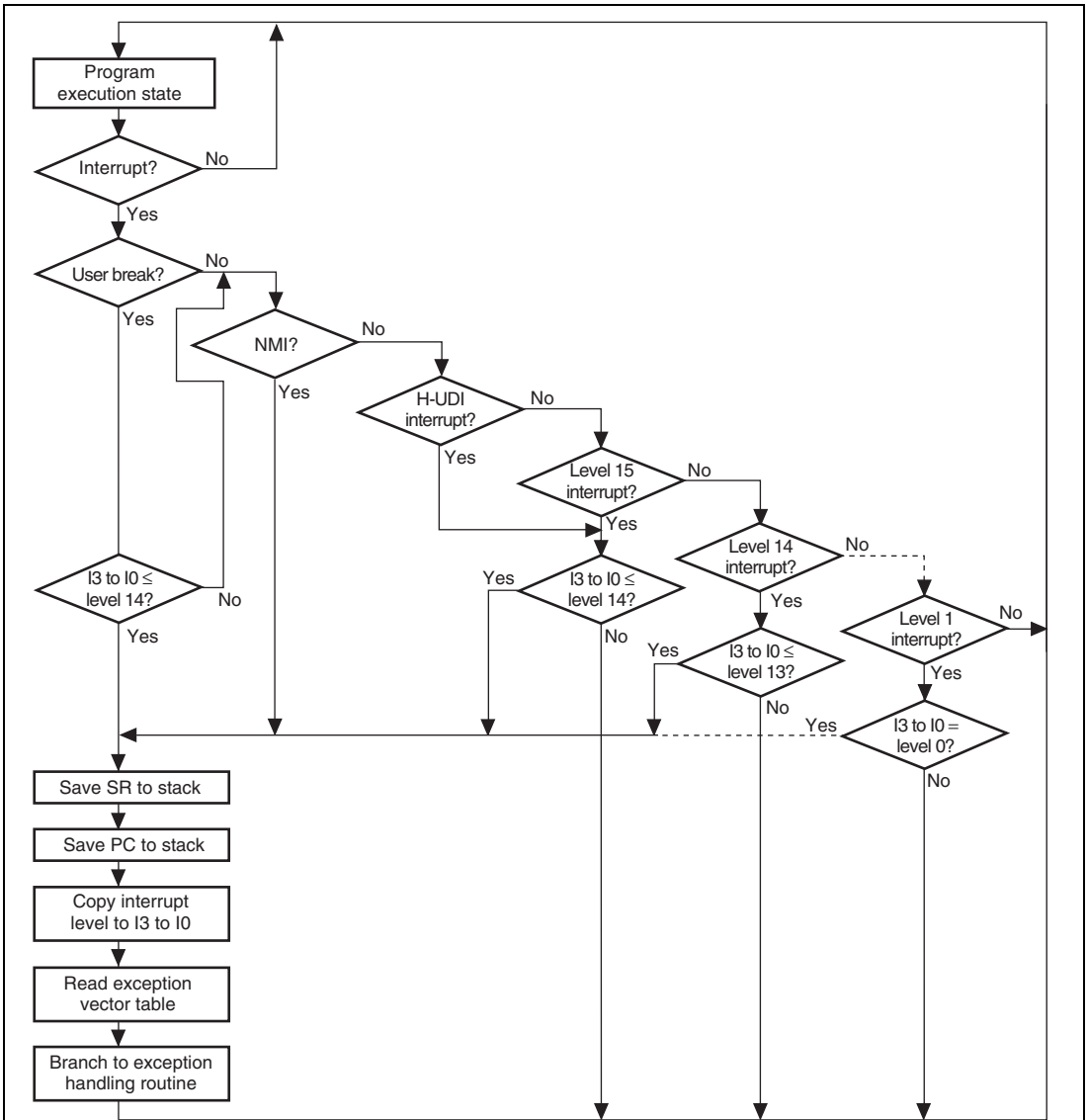
6.6 Interrupt Operation

6.6.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 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 interrupt requests sent, according to the priority levels set in interrupt priority level setting registers A to G (IPRA to IPRG). Interrupts that have lower-priority than that of the selected interrupt are ignored*. If interrupts that have the same priority level or interrupts within a same module occur simultaneously, the interrupt with the highest priority is selected according to the priority shown in table 6.2.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the priority level of the selected request is equal to or less than the level set in bits I3 to I0, the request is ignored. If the priority level of the selected request is higher than the level in bits I3 to I0, the interrupt controller accepts the request and sends an interrupt request signal to the CPU.
4. The CPU detects the interrupt request sent from the interrupt controller in the decode stage of an instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling.
5. SR and PC are saved onto the stack.
6. The priority level of the accepted interrupt is copied to bits (I3 to I0) in SR.
7. The CPU reads the start address of the exception handling routine from the exception vector table for the accepted interrupt, branches to that address, and starts executing the program. This branch is not a delayed branch.

Note: * Interrupt requests that are designated as edge-detect type are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ status register (IRQSR). Interrupts held pending due to edge detection are cleared by a power-on reset or an H-UDI reset.

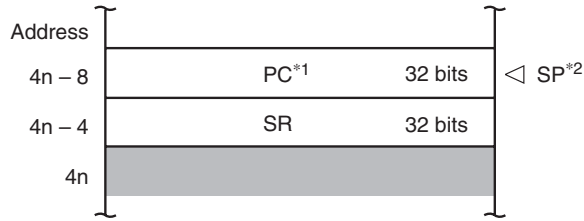


Note: I3 to I0 are Interrupt mask bits in the status register (SR) of the CPU

Figure 6.3 Interrupt Sequence Flowchart

6.6.2 Stack after Interrupt Exception Handling

Figure 6.4 shows the stack after interrupt exception handling.



Notes: 1. PC is the start address of the next instruction (instruction at the return address) after the executed instruction.
2. Always make sure that SP is a multiple of 4

Figure 6.4 Stack after Interrupt Exception Handling

6.7 Interrupt Response Time

Table 6.3 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction of the interrupt handling routine begins.

Table 6.3 Interrupt Response Time

Item		Number of Cycles: NMI, H-UDI	Number of Cycles: IRQ, Peripheral Modules	Remarks
Interrupt priority decision and comparison with mask bits in SR		1 × Icyc + 2 × Pcyc	1 × Icyc + 3 × Pcyc	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)	X (≥ 0)	The longest sequence is for interrupt or address-error exception handling (X = 7 × Icyc + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception handling until fetch of first instruction of exception handling routine starts		8 × Icyc + m1 + m2 + m3	8 × Icyc + m1 + m2 + m3	Performs the saving PC and SR, and vector address fetch.
Interrupt response time	Total:	9 × Icyc + 2 × Pcyc + m1 + m2 + m3 + X	9 × Icyc + 3 × Pcyc + m1 + m2 + m3 + X	
	Minimum*:	12 × Icyc + 2 × Pcyc	12 × Icyc + 3 × Pcyc	SR, PC, and vector table are all in on-chip RAM, or cache hit occurs (in write back mode).
	Maximum:	16 × Icyc + 2 × Pcyc + 2 × (m1 + m2 + m3) + m4	16 × Icyc + 3 × Pcyc + 2 × (m1 + m2 + m3) + m4	

Notes: * In the case that m1 = m2 = m3 = m4 = 1 × Icyc.
m1 to m4 are the number of cycles needed for the following memory accesses.
m1: SR save (longword write)
m2: PC save (longword write)
m3: Vector address read (longword read)
m4: Fetch first instruction of interrupt service routine

Section 7 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.

7.1 Features

The BSC has the following features.

- External address space
  - A maximum 32 or 64 Mbytes for each of the areas, CS0, CS3, CS4, CS5B, and CS6B, totally 256 Mbytes (divided into five areas)
  - A maximum 64 Mbytes for each of the six areas, CS0, CS3, CS4, CS5, and CS6, totally 320 Mbytes (divided into five areas)
  - Can specify the normal space interface, byte-selection SRAM, SDRAM, PCMCIA for each address space
  - Can select the data bus width (8, 16, or 32 bits) for each address space. (The CS0 data bus width can only be selected from 8 or 16 bits.)
  - Can control the insertion of wait cycles for each address space
  - Can control the insertion of wait cycles for each read access and write access
  - Can control the insertion of idle cycles in the consecutive access for five cases independently: read-write (in same space/different space), read-read (in same space/different space), or the first cycle is a write access
- Normal space interface
  - Supports the interface that can directly connect to the SRAM
- SDRAM interface
  - Can connect directly to SDRAM in area 3
  - Multiplex output for row address/column address
  - Efficient access by single read/single write
  - High-speed access by bank-active mode
  - Supports auto-refreshing and self-refreshing

- Byte-selection SRAM interface
  - Can connect directly to byte-selection SRAM
- 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 wait cycles by software
  - Supports the bus sizing function of the I/O bus width (only in little endian mode)
- Refresh function
  - Supports the auto-refreshing and self-refreshing functions
  - Specifies the refresh interval by setting the refresh counter and clock selection
  - Can execute consecutive refresh cycles by specifying the refresh counts (1, 2, 4, 6, or 8)

The block diagram of the BSC is shown in figure 7.1.

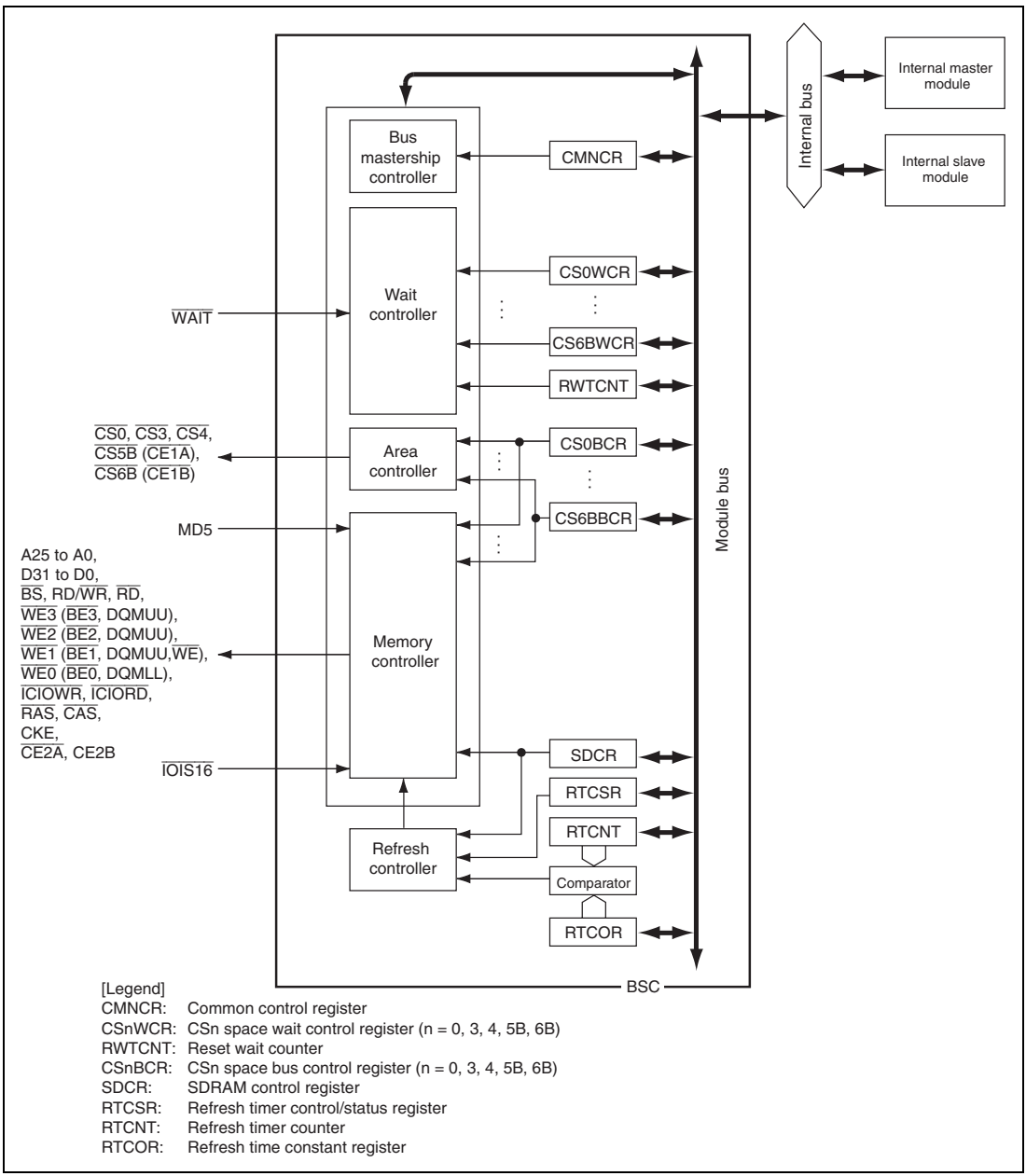


Figure 7.1 Block Diagram of BSC

7.2 Input/Output Pins

table 7.1 lists the pin configuration of the BSC.

Table 7.1 Pin Configuration

Abbreviation	I/O	Function
A25 to A0	Output	Address Bus*
D31 to D0	I/O	Data Bus
$\overline{BS}$	Output	Bus Cycle Start Asserted when a normal space, burst ROM (clock synchronous /asynchronous), or PCMCIA is accessed. Asserted at the same timing as $\overline{CAS}$ assertion in SDRAM access.
$\overline{CS0}$, $\overline{CS3}$, $\overline{CS4}$	Output	Chip Select
$\overline{CS5B}$/$\overline{CE1A}$	Output	Chip Select Chip enable for PCMCIA allocated to area 5 when PCMCIA is in use
$\overline{CE2A}$	Output	Chip enable for PCMCIA allocated to area 5 when PCMCIA is in use
$\overline{CS6B}$/$\overline{CE1B}$	Output	Chip Select Chip enable for PCMCIA allocated to area 6 when PCMCIA is in use
$\overline{CE2B}$	Output	Chip enable for PCMCIA allocated to area 6 when PCMCIA is in use
RD/$\overline{WR}$	Output	Read/Write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is used.
$\overline{RD}$	Output	Read Pulse Signal (read data output enable signal) Strobe signal to indicate a memory read cycle when PCMCIA is in use.
$\overline{ICIOWR}$	Output	Strobe signal to indicate I/O write when PCMCIA is in use.
$\overline{ICIORD}$	Output	Strobe signal to indicate I/O read when PCMCIA is in use.
$\overline{WE3(BE3)}$	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when byte-selection SRAM is in use.
$\overline{WE2(BE2)}$	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when byte-selection SRAM is in use.
$\overline{WE1(BE1)}$/$\overline{WE}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when byte-selection SRAM is in use. Strove signal to indicate a memory write cycle when PCMCIA is in use.

Abbreviation	I/O	Function
$\overline{\text{WE0(BE0)}}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is in use.
$\overline{\text{RAS}}$	Output	Connected to $\overline{\text{RAS}}$ pin when SDRAM is in use.
$\overline{\text{CAS}}$	Output	Connected to $\overline{\text{CAS}}$ pin when SDRAM is in use.
CKE	Output	Connected to CKE pin when SDRAM is in use.
$\overline{\text{IOIS16}}$	Input	PCMCIA 16-bit I/O Signal Enabled only in little endian mode. Drive this signal low in big endian mode.
DQMUU, DQMUL, DQMLU, DQMLL	Output	Connected to the DQMxx pin when SDRAM is in use. DQMUU: Select signal for D31 to D24 DQMUL: Select signal for D23 to D16 DQMLU: Select signal for D15 to D8 DQMLL: Select signal for D7 to D0
$\overline{\text{WAIT}}$	Input	External wait input
MD5, MD3	Input	MD5: Selects data alignment (big endian or little endian) MD3: Specifies area 0 bus width (8/16 bits)

Note: * As pins A25 to A16 act as general I/O ports immediately after a power-on reset, pull-up or pull-down these pins outside the LSI as needed.

7.3 Area Overview

7.3.1 Area Division

The architecture of this LSI has 32-bit address space. The upper three address bits divide the space into areas P0 to P4, and the cache access methods can be specified for each area. For details, see section 3, Cache. Each area indicated by the remaining 29 bits is divided into ten areas (five areas are reserved) when address map 1 is selected or eight areas (three areas are reserved) when address map 2 is selected. The address map is selected by the MAP bit in CMNCR. The BSC controls the areas indicated by the 29 bits.

As listed in tables 7.2 and 7.3, memory can be connected directly to five physical areas of this LSI, and the chip select signals ($\overline{\text{CS0}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$, $\overline{\text{CS5B}}$, and $\overline{\text{CS6B}}$) are output for each area. $\overline{\text{CS0}}$ is asserted during area 0 access.

7.3.2 Shadow Area

Areas 0, 3, 4, 5B, and 6B are divided by decoding physical address bits 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 spaces are reserved, so do not use it.

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

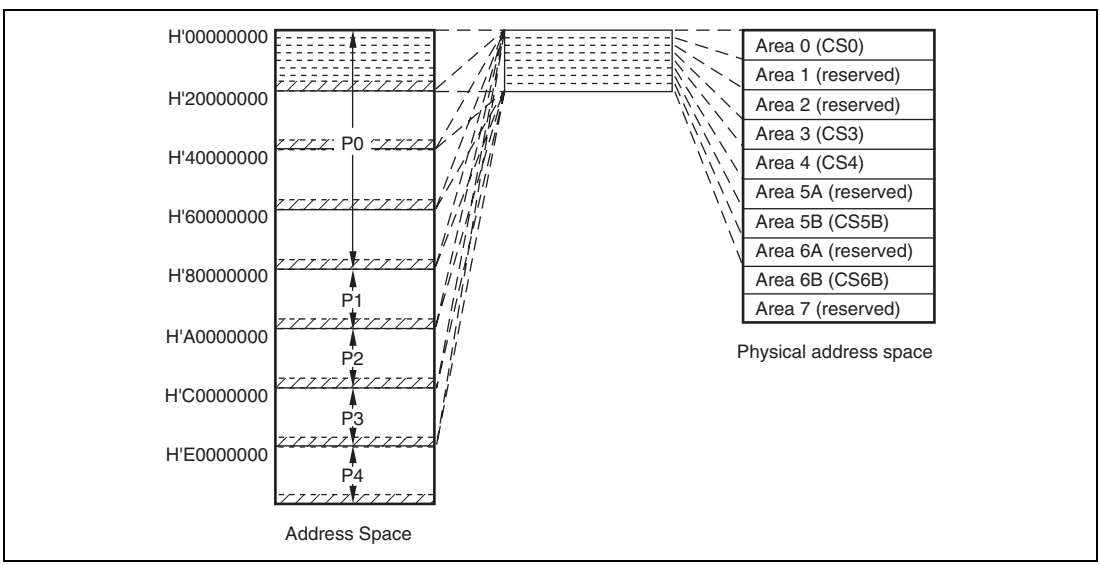


Figure 7.2 Address Space

7.3.3 Address Map

The external address space has a capacity of 256 Mbytes and is divided into five areas. Types of memory to be connected and the data bus width are specified for individual areas. The address map for the external address space is shown in table 7.2.

Table 7.2 Address Map 1 (CMNCR.MAP = 0)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Reserved area*	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Reserved area*	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	64 Mbytes
H'10000000 to H'13FFFFFF	Area 4	Normal memory Byte-selection SRAM	64 Mbytes
H'14000000 to H'15FFFFFF	Area 5A	Reserved area*	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory Byte-selection SRAM	32 Mbytes
H'18000000 to H'19FFFFFF	Area 6A	Reserved area*	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory Byte-selection SRAM	32 Mbytes
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*	64 Mbytes

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

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

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

Physical Address	Area	Memory to be Connected	Capacity
H'18000000 to H'1BFFFFFF	Area 6*[2]	Normal memory 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 enabled.
For area 6, CS6BBCR and CS6BWCR are enabled.

7.3.4 Area 0 Memory Type and Memory Bus Width

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

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

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

7.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data alignment is specified using the external pin (MD5) at a power-on reset as shown in table 7.5.

Table 7.5 Correspondence between External Pin (MD5) and Endians

MD5	Endian
0	Big endian
1	Little endian

7.4 Register Descriptions

The BSC has the following registers. For the addresses and access size for these registers, see section 24, List of Registers.

Do not access spaces other than CS0 until setting the memory interfaces is complete.

- Common control register (CMNCR)
- CS0 space bus control register for area 0 (CS0BCR)
- CS3 space bus control register for area 3 (CS3BCR)
- CS4 space bus control register for area 4 (CS4BCR)
- CS5B space bus control register for area 5B (CS5BBCR)
- CS6B space bus control register for area 6B (CS6BBCR)
- CS0 space wait control register for area 0 (CS0WCR)
- CS3 space wait control register for area 3 (CS3WCR)
- CS4 space wait control register for area 4 (CS4WCR)
- CS5B space wait control register for area 5B (CS5BWCR)
- CS6B space wait control register for area 6B (CS6BWCR)
- SDRAM control register (SDCR)
- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)

7.4.1 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 setting CMNCR is complete.

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	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 7.2 and 7.3. 0: Selects address map 1 1: Selects address map 2
11 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 Fetches the external pin (MD5) state for specifying endian at a power-on reset. The endian setting for all the address spaces are set by this bit. This is a read-only bit. 0: External pin (MD5) for specifying endian was driven low at a power-on reset. This LSI is operated as big endian. 1: External pin (MD5) for specifying endian was driven high at a power-on reset. This LSI is being operated as little endian.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
1	HIZMEM	0	R/W	Hi-Z Memory Control Specifies the pin state in standby mode for pins A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$ ($\overline{BEn}$)/DQMxx, and $\overline{RD}$. 0: High impedance in standby mode 1: Driven in standby mode
0	HIZCNT	0	R/W	Hi-Z Control Specifies the pin state in standby mode for the CKIO, CKE, $\overline{RAS}$, and $\overline{CAS}$ pins. 0: High impedance in standby mode 1: Driven in standby mode

Note: * The external pin (MD5) state for specifying endian is sampled at a power-on reset. When big endian is specified, this bit is read as 0 and when little endian is specified, this bit is read as 1.

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29 28	IWW1 IWW0	1 1	R/W R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The write and read cycles or write and write cycles performed consecutively are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26	IWRWD1	1	R/W	Idle Cycles for Another Space Read-Write
25	IWRWD0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and write cycles which are performed consecutively and are accessed to different areas are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	IWRWS1	1	R/W	Idle Cycles for Read-Write in Same Space
22	IWRWS0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and write cycles which are performed consecutively and are accessed to the same area are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
20	IWRRD1	1	R/W	Idle Cycles for Read-Read in Another Space
19	IWRRD0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and read cycles which are performed consecutively and are accessed to different areas are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	IWRRS1	1	R/W	Idle Cycles for Read-Read in Same Space
16	IWRRS0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and read cycles which are performed consecutively and are accessed to the same area are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 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 the area.
13	TYPE1	0	R/W	0000: Normal space
12	TYPE0	0	R/W	0001: Reserved (setting prohibited) 0010: Reserved (setting prohibited) 0011: Byte-selection SRAM 0100: SDRAM 0101: PCMCIA 0110: Reserved (setting prohibited) 0111: Reserved (setting prohibited) 1000: Reserved (setting prohibited) 1001: Reserved (setting prohibited) 1010: Reserved (setting prohibited) 1011: Reserved (setting prohibited) 1100: Reserved (setting prohibited) 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited) For details on memory type in each area, see tables 7.2 and 7.3.
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*	R/W	Data Bus Size
9	BSZ0	1*	R/W	Specify the data bus width of each area. 00: Reserved (setting prohibited) 01: 8 bits 10: 16 bits 11: 32 bits Notes: 1. The data bus width for area 0 is specified by the external pin. These bits are ignored. 2. When area 5 or 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 3. If area 3 is specified as SDRAM space, the bus width cannot be specified as 8 bits. 4. These bits must be specified to either 01 or 11 before accessing to memory in other than area 0.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * CS0BCR fetches the external pin state (MD3) that specify the bus width at a power-on reset.

7.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 3, 4, 5B, 6B)

CSnWCR 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

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 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 cycles 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 or 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.
1	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to Address, $\overline{\text{CSn}}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- 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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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 (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (used as status) and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (used as strobe)
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 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)

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 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 (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (signal used as status)and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (signal used as strobe)
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: Same number of 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, $\overline{\text{CSn}}$ 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. 00: 0.5 cycles 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 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 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 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS5BWCR

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
17	WW1	0	R/W	Specify the number of cycles that are necessary for write access.
16	WW0	0	R/W	000: Same number of 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, $\overline{\text{CSn}}$ 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. 00: 0.5 cycles 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 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 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 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- 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 (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (used as status) and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (used as strobe)
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value 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 cycles 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 or 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.
1	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to Address, $\overline{\text{CSn}}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

SDRAM:

- 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 13	WTRP1 WTRP0	0 0	R/W R/W	Wait Cycle Number for Precharge Completion Specify the number of minimum wait cycles inserted to wait for the completion of precharge in the following cases. • From the start of auto-precharge to the issuing of the ACTV command for the same bank. • From the issuing of the PRE/PALL command to the issuing of the ACTV command for the same bank. • From the issuing of the PALL command during auto-refreshing to the issuing of the REF command. • From the issuing of the PALL command during self-refreshing to the issuing of the SELF command. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 10	WTRCD1 WTRCD0	0 1	R/W R/W	Wait Cycle Number from ACTV Command to READ(A)/WRIT(A) Command Specify the number of minimum wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
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: Reserved (setting prohibited)
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TRWL1	0	R/W	Wait Cycle Number for Precharge Start Wait
3	TRWL0	0	R/W	Specify the number of minimum wait cycles inserted to wait for the start of precharge in the following cases. • From the issuing of the WRITA command by this LSI to the start of the auto-precharge in the SDRAM. The ACTV command for the same bank is issued after issuing the WRITA command in non-bank active mode. To confirm how many cycles should be needed in the SDRAM between receiving the WRITA command and the auto-precharge start, refer to the data sheets for each SDRAM. Set this bit so that the cycle number in that data sheets should not exceed the cycle number set by this bit. • From the issuing of the WRIT command by this LSI to the issuing of the PRE command. A different row address in the same bank is accessed in bank active mode. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1 0	WTRC1 WTRC0	0 0	R/W R/W	Idle Cycle Number from REF Command/Self-Refreshing Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the following cases. • From the issuing of the REF command to the issuing of the ACTV/REF/MRS command. • From the self-refreshing release to the issuing of the ACTV/REF/MRS command. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

PCMCIA:

- CS5BWCR, CS6BWCR

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21 20	SA1 SA0	0 0	R/W R/W	Space Attribute Specification Specify memory card interface or I/O 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

Bit	Bit Name	Initial Value	R/W	Description
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 $\overline{WE}$ assertion in PCMCIA interface.
12	TED1	0	R/W	
11	TED0	0	R/W	0000: 0.5 cycles 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: 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)

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 $\overline{WE}$ negation 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

7.4.4 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 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refreshing SDRAM is performed. 0: Refreshing is not performed 1: Refreshing is performed
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refreshing or self-refreshing when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refreshing starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refreshing starts according to the contents that are set in RTCSR, RTCNT, and RTCOR. 0: Auto-refreshing is performed 1: Self-refreshing is performed
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	BACTV	0	R/W	Bank Active Mode Specifies whether to access 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)
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	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)

7.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

When RTCSR is written to, the upper 16 bits of the write data must be H'A55A to 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.

Bit	Bit Name	Initial Value	R/W	Description
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). [Clearing condition] When 0 is written to this bit after reading RTCSR with CMF = 1. [Setting condition] When RTCNT value matches RTCOR value
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	CKS2	0	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
4	CKS1	0	R/W	
3	CKS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	Specify the number of consecutive 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). Using consecutive refresh cycles can prolong cycles between refreshing.
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)

7.4.6 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 RTCNT is written to, 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

7.4.7 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. The 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.

When the RTCOR is written to, 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

7.5 Operation

7.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the most significant byte (MSByte) of multiple byte data is stored in the lower address, and little endian, in which the least significant byte (LSByte) of multiple byte data is stored in the lower address. Endian is specified at a power-on reset by the external pin (MD5). When pin MD5 is driven low at a power-on reset, the endian will become big endian and when pin MD5 is driven high at a power-on reset, the endian will become little endian.

Three data bus widths (8, 16, and 32 bits) are available for normal memory and byte-selection SRAM. Two data bus widths (16 and 32 bits) are available for SDRAM. Two data bus widths (8 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 7.6 to 7.11 show the relationship between endian, device data width, and access unit.

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

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{\text{WE3(BE3)}}$, 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 7.7 16-Bit External Device/Big 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 15 to 8	—	—	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 7.8 8-Bit External Device/Big 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	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 7.9 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 7.10 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 7.11 8-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	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

7.5.2 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 7.5.6, Byte-Selection SRAM Interface. Figure 7.3 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.

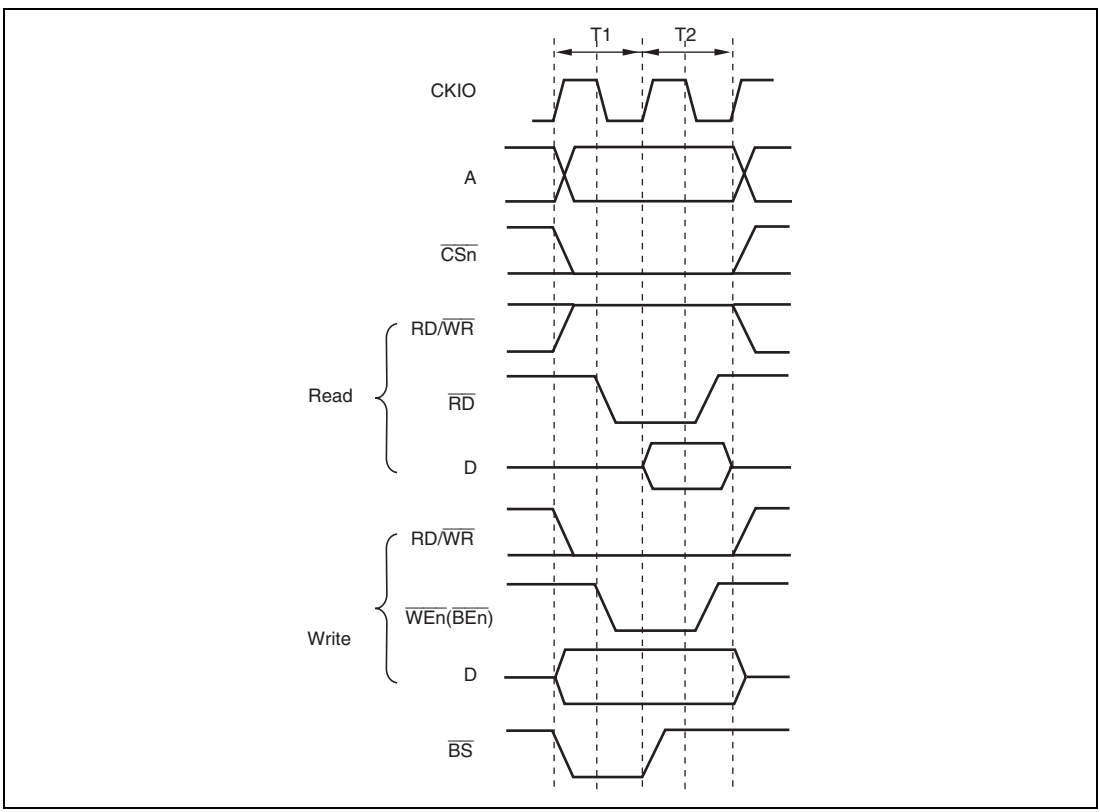


Figure 7.3 Normal Space Basic Access Timing (No-Wait Access)

There is no output signal which informs external devices of the access size when reading. Although the least significant bit of the address indicates the correct address when the access starts, 16-bit data is always read from a 16-bit device. When writing, only the $\overline{WEn}$ ($\overline{BEn}$) signal for the byte to be written to is asserted.

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

carried out. Therefore, care must be taken when controlling the buffers with the RD/$\overline{\text{WR}}$ signal, to avoid data conflict.

Figures 7.4 and 7.5 show the basic timings of normal space consecutive access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted to check the external wait (figure 7.4). If the WM bit in CSnWCR is set to 1, an external wait request is ignored and no Tnop cycle is inserted (figure 7.5).

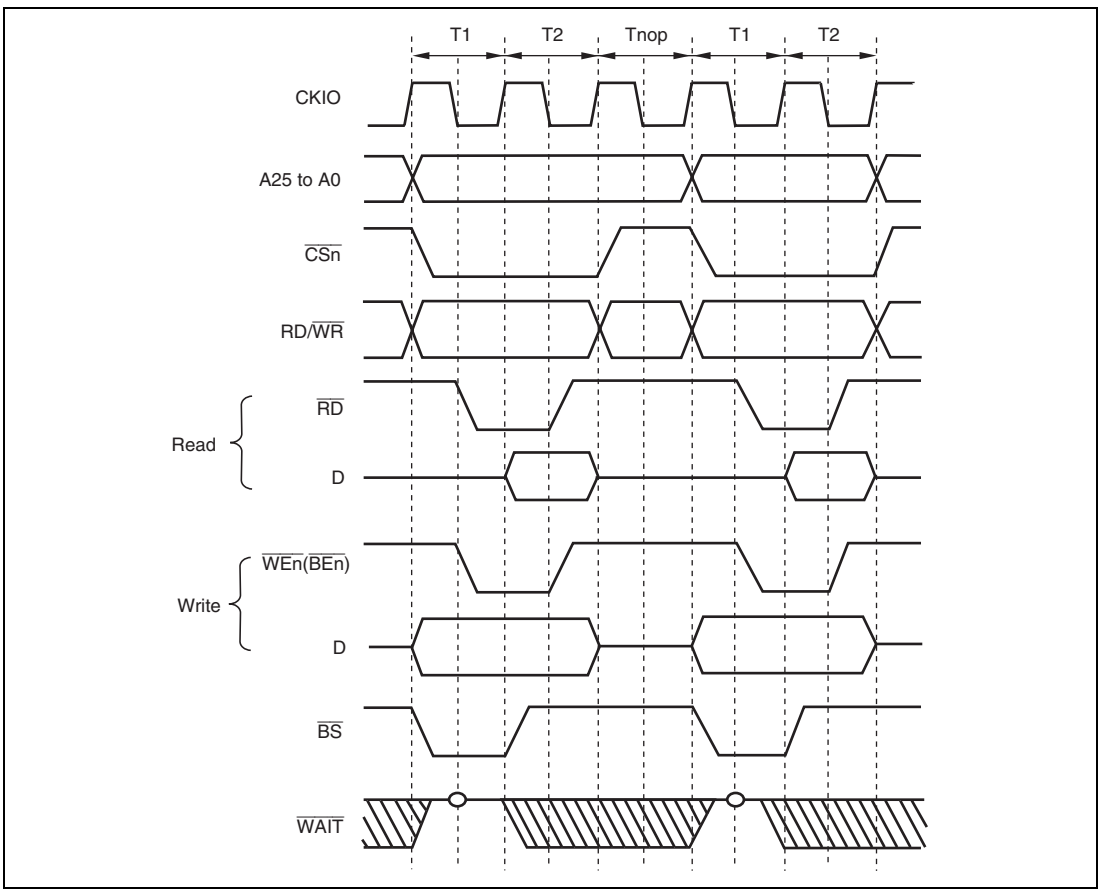


Figure 7.4 Consecutive Access to Normal Space (1): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 0 (Access Wait = 0, Cycle Wait = 0)

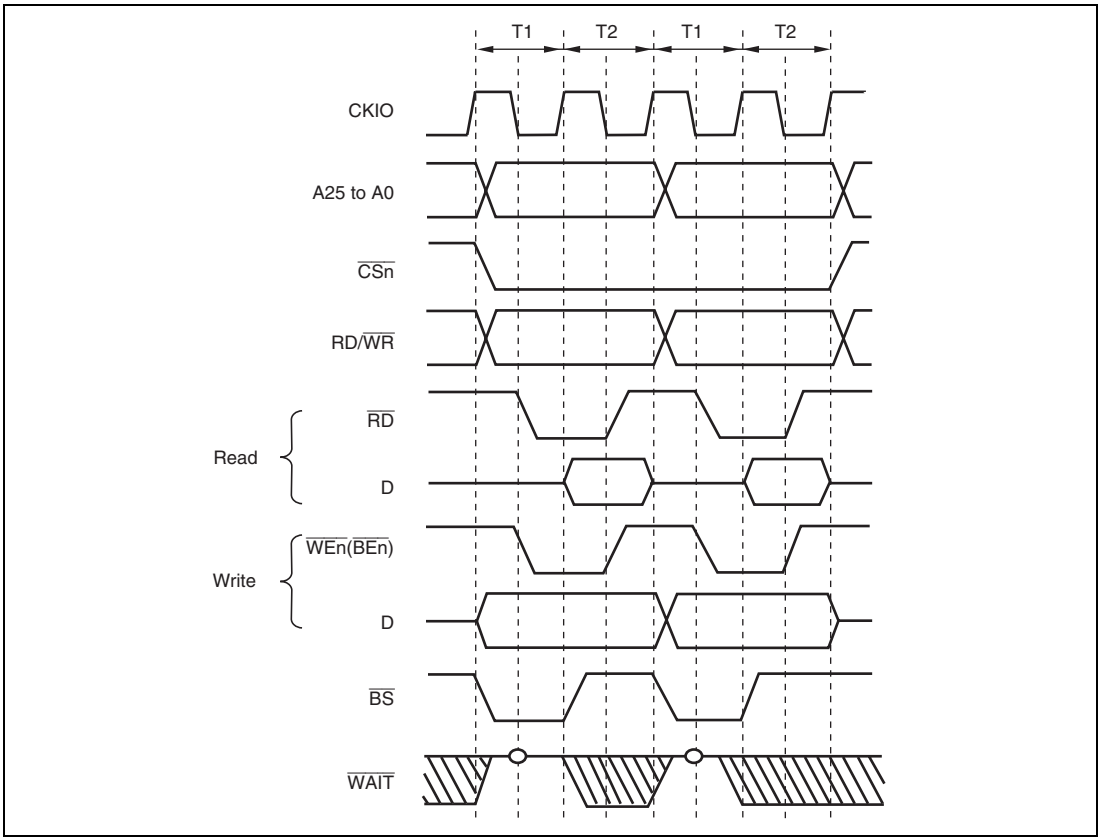


Figure 7.5 Consecutive Access to Normal Space (2): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 1 (Access Wait = 0, Cycle Wait = 0)

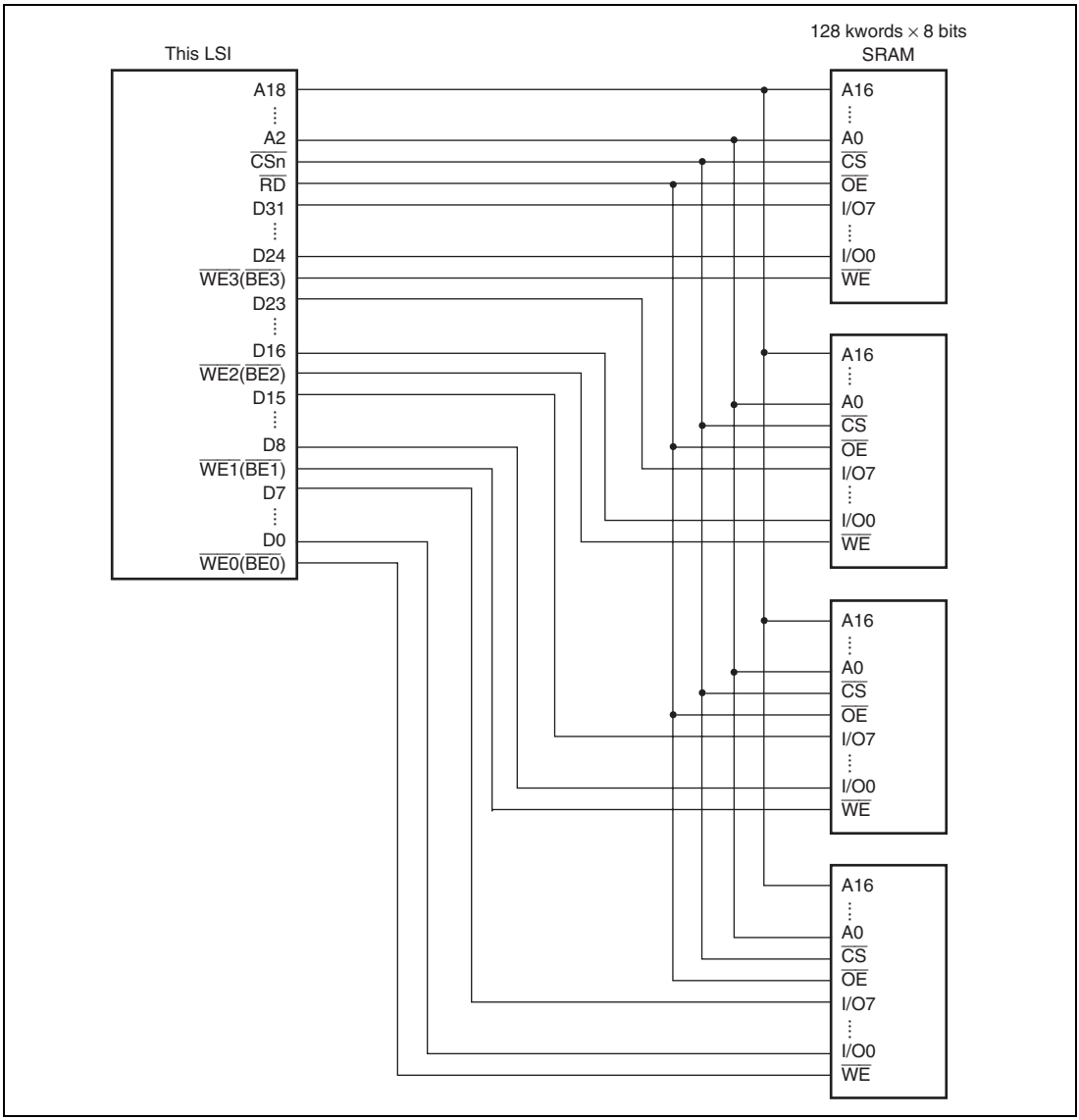


Figure 7.6 Example of 32-Bit Data-Width SRAM Connection

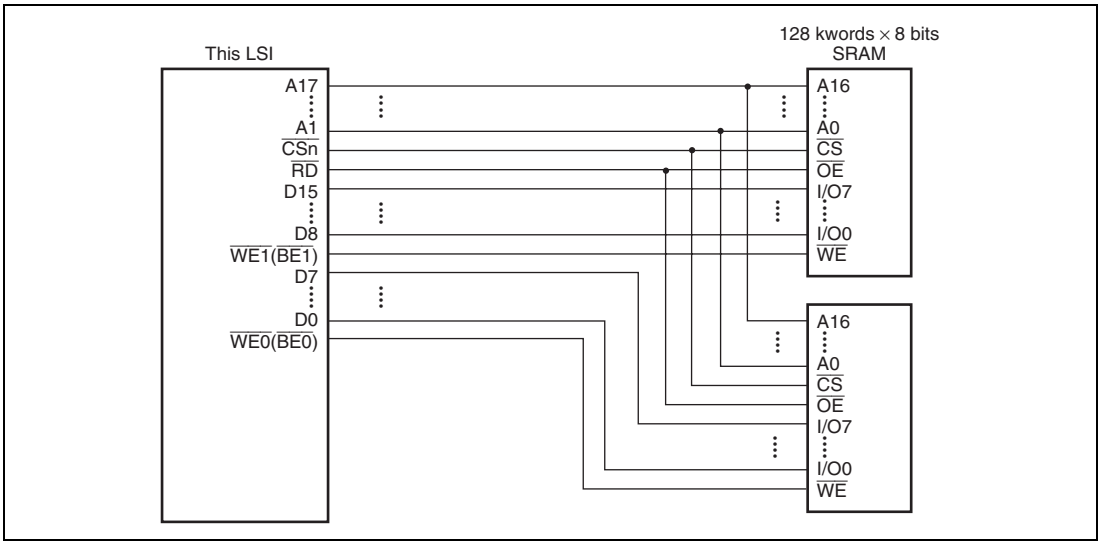


Figure 7.7 Example of 16-Bit Data-Width SRAM Connection

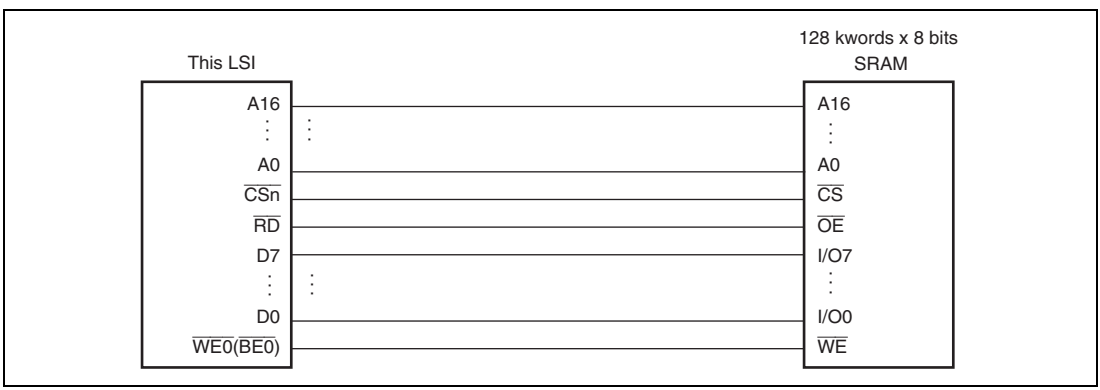


Figure 7.8 Example of 8-Bit Data-Width SRAM Connection

7.5.3 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 the same 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 7.9.

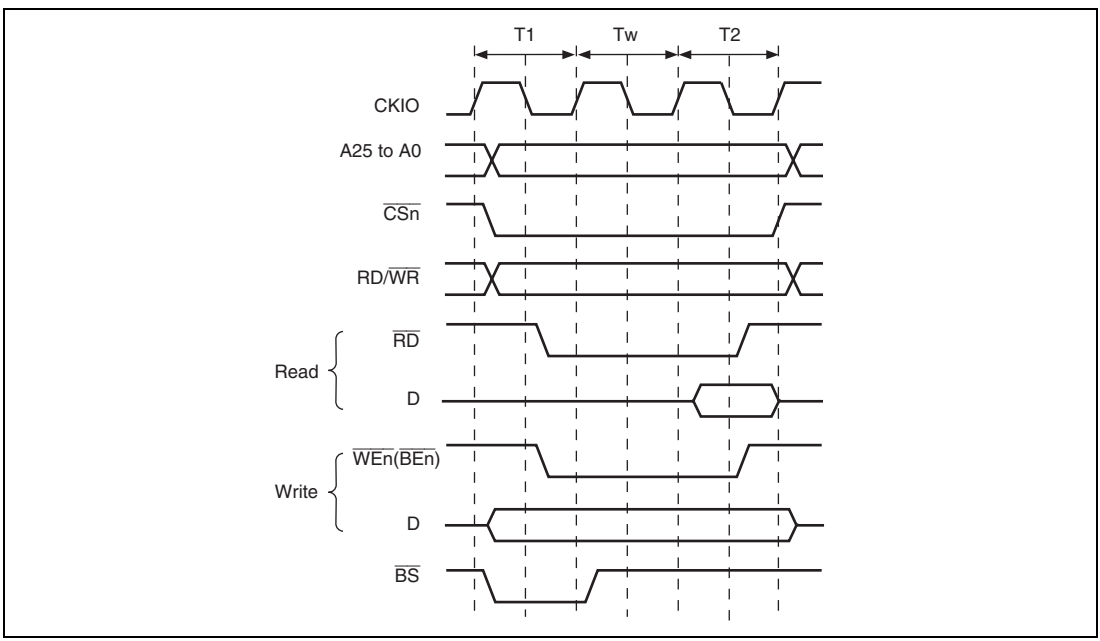


Figure 7.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait signal ($\overline{\text{WAIT}}$) is also sampled. The $\overline{\text{WAIT}}$ pin sampling is shown in figure 7.10. In this example, two wait cycles are inserted as software wait. The $\overline{\text{WAIT}}$ signal is sampled at the falling edge of the CKIO signal in the cycle immediately before the T2 cycle (T1 or Tw cycle).

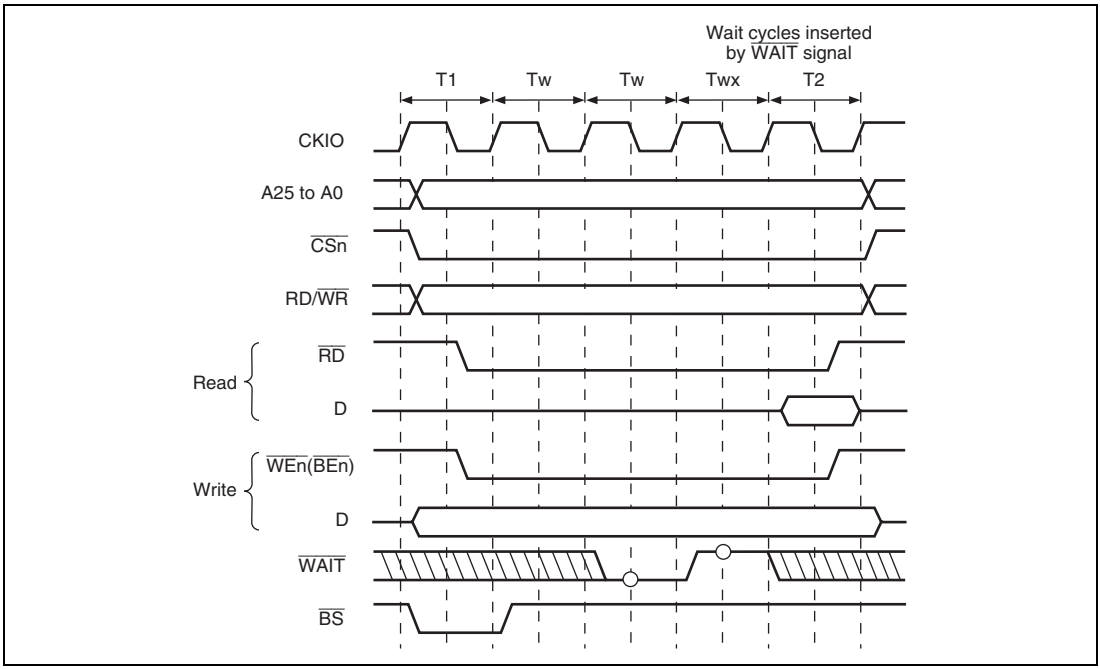


Figure 7.10 Wait Cycle Timing for Normal Space Access (Wait cycle Insertion using $\overline{\text{WAIT}}$)

7.5.4 Extension of Chip Select ($\overline{CSn}$) Assertion Period

The number of cycles from $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to $\overline{CSn}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 7.11 shows an example. A Th cycle and a Tf cycle are added before and after a normal cycle, respectively. In these cycles, $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) 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.

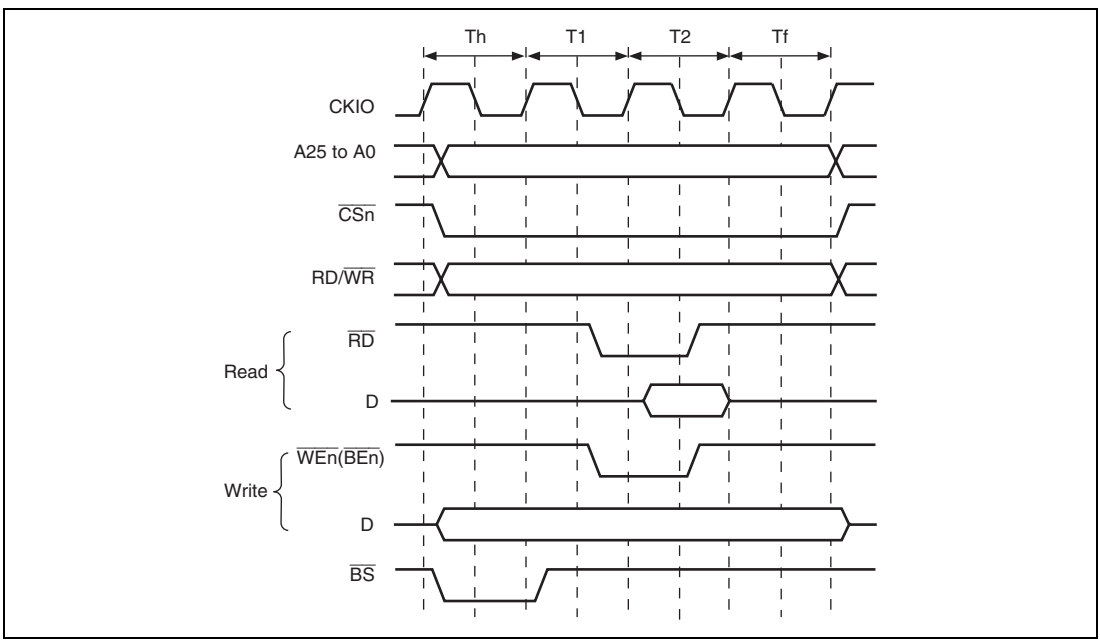


Figure 7.11 Example of Timing when $\overline{CSn}$ Assertion Period is Extended

7.5.5 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, and $\overline{CS3}$. Signals other than CKE are valid when $\overline{CS3}$ is asserted. SDRAM can be connected to area 2. The data bus width of the area that is connected to SDRAM can be set to 16 bits or 32 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-refreshing (REF)
- Self-refreshing (SELF)
- All banks precharge (PALL)
- Specified bank precharge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with precharge (READA)
- Write (WRIT)
- Write with precharge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU and DQMLL. 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 7.5.1, Endian/Access Size and Data Alignment.

Figures 7.12 and 7.13 show an example of the connection of the SDRAM with the LSI.

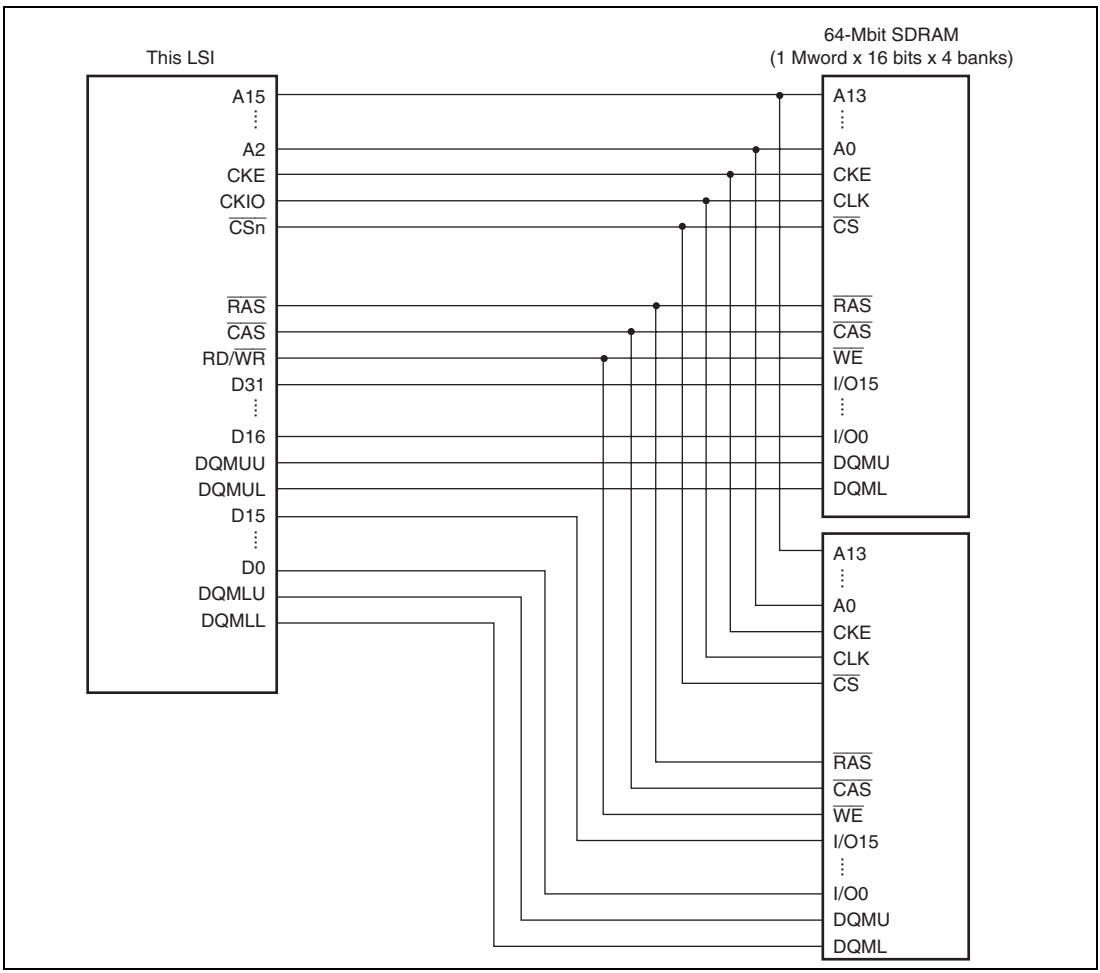


Figure 7.12 Example of 32-Bit Data-Width SDRAM Connection

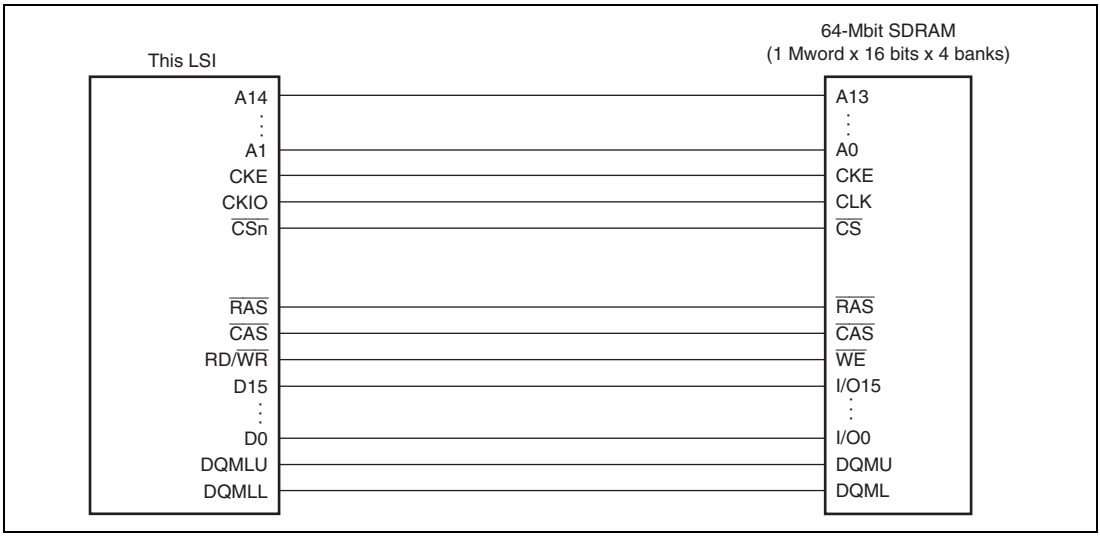


Figure 7.13 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ1 and BSZ0 in CSnBCR, AnROW1 and AnROW0 and AnCOL1 AnCOL0 in SDCR. Tables 7.12 to 7.17 show the relationship between those settings and the bits output on 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 on these pins.

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

Table 7.12 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (1)

Setting				
BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**		
11 (32 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*[2]*[3]	A22*[2]*[3]	A12 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20	L/H*[1]	A10/AP	Specifies address/ precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	

Setting				
BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**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	

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
11 (32 bits)	00 (11 bits)	00 (8 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A1	A9	A1		Unused
A0	A8	A0		
Example of memory connection				
One 64-Mbit product (512 kwords x 32 bits x 4 banks, 8-bit column product)				
Two 16-Mbit products (512 kwords x 16 bits x 2 banks, 8-bit column product)				

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
11 (32 bits)	01 (12 bits)	00 (8 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A1	A9	A1		Unused
A0	A8	A0		
Example of memory connection				
One 128-Mbit product (1 Mword x 32 bits x 4 banks, 8-bit column product)				
Two 64-Mbit product (1 Mword x 16 bits x 4 banks, 8-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification
3. Applicable only to a 64-Mbit product

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

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

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

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
11 (32 bits)	01 (12 bits)	01 (9 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
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 memory connection				
One 256-Mbit product (2 Mwords x 32 bits x 4 banks, 9-bit column product)				
Two 128-Mbit products (2 Mwords x 16 bits x 4 banks, 9-bit column product)				

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
11 (32 bits)	01 (12 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
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	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of memory connection				
One 512-Mbit product (4 Mwords x 32 bits x 4 banks, 10-bit column product)				
Two 256-Mbit product (4 Mwords x 16 bits x 4 banks, 10-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.

2. Bank address specification

Table 7.14 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (3)

Setting				
BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**		
11 (32 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**Function**
A17	A26	A17		Unused
A16	A25*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 memory connection				
One 512-Mbit product (4 Mwords x 32 bits x 4 banks, 9-bit column product)				
Two 256-Mbit products (4 Mwords x 16 bits x 4 banks, 9-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.15 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (4)

Setting					Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]			BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	00 (11 bits)	00 (8 bits)			10 (16 bits)	01 (12 bits)	00 (8 bits)		
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**Function**	**Output Pins of This LSI**	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**Function**
A17	A25	A17		Unused	A17	A25	A17		Unused
A16	A24	A16			A16	A24	A16		
A15	A23	A15			A15	A23	A15		
A14	A22	A14			A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21	A21			A13	A21*2	A21*2	A12 (BA0)	
A12	A20*2	A20*2	A11 (BA0)	Specifies bank	A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/ precharge	A11	A19	L/H*1	A10/AP	Specifies address/ precharge
A10	A18	A10	A9	Address	A10	A18	A10	A9	Address
A9	A17	A9	A8		A9	A17	A9	A8	
A8	A16	A8	A7		A8	A16	A8	A7	
A7	A15	A7	A6		A7	A15	A7	A6	
A6	A14	A6	A5		A6	A14	A6	A5	
A5	A13	A5	A4		A5	A13	A5	A4	
A4	A12	A4	A3		A4	A12	A4	A3	
A3	A11	A3	A2		A3	A11	A3	A2	
A2	A10	A2	A1		A2	A10	A2	A1	
A1	A9	A1	A0		A1	A9	A1	A0	
A0	A8	A0		Unused	A0	A8	A0		Unused

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	00 (11 bits)	00 (8 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
Example of memory connection				
One 16-Mbit product (512 kwords x 16 bits x 2 banks, 8-bit column product)				

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	00 (8 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
Example of memory connection				
One 64-Mbit products (1 Mword x 16 bits x 4 banks, 8-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

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

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

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	01 (9 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
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 memory connection				
One 128-Mbit product (2 Mwords x 16 bits x 4 banks, 9-bit column product)				

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
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 memory connection				
One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 10-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.17 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (6)

Setting					Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]			BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	10 (13 bits)	01 (9 bits)			10 (16 bits)	10 (13 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function	Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A17	A26	A17		Unused	A17	A27	A17		Unused
A16	A25	A16			A16	A26	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank	A15	A25*2	A25*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)		A14	A24*2	A24*2	A13 (BA0)	
A13	A22	A13	A12	Address	A13	A23	A13	A12	Address
A12	A21	A12	A11		A12	A22	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/ precharge	A11	A21	L/H*1	A10/AP	Specifies address/ precharge
A10	A19	A10	A9	Address	A10	A20	A10	A9	Address
A9	A18	A9	A8		A9	A19	A9	A8	
A8	A17	A8	A7		A8	A18	A8	A7	
A7	A16	A7	A6		A7	A17	A7	A6	
A6	A15	A6	A5		A6	A16	A6	A5	
A5	A14	A5	A4		A5	A15	A5	A4	
A4	A13	A4	A3		A4	A14	A4	A3	
A3	A12	A3	A2		A3	A13	A3	A2	
A2	A11	A2	A1		A2	A12	A2	A1	
A1	A10	A1	A0		A1	A11	A1	A0	
A0	A9	A0		Unused	A0	A10	A0		Unused

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	10 (13 bits)	01 (9 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
Example of memory connection				
One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 9-bit column product)				

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	10 (13 bits)	10 (10 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
Example of memory connection				
One 512-Mbit product (8 Mwords x 16 bits x 4 banks, 10-bit column product)				

Notes: 1. L/H is a bit used in the command specification; it is fixed 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 by DMAC and E-DMAC (access to non-cacheable area)

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

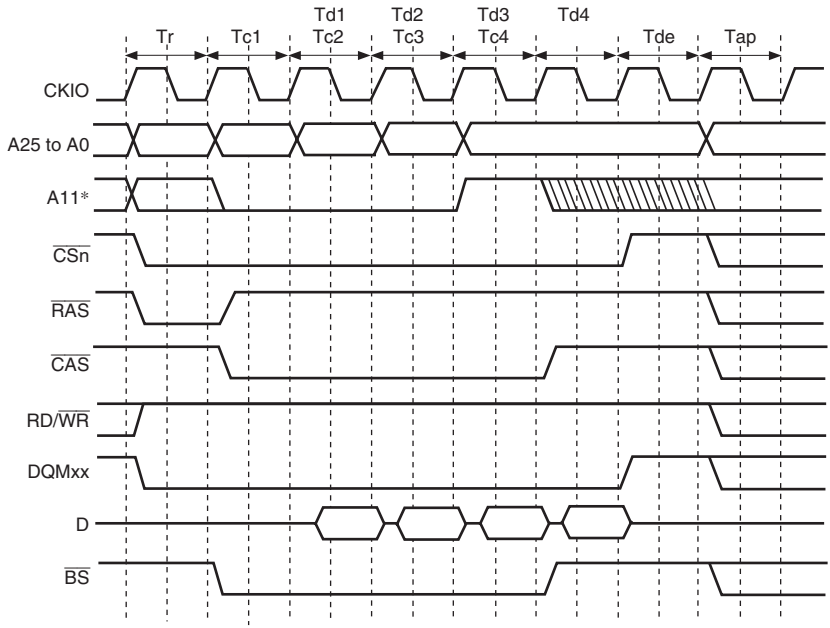
Table 7.18 shows the relationship between the access size and the number of bursts.

Table 7.18 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

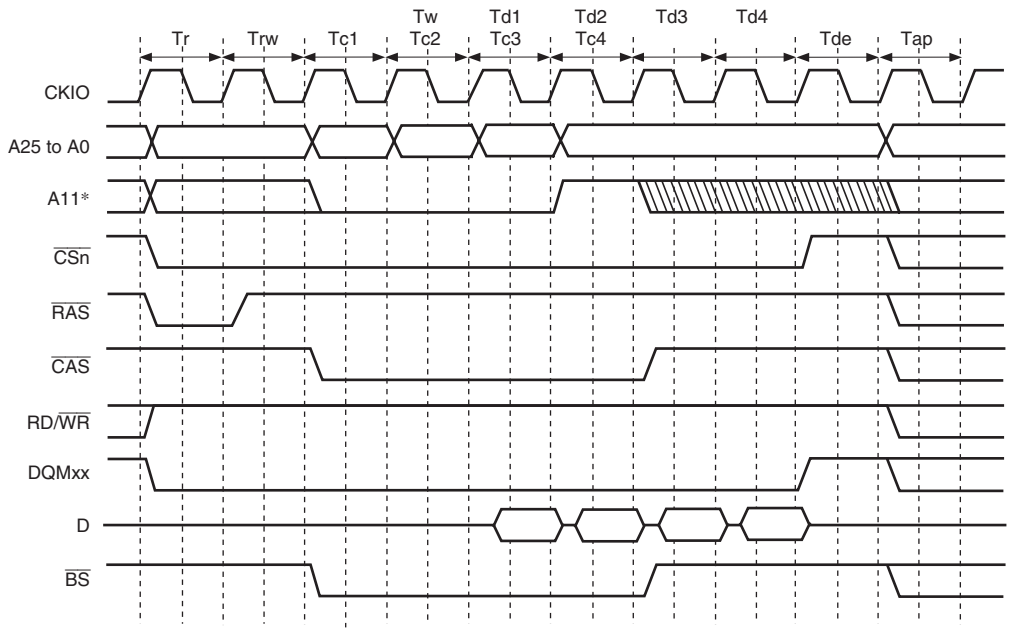
Figures 7.14 and 7.15 show timing charts in burst read. In burst read, the ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is latched at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READ command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, other banks can be accessed. The number of Tap cycles is specified by bits WTRP1 and WTRP0 in CS3WCR.

In this LSI, wait cycles can be inserted by specifying bits in CSnWCR to connect the SDRAM with variable frequencies. Figure 7.15 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 bits WTRCD1 and WTRCD0 in CS3WCR. When bits WTRCD1 and WTRCD0 is set to 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 by bits A3CL1 and A3CL0 bits in CS3WCR in CS3WCR. This number of cycles 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 one to four cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the synchronous DRAM.



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

Figure 7.14 Burst Read Basic Timing (Auto Precharge)

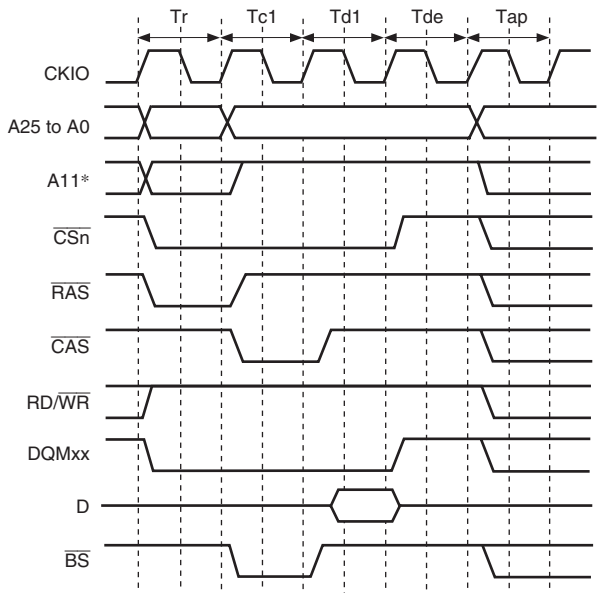


Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.15 Burst Read Wait Specification Timing (Auto Precharge)

Single Read: A read access ends in one cycle when data exists in non-cacheable area and the data bus width is larger than or equal to access size. Since 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 7.16 shows the single read basic timing.



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

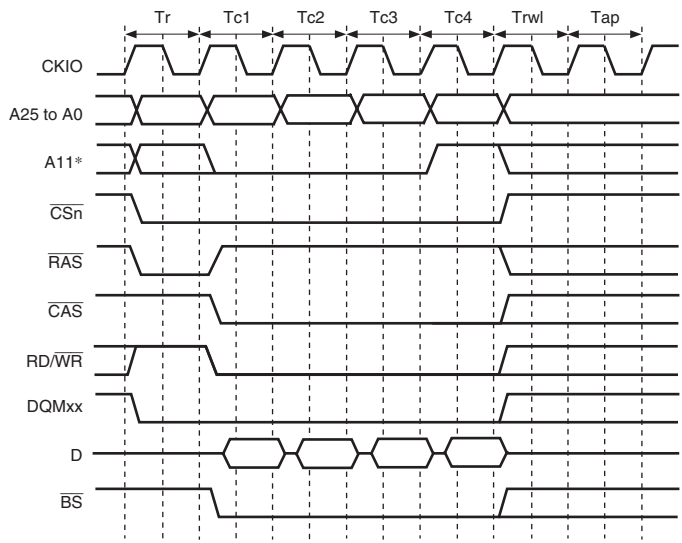
Figure 7.16 Basic Timing for Single Read (Auto Precharge)

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

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

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed consecutively four times to write 16-byte consecutive 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 7.18.

Figure 7.17 shows a timing chart for burst writes. In burst write, the 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, other CS areas and other banks can be accessed. The number of Trw1 cycles is specified by bits TRWL1 and TRWL0 in CS3WCR. The number of Tap cycles is specified by bits WTRP1 and WTRP0 in CS3WCR.

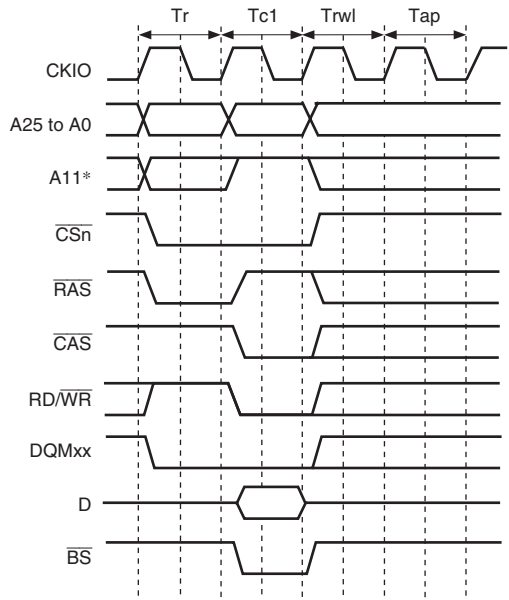


Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.17 Basic Timing for Burst Write (Auto Precharge)

Single Write: A write access ends in one cycle when data is written in non-cacheable area and the data bus width is larger than or equal to access size.

Figure 7.18 shows the single write basic timing.



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

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

Bank Active: The synchronous DRAM bank function is used to support high-speed 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.

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. Since synchronous DRAM is internally divided into several banks, it is possible to keep one row address in each bank activated. 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 bits WTRP1 and WTRP0 in CSnWCR.

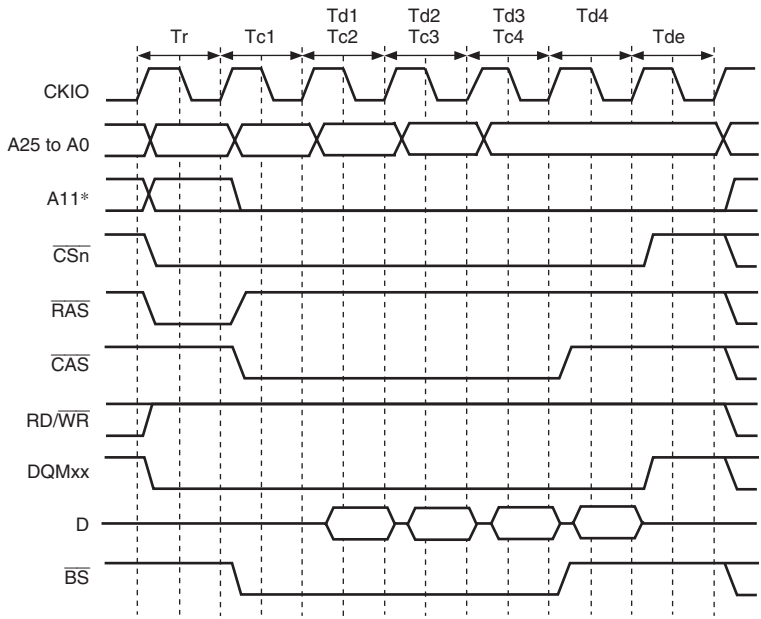
In a write access, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT command can be issued successively if the row address is the same. The number of cycles 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 another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refreshing 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 7.19, a burst read cycle for the same row address in figure 7.20, and a burst read cycle for different row addresses in figure 7.21. Likewise, a single write cycle without auto-precharge is shown in figure 7.22, a single write cycle for the same row address in figure 7.23, and a single write cycle for different row addresses in figure 7.24.

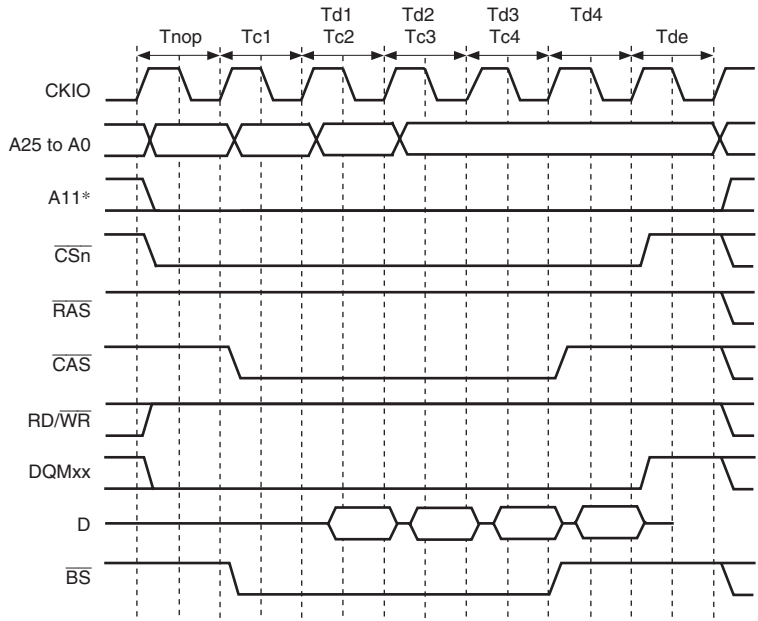
In figure 7.20, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to secure two cycles of CAS latency for the DQMxx signal that specifies which byte data is read from SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be secured even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 7.19 or 7.22, followed by repetition of the cycle in figure 7.20 or 7.23. An access to a different area during this time has no effect. When a different row address is accessed in the bank active state, the bus cycle shown in figure 7.21 or 7.24 is executed instead of that in figure 7.20 or 7.23. In bank active mode, too, all banks become inactive after a refresh cycle.



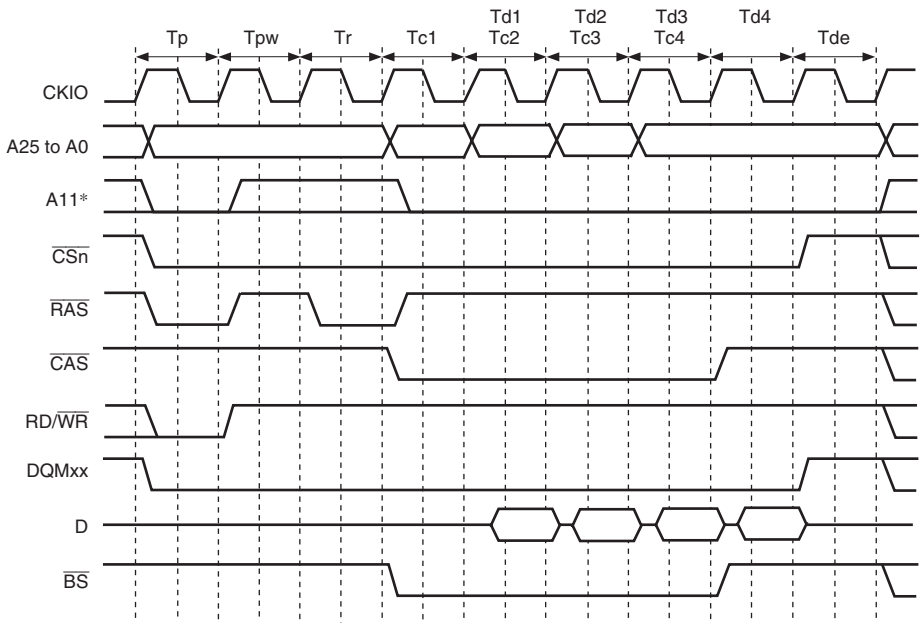
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.19 Burst Read Timing (No Auto Precharge)



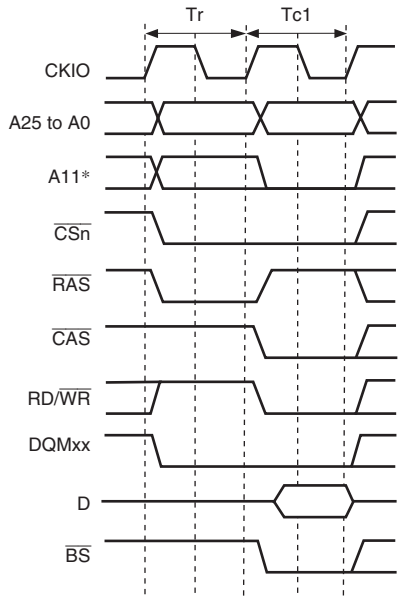
Note: * Address pin to be connected to pin A10 of SDRAM.

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



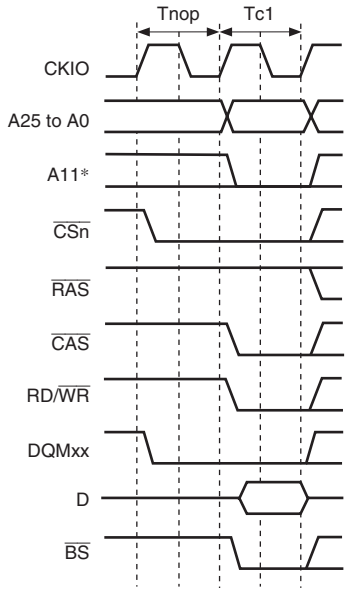
Note: * Address pin to be connected to pin A10 of SDRAM.

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



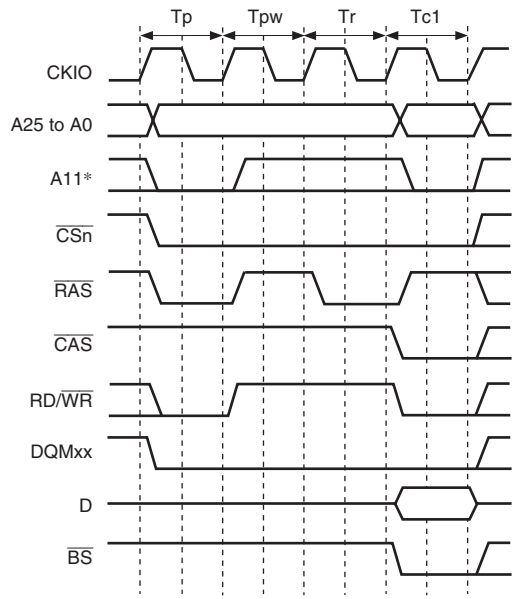
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.22 Single Write Timing (No Auto Precharge)



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

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



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

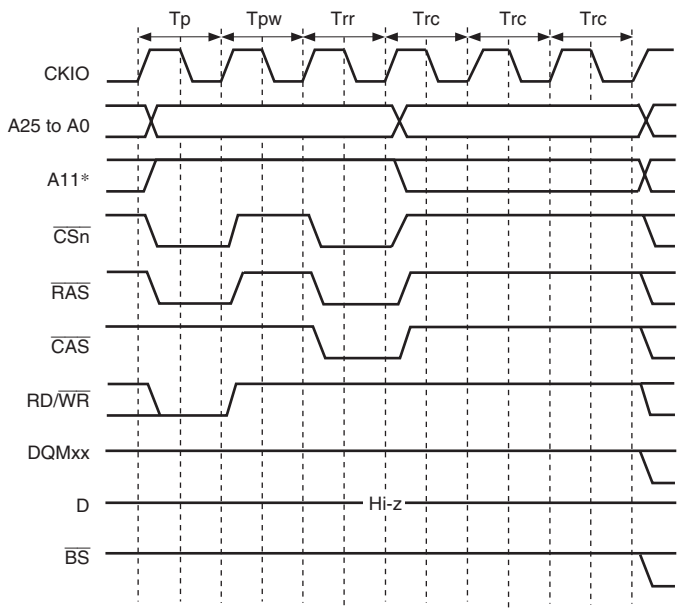
Figure 7.24 Single Write Timing (Bank Active, Different Row Addresses)

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A consecutive refreshing can be performed by setting bits RRC2 to RRC0 in RTCSR. If synchronous DRAM is not accessed for a long period, self-refreshing 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 CKS2 to CKS0 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 given refresh interval for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE, 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-refreshing 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 7.25 shows the auto-refreshing 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 the REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by bits WTRP1 and WTRP0 in CSnWCR. A new command is not issued for the duration of the number of cycles specified by bits WTRC1 and WTRC0 in CSnWCR after the Trr cycle. Bits WTRC1 and WTRC0 in CSnWCR must be set so as to satisfy the SDRAM refreshing cycle time (tRC). A Tpw cycle is inserted between the Tp cycle and Trr cycle when the setting of bits WTRP1 and WTRP0 in CSnWCR is longer than or equal to two cycles.



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

Figure 7.25 Auto-Refreshing Timing

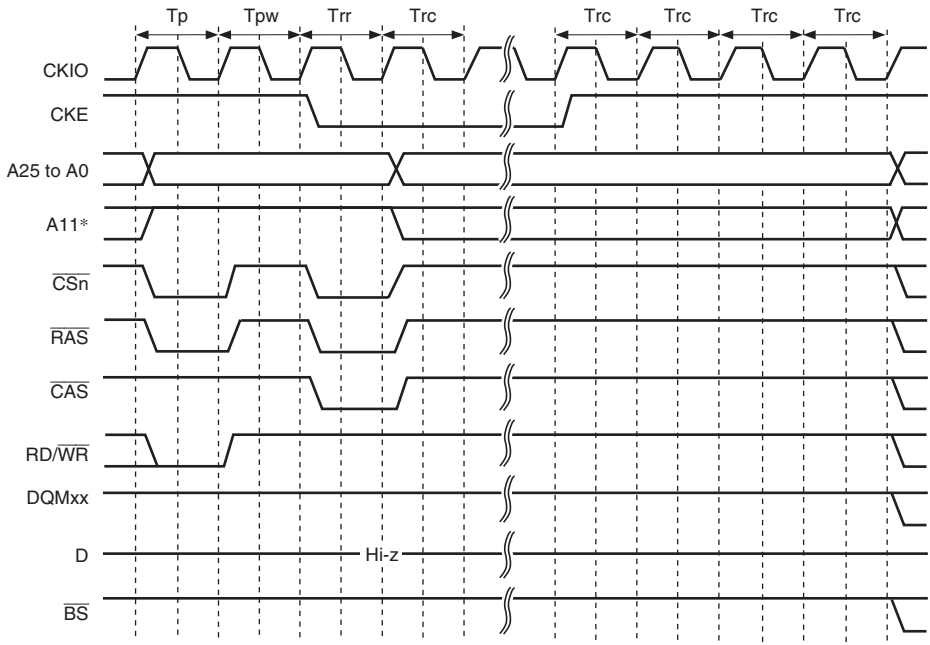
2. Self-refreshing

   When self-refreshing mode is selected, 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, the PALL command is issued in the Tp cycle after the completion of pre-charging the bank. The SELF command is then issued after inserting idle cycles of which the number is specified by bits WTRP1 and WTRP0 in CSnWSR. Synchronous DRAM cannot be accessed while self-refreshing. Self-refreshing mode is cleared by clearing the RMODE bit to 0. After self-refreshing mode has been cleared, command issuance is disabled for the number of cycles specified by bits WTRC1 and WTRC0 in CSnWCR.

   Self-refreshing timing is shown in figure 7.26. Settings must be made immediately after clearing self-refreshing mode so that auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the auto-refreshing mode, only clearing the RMODE bit to 1 resumes auto-refreshing mode. If it takes long time to start the auto-refreshing, setting RTCNT to the value of RTCOR − 1 starts the auto-refreshing immediately.

   After self-refreshing has been set, the self-refreshing mode continues even in standby mode, and is maintained even after recovery from standby mode by an interrupt.

   Since the BSC registers are initialized at a power-on reset, the self-refreshing mode is cleared.



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

Figure 7.26 Self-Refreshing Timing

Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed.

If a new refresh request occurs while the previous refresh request is not performed, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus busy must be prevented.

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after turning the power on. To perform synchronous DRAM initialization correctly, the BSC registers must first be set, followed by writing to the synchronous DRAM mode register. When writing to the synchronous DRAM mode register, 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, write to the address of H'F8FD5000 + X in words. In this operation, the data is ignored. To set burst read/single write, burst read/burst write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written to the addresses shown in table 7.19 in bytes. In this case, 0s are output at the external address pins of A12 or later.

Table 7.19 Access Address for SDRAM Mode Register Write

- Burst read/single write (burst length 1)

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'F8FD5440	H'0000440
	3	H'F8FD5460	H'0000460
32 bits	2	H'F8FD50880	H'0000880
	3	H'F8FD58C0	H'00008C0

- Burst read/burst write (burst length 1)

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

Mode register setting timing is shown in figure 7.27. The PALL command (all bank precharge command) is firstly issued. The REF command (auto-refreshing command) is then issued eight times. The MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by bits WTRP1 and WTRP0 in CSnWCR, are inserted between the PALL and the first REF commands. Idle cycles, of which number is specified by bits WTRC1 and WTRC0 in CSnWCR, are inserted between the REF and REF commands, and between the 8th REF and MRS commands. In addition, one or more idle cycles are inserted between the MRS and the next command.

It is necessary to keep idle time of certain cycles for SDRAM before issuing the PALL command after turning the 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.

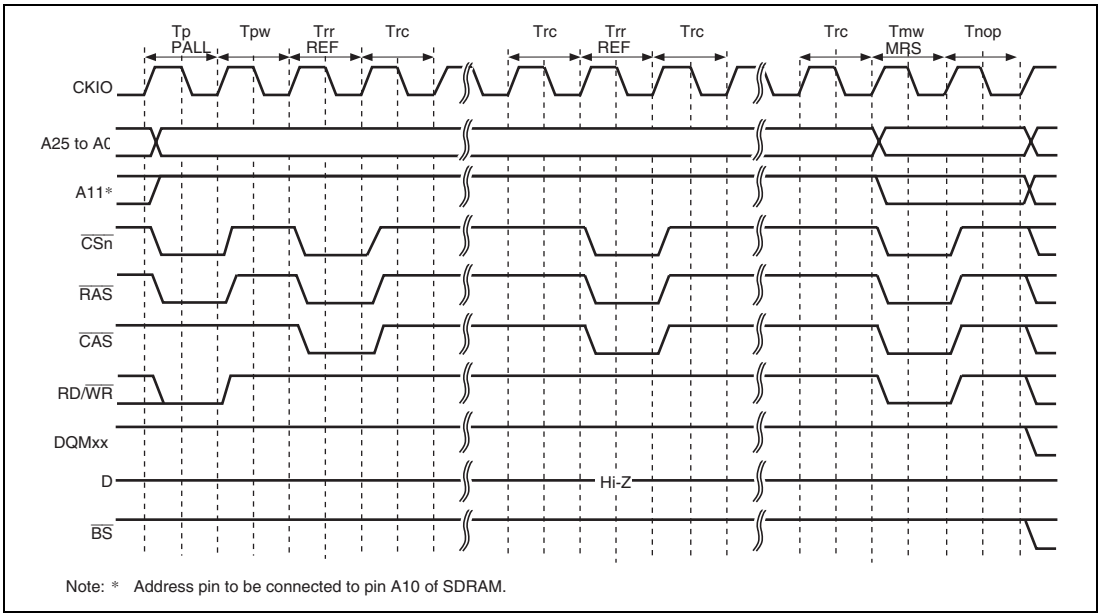


Figure 7.27 Write Timing for SDRAM Mode Register (Based on JEDEC)

7.5.6 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to SRAM which has a byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). This interface is used to access to SRAM which has 16-bit data pins and 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 7.28. 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. The basic access timing is shown in figure 7.29. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be secured by setting bits HW1 to HW0 in CSnWCR. Figure 7.30 shows the access timing when a software wait is specified.

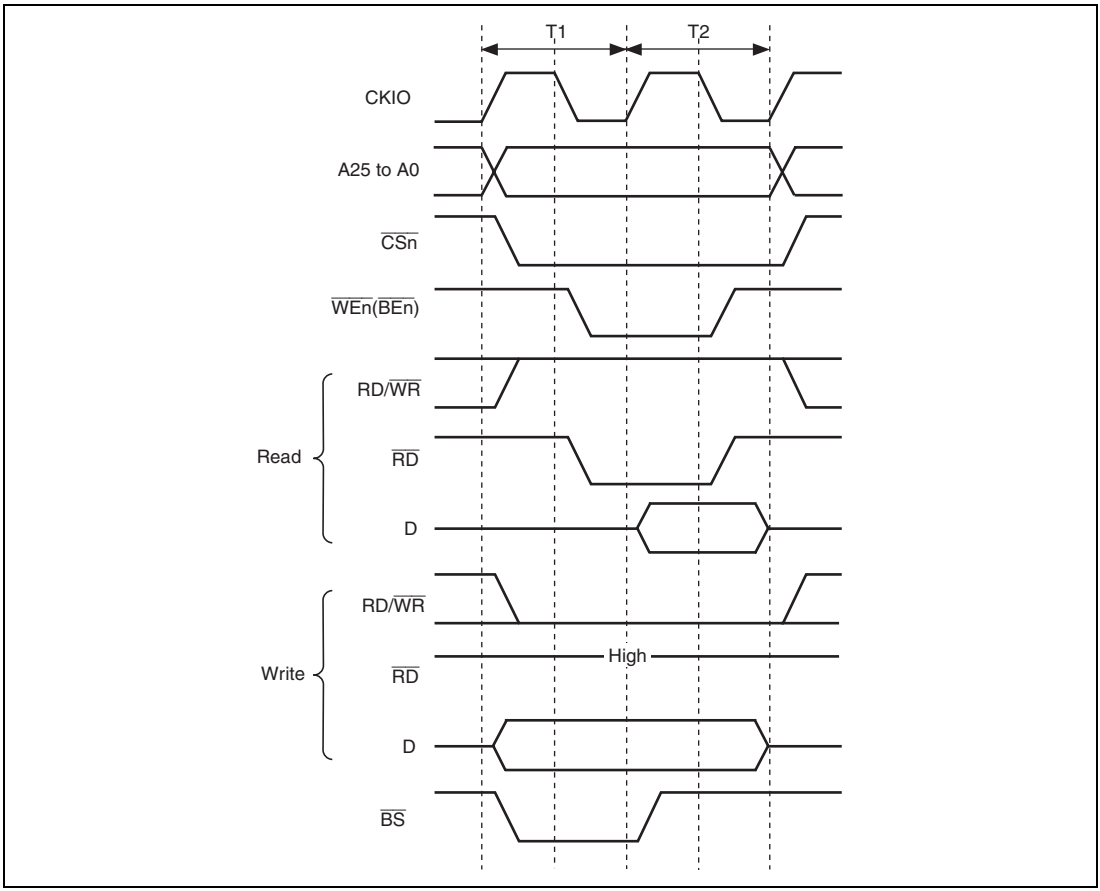


Figure 7.28 Basic Access Timing for Byte-Selection SRAM (BAS = 0)

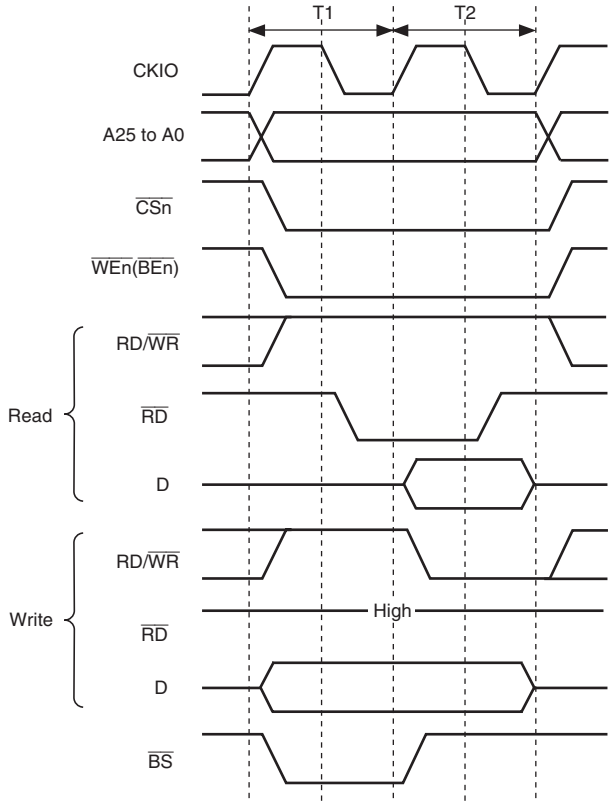


Figure 7.29 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

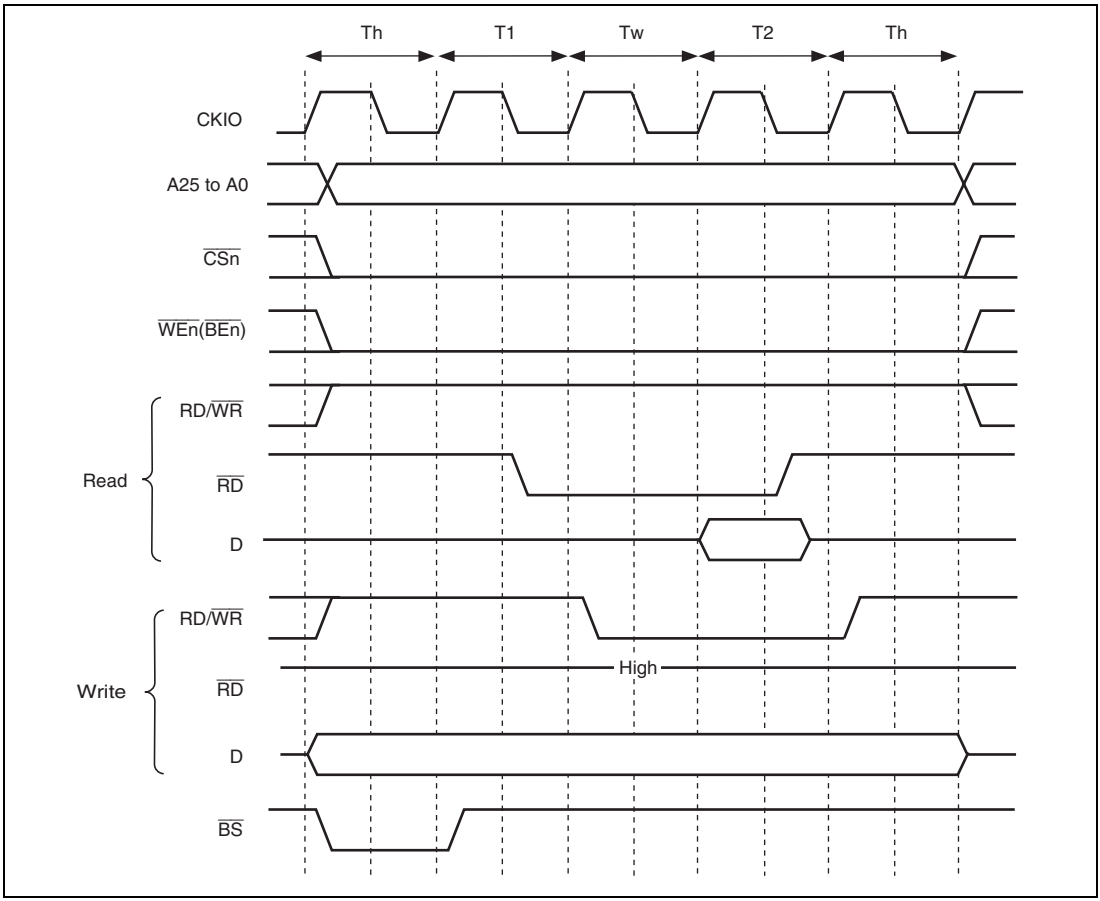


Figure 7.30 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only)

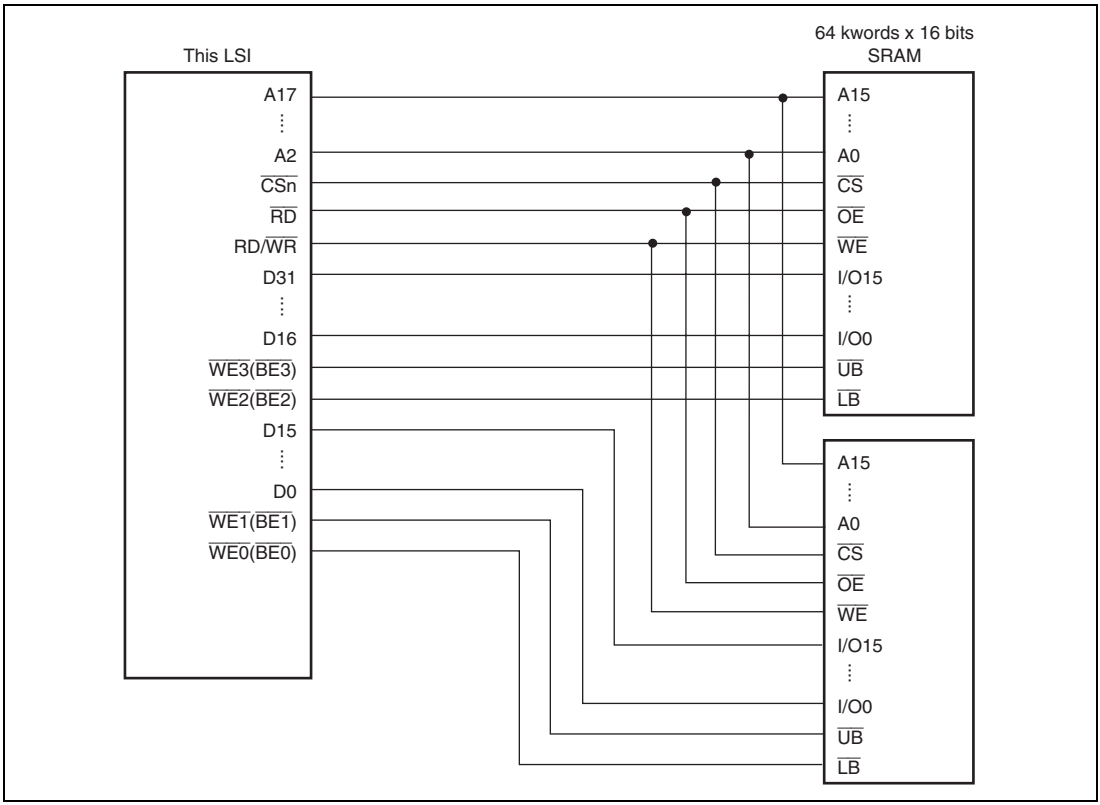


Figure 7.31 Example of Connection with 32-Bit Data-Width Byte-Selection SRAM

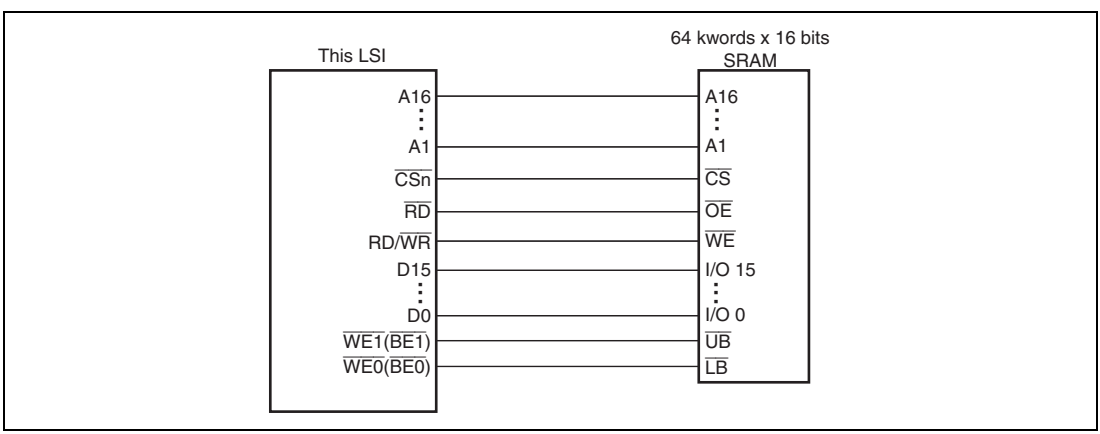


Figure 7.32 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

7.5.7 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 bits TYPE3 to TYPE0 in CSnBCR (n = 5B and 6B) to B'0101. In addition, bits SA1 and SA0 in CSnWCR (n = 5B and 6B) assign the upper or lower 32 Mbytes of each area to an IC memory card or I/O card interface. For example, if bits SA1 and SA0 in 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 bits BSZ1 and BSZ0 in CS5BBCR or CS6BBCR.

Figure 7.33 shows an example of a connection between this LSI and the PCMCIA card. To enable insertion and removal of the PCMCIA card with the system power turned on, tri-state buffers must be connected between the LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in big endian mode is not clearly defined. Consequently, the provided PCMCIA interface in big endian mode is available only for this LSI.

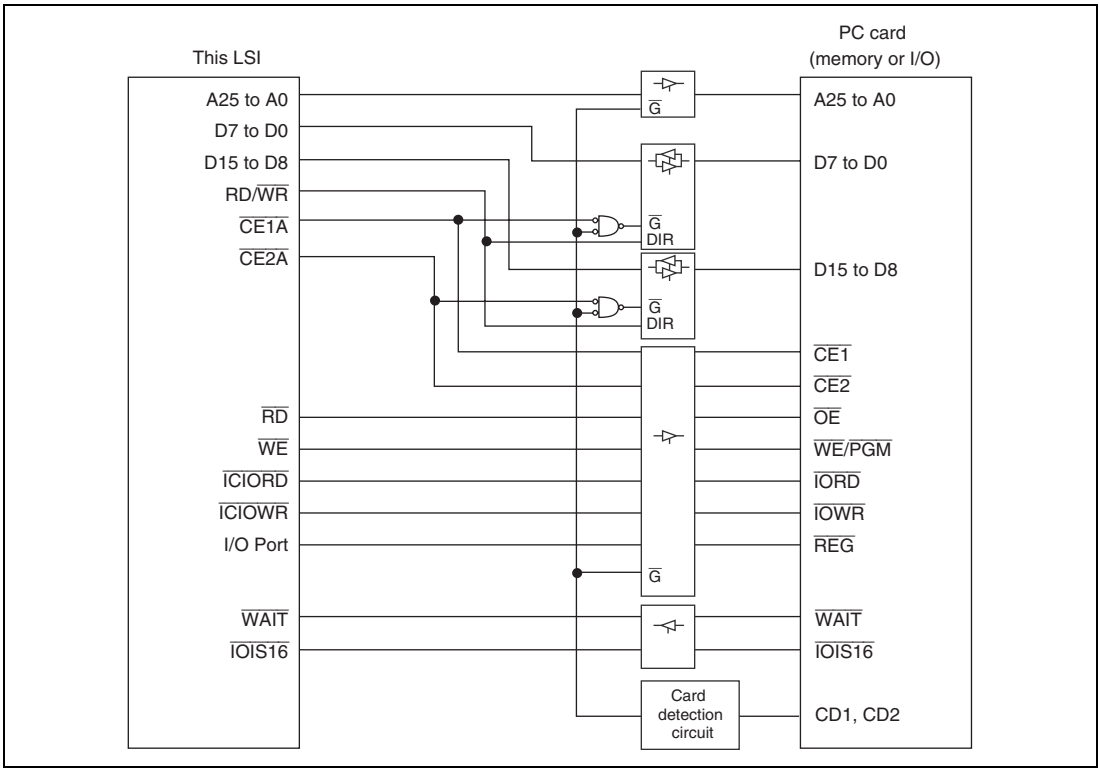


Figure 7.33 Example of PCMCIA Interface Connection

Basic Timing for Memory Card Interface: Figure 7.34 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 with 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), card enable signals ($\overline{CE1A}$, $\overline{CE2A}$, $\overline{CE1B}$, $\overline{CE2B}$), and write data (D15 to D0) to the $\overline{RD}$ and $\overline{WE}$ signals become insufficient. To prevent this error, this 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 $\overline{WAIT}$ pin. Figure 7.35 shows the PCMCIA memory bus wait timing.

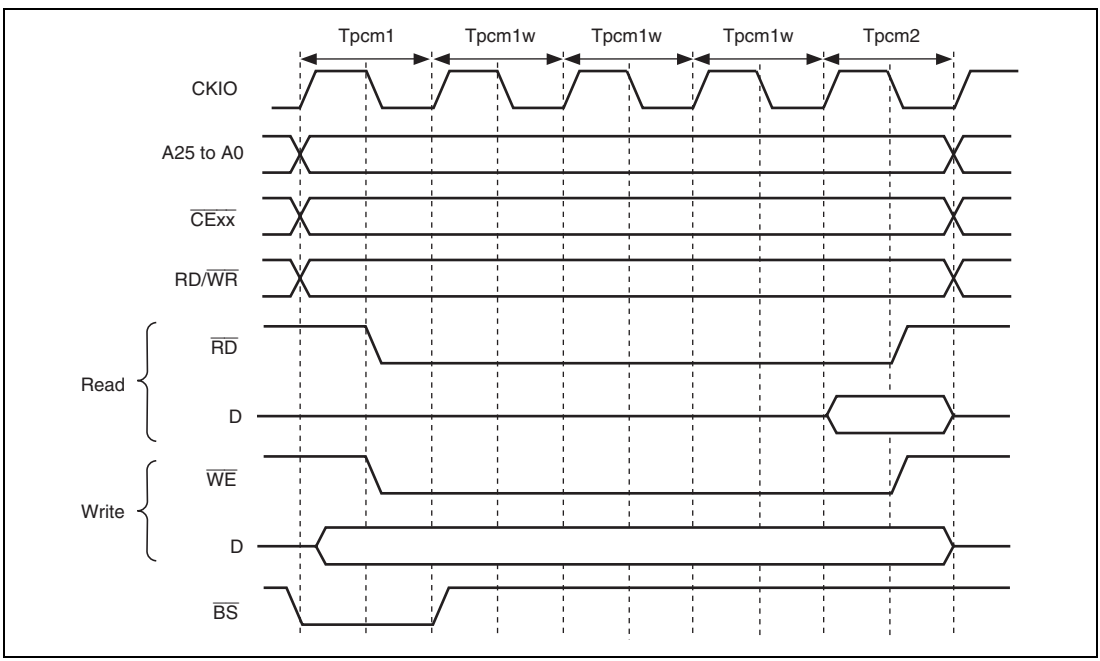


Figure 7.34 Basic Access Timing for PCMCIA Memory Card Interface

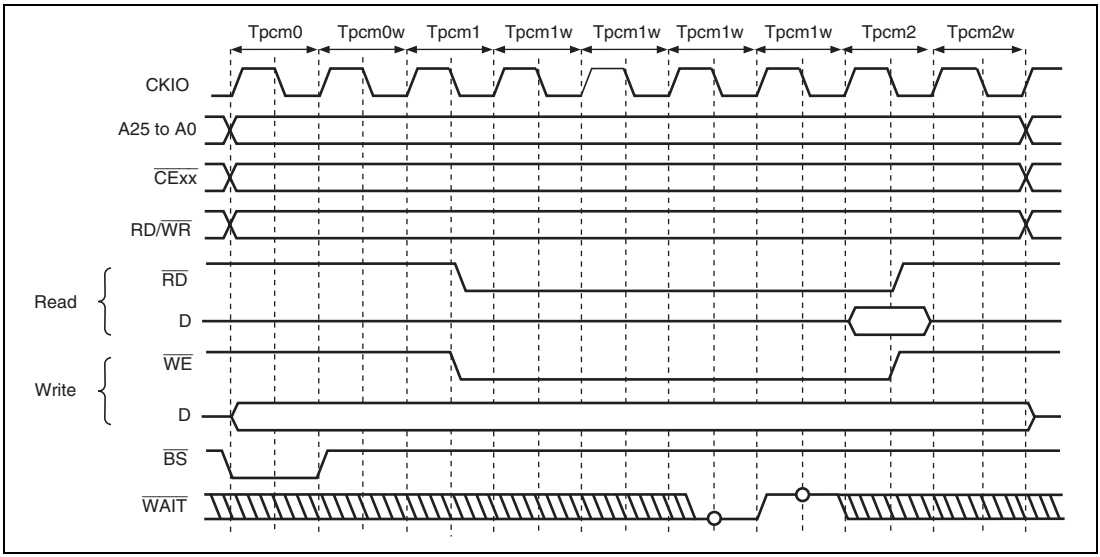


Figure 7.35 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

When 32 Mbytes of the memory space are used as an IC memory card interface, a port is used to generate the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory. When the memory space used for the IC memory card interface is 16 Mbytes or less, pin A24 can be used as the $\overline{\text{REG}}$ signal by allocating 16-Mbyte memory space to each the common memory space and the attribute memory space.

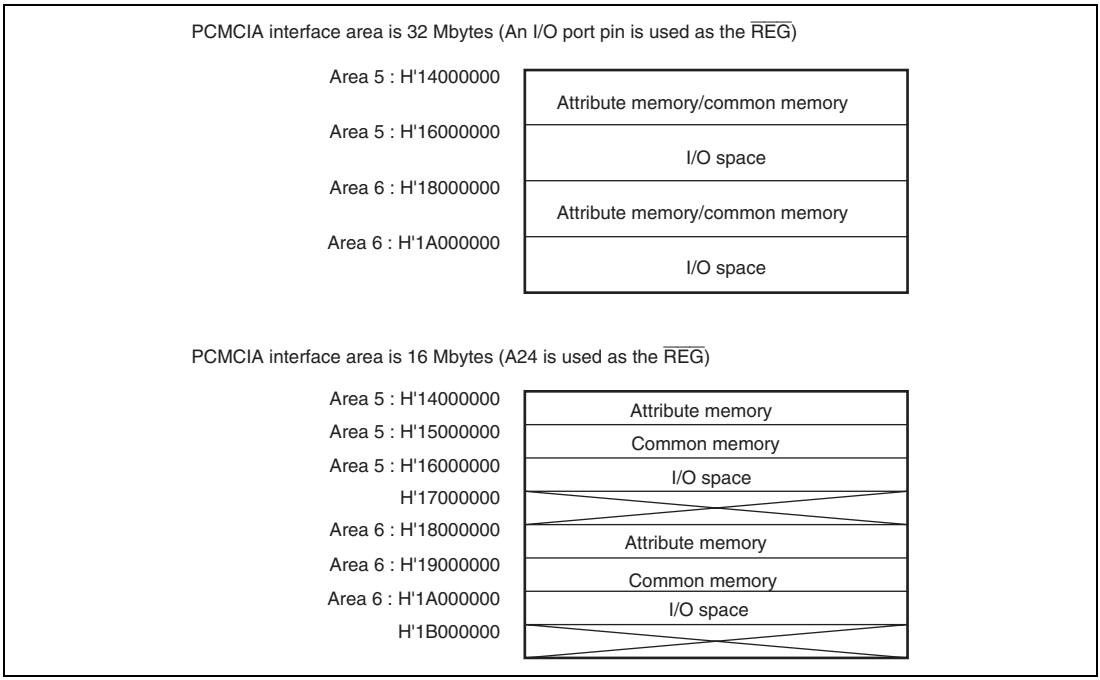


Figure 7.36 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B'10, CS6BWCR.SA[1:0] = B'10)

Basic Timing for I/O Card Interface: Figures 7.37 and 7.38 show the basic timings for the PCMCIA I/O card interface.

The I/O card and IC memory card interfaces are specified by an address to be accessed. When area 5 of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in CS5BWCR are set to 1, the I/O card interface can automatically be specified by accessing the physical addresses from H'16000000 to H'17FFFFFF and from H'14000000 to H'15FFFFFF. When area 6 of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in CS6BWCR are set to 1, the I/O card interface can automatically be specified by accessing the physical addresses from H'1A000000 to H'1BFFFFFF and from H'18000000 to H'19FFFFFF.

Note that areas to be accessed as the PCMCIA I/O card must be non-cached (space P2).

If the PCMCIA card is accessed as an I/O card in little endian mode, dynamic bus sizing for the I/O bus can be achieved using the $\overline{\text{IOIS16}}$ signal. If the $\overline{\text{IOIS16}}$ signal is driven 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 units of eight bits in the I/O bus cycle to be executed.

The $\overline{\text{IOIS16}}$ signal is sampled at the falling edge of the CKIO signal in the Tpci0, Tpci0w, and Tpci1 cycles when bits TED3 to TED0 are specified as 1.5 cycles or more, and is reflected in the CE2 signal 1.5 cycles after the CKIO sampling point. Bits TED3 to TED0 must be specified appropriately to satisfy the setup time of the PC card from $\overline{\text{ICIORD}}$ and $\overline{\text{ICIOWR}}$ to CEn.

Figure 7.39 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.

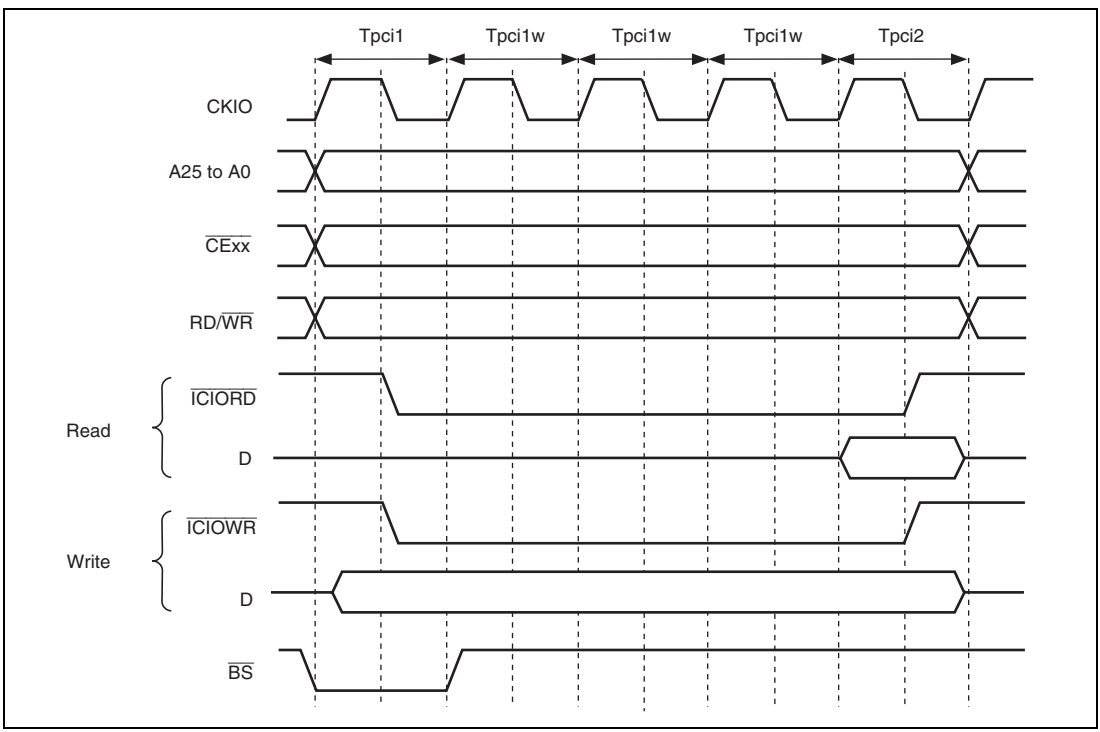


Figure 7.37 Basic Timing for PCMCIA I/O Card Interface

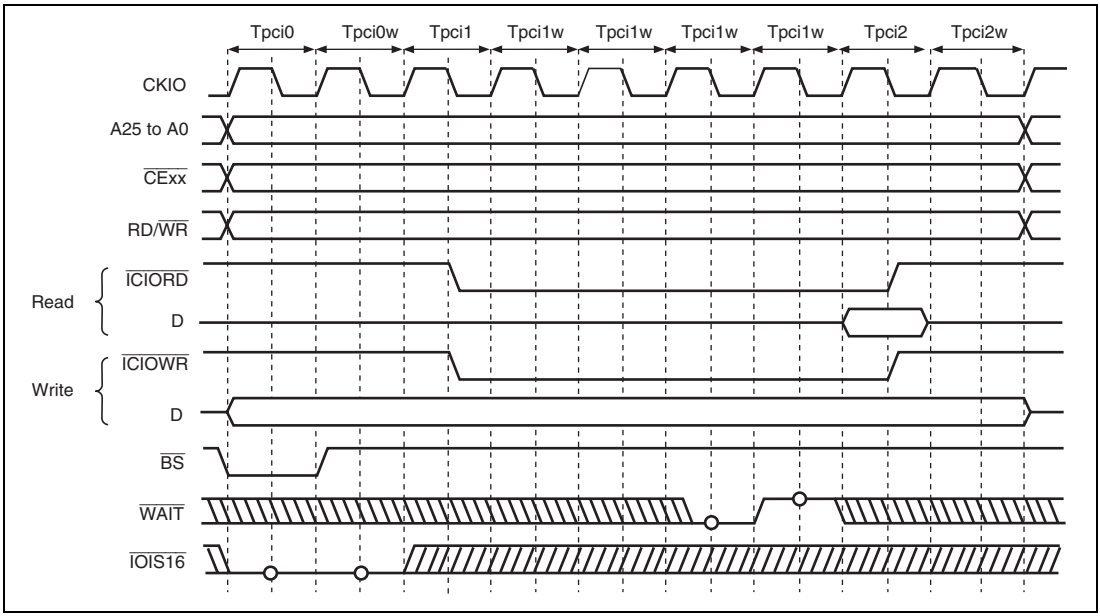


Figure 7.38 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

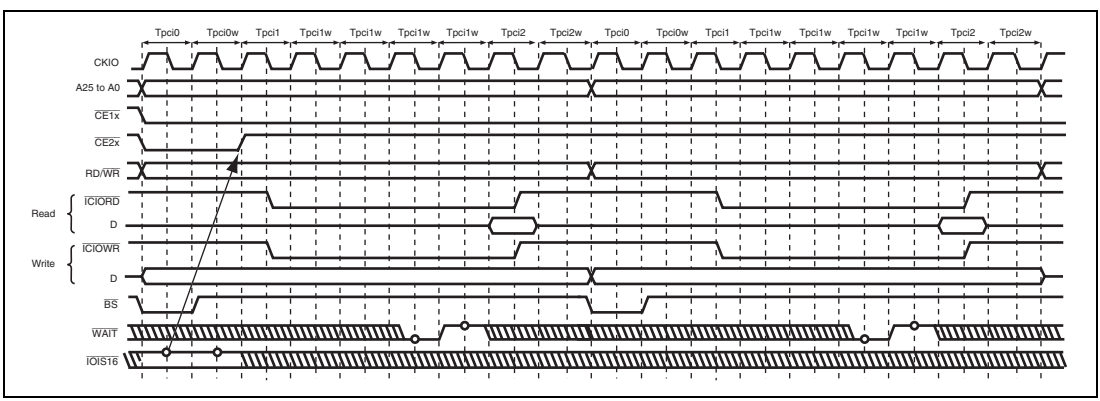


Figure 7.39 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

7.5.8 Wait between Access Cycles

Data output in the previous cycle may conflict with that in the next cycle because the buffer-off timing of devices with slow access speed cannot be operated to satisfy the higher operating frequency of LSIs. As a result of these conflict, the reliability of the device is low and malfunctions may occur. This LSI has a function that avoids data conflicts by inserting wait cycles between consecutive access cycles.

The number of wait cycles between access cycles can be set by bits IWW1 and IWW0, bits IWRWD1 and IWRWD0, bits IWRWS1 and IWRWS0, bits IWRRD1 and IWRRD0, and bits IWRRS1 and IWRRS0 in CSnBCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Consecutive accesses are write-read or write-write
2. Consecutive accesses are read-write for different areas
3. Consecutive accesses are read-write for the same area
4. Consecutive accesses are read-read for different areas
5. Consecutive accesses are read-read for the same area

7.5.9 Others

Reset: The bus state controller (BSC) can be initialized completely only by a power-on 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 mode and sleep mode, control registers of the BSC are not initialized.

Some flash memories may stipulate a minimum time from reset release to the first access. To ensure this minimum time, the BSC supports a 7-bit counter (RWTCNT). At a power-on reset, the RWTCNT contents are cleared to 0. After a power-on reset, RWTCNT is counted up in synchronization with the CKIO signal and an external access will not be generated until RWTCNT is counted up to H'007F.

Access from the Site of the LSI Internal Bus Master: There are three types of LSI 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 BSC are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memory other than the cache memory and debugging modules such as the UBC are connected to both 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 other than the CPU such as the DMAC and E-DMAC 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 consecutive 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 consecutive longword accesses to perform a cache fill operation on the external interface. For a cache-through 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 updated. In this case, data to be updated 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 updated. 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 updated but an actual write is performed via the internal bus.

Since the BSC incorporates a 1-stage write buffer, the BSC can execute an access via the 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 cycle 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 8 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a peripheral clock (Pϕ), a bus clock (Bϕ), and a clock (Mϕ) for the on-chip IEEE802.3-PHY (physical layer device, hereinafter called PHY). The CPG consists of an oscillator, PLL circuits, and divider circuits.

8.1 Features

- Four clock modes

  Selection of four clock modes depending on the frequency of a clock source and whether a crystal resonator or external clock input is in use.
- Four clocks generated independently

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

  Frequencies of the internal clock, peripheral clock, and clock for the PHY can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using the frequency control register (FRQCR) and PHY clock frequency control register (MCLKCR) settings.
- Power-down mode control

  The clock can be stopped in sleep mode and software standby mode and specific modules can be stopped using the module standby function.

A block diagram of the CPG is shown in figure 8.1.

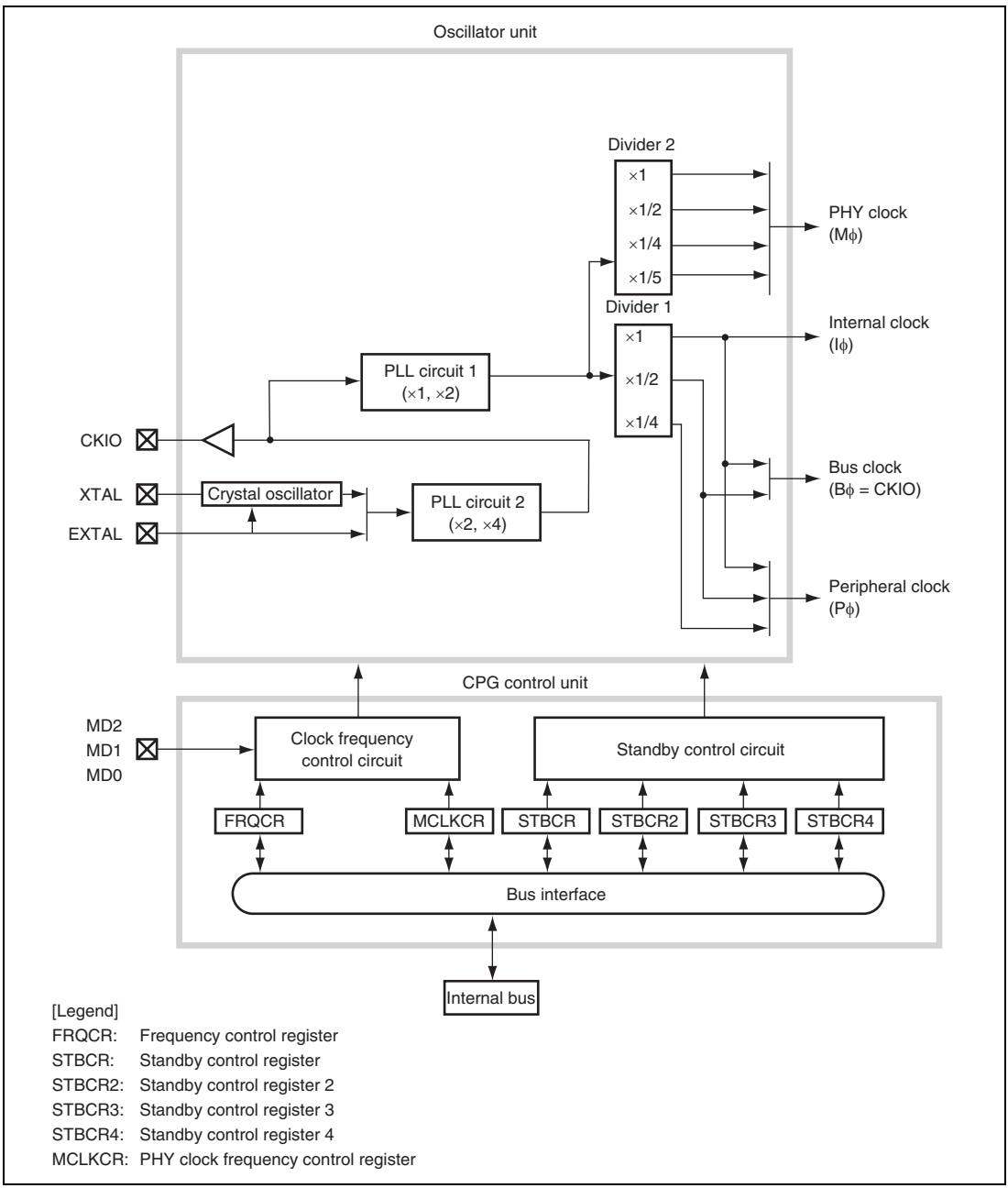


Figure 8.1 Block Diagram of CPG

The clock pulse generator blocks function as follows:

PLL Circuit 1: PLL circuit 1 leaves the input clock frequency from the PLL circuit 2 unchanged or doubles it. The multiplication ratio is set by the frequency control register. The phase of the rising edge of the internal clock is controlled so that it will match the phase of the rising edge of the CKIO pin.

PLL Circuit 2: PLL circuit 2 doubles or quadruples the clock frequency input from the crystal oscillator or the EXTAL pin. The multiplication ratio is fixed for each clock operating mode. The clock operating mode is set with pins MD0, MD1, or MD2.

Crystal Oscillator: The crystal oscillator is an oscillator circuit when a crystal resonator is connected to the XTAL and EXTAL pins. The crystal oscillator can be used by setting the clock operating mode.

Divider 1: Divider 1 generates clocks with the frequencies used by the internal clock, peripheral clock, and bus clock. The frequency output as the internal clock is always the same as that of the devider1 output. The frequency output as the bus clock is automatically selected so that it is the same as the frequency of the CKIO signal according to the multiplication ratio of PLL circuit 1. The frequencies can be 1, 1/2, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the frequency of the CKIO pin. The division ratio is set in the frequency control register.

Divider 2: Divider 2 generates a clock that is supplied to the on-chip PHY. Divider 2 must output 25-MHz frequency for the on-chip PHY that requires 25-MHz clock. The output clock of divider 2 can be 1, 1/2, 1/4, or 1/5 times the output frequency of PLL circuit 1. The division ratio is set in the PHY clock frequency control register.

Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using pins MD0, MD1, and MD2, the frequency control register, and PHY clock frequency control register.

Standby Control Circuit: The standby control circuit controls the state of the on-chip oscillator circuit and other modules during clock switching and in software standby mode.

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 peripheral clock.

Standby Control Register: The standby control register has bits for controlling the power-down modes. For details, see section 10, Power-Down Modes.

PHY Clock Frequency Control Register: The PHY clock frequency control register sets the frequency division ratio of the PHY clock.

8.2 Input/Output Pins

Table 8.1 shows the CPG pin configuration.

Table 8.1 Pin Configuration

Pin Name	Abbr.	I/O	Description
Mode control pins*	MD0	Input	Set the clock operating mode.
	MD1	Input	Set the clock operating mode.
	MD2	Input	Set the clock operating mode.
Clock input pins	XTAL	Output	Connects a crystal resonator.
	EXTAL	Input	Connects a crystal resonator or an external clock.
Clock output pin	CKIO	Output	Outputs an external clock.

Note: * The values of these mode control pins are sampled only at a power-on reset or in a software standby with the MDCHG bit in STBCR to 1. This can prevent the erroneous operation of this LSI.

8.3 Clock Operating Modes

Table 8.2 shows the relationship between the mode control pins (MD2 to MD0) combinations and the clock operating modes. Table 8.3 shows the usable frequency ranges in the clock operating modes and the frequency range of the input clock.

Table 8.2 Mode Control Pins and Clock Operating Modes

Clock Operating Mode	Pin Values			Clock I/O				
	MD2	MD1	MD0	Source	Output	PLL2	PLL1	CKIO Frequency
1	0	0	1	EXTAL	CKIO	ON (×4)	ON (×1, ×2)	(EXTAL) × 4
2	0	1	0	Crystal resonator	CKIO	ON (×4)	ON (×1, ×2)	(Crystal resonator) × 4
5	1	0	1	EXTAL	CKIO	ON (×2)	ON (×1, ×2)	(EXTAL) × 2
6	1	1	0	Crystal resonator	CKIO	ON (×2)	ON (×1, ×2)	(Crystal resonator) × 2

Mode 1: The frequency of the external clock input from the EXTAL pin is quadrupled by PLL circuit 2, and then the clock is supplied to this LSI. Since the input clock frequency ranging 10 MHz to 15.625 MHz can be used, the CKIO frequency ranges from 40 MHz to 62.5 MHz.

Mode 2: The frequency of the on-chip crystal oscillator output is quadrupled by PLL circuit 2, and then the clock is supplied to this LSI. Since the crystal resonator frequency ranging 10 MHz to 15.625 MHz can be used, the CKIO frequency ranges from 40 MHz to 62.5 MHz.

Mode 5: The frequency of the external clock from the EXTAL pin is doubled by PLL circuit 2, and then the clock is supplied to this LSI. Since the input clock frequency ranging 10 MHz to 25 MHz, the CKIO frequency ranges from 20 MHz to 50 MHz.

Mode 6: The frequency of the on-chip crystal oscillator output is doubled by PLL circuit 2, and then the clock is supplied to the LSI. Since the crystal oscillation frequency ranging 10 MHz to 25 MHz can be used, the CKIO frequency ranges from 20 MHz to 50 MHz.

Table 8.3 Possible Combination of Clock Modes and FRQCR Values

Mode	FRQCR Register Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Input Clock Frequency Range (Pϕ must be equal to or lower than 31.25 MHz)	CKIO Pin Frequency Range
1 or 2	H'1000	ON (×1)	ON (×4)	4:4:4	10 MHz to 15.625 MHz	40 MHz to 62.5 MHz
	H'1001	ON (×1)	ON (×4)	4:4:2		
	H'1003	ON (×1)	ON (×4)	4:4:1		
	H'1101	ON (×2)	ON (×4)	8:4:4		
	H'1103	ON (×2)	ON (×4)	8:4:2		
5 or 6	H'1000	ON (×1)	ON (×2)	2:2:2	10 MHz to 25 MHz	20 MHz to 50 MHz
	H'1001	ON (×1)	ON (×2)	2:2:1		
	H'1003	ON (×1)	ON (×2)	2:2:1/2		
	H'1101	ON (×2)	ON (×2)	4:2:2		
	H'1103	ON (×2)	ON (×2)	4:2:1		

Note: * Input clock is assumed to be 1.

Cautions:

1. The internal clock frequency is the product of the frequency of the CKIO pin and the frequency multiplication ratio of PLL circuit 1.
2. The peripheral clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1.

   Do not set the peripheral clock frequency lower than the CKIO pin frequency.
3. The PHY clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2.
4. ×1, ×1/2, or ×1/4 can be used as the division ratio of divider 1. This is set by the frequency control register.
5. The division ratio of divider 2 is selected from ×1, ×1/2, ×1/4, or ×1/5. This is set by the PHY clock frequency control register.
6. The output frequency of PLL circuit 1 is the product of the frequency of the CKIO pin and the multiplication ratio of PLL circuit 1. It is set by the frequency control register.
7. The bus clock frequency is always set to be equal to the frequency of the CKIO pin.
8. The clock mode, the FRQCR register value, and the frequency of the input clock should be decided to satisfy the range of operating frequency specified in section 25, Electrical Characteristics, with referring to table 8.3.

8.4 Register Descriptions

The CPG has the following registers.

For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Frequency control register (FRQCR)
- PHY clock frequency control register (MCLKCR)

8.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register that specifies whether a clock is output from the CKIO pin in standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the peripheral clock. Only word access can be used on FRQCR.

FRQCR is initialized by a power-on reset due to the external input signal. However, it is not initialized by a power-on reset due to a WDT overflow.

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 Specifies whether a clock continues to be output from the CKIO pin or the output level of the CKIO signal is fixed when leaving software standby mode. The CKIO output is fixed low when this bit is set to 0. Therefore, the malfunction of external circuits because of an unstable CKIO clock when leaving software standby mode can be prevented. 0: Output level of the CKIO signal is fixed low in software standby mode. 1: Clock input to the EXTAL pin is output to the CKIO pin during software standby mode in clock mode 1 or 5. However, the output level of the CKIO pin is fixed low for two cycles of Pϕ when changing from the normal mode to the standby mode. This prevents hazard which occurs when the source of the CKIO signal is changed from the PLL2 output to the EXTAL signal.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	STC2	0	R/W	PLL Circuit 1 Frequency Multiplication Ratio
9	STC1	0	R/W	000: ×1
8	STC0	0	R/W	001: ×2 Other values: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	1	R/W	Specify the division ratio of the peripheral clock frequency with respect to the output frequency of PLL circuit 1.
0	PFC0	1	R/W	
				000: ×1
				001: ×1/2
				011: ×1/4
				Other values: Setting prohibited

8.4.2 PHY Clock Frequency Control Register (MCLKCR)

MCLKCR is an 8-bit readable/writable register. This register must be written to in words. The upper byte of the word data must be H'5A and the lower byte is the write data.

Bit	Bit Name	Initial Value	R/W	Description
7	FLSCS1	0	R/W	Source Clock Select
6	FLSCS0	1	R/W	Select the source clock.
				00: PLL1 output clock
				01: PLL1 output clock
				10: Setting prohibited
				11: Setting prohibited
5 to 3	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.
2	FLDIVS2	0	R/W	Divider Select
1	FLDIVS1	1	R/W	Set the division ratio of PLL1 output.
0	FLDIVS0	1	R/W	000: ×1
				001: ×1/2
				011: ×1/4
				100: ×1/5
				Other values: Setting prohibited

8.4.3 Usage Notes

- MCLKCR is used only to generate clocks for the on-chip PHY. However, dedicated clocks input from the CK_PHY pin can be used instead of Mϕ, which is set by MCLKCR as clocks for the on-chip PHY.
- When changing the contents of MCLKCR or FRQCR, make sure that the on-chip PHY is in the reset state. Otherwise, a hazard may be temporarily generated on Mϕ output. To use Mϕ as clocks for the on-chip PHY, assert the module reset of the on-chip PHY after the contents of MCLKCR or FRQCR have been changed.

8.5 Changing Frequency

The internal clock frequency can be changed by changing the multiplication ratio of PLL circuit 1. The peripheral clock frequency can be changed either by changing the multiplication ratio of PLL circuit 1 or by changing the division ratio of divider 1. All of these are controlled by software through the frequency control register. The methods are described below.

8.5.1 Changing Multiplication Ratio

The PLL lock time must be preserved when the multiplication ratio of PLL circuit 1 is changed. The on-chip WDT counts for preserving the PLL lock time.

1. In the initial state, the multiplication ratio of PLL circuit 1 is 1.
2. Set a value that satisfies the given PLL lock time in the WDT and stop the WDT. The following must be set.
   — TME bit in WTCSR = 0: WDT stops
   — Bits CKS2 to CKS0 in WTCSR: Division ratio of WDT count clock
   — WTCNT: Initial counter value
3. Set the desired value in bits STC2 to STC0 while the MDCHG bit in STBCR is 0. The division ratio can also be set in bits PFC2 to PFC0.
4. This LSI pauses internally and the WDT starts incrementing. The internal and peripheral clocks both stop and only the WDT is supplied with the clock. The clock will continue to be output on the CKIO pin.
5. Supply of the specified clock starts at a WDT count overflow, and this LSI starts operating again. The WDT stops after it overflows.

Notes: 1. When the MDCHG bit in STBCR is set to 1, changing the FRQCR value has no effect on the operation immediately. For details, see section 8.5.3, Changing Clock Operating Mode.
2. The multiplication ratio should be changed after completion of the operation, if the on-chip peripheral module is operating. The internal and peripheral clocks are stopped during the multiplication ratio is changed. The communication error may occur by the peripheral module communicating to the external IC, and the time error may occur by the timer unit (except the WDT). The edge detection of external interrupts (NMI and IRQ7 to IRQ0) cannot be performed.

8.5.2 Changing Division Ratio

The WDT will not count unless the multiplication ratio is changed simultaneously.

1. In the initial state, PFC2 to PFC0 = 011.
2. Set the desired values in bits PFC2 to PFC0 while the MDCHG bit in STBCR is 0. The values that can be set are limited by the clock mode and the multiplication ratio of PLL circuit 1. Note that if the wrong value is set, this LSI will malfunction.
3. The clock is immediately changed to the new division ratio.

Note: When the MDCHG bit in STBCR is set to 1, changing the FRQCR value has no effect on the operation immediately. For details, see section 8.5.3, Changing Clock Operating Mode.

8.5.3 Changing Clock Operating Mode

The values of the mode control pins (MD2 to MD0) that define a clock operating mode are fetched at a power-on reset and software standby while the MDCHG bit in STBCR is set to 1 register.

Even if changing the FRQCR with the MDCHG bit set to 1, the clock mode cannot immediately be changed to the specified clock mode. This change can be reflected as a multiplication ratio or a division ratio after leaving software standby mode to change operating modes. Reducing the PLL settling time without changing again the multiplication ratio after the operating mode changing is possible by the use of this.

The procedures for the mode change using software standby mode are described below.

1. Set bits MD2 to MD0 to the desired clock operating mode.
2. Set both the STBY and MDCHG bits in STBCR to 1.
3. Set the adequate value to the WDT so that the given oscillation settling time can be satisfied. Then stop the WDT.
4. Set FRQCR to the desired mode. Set bits STC2 to STC0 to the desired multiplication ratio. At this time, a division ratio can be set in bits PFC2 to PFC0. During the operation before the mode change, the clock cannot be changed to the specified clock.
5. Enter software standby mode using the SLEEP instruction.
6. Leave software standby mode using an interrupt.
7. After leaving software standby mode, this LSI starts the operation with the value of FRQCR that has been set before the mode change.

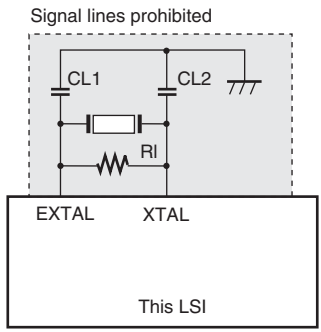
Notes: 1. Pins MD2 to MD0 should be set during the operation before the mode change or during software standby mode before requesting an interrupt.

2. Clear the STBY bit in STBCR in the exception handling routine for the interrupt in step 6. Otherwise, software standby mode is entered again. For details, see section 10.5.2, Canceling Software Standby Mode.

3. Once bits STC2 to STC0 are changed, the clock is not switched to the specified clock even if only bits PFC2 to PFC0 are changed. When bits STC2 to STC0 are changed after the MDCHG bit has been set to 1, the FRQCR setting must not be made until the clock mode is changed.

8.6 Notes on Board Design

Note on Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and feedback resistor R1 as close to the XTAL and EXTAL pins as possible. In addition, to minimize induction and thus obtain oscillation at the correct frequency, the capacitors to be attached to the resonator must be grounded to the same ground. Do not bring wiring patterns close to these components.



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

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

Figure 8.2 Note on Using a Crystal Resonator

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

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

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

Notes on Using a PLL Oscillator Circuit: In the Vcc and Vss connection pattern for the PLL, signal lines from the board power supply pins must be as short as possible and pattern width must be as wide as possible to reduce inductive interference.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin VccQ should not supply the same resources on the board if at all possible.

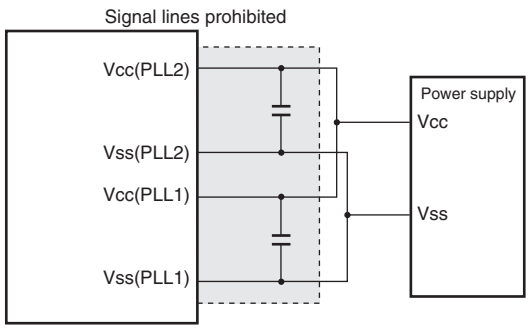


Figure 8.3 Note on Using a PLL Oscillator Circuit

Section 9 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT) that can reset this LSI by the overflow of the counter when the value of the counter has not been updated because of a system runaway.

The WDT is a single-channel timer that uses a peripheral clock as an input and counts the clock settling time when leaving software standby mode and temporary standby state, such as frequency changes. It can also be used as an interval timer.

9.1 Features

The WDT has the following features:

- Can be used to ensure the clock settling time.
  The WDT can be used when leaving software standby mode and the temporary standby state which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Internal resets in watchdog timer mode
  Internal resets are generated when the counter overflows.
- Interrupts are generated in interval timer mode
  Interval timer interrupts are generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock can be chosen.

Figures 9.1 is a block diagram of the WDT.

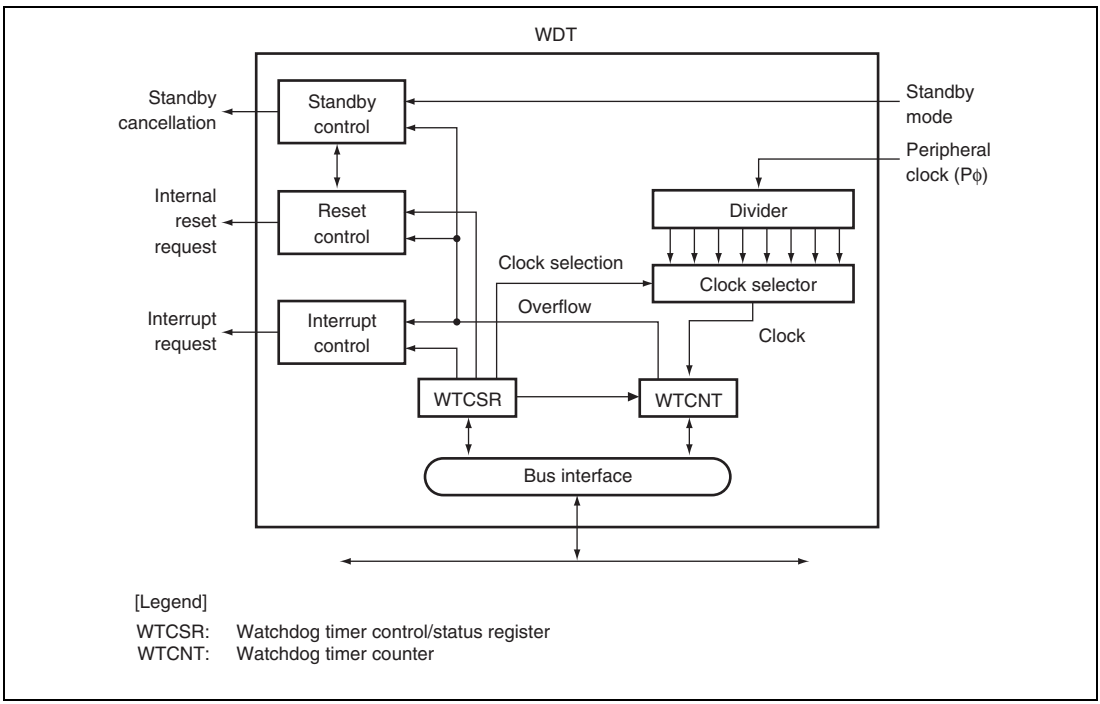


Figure 9.1 Block Diagram of WDT

9.2 Register Descriptions

The WDT has the following two registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

9.2.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that 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 not initialized by an internal power-on reset due to the WDT overflow. WTCNT is initialized to H'00 by a power-on reset input to the pin and an H-UDI reset.

Use a word access to write to WTCNT, with H'5A in the upper byte. Use a byte access to read WTCNT.

Note: The writing method for WTCNT differs from other registers so that the WTCNT value cannot be changed accidentally. For details, see section 9.2.3, Notes on Register Access.

9.2.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the counting, bits to select the timer mode and overflow flags, and enable bits.

WTCSR holds its value in the internal reset state due to the WDT overflow. WTCSR is initialized to H'00 by a power-on reset input to the pin and an H-UDI reset. To use it for counting the clock settling time when leaving software standby mode, WTCSR holds its value after a 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: The writing method for WTCNT differs from other registers so that the WTCNT value cannot be changed accidentally. For details, see section 9.2.3, Notes on Register Access.

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 software standby mode or when changing the clock frequency. 0: Timer disabled: Count-up stops and WTCNT value is retained 1: Timer enabled
6	WT/IT	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Interval timer mode 1: Watchdog timer mode Note: If WT/IT is modified when the WDT is operating, the up-count may not be performed correctly.
5	—	0	R	Reserved This bit is always red as 0. The write value should always be 0.
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that 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 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 (Pϕ). The overflow period that is shown inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 25 MHz.
0	CKS0	0	R/W	000: Pϕ (10 μs) 001: Pϕ /4 (41 μs) 010: Pϕ /16 (164 μs) 011: Pϕ /32 (328 μs) 100: Pϕ /64 (655 μs) 101: Pϕ /256 (2.62 ms) 110: Pϕ /1024 (10.49 ms) 111: Pϕ /4096 (41.94 ms) Note: If bits CKS2 to CKS0 are modified when the WDT is operating, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not operating.

9.2.3 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 is 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 9.2. 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.

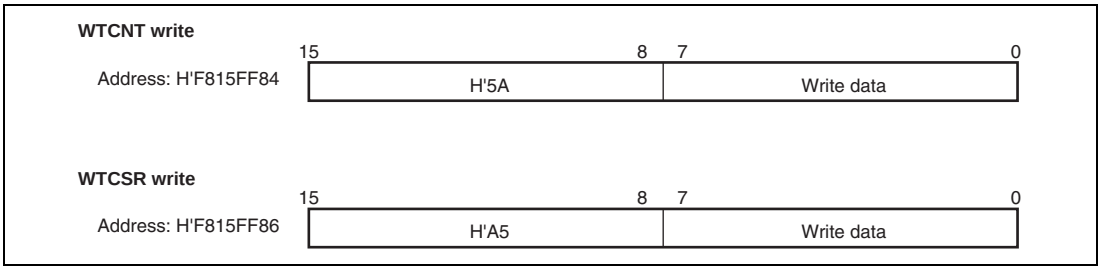


Figure 9.2 Writing to WTCNT and WTCSR

9.3 WDT Operation

9.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI interrupt or external interrupt (IRQ). The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RES}}$ pin low until the clock stabilizes.)

1. Before transition to software standby mode, always clear the TME bit in WTCSR to 0. 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 software standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the change of input levels of the NMI or IRQ pin.
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 when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in STBCR to 0 in the interrupt processing program and this will stop the WDT to count. When the STBY bit remains 1, the LSI again enters software standby mode when the WDT has counted up to H'80. This software standby mode can be canceled by a power-on reset.

9.3.2 Changing Frequency

To change the multiplication ratio of PLL circuit 1, 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 bits STC2 to STC0 in the frequency control register (FRQCR) is written, the processor stops 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 when this happens.
5. WTCNT stops at the value of H'00.
6. Before changing WTCNT after the execution of the frequency change instruction, always confirm that the value of WTCNT is H'00 by reading WTCNT.

9.3.3 Using Watchdog Timer Mode

1. Set the WT/IT bit in WTCSR to 1, set the type of count clock in bits CKS2 to CKS0, 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 a power-on reset. WTCNT then resumes counting.

9.3.4 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/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 WTCNT overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The WTCNT then resumes counting.

9.4 Usage Notes

Pay attention to the following points when using the WDT.

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

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

This phenomenon does not occur when the WDT is used in watchdog timer mode.

Section 10 Power-Down Modes

This LSI supports the following power-down modes: sleep mode, software standby mode, module standby mode.

10.1 Features

- Supports sleep mode, software standby mode, and module standby

10.1.1 Types of Power-Down Modes

This LSI has the following power-down modes.

- Sleep mode
- Software standby mode
- Module standby mode (cache, U-memory, UBC, H-UDI, and on-chip peripheral modules)

Table 10.1 shows the methods to make a transition 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 Method**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**Canceling Procedure**
Sleep	Execute SLEEP instruction with STBY bit in STBCR cleared to 0.	Runs	Halts	Held	Halts (contents remained)	Run	Held	• Interrupt other than user break • Reset
Software standby	Execute SLEEP instruction with STBY bit in STBCR set to 1.	Halts	Halts	Held	Halts (contents remained)	Halt	Held	• NMI, IRQ • Reset
Module standby	Set MSTP bits in STBCR2 to STBCR4 to 1.	Runs	Runs	Held	Specified module halts (contents remained)	Specified module halts	Held	• Clear MSTP bit to 0 • Power-on reset

10.2 Input/Output Pins

Table 10.2 lists the pins used for the power-down modes.

Table 10.2 Pin Configuration

Pin Name	Abbr.	I/O	Description
Reset input pin	$\overline{\text{RES}}$	Input	Reset input signal. Reset by low level.

10.3 Register Descriptions

There are following registers used for the power-down modes. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)
- Standby control register 4 (STBCR4)

10.3.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction makes this LSI sleep mode 1: Executing SLEEP instruction makes this LSI software standby mode
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDCHG	0	R/W	MD2 to MD0 Pin Control Specifies whether or not the values of pins MD2 to MD0 are reflected in software standby mode. The values of pins MD2 to MD0 are reflected at returning from software standby mode by an interrupt when the MDCHG bit has been set to 1. 0: The values of pins MD2 to MO0 are not reflected in software standby mode. 1: The values of pins MD2 to MD0 are reflected in software standby mode.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.3.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop Bit 10 When this bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI operates 1: Clock supply to H-UDI halted
6	MSTP9	0	R/W	Module Stop Bit 9 When this bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC operates 1: Clock supply to UBC halted
5	MSTP8	0	R/W	Module Stop Bit 8 When this bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC operates 1: Clock supply to DMAC halted
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MSTP5	0	R/W	Module Stop Bit 5 When this bit is set to 1, the supply of the clock to the cache memory is halted. 0: Cache memory operates 1: Clock supply to cache memory halted
1	MSTP4	0	R/W	Module Stop Bit 4 When this bit is set to 1, the supply of the clock to the U memory is halted. 0: U memory operates 1: Clock supply to the U memory halted

Bit	Bit Name	Initial Value	R/W	Description
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

10.3.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	MSTP15	0	R/W	Module Stop Bit 15 When this bit is set to 1, the supply of the clock to the CMT is halted. 0: CMT operates 1: Clock supply to CMT halted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP13	0	R/W	Module Stop Bit 13 When this bit is set to 1, the supply of the clock to the SCIF2 is halted. 0: SCIF2 operates 1: Clock supply to SCIF2 halted
1	MSTP12	0	R/W	Module Stop Bit 12 When this bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 halted

Bit	Bit Name	Initial Value	R/W	Description
0	MSTP11	0	R/W	Module Stop Bit 11 When this bit is set to 1, the supply of the clock to the SCIF0 is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 halted

10.3.4 Standby Control Register 4 (STBCR4)

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

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

Bit	Bit Name	Initial Value	R/W	Description
0	MSTP19	0	R/W	Module Stop Bit 19 When this bit is set to 1, the supply of the clock to the EtherC and E-DMAC is halted. 0: EtherC and E-DMAC operate 1: Clock supply to EtherC and E-DMAC halted

10.4 Sleep Mode

10.4.1 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 operate in sleep mode and the clock continues to be output to the CKIO pin.

10.4.2 Canceling Sleep Mode

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

Canceling with Interrupt: When a user-break, NMI, H-UDI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. When the priority level of an IRQ or on-chip peripheral module interrupt is lower than the interrupt mask level set in the status register (SR) of the CPU, an interrupt request is not accepted preventing sleep mode from being canceled.

Canceling with Reset: Sleep mode is canceled by a power-on reset or an H-UDI reset.

10.5 Software Standby Mode

10.5.1 Transition to Software Standby Mode

This LSI switches from a program execution state to software standby mode by executing the SLEEP instruction when the STBY bit in STBCR is 1. In 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
Clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Direct memory access controller (DMAC)	—	All registers
Ethernet controller (EtherC)	—	All registers
Direct memory access controller for Ethernet controller (E-DMAC)	—	All registers
I/O port	—	All registers
User debugging interface (H-UDI)	—	All registers
Serial communication interface with FIFO (SCIF0 to SCIF2)	—	All registers
Compare match timer (CMT0 and CMT1)	All registers	—
Host interface (HIF)	—	All registers
Serial IO with FIFO (SIOF)	—	All registers
Ethernet physical layer transceiver (PHY)	Some registers*	Some registers*

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

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the timer control register (WTCSR) of the WDT to 0 to stop the WDT.
2. Set the timer counter (WTCNT) of the WDT to 0 and bits CKS2 to CKS0 in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY bit in STBCR to 1, execute the SLEEP instruction.
4. Software standby mode is entered and the clocks within this LSI are halted.

10.5.2 Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI, IRQ) or a reset.

Canceling with Interrupt: The WDT can be used for hot starts. When an NMI or IRQ interrupt is detected, the clock will be supplied to the entire LSI and software standby mode will be canceled after the time set in the timer control/status register of the WDT has elapsed. Interrupt exception handling is then executed. After the branch to the interrupt handling routine, clear the STBY bit in STBCR. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continues operation and a transition is made to software standby mode* when it reaches H'80. This function prevents data destruction due to the voltage rise by an unstable power supply voltage.

IRQ cancels the software standby mode when the input condition matches the specified detect condition while the IRQn1S and IRQn0S bits in IRQCR are not B'00 (settings other than the low level detection). When the priority level of an IRQ interrupt is lower than the interrupt mask level set in the status register (SR) of the CPU, the execution of the instruction following the SLEEP instruction starts again after the cancellation of software standby mode. When the priority level of an IRQ interrupt is higher than the interrupt mask level set in the status register (SR) of the CPU, IRQ interrupt exception handling is executed after the cancellation of software standby mode.

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

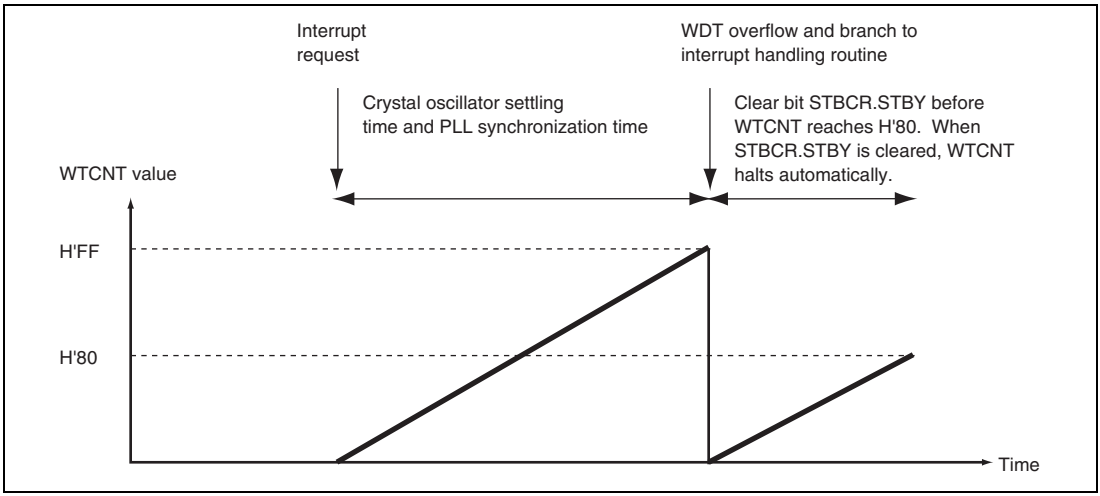


Figure 10.1 Canceling Standby Mode with STBY Bit in STBCR

Canceling with Reset: Software standby mode is canceled by a power-on reset. Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

10.6 Module Standby Mode

10.6.1 Transition to Module Standby Mode

Setting the MSTP bits in the standby control registers (STBCR2 to STBCR4) to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal mode.

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

10.6.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits in STBCR2 to STBCR4 to 0, or by a power-on reset.

Section 11 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. This LSI has one MAC layer interface.

The Ethernet controller is connected to the direct memory access controller for Ethernet controller (E-DMAC) inside this LSI, and carries out high-speed data transfer to and from the memory.

Figure 11.1 shows a configuration of the EtherC.

11.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps receive/transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard MII (Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Conforms to IEEE802.3x flow control

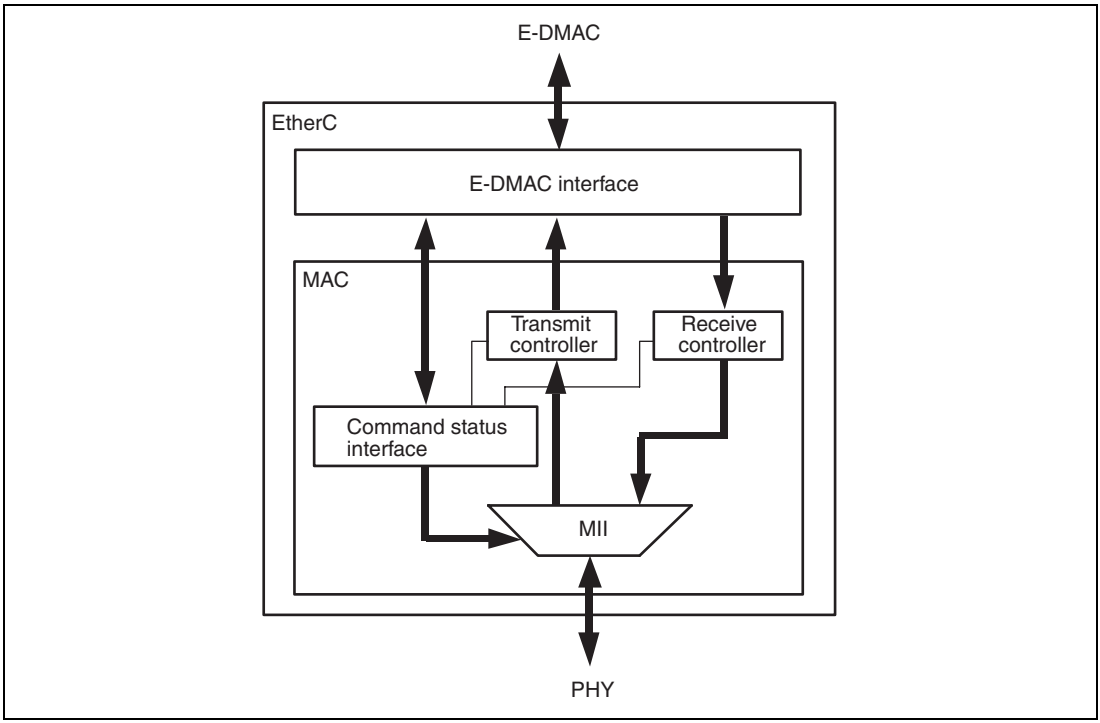


Figure 11.1 Configuration of EtherC

11.2 Input/Output Pins

Table 11.1 lists the pin configuration of the EtherC.

Table 11.1 Pin Configuration

Port	Abbreviation	I/O	Function
0	TX-CLK*	Input	Transmit Clock Timing reference signal for the TX-EN, MII_TXD3 to MII_TXD0, TX-ER signals
0	RX-CLK*	Input	Receive Clock Timing reference signal for the RX-DV, MII_RXD3 to MII_RXD0, RX-ER signals
0	TX-EN*	Output	Transmit Enable Indicates that transmit data is ready on pins MII_TXD3 to MII_TXD0.
0	MII_TXD3 to MII_TXD0*	Output	Transmit Data 4-bit transmit data
0	TX-ER*	Output	Transmit Error Notifies the PHY-LSI of error during transmission
0	RX-DV*	Input	Receive Data Valid Indicates that valid receive data is on pins MII_RXD3 to MII_RXD0.
0	MII_RXD3 to MII_RXD0*	Input	Receive Data 4-bit receive data
0	RX-ER*	Input	Receive Error Identifies error state occurred during data reception.
0	CRS	Input	Carrier Detection Carrier detection signal
0	COL	Input	Collision Detection Collision detection signal
0	MDC	Output	Management Data Clock Reference clock signal for information transfer via MDIO
0	MDIO	Input/ Output	Management Data I/O Bidirectional signal for exchange of management information between this LSI and PHY

Port	Abbreviation	I/O	Function
0	LNKSTA	Input	Link Status Inputs link status from PHY
0	EXOUT	Output	General-Purpose External Output Signal indicating value of register-bit (ECMR0-ELB)
0	WOL	Output	Wake-On-LAN Signal indicating reception of Magic Packet

Note: * MII signal conforming to IEEE802.3u

11.3 Register Description

The EtherC has the following registers. For details on addresses and access sizes of registers, see section 24, List of Registers.

MAC Layer Interface Control Registers:

- EtherC mode register (ECMR)
- EtherC status register (ECSR)
- EtherC interrupt permission register (ECSIPR)
- PHY interface register (PIR)
- MAC address high register (MAHR)
- MAC address low register (MALR)
- Receive frame length register (RFLR)
- PHY status register (PSR)
- Transmit retry over counter register (TROCR)
- Delayed collision detect counter register (CDCR)
- Lost carrier counter register (LCCR)
- Carrier not detect counter register (CNDCR)
- CRC error frame counter register (CEFCR)
- Frame receive error counter register (FRECR)
- Too-short frame receive counter register (TSFRCR)
- Too-long frame receive counter register (TLFRCR)
- Residual-bit frame counter register (RFCR)
- Multicast address frame counter register (MAFCR)
- IPG register (IPGR)
- Automatic PAUSE frame set register (APR)
- Manual PAUSE frame set register (MPR)
- PAUSE frame retransfer count set register (TPAUSER)

11.3.1 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 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	ZPF	0	R/W	0 time parameter PAUSE Frame Use Enable 0: Disables PAUSE frame control in which the TIME parameter is 0. The next frame is transmitted after the time indicated by the Timer value has elapsed. When the EtherC receives a PAUSE frame with the time indicated by the Timer value set to 0, the PAUSE frame is discarded. 1: Enables PAUSE frame control in which the TIME parameter is 0. A PAUSE frame with the Timer value set to 0 is transmitted when the number of data in the receive FIFO is less than the FCFTR value before the time indicated by the Timer value has not elapsed. When the EtherC receives a PAUSE frame with the time indicated by the Timer value set to 0, the transmit wait state is canceled.
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the E-DMAC 1: PAUSE frame is transferred to the E-DMAC
17	RXF	0	R/W	Receive Flow Control Operating Mode 0: PAUSE frame detection function is disabled 1: Receive flow control function is enabled

Bit	Bit Name	Initial Value	R/W	Description
16	TXF	0	R/W	Transmit Flow Control Operating mode 0: Transmit flow control function is disabled 1: Transmit flow control function is enabled
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PRCEF	0	R/W	Permit Receive CRC Error Frame 0: A frame with a CRC error is received as a frame with an error. 1: A frame with a CRC error is received as a frame without an error. For a frame with an error, a CRC error is reflected in the ECSR of the E-DMAC and the status of the receive descriptor. For a frame without an error, the frame is received as normal frame.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RE	0	R/W	Reception Enable If a frame is being received when this bit is switched from receive function enabled (RE = 1) to disabled (RE = 0), the receive function will be enabled until reception of the corresponding frame is completed. 0: Receive function is disabled 1: Receive function is enabled

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmission Enable If a frame is being transmitted when this bit is switched from transmit function enabled (TE = 1) to disabled (TE = 0), 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: When DM = 1, data loopback is performed inside the MAC in the EtherC.
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 LSI, using the EXOUT pin. In order for LSI loopback to be implemented using this function, the 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

Bit	Bit Name	Initial Value	R/W	Description
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

11.3.2 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 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 the ECI bit in EESR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PSRTO	0	R/W	PAUSE Frame Retransfer Retry Over Indicates that during the retransfer of PAUSE frames when the flow control is enabled, the number of retries has exceeded the upper limit set in the automatic PAUSE frame retransfer count set register (TPAUSER). 0: Number of PAUSE frame retransfers has not exceeded the upper limit 1: Number of PAUSE frame retransfers has exceeded the upper limit
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	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from the PHY has changed from high to low or low to high. To check the current Link state, refer to the LMON bit in the PHY status register (PSR). 0: Changes in the LNKSTA signal are not detected 1: Changes in the LNKSTA signal are 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
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY has detected an illegal carrier on the line. If a change in the signal input from the PHY occurs before the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY used. 0: LSI has not detected an illegal carrier on the line 1: LSI has detected an illegal carrier on the line

11.3.3 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 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PSRTOIP	0	R/W	PAUSE Frame Retransfer Retry Over Interrupt Enable 0: Interrupt notification by the PSRTO bit is disabled 1: Interrupt notification by the PSRTO bit is enabled
3	—	0	R	Reserved This bit is 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

11.3.4 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY 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 11.4.4, Accessing MII Registers.

11.3.5 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.

11.3.6 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit 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.

11.3.7 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 and including the CRC data. Frame contents from the destination address up to and including 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

11.3.8 PHY Status Register (PSR)

PSR is a read-only register that can read interface signals from the PHY.

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 to the LNKSTA pin. For the polarity, refer to the PHY specifications to be connected.

11.3.9 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be 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.

11.3.10 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 32-bit counter that indicates the number of delayed collisions on all lines from a start of 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 delayed collisions on all lines from a start of transmission.

11.3.11 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during 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.

11.3.12 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier could not be 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.

11.3.13 CRC Error Frame 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.

11.3.14 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames input from the PHY for which a receive error was indicated by the RX-ER pin. 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.

11.3.15 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames of fewer than 64 bytes 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.

11.3.16 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.

11.3.17 Residual-Bit Frame 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.

11.3.18 Multicast Address Frame Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specified 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.

11.3.19 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 11.4.6, 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 (Initial value) : : H'1F: 144-bit time

11.3.20 Automatic PAUSE Frame Set Register (APR)

APR sets the TIME parameter value of the automatic PAUSE frame. When transmitting the automatic PAUSE frame, the value set in this register is used as the TIME parameter of the PAUSE frame.

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

11.3.21 Manual PAUSE Frame Set Register (MPR)

MPR sets the TIME parameter value of the manual PAUSE frame. When transmitting the manual PAUSE frame, the value set to this register is used as the TIME parameter of the PAUSE frame.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MP15 to MP0	All 0	R/W	Manual PAUSE Sets the TIME parameter value of the manual PAUSE frame. At this time, 1 bit means 512-bit time. Read values are undefined.

11.3.22 PAUSE Frame Retransfer Count Set Register (TPAUSER)

TPAUSER sets the upper limit of the number of times of the PAUSE frame retransfer. TPAUSER must not be changed while the transmitting function is enabled.

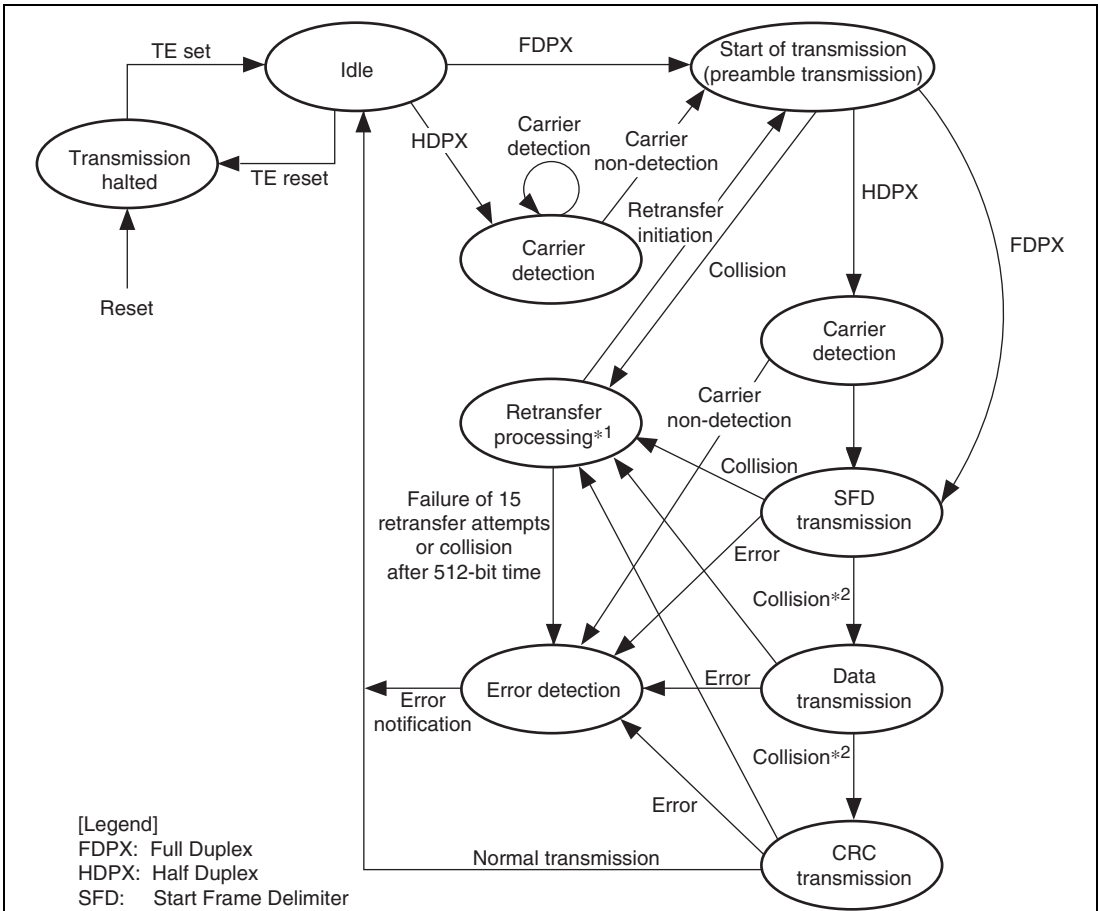
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE15 to TPAUSE0	All 0	R/W	Upper Limit of the Number of Times of PAUSE Frame Retransfer H'0000: Unlimited number of times of retransfer H'0001: Retransfer once : : H'FFFF: Number of times of retransfer is 65535

11.4 Operation

The overview of the Ethernet controller (EtherC) are shown below. The EtherC transmits and receives PAUSE frames conforming to the Ethernet/IEEE802.3x frames.

11.4.1 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 11.3 shows the state transition of the EtherC transmitter.



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 11.2 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit E-DMAC, the EtherC sends the 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.

11.4.2 Reception

The EtherC receiver separates the frame data (MII into preamble, SFD, DA (destination address), SA (Source address), type/length, Data, and CRC data) and outputs DA, SA, type/length, Data to the E-DMAC. Figure 11.3 shows the state transitions of the EtherC receiver.

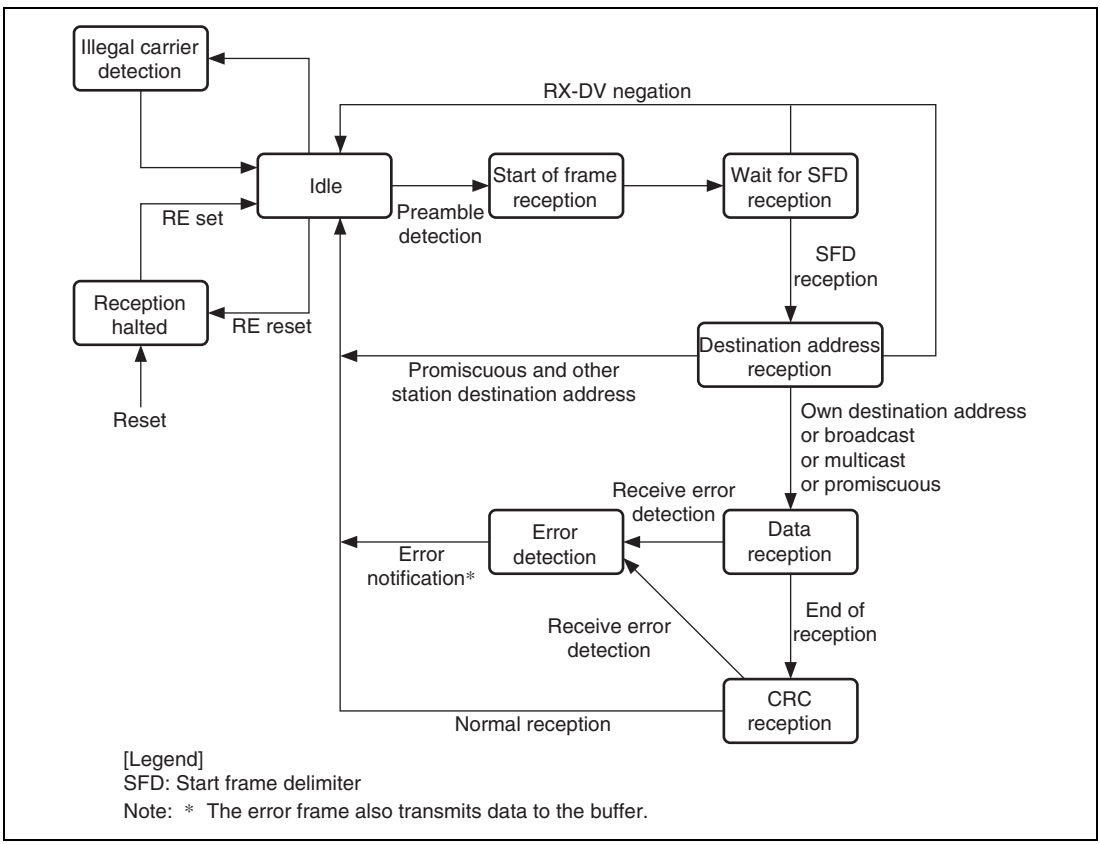


Figure 11.3 EtherC Receiver State Transmissions

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.

11.4.3 MII Frame Timing

Each MII Frame timing is shown in figure 11.4.

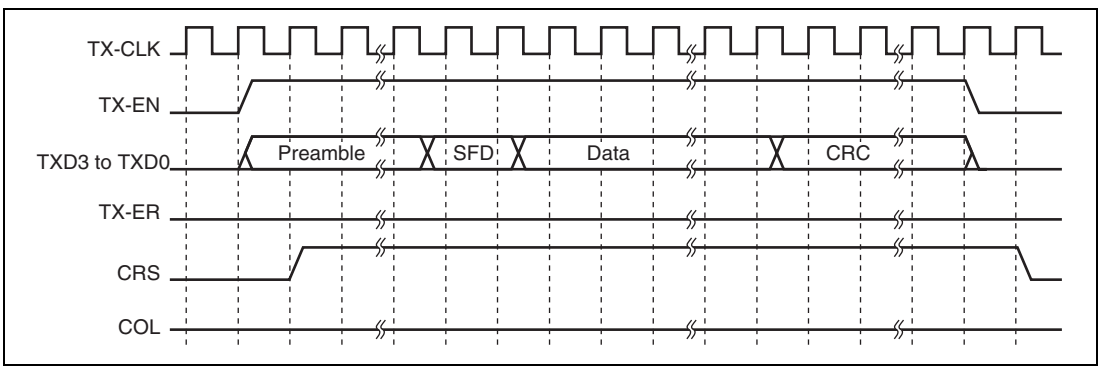


Figure 11.4 (1) MII Frame Transmit Timing (Normal Transmission)

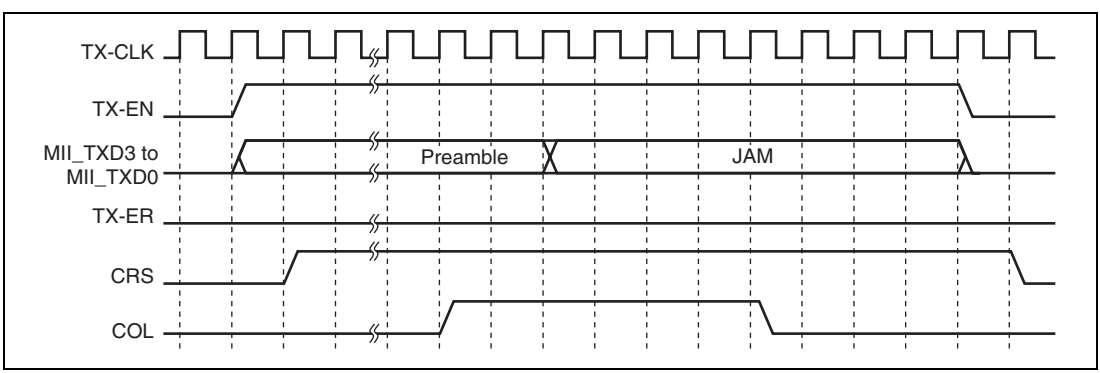


Figure 11.4 (2) MII Frame Transmit Timing (Collision)

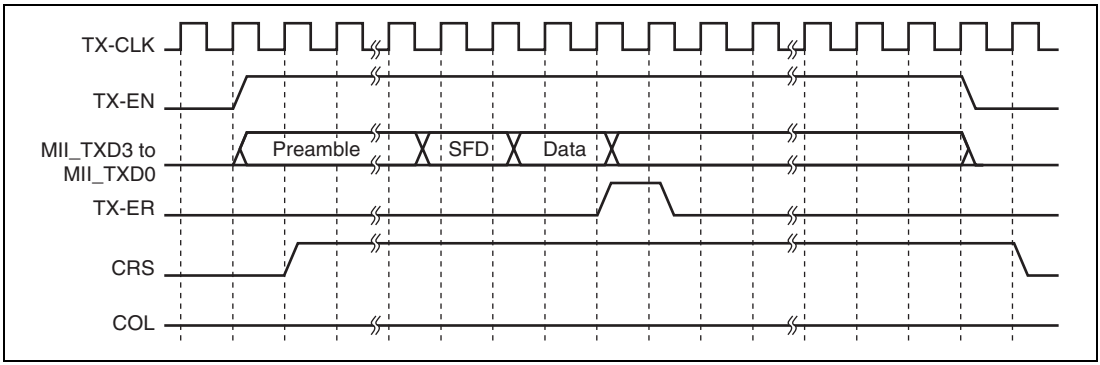


Figure 11.4 (3) MII Frame Transmit Timing (Transmit Error)

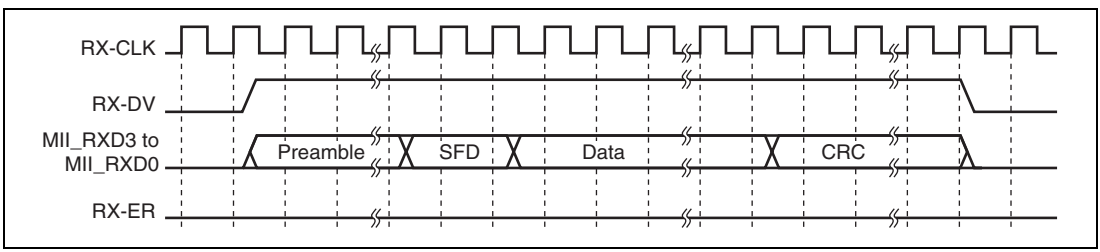


Figure 11.4 (4) MII Frame Receive Timing (Normal Reception)

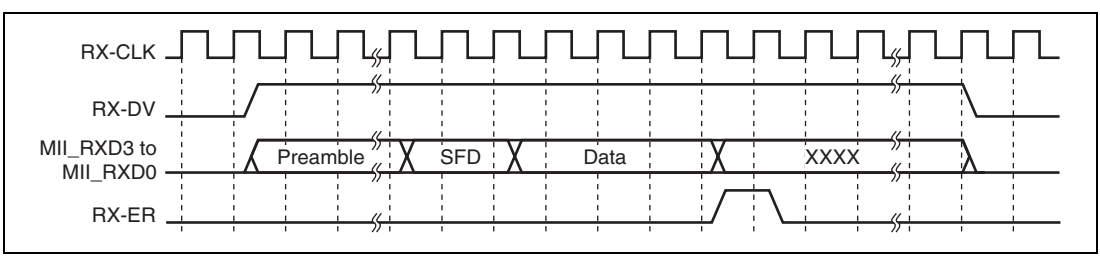


Figure 11.4 (5) MII Frame Receive Timing (Reception Error (1))

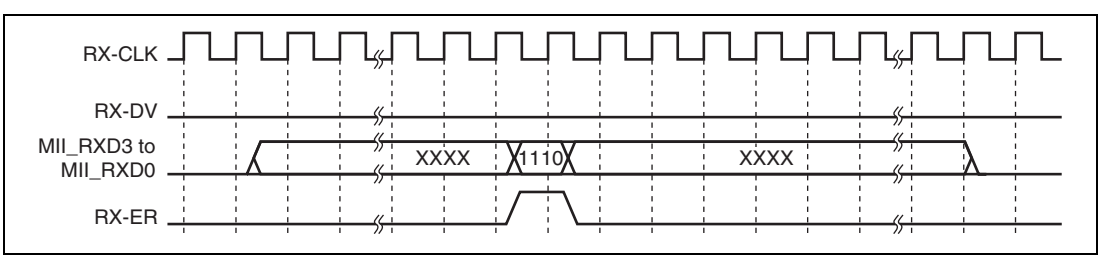


Figure 11.4 (6) MII Fame Receive Timing (Reception Error (2))

11.4.4 Accessing MII Registers

MII registers in the PHY 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 11.8. 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 address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY address.
REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(b) Read: Bus already released in TA; data control unnecessary

Figure 11.5 MII Management Frame Format

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 11.9 shows the MII register access timing. The timing will differ depending on the PHY type.

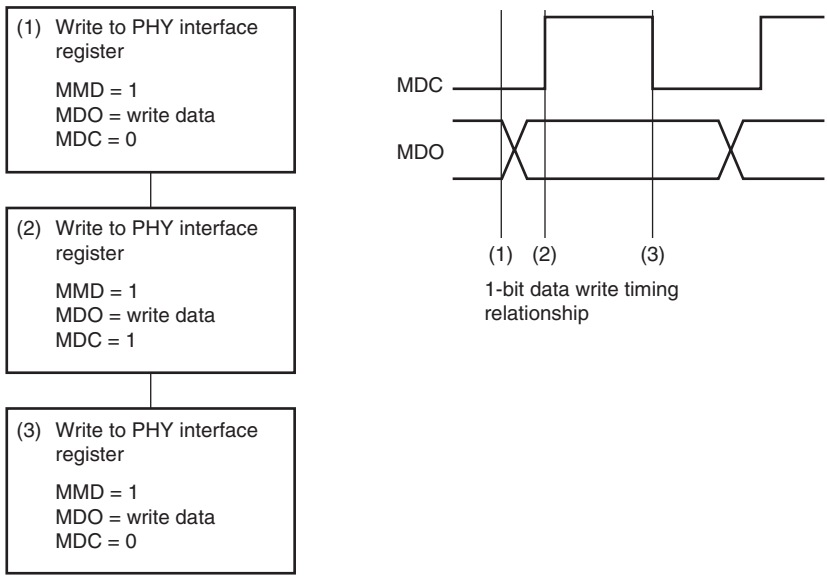


Figure 11.6 (1) 1-Bit Data Write Flowchart

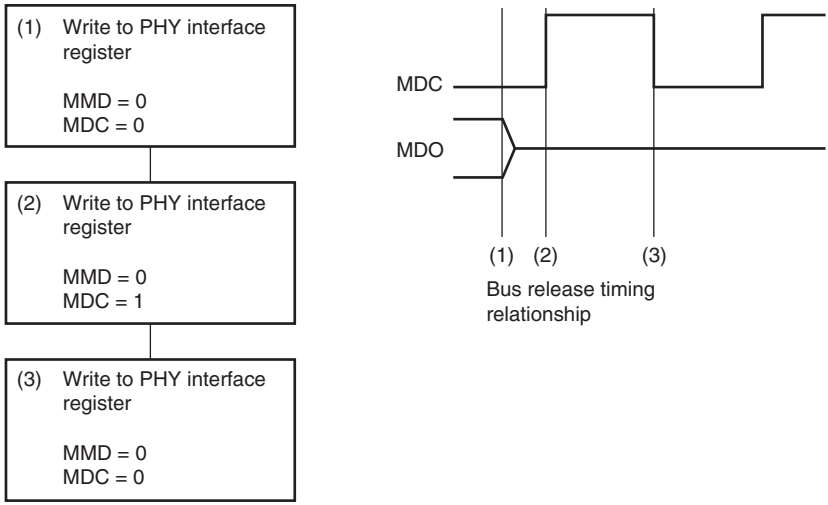


Figure 11.6 (2) Bus Release Flowchart (TA in Read in Figure 11.5)

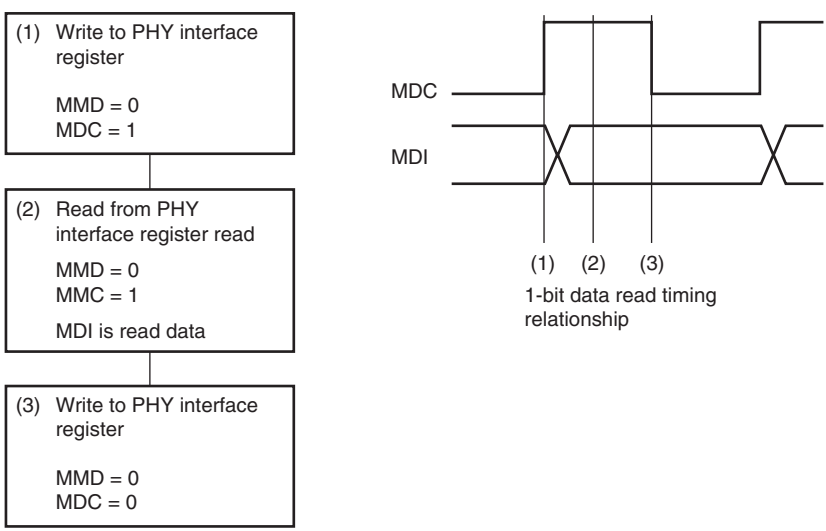


Figure 11.6 (3) 1-Bit Data Read Flowchart

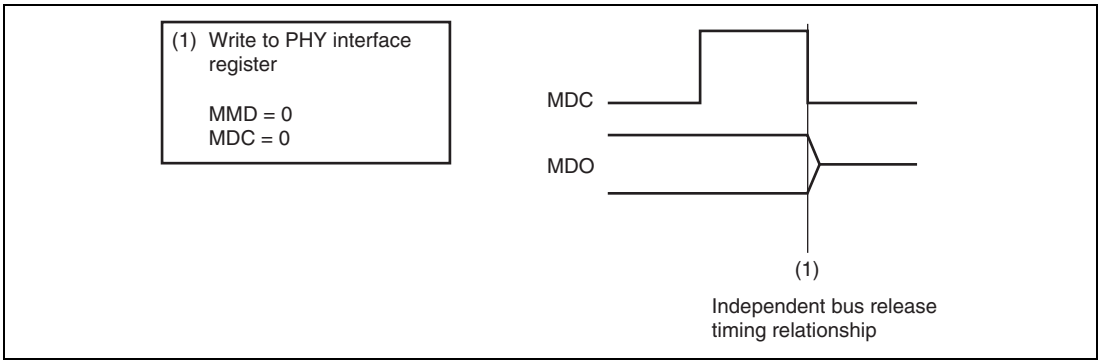


Figure 11.6 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 11.5)

11.4.5 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 the SWR bit in the E-DMAC mode register (EDMR).

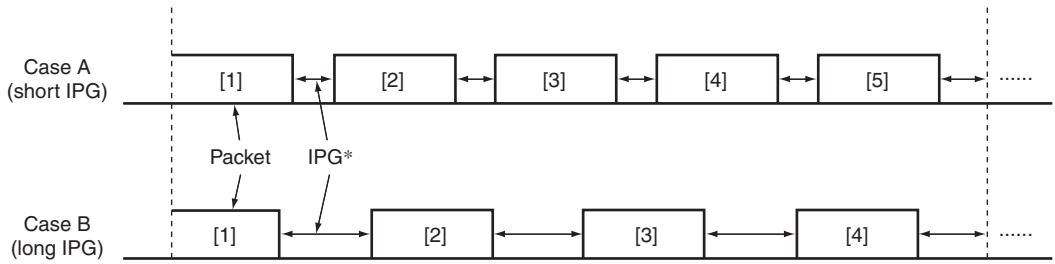
With a Magic Packet, reception is performed regardless of the destination address. As a result, this function is valid, and the WOL pin enabled, only in the case of a match with the destination address specified by the format in the Magic Packet. Further information on Magic Packets can be found in the 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 supporting functions 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.

11.4.6 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.



Note: * IPG may be longer than the set value, depending on the state of the circuit and the system bus.

Figure 11.7 Changing IPG and Transmission Efficiency

11.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x in full-duplex operations. Flow control can be applied to both receive and transmit operations. The methods for transmitting PAUSE frames when controlling flow are as follows:

Automatic PAUSE Frame Transmission: For receive frames, PAUSE frames are automatically transmitted when the number of data in the receive FIFO (included in E-DMAC) reaches the value set in the flow control FIFO threshold register (FCFTR) of the E-DMAC. The TIME parameter included in the PAUSE frame at this time is set by the automatic PAUSE frame setting register (APR). The automatic PAUSE frame transmission is repeated until the number of data in the receive FIFO becomes less than the FCFTR setting as the receive data is read from the FIFO.

The upper limit of the number of retransfers of the PAUSE frame can also be set by the automatic PAUSE frame retransfer count set register (TPAUSER). In this case, PAUSE frame transmission is repeated until the number of data becomes FCFTR value set or below, or the number of transmits reaches the value set by TPAUSER. The automatic PAUSE frame transmission is enabled when the TXF bit in the EtherC mode register (ECMR) is 1.

Manual PAUSE Frame Transmission: PAUSE frames are transmitted by directives from the software. When writing the Timer value to the manual PAUSE frame set register (MPR), manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

PAUSE Frame Reception: The next frame is not transmitted until the time indicated by the Timer value elapses after receiving a PAUSE frame. However, the transmission of the current frame is continued. A received PAUSE frame is valid only when the RXF bit in the EtherC mode register (ECMR) is set to 1.

11.5 Connection to PHY-LSI

Figure 11.8 shows the example of connection to a DP83846AVHG by National Semiconductor Corporation.

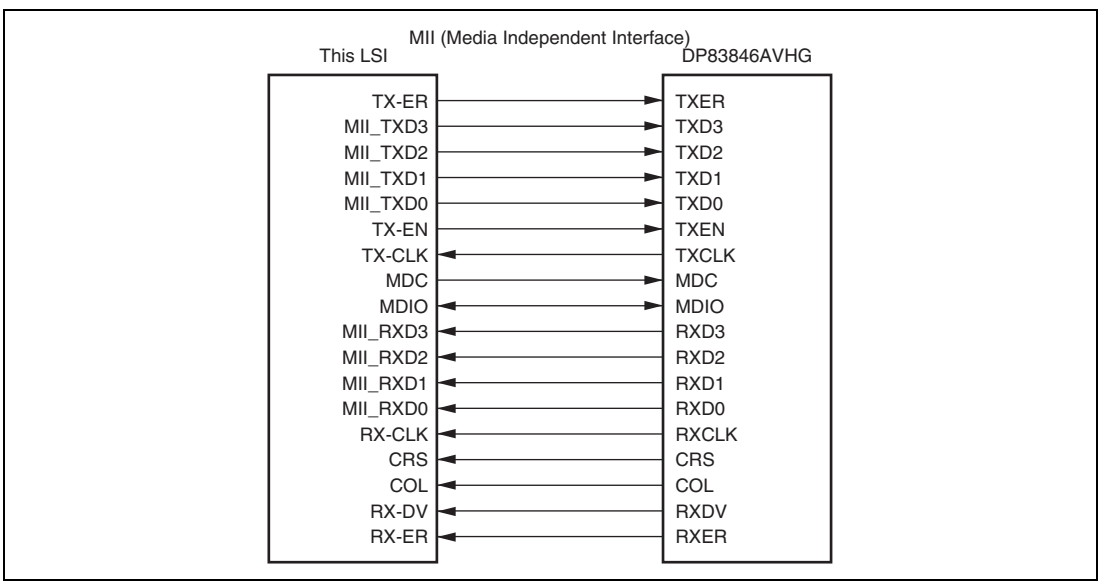


Figure 11.8 Example of Connection to DP83846AVHG

11.6 Usage Notes

- Conditions for Setting LCHNG Bit

  Even if the level of the signal input to the LNKSTA pin is not changed, the LCHNG bit in ECSR may be set. It may happen when the pin function is changed from port to LNKSTA by PCCRH2 of the PFC or when a software reset caused by the SWR bit in EDMR is cleared while the LNKSTA pin is being driven high.

  This is because the LNKSTA signal is internally fixed low when the pin functions as a port or during the software reset state regardless of the external pin level.

  Clear the LCHNG bit before setting the LCHNGIP bit in ECSIPR not to request a LINK signal changed interrupt accidentally.

- Flow Control Defect 1

  Once a PAUSE frame is received while the receiving flow control is enabled in full-duplex mode (the RXF bit in ECMR = 1), each time when the local station receives a normal unicast frame (non-PAUSE frame without a CRC error), the TIME parameter specified by the PAUSE frame that has been previously received is incorrectly applied. As a result, unnecessary waiting time is generated to slow down the transmission throughput. The TIME parameter value is maintained until another PAUSE frame is received.

  This defect can be prevented if the destination station supports the function to transmit the 0 time PAUSE frame as the same as this LSI does. Enable the use of 0 time PAUSE frame in this LSI (the ZPF bit in ECMR = 1) before the 0 time PAUSE frame is received from the destination station. This clears the TIME parameter incorrectly maintained in the EtherC and prevents the unnecessary waiting time for transmission to be generated.

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

- Flow Control Defect 2

  When a PAUSE period is generated while the transmitting/receiving flow control is enabled in full-duplex mode (the TXF/RXF bit in ECMR = 1), non-PAUSE frames are waited for transmission (this is a normal operation) whereas PAUSE frames are incorrectly waited for transmission. The transmission of non-PAUSE frames in a PAUSE period is prohibited, though the transmission of PAUSE frames is enabled in IEEE802.3.

  When a PAUSE period is generated by the request from the destination station (that is, a PAUSE frame is received from the destination station), the load of the destination station is high and that of the local station is not so high. Therefore, the transmission of PAUSE frames during this period is less likely to happen. The ratio that this defect actually affects the operation in this LSI is rather low.

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

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

This LSI includes a direct memory access controller (E-DMAC) directly connected to the Ethernet controller (EtherC). A large proportion of buffer management is controlled by the E-DMAC itself using descriptors. This lightens the load on the CPU and enables efficient control of data transfer.

Figure 12.1 shows the configuration of the E-DMAC, and the descriptors and transmit/receive buffers in memory.

12.1 Features

The E-DMAC has the following features:

- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of block transfer (16-byte units)
- Supports single-frame/multi-buffer operation

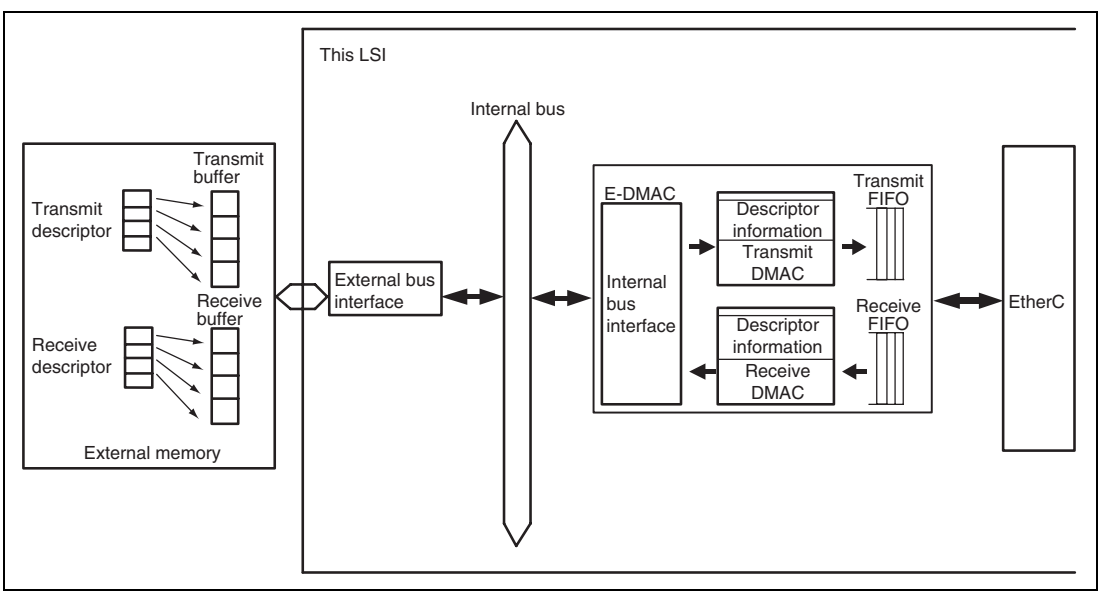


Figure 12.1 Configuration of E-DMAC, and Descriptors and Buffers

12.2 Register Descriptions

The E-DMAC has the following registers. For addresses and access sizes of these registers, see section 24, List of Registers.

- E-DMAC mode register (EDMR)
- E-DMAC transmit request register (EDTRR)
- E-DMAC receive request register (EDRRR)
- Transmit descriptor list address register (TDLAR)
- Receive descriptor list address register (RDLAR)
- EtherC/E-DMAC status register (EESR)
- EtherC/E-DMAC status interrupt permission register (EESIPR)
- Transmit/receive status copy enable register (TRSCER)
- Receive missed-frame counter register (RMFCR)
- Transmit FIFO threshold register (TFTR)
- FIFO depth register (FDR)
- Receiving method control register (RMCR)
- E-DMAC operation control register (EDOCR)
- Receive buffer write address register (RBWAR)
- Receive descriptor fetch address register (RDFAR)
- Transmit buffer read address register (TBRAR)
- Transmit descriptor fetch address register (TDFAR)
- Flow control FIFO threshold register (FCFTR)
- Transmit interrupt register (TRIMD)

12.2.1 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies the operating mode of the E-DMAC. The settings in this register are normally made in the initialization process following a reset. If the EtherC and E-DMAC are initialized by means of this register during data transmission, abnormal data may be sent onto the line. Operating mode settings must not be changed while the transmit and receive functions are enabled. To change the operating mode, the EtherC and E-DMAC modules are got into at their initial state by means of the software reset bit (SWR) in this register, then make new settings. It takes 64 cycles of the internal bus clock Bϕ to initialize the EtherC and E-DMAC. 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 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	DE	0	R/W	E-DMAC Data Endian Convert Selects whether or not the endian format is converted on data transfer by the E-DMAC. However, the endian format of the descriptors and E-DMAC register values are not converted regardless of this bit setting. 0: Endian format not converted (big endian) 1: Endian format converted (little endian)
5	DL1	0	R/W	Descriptor Length
4	DL0	0	R/W	These bits specify the descriptor length. 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Reserved (setting prohibited)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
0	SWR	0	R/W	Software Reset Writing 1 in this bit initializes registers of the E-DMAC other than TDLAR, RDLAR, and RMFCR and registers of the EtherC. While a software reset is being executed (64 cycles of the internal bus clock Bϕ), accesses to the all Ethernet-related registers are prohibited. Software reset period (example): When Bϕ = 62.5 MHz: 1.03 μS When Bϕ = 33 MHz: 1.94 μS This bit is always read as 0. 0: Writing 0 is ignored (E-DMAC operation is not affected) 1: Writing 1 resets the EtherC and E-DMAC and then automatically cleared

12.2.2 E-DMAC Transmit Request Register (EDTRR)

The EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. When transmission of one frame is completed, the next descriptor is read. If the transmit descriptor active bit in this descriptor has the "active" setting, transmission is continued. If the transmit descriptor active bit has the "inactive" setting, the TR bit is cleared and operation of the transmit DMAC is halted.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write 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 active bit in the transmit descriptor 1: Start of transmission. The relevant descriptor is read and a frame is sent with the transmit active bit set to 1

12.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. When the receive request bit is set, the E-DMAC reads the relevant receive descriptor. If the receive descriptor active bit in the descriptor has the "active" setting, the E-DMAC prepares for a receive request from the EtherC. When one receive buffer of data has been received, the E-DMAC reads the next descriptor and prepares to receive the next frame. If the receive descriptor active bit in the descriptor has the "inactive" setting, the RR bit is cleared and operation of the receive DMAC is halted.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write 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 cannot operate successfully. In this case, to make the E-DMAC reception enabled again, execute a software reset by the SWR bit in EDMR. To make the E-DMAC reception disabled without executing a software reset, set the RE bit in ECMR. 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.

12.2.4 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 to TDLA0 = 0000 32-byte boundary: TDLA4 to TDLA0 = 00000 64-byte boundary: TDLA5 to TDLA0 = 000000

12.2.5 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 to RDLA0 = 0000 32-byte boundary: RDLA4 to RDLA0 = 00000 64-byte boundary: RDLA5 to RDLA0 = 000000

12.2.6 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 interrupts. 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 interrupt priority, see section 6.5, Interrupt Exception Handling Vector Table.

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	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 frame transmission by the EtherC has been aborted because of an error during transmission. 0: Frame transmission has not been aborted or no transmit directive 1: Frame transmit has been aborted

Bit	Bit Name	Initial value	R/W	Description
25	RABT	0	R/W	Receive Abort Detection Indicates that frame reception by the EtherC has been aborted because of an error during reception. 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
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, set the E-DMAC again after software reset by means of 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

Bit	Bit Name	Initial value	R/W	Description
21	TC	0	R/W	Frame Transmit Complete Indicates that all the data specified by the transmit descriptor has been transmitted to the EtherC. The transfer status is written back to the relevant descriptor. For 1-frame/1-buffer processing, when 1-frame transmission is completed and the transmission descriptor valid bit (TACT) in the next descriptor is not set, transmission is completed and this bit is set to 1. Likewise, for multiple-frame buffer processing, when the last data in the frame is transmitted and the transmission descriptor valid bit (TACT) in the next descriptor is not set, transmission is completed and this bit is set to 1. After frame transmission, the E-DMAC writes the transmission status back to the descriptor. 0: Transfer not complete, or no transfer directive 1: Transfer complete
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmission descriptor valid bit (TACT) in the descriptor is not set when the E-DMAC reads the transmission descriptor when the previous descriptor is not the last one of the frame for multiple-buffer frame processing. 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

Bit	Bit Name	Initial value	R/W	Description
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
17	RDE	0	R/W	Receive Descriptor Empty When receive descriptor empty (RDE = 1) occurs, receiving can be restarted by setting RACT = 1 in the receive descriptor and initiating receiving. 0: Receive descriptor active bit RACT = 1 not 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. 0: A carrier is detected when transmission starts 1: A carrier is not detected when transmission starts
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 of the EtherC has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 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 Receive Error 0: PHY receive error not detected 1: PHY receive error detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error not detected 1: CRC error detected

12.2.7 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. In the initial state, interrupts are not enabled.

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 Permission 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Permission 0: Transmit abort detection interrupt is disabled 1: Transmit abort detection interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detection Interrupt Permission 0: Receive abort detection interrupt is disabled 1: Receive abort detection interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Permission 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	ADEIP	0	R/W	Address Error Interrupt Permission 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Interrupt Permission 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Permission 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 Permission 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Permission 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Received Interrupt Permission 0: Frame received interrupt is disabled 1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Permission 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Permission 0: Receive FIFO overflow interrupt is disabled 1: Receive FIFO overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Permission 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Permission 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Permission 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Permission 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 Permission 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6, 5	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Permission 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 Permission 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Permission 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Permission 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame 0: CRC error on received frame interrupt is disabled 1: CRC error on received frame interrupt is enabled

12.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether or not transmit and receive status information reported by bits in the EtherC/E-DMAC status register is to be indicated in bits TFS26 to TFS0 and RFS26 to RFS0 in the corresponding descriptor. Bits in this register correspond to bits 11 to 0 in the EtherC/E-DMAC status register (EESR). When a bit is cleared to 0, the transmit status (bits 11 to 8 in EESR) is indicated in bits TFS3 to TFS0 in the transmit descriptor, and the receive status (bits 7 to 0 in EESR) is indicated in bits RFS7 to RFS0 of the receive descriptor. When a bit is set to 1, the occurrence of the corresponding interrupt is not indicated in the descriptor. After this LSI is reset, all bits are cleared to 0.

Bit	Bit Name	Initial value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDCE	0	R/W	CND Bit Copy Directive 0: Indicates the CND bit state in bit TFS3 in the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS3 of the transmit descriptor
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Indicates the DLC bit state in bit TFS2 of the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS2 of the transmit descriptor
9	CDCE	0	R/W	CD Bit Copy Directive 0: Indicates the CD bit state in bit TFS1 of the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS1 of the transmit descriptor
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Indicates the TRO bit state in bit TFS0 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS0 of the receive descriptor

Bit	Bit Name	Initial value	R/W	Description
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Indicates the RMAF bit state in bit RFS7 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS7 of the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Indicates the RRF bit state in bit RFS4 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS4 of the receive descriptor
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Indicates the RTLF bit state in bit RFS3 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS3 of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Indicates the RTSF bit state in bit RFS2 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS2 of the receive descriptor
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Indicates the PRF bit state in bit RFS1 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS1 of the receive descriptor
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Indicates the CERF bit state in bit RFS0 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS0 of the receive descriptor

12.2.9 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.

12.2.10 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'81 to H'200: Setting prohibited

Note: When starting transmission before one frame of data write has completed, take care the generation of the underflow.

12.2.11 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the depth 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	B'001	R/W	Transmit FIFO Depth These bits specify the depth of the transmit FIFO. After the start of the transmission and reception, the setting cannot be changed. 000: 256 bytes 001: 512 bytes Other than above: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	RFD2 to RFD0	B'001	R/W	Receive FIFO Depth These bits specify the depth of the receive FIFO. After the start of the transmission and reception, the setting cannot be changed. 000: 256 bytes 001: 512 bytes Other than above: Setting prohibited

12.2.12 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RR bit in EDRRR 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 0: When reception of one frame is completed, the E-DMAC writes the receive status into the descriptor and clears the RR bit in EDRRR 1: When reception of one frame is completed, the E-DMAC writes the receive status into the descriptor, reads the next descriptor, and prepares to receive the next frame

12.2.13 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: E-DMAC stops its operation due to illegal memory address detection Note: To resume the operation, set the E-DMAC again after software reset by means of the SWR bit in EDMR.
1	EDH	0	R/W	E-DMAC Halted 0: The E-DMAC is operating normally 1: The E-DMAC has been halted by NMI pin assertion. E-DMAC operation is restarted by writing 0
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

12.2.14 Receiving-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.

12.2.15 Receiving-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.

12.2.16 Transmission-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.

12.2.17 Transmission-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.

12.2.18 Flow Control FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC (setting the threshold on automatic PAUSE transmission). The threshold can be specified by the depth of the receive FIFO data (RFD2 to RFD0) and the number of receive frames (RFF2 to RFF0). The condition to start the flow control is decided by taking OR operation on the two thresholds. Therefore, the flow control by the two thresholds is independently started.

When flow control is performed according to the RFD bits setting, if the setting is the same as the depth of the receive FIFO specified by the FIFO depth register (FDR), flow control is started when the remaining FIFO is (FIFO data – 64) bytes. For instance, when RFD in FDR = 1 and RFD in FCFTR = 1, flow control is started when (512 – 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.

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.

Bit	Bit Name	Initial value	R/W	Description
18	RFF2	1	R/W	Receive Frame Number Flow Control Threshold
17	RFF1	1	R/W	000: When one receive frame has been stored in the receive FIFO
16	RFF0	1	R/W	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	—	Reserved These bits are always read as 0. The write value should always be 0.
2	RFD2	0	R/W	Receive Byte Flow Control Threshold
1	RFD1	0	R/W	000: When (256 – 64) bytes of data is stored in the receive FIFO
0	RFD0	0	R/W	001: When (512 – 64) bytes of data is stored in the receive FIFO
				Other than above: Setting prohibited

12.2.19 Transmit Interrupt Register (TRIMD)

TRIMD is a 32-bit readable/writable register that specifies whether or not to notify write-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-back completion for each frame is not notified 1: Write-backed completion for each frame using the TWB bit in EESR is notified

12.3 Operation

The E-DMAC is connected to the EtherC, and performs efficient transfer of transmit/receive data between the EtherC and memory (buffers) without the intervention of the CPU. The E-DMAC itself reads control information, including buffer pointers called descriptors, relating to the buffers. The E-DMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer in accordance with this control information. By setting up a number of consecutive descriptors (a descriptor list), it is possible to execute transmission and reception continuously.

12.3.1 Descriptor List and Data Buffers

Before starting transmission/reception, the communication program creates transmit and receive descriptor lists in memory. The start addresses of these lists are then set in the transmit and receive descriptor list start address registers.

The descriptor start address must be aligned so that it matches the address boundary according to the descriptor length set by the E-DMAC mode register (EDMR). The transmit buffer start address can be aligned with a byte, a word, and a longword boundary.

(1) Transmit Descriptor

Figure 12.2 shows the relationship between a transmit descriptor and the transmit buffer. According to the specification in this descriptor, the relationship between the transmit frame and transmit buffer can be defined as one frame/one buffer or one frame/multi-buffer.

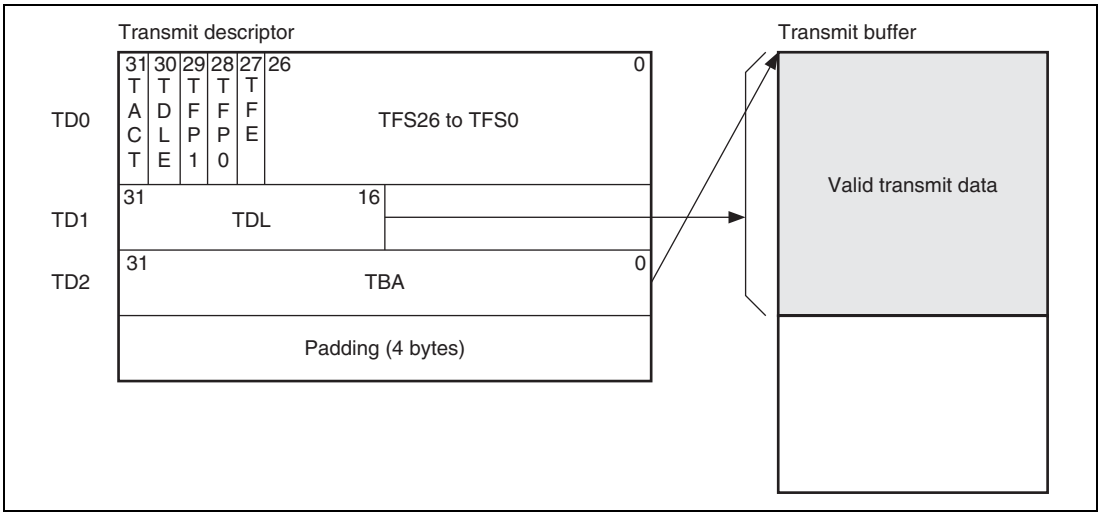


Figure 12.2 Relationship between Transmit Descriptor and Transmit Buffer

(a) Transmit Descriptor 0 (TD0)

TD0 indicates the transmit frame status. The CPU and E-DMAC use TD0 to report the frame transmission status.

Bit	Bit Name	Initial value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Active Indicates that this descriptor is active. The CPU sets this bit after transmit data has been transferred to the transmit buffer. The E-DMAC resets this bit on completion of a frame transfer or when transmission is suspended. 0: The transmit descriptor is invalid. Indicates that valid data has not been written to this bit by the CPU, or this bit has been reset by a write-back operation on termination of E-DMAC frame transfer processing (completion or suspension of transmission) If this state is recognized in an E-DMAC descriptor read, the E-DMAC terminates transmit processing and transmit operations cannot be continued (a restart is necessary) 1: The transmit descriptor is valid. Indicates that valid data has been written to the transmit buffer by the CPU and frame transfer processing has not yet been executed, or that frame transfer is in progress When this state is recognized in an E-DMAC descriptor read, the E-DMAC continues with the transmit operation
30	TDLE	0	R/W	Transmit Descriptor List End After completion of the corresponding buffer transfer, the E-DMAC references the first descriptor. This specification is used to set a ring configuration for the transmit descriptors. 0: This is not the last transmit descriptor list 1: This is the last transmit descriptor list

Bit	Bit Name	Initial value	R/W	Description
29	TFP1	0	R/W	Transmit Frame Position 1, 0
28	TFP0	0	R/W	These two bits specify the relationship between the transmit buffer and transmit frame. In the preceding and following descriptors, a logically positive relationship must be maintained between the settings of this bit and the TDLE bit. 00: Frame transmission for transmit buffer indicated by this descriptor continues (frame is not concluded) 01: Transmit buffer indicated by this descriptor contains end of frame (frame is concluded) 10: Transmit buffer indicated by this descriptor is start of frame (frame is not concluded) 11: Contents of transmit buffer indicated by this descriptor are equivalent to one frame (one frame/one buffer)
27	TFE	0	R/W	Transmit Frame Error Indicates that one or other bit of the transmit frame status indicated by bits 26 to 0 is set. Whether or not the transmit frame status information is copied into this bit is specified by the transmit/receive status copy enable register. 0: No error during transmission 1: An error occurred during transmission
26 to 0	TFS26 to TFS0	All 0	R/W	Transmit Frame Status TFS26 to TFS4: Reserved (The write value should always be 0.) TFS3: Carrier Not Detect (corresponds to CND bit in EESR) TFS2: Detect Loss of Carrier (corresponds to DLC bit in EESR) TFS1: Delayed Collision Detect (corresponds to CD bit in EESR) TFS0: Transmit Retry Over (corresponds to TRO bit in EESR)

(b) Transmit Descriptor 1 (TD1)

TD1 specifies the transmit buffer length (maximum 64 kbytes).

Bit	Bit Name	Initial value	R/W	Description
31 to 16	TDL	All 0	R/W	Transmit Buffer Data Length These bits specify the valid transfer byte length in the corresponding transmit buffer. When the one frame/multi-buffer system is specified (TD0 and TFP = 10 or 00), the transfer byte length specified in the descriptors at the start and midway can be set in byte units.
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 specifies the 32-bit transmit buffer start address. The transmit buffer start address setting can be aligned with a byte, a word, or a longword boundary.

(2) Receive Descriptor

Figure 12.3 shows the relationship between a receive descriptor and the receive buffer. In frame reception, the E-DMAC performs data rewriting up to a receive buffer 16-byte boundary, regardless of the receive frame length. Finally, the actual receive frame length is reported in the lower 16 bits of RD1 in the descriptor. Data transfer to the receive buffer is performed automatically by the E-DMAC to give a one frame/one buffer or one frame/multi-buffer configuration according to the size of one received frame.

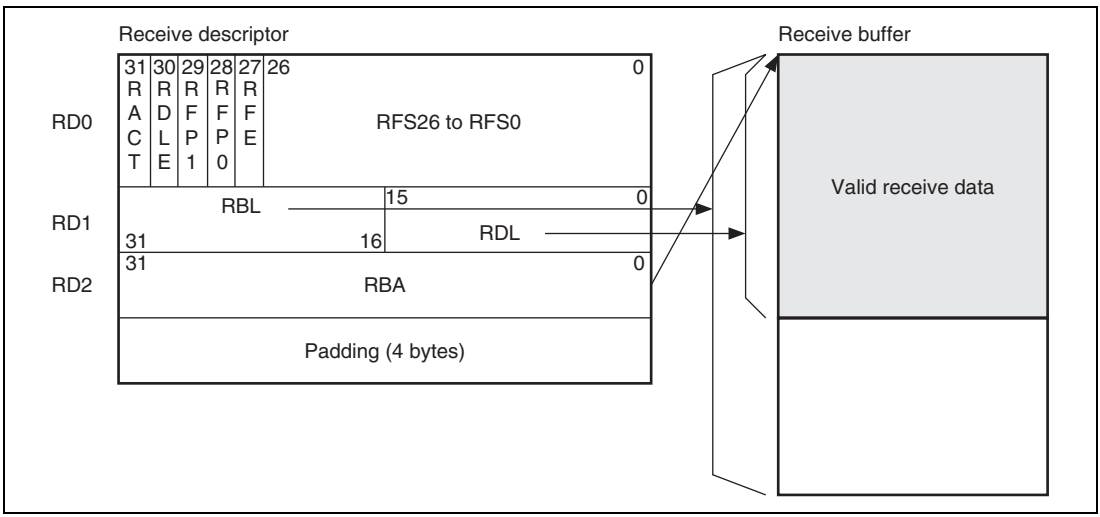


Figure 12.3 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

RD0 indicates the receive frame status. The CPU and E-DMAC use RD0 to report the frame receive status.

Bit	Bit Name	Initial value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Active Indicates that this descriptor is active. The E-DMAC resets this bit after receive data has been transferred to the receive buffer. On completion of receive frame processing, the CPU sets this bit to prepare for reception. 0: The receive descriptor is invalid. Indicates that the receive buffer is not ready (access disabled by E-DMAC), or this bit has been reset by a write-back operation on termination of E-DMAC frame transfer processing (completion or suspension of reception). If this state is recognized in an E-DMAC descriptor read, the E-DMAC terminates receive processing and receive operations cannot be continued. Reception can be restarted by setting RACT to 1 and executing receive initiation. 1: The receive descriptor is valid Indicates that the receive buffer is ready (access enabled) and processing for frame transfer from the FIFO has not been executed, or that frame transfer is in progress. When this state is recognized in an E-DMAC descriptor read, the E-DMAC continues with the receive operation.
30	RDLE	0	R/W	Receive Descriptor List Last After completion of the corresponding buffer transfer, the E-DMAC references the first receive descriptor. This specification is used to set a ring configuration for the receive descriptors. 0: This is not the last receive descriptor list 1: This is the last receive descriptor list

Bit	Bit Name	Initial value	R/W	Description
29	RFP1	0	R/W	Receive Frame Position
28	RFP0	0	R/W	These two bits specify the relationship between the receive buffer and receive frame. 00: Frame reception for receive buffer indicated by this descriptor continues (frame is not concluded) 01: Receive buffer indicated by this descriptor contains end of frame (frame is concluded) 10: Receive buffer indicated by this descriptor is start of frame (frame is not concluded) 11: Contents of receive buffer indicated by this descriptor are equivalent to one frame (one frame/one buffer)
27	RFE	0	R/W	Receive Frame Error Indicates that one or other bit of the receive frame status indicated by bits 26 to 0 is set. Whether or not the receive frame status information is copied into this bit is specified by the transmit/receive status copy enable register. 0: No error during reception 1: A certain kind of error occurred during reception

Bit	Bit Name	Initial value	R/W	Description
26 to 0	RFS26 to RFS0	All 0	R/W	Receive Frame Status These bits indicate the error status during frame reception. RFS26 to RFS10: Reserved (The write value should always be 0.) RFS9: Receive FIFO overflow (corresponds to RFOF bit in EESR) RFS8: Reserved (The write value should always be 0.) RFS7: Multicast address frame received (corresponds to RMAF bit in EESR) RFS6: CAM entry unregistered frame received (corresponds to the RUAF bit in EESR) RSF5: Reserved (The write value should always be 0.) RFS4: Receive residual-bit frame error (corresponds to RRF bit in EESR) RFS3: Receive too-long frame error (corresponds to RTLF bit in EESR) RFS2: Receive too-short frame error (corresponds to RTSF bit in EESR) RFS1: PHY-LSI receive error (corresponds to PRE bit in EESR) RFS0: CRC error on received frame (corresponds to CERF bit in EESR)

(b) Receive Descriptor 1 (RD1)

RD1 specifies the receive buffer length (maximum 64 kbytes).

Bit	Bit Name	Initial value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Length These bits specify the maximum reception byte length in the corresponding receive buffer. The transfer byte length must align with a 16-byte boundary (bits 19 to 16 cleared to 0). The maximum receive frame length with one frame per buffer is 1,514 bytes, excluding the CRC data. Therefore, for the receive buffer length specification, a value of 1,520 bytes (H'05F0) that takes account of a 16-byte boundary is set as the maximum receive frame length.
15 to 0	RDL	All 0	R/W	Receive Data Length These bits specify the data length of a receive frame stored in the receive buffer. The receive data transferred to the receive buffer does not include the 4-byte CRC data at the end of the frame. The receive frame length is reported as the number of words (valid data bytes) not including this CRC data.

(c) Receive Descriptor 2 (RD2)

RD2 specifies the 32-bit receive buffer start address. The receive buffer start address must be aligned with a longword boundary. However, when SDRAM is connected, it must be aligned with a 16-byte boundary.

12.3.2 Transmission

When the transmit function is enabled and the transmit request bit (TR) is set in the E-DMAC transmit request register (EDTRR), the E-DMAC reads the descriptor used last time from the transmit descriptor list (in the initial state, the descriptor indicated by the transmission descriptor start address register (TDLAR)). If the setting of the TACT bit in the read descriptor is active, the E-DMAC reads transmit frame data sequentially from the transmit buffer start address specified by TD2, and transfers it 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 01 (frame continuation):
   Descriptor write-back is performed after DMA transfer.
2. TFP = 01 or 11 (frame end):
   Descriptor write-back is performed after completion of frame transmission.

The E-DMAC continues reading descriptors and transmitting frames as long as the setting of the TACT bit in the read descriptors is "active." When a descriptor with an "inactive" TACT bit is read, the E-DMAC resets the transmit request bit (TR) in the transmit register and ends transmit processing (EDTRR).

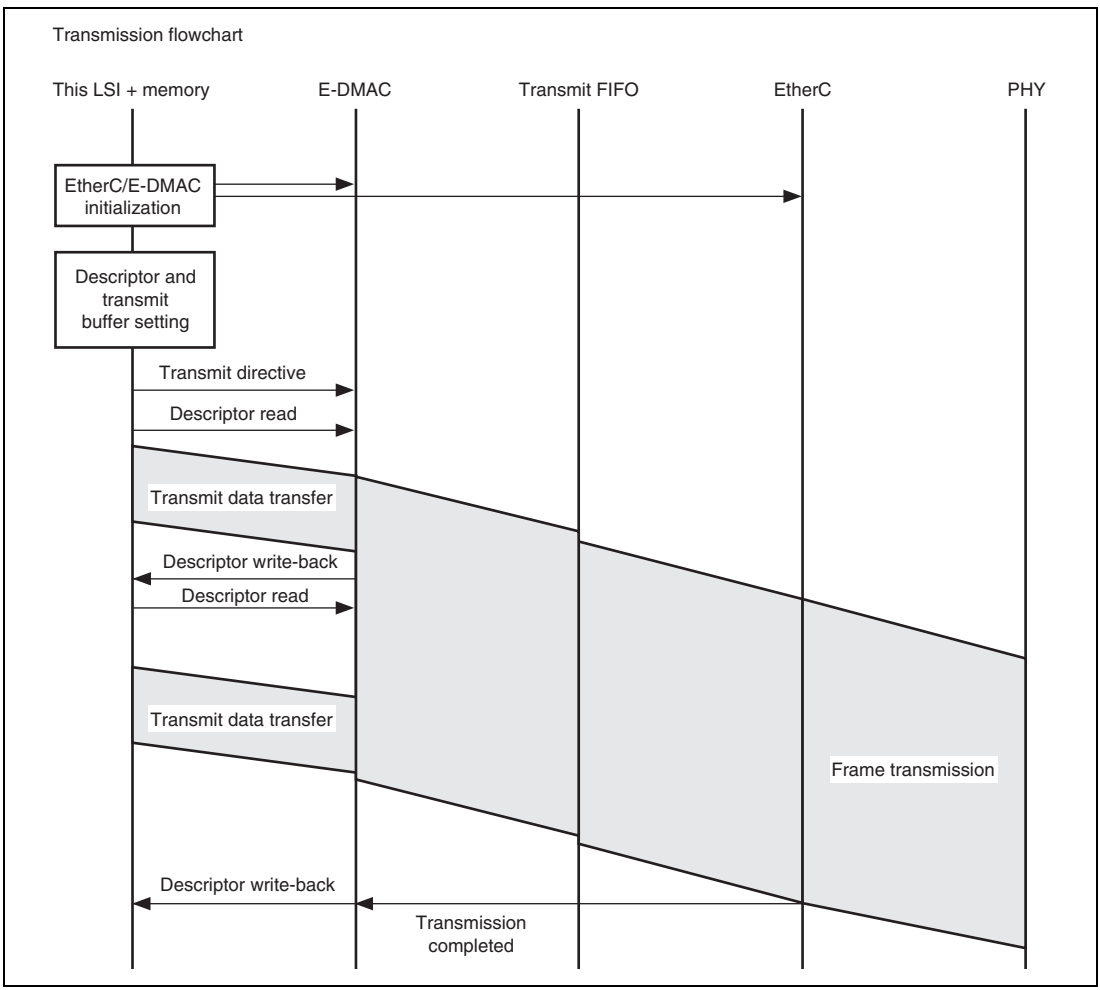


Figure 12.4 Sample Transmission Flowchart

12.3.3 Reception

When the receive function is enabled and the CPU sets the receive request bit (RR) in the E-DMAC receive request register (EDRRR), the E-DMAC reads the descriptor following the previously used one from the receive descriptor list (in the initial state, the descriptor indicated by the transmission descriptor start address register (TDLAR)), and then enters the receive-standby state. If the setting of the RACT bit is "active" and an own-address frame is received, the E-DMAC transfers the frame to the receive buffer specified by RD2. If the data length of the received frame is greater than the buffer length given by RD1, the E-DMAC performs write-back to the descriptor when the buffer is full (RFP = 10 or 00), 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 (RFP = 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 control register (RCR). After initialization, this bit is cleared to 0.

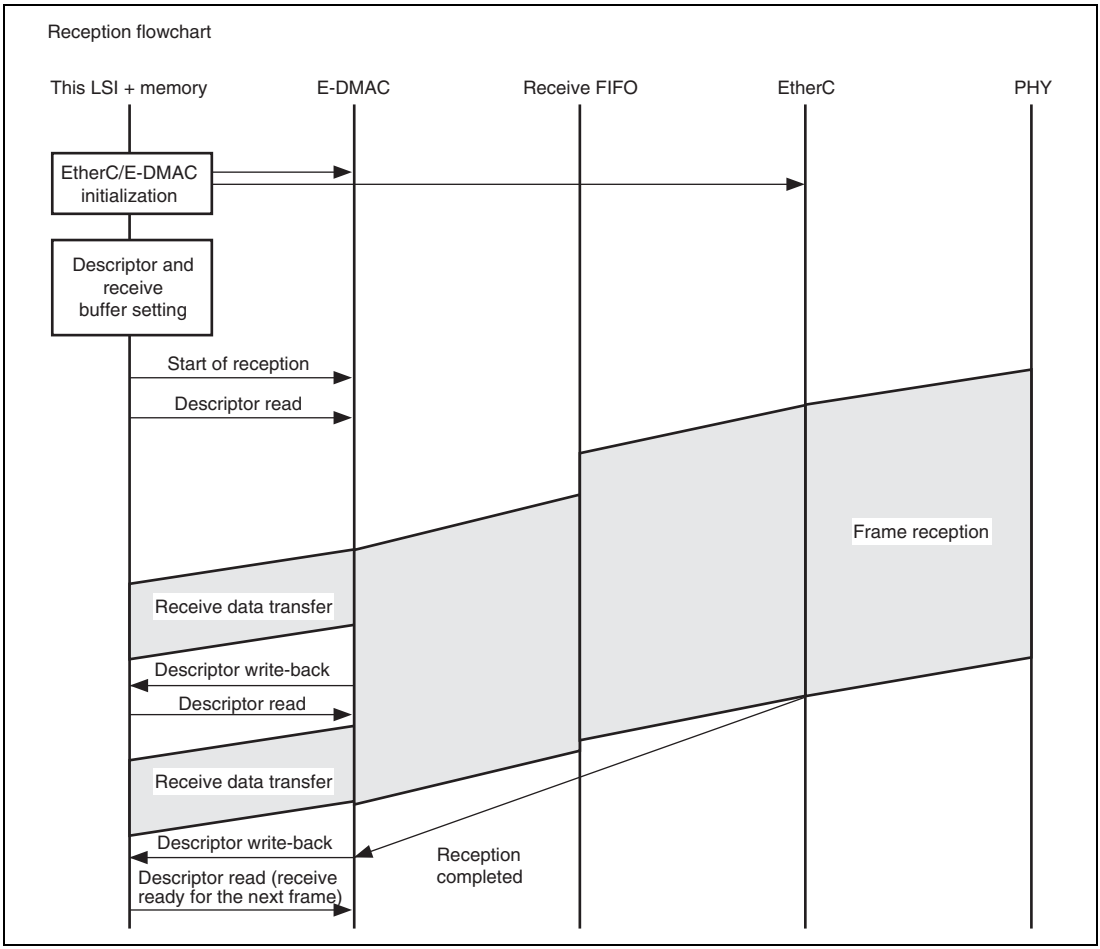


Figure 12.5 Sample Reception Flowchart

12.3.4 Multi-Buffer Frame Transmit/Receive Processing

Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in figure 12.6 is carried out by the E-DMAC.

Where the transmit descriptor is shown as inactive (TACT bit = 0) in the figure, 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.

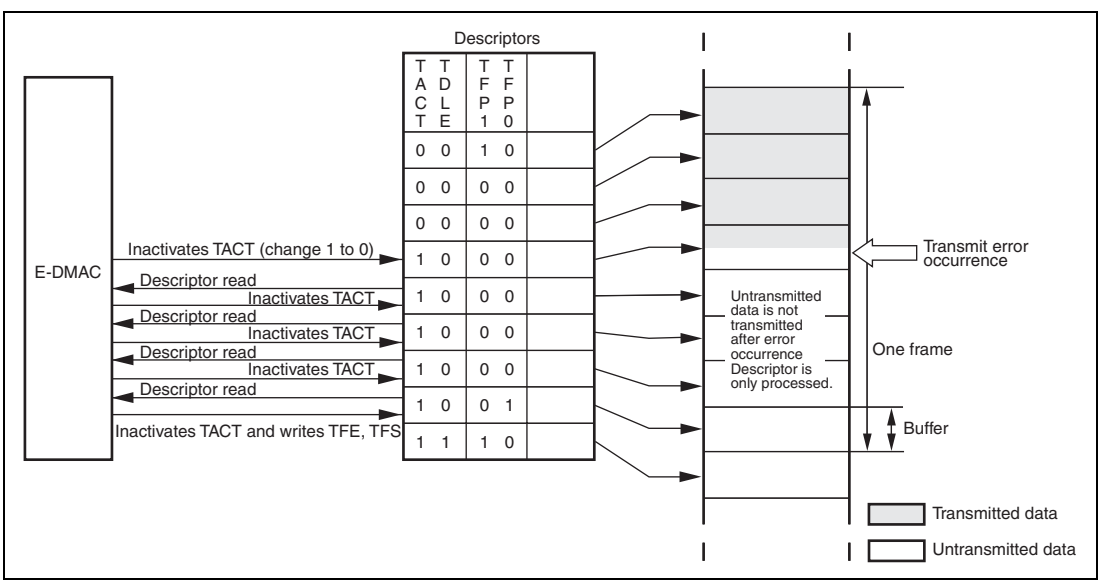


Figure 12.6 E-DMAC Operation after Transmit Error

Multi-Buffer Frame Receive Processing

If an error occurs during multi-buffer frame reception, the processing shown in figure 12.7 is carried out by the E-DMAC.

Where the receive descriptor is shown as inactive (RACT bit = 0) in the figure, buffer data has already been received normally, and where the receive descriptor is shown as active (RACT bit = 1), this indicates a buffer for which reception has not yet been performed. If a frame receive error occurs in the first descriptor part where the RACT bit = 1 in the figure, reception is halted immediately and a status write-back to the descriptor is performed.

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.

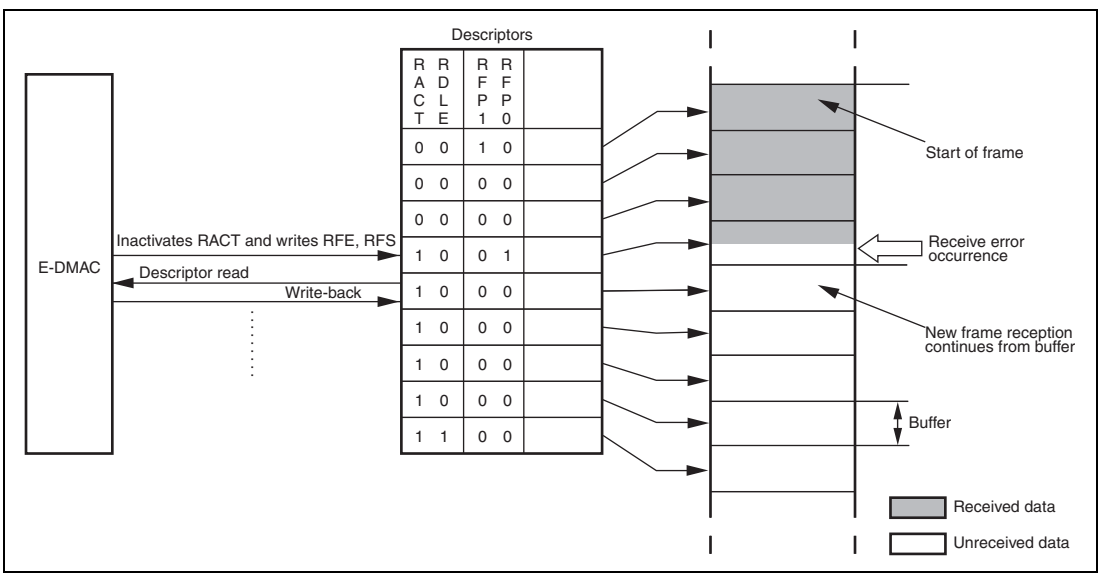


Figure 12.7 E-DMAC Operation after Receive Error

12.4 Usage Notes

12.4.1 Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR)

When the status bits in EESR of the on-chip E-DMAC of the SH-Ether chip are used as interrupt sources, setting of the interrupt source may fail if software writes a 1 to the corresponding status bit in EESR to clear the bit and this coincides with setting of the status interrupt source in EESR by the EtherC or E-DMAC. Figure 12.8 shows an example of timing in the case where setting of the interrupt source in EESR has failed.

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

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

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

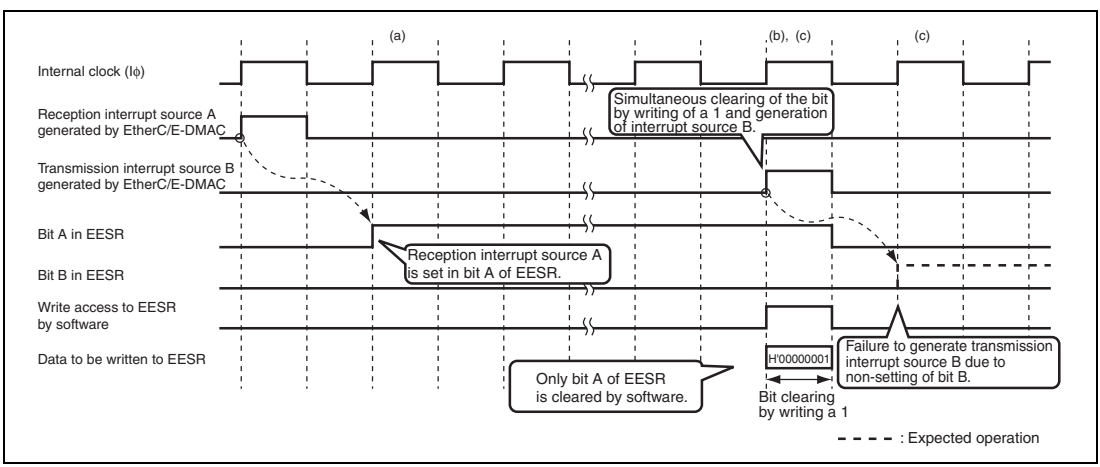


Figure 12.8 Timing of the Case where Setting of the Interrupt Source Bit in EESR by the E-DMAC Fails

(1) Countermeasure

This problem does not occur with all of the bits in EESR. The description applies to some bits but not others. Table 12.1 shows whether the problem can occur with the individual bits and whether the state of the individual interrupt source is reflected in the descriptor.

Table 12.1 EESR Bits for which This Problem can Occur and Reflection of Interrupt Sources in the Descriptor

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

Bit	Bit Name	Status	Possibility of Problem	Reflection in Descriptor	Interrupt Source
15	—	Reserved	—	—	—
14	—	Reserved	—	—	—
13	—	Reserved	—	—	—
12	—	Transmit frame length error	Yes	Reflected in TD0 bit4 (TFS4)	Transmit
11	CND	Carrier not detected	Yes	Reflected in TD0 bit3 (TFS3)	Transmit
10	DLC	Loss of carrier detected	Yes	Reflected in TD0 bit2 (TFS2)	Transmit
9	CD	Delayed collision detected	Yes	Reflected in TD0 bit1 (TFS1)	Transmit
8	TRO	Transmit retry over	Yes	Reflected in TD0 bit0 (TFS0)	Transmit
7	RMAF	Multicast address frame received	No	Reflected in RD0 bit7 (RFS7)	Reception
6	—	Reserved	—	—	—
5	—	Receive frame discard request asserted	No	Reflected in RD0 bit5 (RFS5)	Reception
4	RRF	Residual-bit frame received	No	Reflected in RD0 bit4 (RFS4)	Reception
3	RTLF	Overly long frame received	No	Reflected in RD0 bit3 (RFS3)	Reception
2	RTSF	Overly short frame received	No	Reflected in RD0 bit2 (RFS2)	Reception
1	PRE	PHY receive error	No	Reflected in RD0 bit1 (RFS1)	Reception

Bit	Bit Name	Status	Possibility of Problem	Reflection in Descriptor	Interrupt Source
0	CERF	CRC error in received frame	No	Reflected in RD0 bit0 (RFS0)	Reception

"Yes": Setting of this interrupt source bit can fail.
"No": Setting of this interrupt source bit does not fail.

Take the following countermeasures for bits where the problem can arise.

- Bit 30 (TWB): Write-back complete interrupt source bit in EESR may not be set.
  Check the TACT bit in the transmit descriptor. TACT = 0 indicates that the transmission is complete.
- Bit 26 (TABT): Transmit abort detection interrupt source bit in EESR may not be set.
  Since the state of the interrupt source is written back to the relevant descriptor, check the transmit descriptor (TD0) to confirm the error status.
- Bit 24 (RFCOF): Receive frame counter overflow interrupt source bit in EESR may not be set.
  However, even if the software is not notified of the interrupt despite the frame counter having overflowed, the upper layer (e.g. TCP/IP) can recognize the error because this LSI discards the frame. After departure from the overflow state, storage in the receive FIFO proceeds normally from the head of the next frame. Therefore, no problem with the system arises.
- Bit 21 (TC): Frame transmission complete interrupt source bit in EESR may not be set.
  For transmission-related processing, either procedure (a) or (b) given below is effective.
  (a) Transmission processing without interrupt handling of the frame transmission complete interrupt
    1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
    2. After setting the transmit descriptors, set bit 0 (TR) in the E-DMAC transmit request register (EDTRR) to start transmission.
    3. Before setting the next frame for transmission in the descriptor (when a transmission task arises), check the TACT bit of the corresponding transmit descriptor.
    4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, do not set the transmit descriptor until the next timing.

  (b) For systems where completion of the transmission of each frame must be confirmed (that is, set frame for transmission → initiate transmission → complete frame transmission → set the next frame for transmission → …)
    1. Check the TACT bit in the last descriptor of the frame for transmission and confirm that TACT = 0, which means that the transmission was completed.

- Bit 19 (TFUF): Transmit FIFO underflow interrupt source bit in EESR may not be set.
  When this bit is used as an interrupt source but is not set when it should be, the software is not notified of the interrupt. However, the upper layer will recognize the error in the form of an underflow of the transmit FIFO.
- Bit 16 (RFOF): Receive FIFO overflow interrupt source bit in EESR may not be set.
  Since the state of the interrupt source is written back to the relevant descriptor, check the receive descriptor (RD0) to confirm the error status.
- Bit 11 (CND), bit 10 (DLC), bit 9 (CD), bit 8 (TRO): The interrupt source bits in EESR for the carrier not detected, loss of carrier detected, delayed collision detected, and transmit retry over interrupts may not be set.
  However, since the states of the interrupt sources are written back to the relevant descriptor, check the transmit descriptor (TD0) to confirm the error status.

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

The following descriptions are of sample countermeasures for cases when software processing is based on the frame transmit complete interrupt (bit 21 (TC) in EESR).

If the TC interrupt source bit (bit 21) in EESR is not set on completion of transmission, the system will continue to wait for the TC interrupt, leading to stoppage of transmission. This situation arises when the interrupt handler writes a 1 to clear the bit. The sample method given as case (a) below takes the above possibility into account and avoids the problem by monitoring the transmit descriptor in interrupt processing for interrupts other than the TC interrupt.

The sample method given as case (b) below avoids the above problem by setting a timeout limit for retry processing when multiple transmit descriptors are in use.

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

(a) Countermeasure by monitoring of the transmit descriptor in the processing of interrupts other than the frame transmit complete (TC) interrupt

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. Provide a "condition flag" for use in step 5 and by interrupt handlers, and then turn off this flag.
   This flag serves as a condition flag into which the TACT bits of transmit descriptors are read out.
3. After setting the frame for transmission in the first descriptor, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
4. Before setting the next frame for transmission in the transmit descriptor (when another transmission task arises), check the TACT bit in the corresponding transmit descriptor.
5. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, turn on the condition flag and make an OS service call (e.g. to acquire the semaphore) to place the transmission task in the waiting state.

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

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

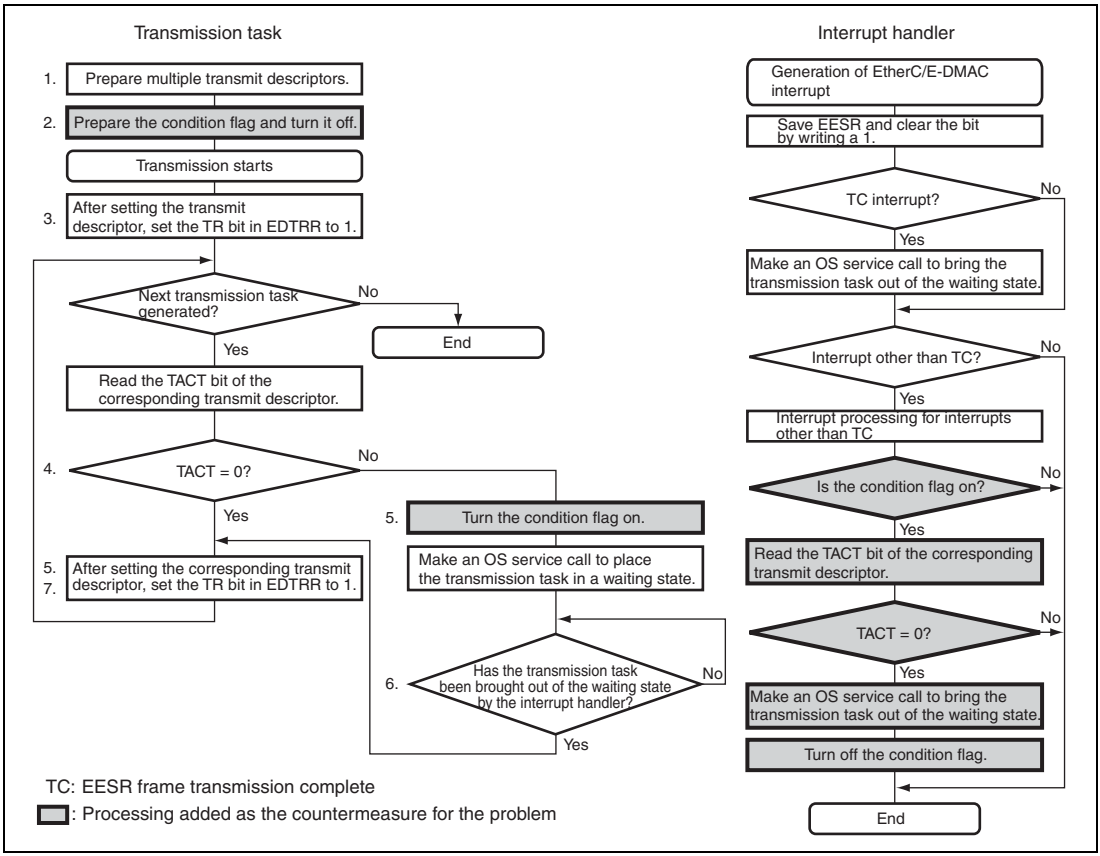


Figure 12.9 Countermeasure by Monitoring the Transmit Descriptor in Processing of Interrupts Other than the Frame Transmit Complete (TC) Interrupt

(b) Countermeasure by adding timeout processing

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. After setting the descriptors, set bit 0 (TR) in the E-DMAC transmit request register (EDTRR) to start transmission.
3. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the corresponding transmit descriptor.
4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, place the transmission task in a waiting state by making an OS service call of a routine with a timeout function (e.g. acquire a semaphore that has a timeout).

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

5. When the transmission task has left the waiting state and entered the execution state within the time limit, set the frame for transmission in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission. Taking the transmission task out of the waiting state should be done by the interrupt handler when the TC interrupt is generated.
6. When the timeout limit is reached, check the TACT bit in the corresponding transmit descriptor. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, place the transmission task in a waiting state by making an OS service call of a routine with a timeout function, or execute a software reset to initialize all of the modules associated with Ethernet functionality.

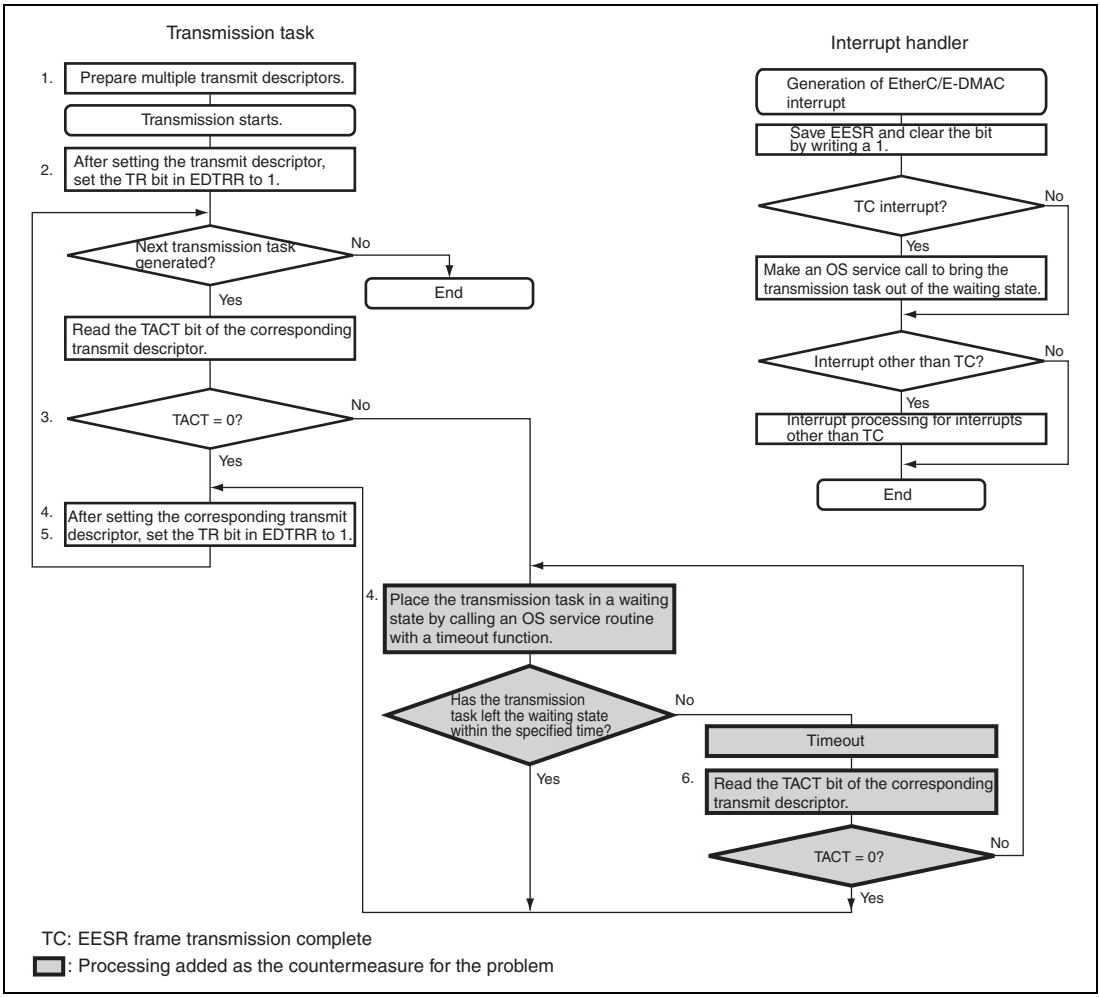


Figure 12.10 Method of Adding Timeout Processing

12.4.2 Usage Notes on SH-Ether Transmit-FIFO Underflow

In the transmission operation of the on-chip E-DMAC of the SH-Ether, if the E-DMAC cannot acquire bus-mastership due to occupancy of the bus by a bus master other than the E-DMAC, data are not writable to the transmit FIFO and an underflow occurs. The expected operation from that point is as follows: on obtaining the bus mastership, the E-DMAC resumes transmission of the remaining data for transmission; on completion of the DMA transfer, it writes back to the corresponding descriptor, and then fetches the next transmit descriptor. However, if the size of the transmit FIFO set by the FIFO depth register (FDR) ≤ maximum frame length for transmission (1518 bytes), the E-DMAC may stop operating even if the transmit request bit (TR) in the E-DMAC transmit request register (EDTRR) is set to 1, according to the relationship between the length of the remaining frame data and the value of the transmit FIFO pointer.

The relationship between the stoppage of E-DMAC operation and the state of the transmit FIFO is shown below.

The data for transmission, which are placed in external memory (transmit buffer), are DMA-transferred by the E-DMAC to the transmit FIFO and output from the MII pin via the EtherC module. The transmit FIFO write pointer (WP) is used when the E-DMAC writes the data for transmission to the transmit FIFO, and the transmit FIFO read pointer (RP) is used when the EtherC module reads the data for transmission from the transmit FIFO.

1. After a software reset, the transmit FIFO will have been initialized, and WP and RP will hold the minimum and maximum values, respectively, of the transmit FIFO capacity.
2. When the E-DMAC starts DMA transfer, WP is incremented when the data for transmission are written to the transmit FIFO. On the other hand, RP is incremented when the data written to the transmit FIFO are read out by the EtherC module.

Note: The transmit FIFO only stores the data of a single frame that is being processed. It does not store data extending over multiple frames. This means that the E-DMAC does not transfer the next frame to the transmit FIFO until the data of the frame being processed are read from the transmit FIFO.

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

However, as stated in step 3, the read access to the transmit FIFO by the EtherC module will have been terminated, and the E-DMAC thus stops operating with the transmit FIFO full.
In short, this problem arises when [initial value of RP – WP value < length of remaining frame data] at the point of the transmit underflow.

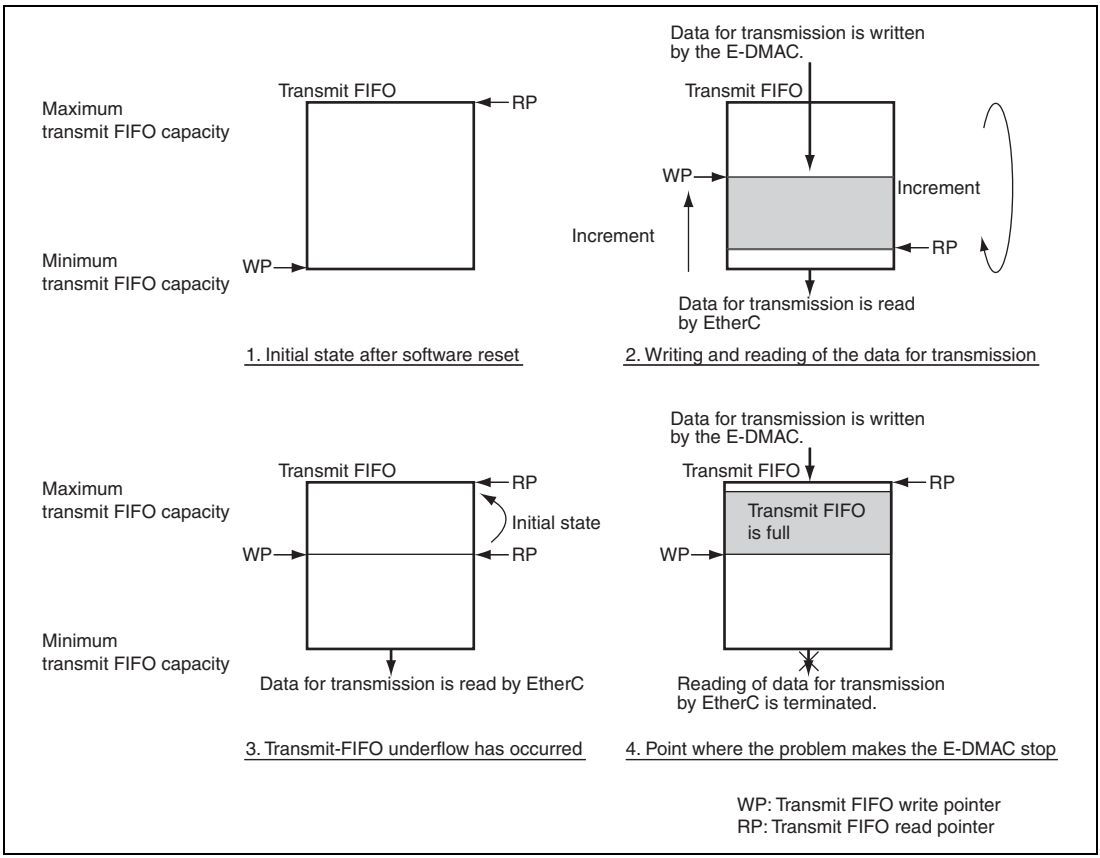


Figure 12.11 Operation when E-DMAC Stops and the Transmit FIFO

(1) Countermeasure

This problem occurs under this condition:
size of transmit FIFO set in the FIFO depth register (FDR) ≤ maximum length of frame for transmission (1518bytes).

To release the E-DMAC from the stopped state due to this problem, execute a software reset to initialize both the E-DMAC and EtherC modules.

Specific countermeasures are given below. An example for the case where the software does not use TC interrupts in transmission processing is given as (2), and an example for the case with TC interrupt-driven software is given as (3). Both methods require the addition of timeout processing with a maximum specified time as the timeout limit, and are based on the countermeasures explained in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).

The constant specified time corresponds to the timeout limit stated in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR). The maximum specified time should be set with reference to the maximum times taking retry processing into consideration, as given in table 12.2. Derive n, the number of repetitions of the constant specified time, from this maximum specified time. If transfer takes more than the maximum specified time, this indicates that the E-DMAC has stopped due to a transmission underflow. In this case, execute a software reset to initialize the EtherC and E-DMAC modules. Since the receiving side will also be initialized by the software reset, the receiving side may require processing in a higher-level layer (e.g. TCP/IP).

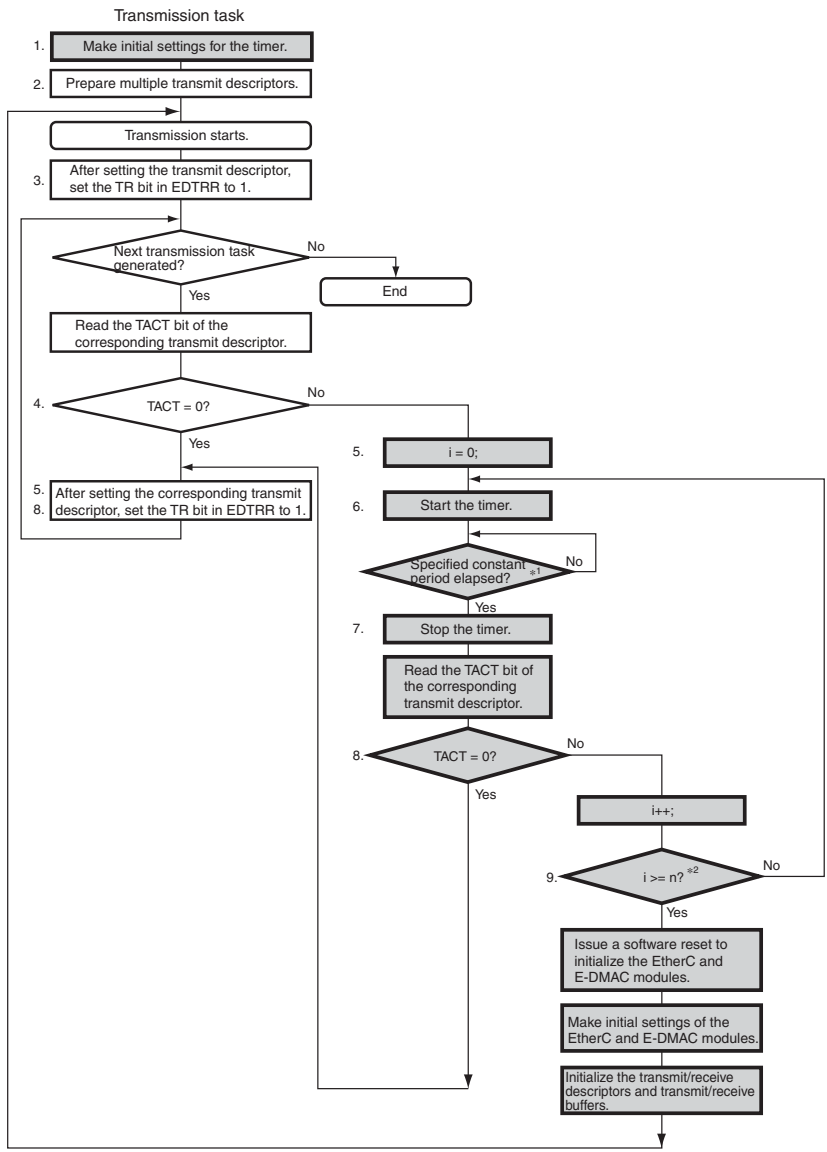
Note: The countermeasure should be the one that best suits the structure of your driver and other software.

(2) Countermeasure for the case where the software handles transmission without the aid of TC interrupts

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

(a) Processing transmission without handling of the frame transmission complete (TC) interrupt

1. Make initial settings for the timer.
2. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
3. After setting the transmit descriptors, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
4. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the corresponding transmit descriptor.
5. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, set counter i to 0 (counter i is the variable that indicates the number of repetitions of the timer operation to measure the specified constant period).
6. Start counting by the timer.
7. When the specified constant period has elapsed, stop the timer counter and check the TACT bit in the corresponding transmit descriptor.
8. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, increment counter i.
9. While the TACT bit is found to be 1 in step 8 and the value of counter i is less than n, repeat steps 6 to 8 until the maximum specified time is reached (the maximum specified time should be set with reference to the maximum times in consideration of retry processing given in table 12.2, and from this maximum specified time, determine n, the number of repetitions of the specified constant period; n is determined by the user with reference to table 12.2).
   If counter i reaches or exceeds n, the maximum specified time has elapsed and we can judge that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E-DMAC modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMR). After re-making initial settings for the Ethernet module, initialize the transmit/receive descriptors and transmit/receive buffers.



Notes: 1. The specified constant period is the timeout period mentioned in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).
2. Set n with reference to the maximum specified time values in table 12.2.

: Processing added as the countermeasure for the problem

Figure 12.12 Processing Transmission without Handling of the TC Interrupt

(3) Countermeasure for the case of TC interrupt-driven software

The sample countermeasure for the case of TC interrupt-driven software shown below is the addition of timeout processing within the limit imposed by the maximum specified time. This is based on the method explained in (b) Countermeasure by adding timeout processing in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).

The maximum specified time should be set with reference to the maximum times in consideration of retry processing (table 12.2). From this maximum specified time, determine n, the number of calls of the OS service routine with a timeout function.

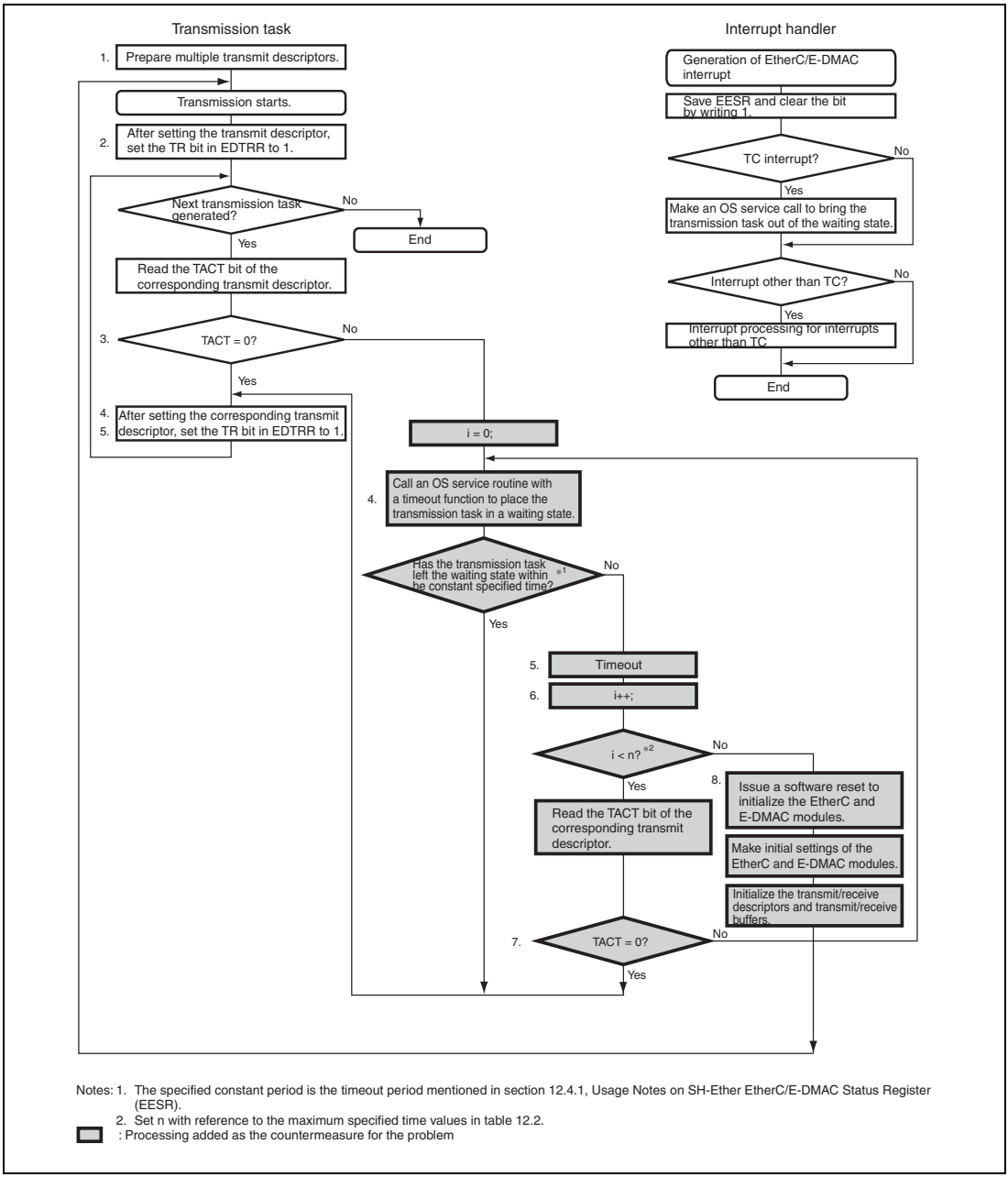
(b) Countermeasure as the addition of timeout processing within the limit imposed by the maximum specified time

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. After setting the transmit descriptors, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
3. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the transmit descriptor.
4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, set counter i to 0 (counter i is the variable that indicates the number of calls of the OS service routine with a timeout function). Then, place the transmission task in a waiting state by calling the OS routine (e.g. acquire a semaphore that has a timeout limit).

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

5. When the transmission task has left the waiting state and entered the execution state within the specified constant period, set the frame for transmission in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission. The transmission task should be taken out of the waiting state by the interrupt handler initiated by generation of the TC interrupt.
6. If the transmission task has not left the waiting state within the specified constant period, increment counter i. Then, if i < n, check the TACT bit in the corresponding transmit descriptor. The value for counting, n, is determined by the user with reference to table 12.2.
7. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, return the transmission task to the waiting state by calling an OS service routine that has a timeout function, and then repeat steps 5 and 6.

8. If counter i reaches or exceeds n, the maximum specified time has elapsed and we can judge that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E-DMAC modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMR). After re-making initial settings for the Ethernet module, initialize the transmit/receive descriptors and transmit/receive buffers.



Notes: 1. The specified constant period is the timeout period mentioned in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).
2. Set n with reference to the maximum specified time values in table 12.2.
: Processing added as the countermeasure for the problem

Figure 12.13 Countermeasure for the Case with TC Interrupt-Driven Software: Addition of Timeout Processing within the Limit Imposed by the Maximum Specified Time

Table 12.2 Reference Values for Maximum Specified Time

Communication Rate		10 Mbps	100 Mbps
Maximum specified time	Full-duplex with no flow control	1.3 ms or longer	130 μs or longer
	Half-duplex with no flow control	183 ms or longer (max. 366 ms)	18.3 ms or longer (max. 36.6 ms)
	With flow control	336 ms or longer	33.6 ms or longer

Note: The maximum specified time refers to the maximum time taken to transmit a single frame or the maximum time for flow control for a single frame.

Section 13 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

13.1 Features

- Four channels (two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode or single address mode can be selected.
- Transfer requests:
  External request, on-chip peripheral module request, or auto request can be selected.
  The following modules can issue an on-chip peripheral module request.
  — SCIF0, SCIF1, SCIF2, and SIOF0
- Selectable bus modes:
  Cycle steal mode (normal mode and intermittent mode) or burst mode can be selected.
- Selectable channel priority levels:
  The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU after transfers end by the specified counts.
- External request detection: There are following four types of DREQ input detection.
  — Low level detection
  — High level detection
  — Rising edge detection
  — Falling edge detection
- Transfer request acknowledge signal:
  Active levels for DACK and TEND can be set independently.

Figure 13.1 shows the block diagram of the DMAC.

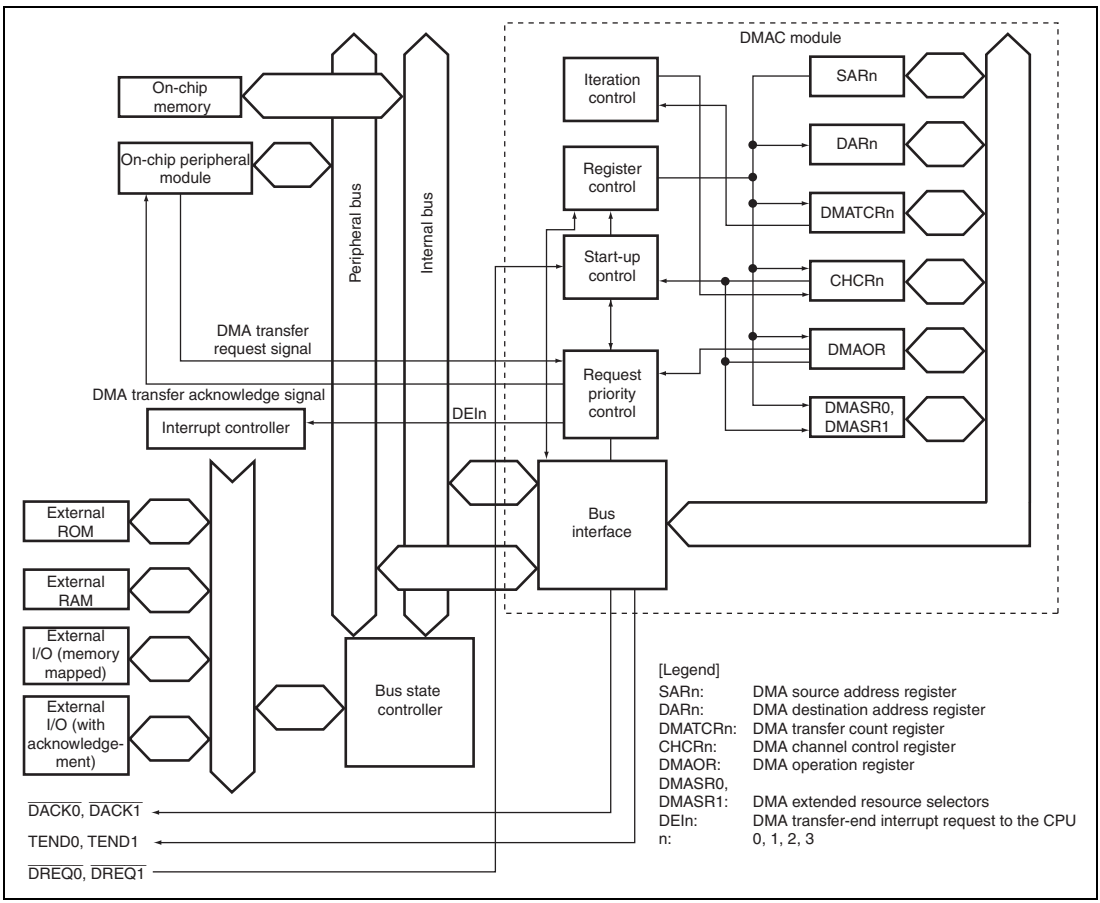


Figure 13.1 Block Diagram of DMAC

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	Pin Name	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	Input	DMA transfer request input from external device to channel 0
	DMA transfer request acknowledge	$\overline{\text{DACK0}}$	Output	DMA transfer request acknowledge output from channel 0 to external device
	DMA transfer end	$\overline{\text{TEND0}}$	Output	DMA transfer end of DMAC channel 0 output of
1	DMA transfer request	$\overline{\text{DREQ1}}$	Input	DMA transfer request input from external device to channel 1
	DMA transfer request acknowledge	$\overline{\text{DACK1}}$	Output	DMA transfer request acknowledge output from channel 1 to external device
	DMA transfer end	$\overline{\text{TEND1}}$	Output	DMA transfer end of DMAC channel 1 output

13.3 Register Descriptions

The DMAC has the following registers. See section 24, List of Registers, for the addresses of these registers and the state of them in each processing status. The SAR for channel 0 is expressed such as SAR_0.

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)

Common:

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

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

SAR are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address. When the data is transferred from an external device with the DACK in single address mode, the 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 must be set for the source address value. The initial value is undefined.

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

DAR are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address. When the data is transferred from an external device with the DACK in single address mode, the 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 must be set for the destination address value. The initial value is undefined.

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

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

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

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

CHCR are 32-bit readable/writable registers that control the DMA transfer mode.

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

Bit	Bit Name	Initial Value	R/W	Descriptions
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read as 0 in CHCR_2 and CHCR_3. The write value should always be 0. 0: Low-active output of DACK 1: High-active output of DACK
15	DM1	0	R/W	Destination Address Mode 1, 0
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 DACK.) 00: Fixed destination address (setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer; setting prohibited in 16-byte transfer) 11: Setting prohibited
13	SM1	0	R/W	Source Address Mode 1, 0
12	SM0	0	R/W	Specify whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address (setting prohibited in 16-byte transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer; setting prohibited in 16-byte transfer) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11	RS3	0	R/W	Resource Select 3 to 0
10	RS2	0	R/W	Specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that the DMA enable bit (DE) is set to 0.
9	RS1	0	R/W	
8	RS0	0	R/W	

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

Note: External request specification is valid only in CHCR_0 and CHCR_1. None of the external request can be selected in CHCR_2 and CHCR_3.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	Specify the detecting method of the DREQ pin input and the detecting level. These bits are valid only in CHCR_0 and CHCR_1. These bits are always reserved and read as 0 in CHCR_2 and CHCR_3. 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, these bits are 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 DMA transfers data. 0: Cycle steal mode 1: Burst mode
4	TS1	0	R/W	Transfer Size 1, 0
3	TS0	0	R/W	Specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte size 01: Word size (2 bytes) 10: Longword size (4 bytes) 11: 16-byte unit (four longword transfers)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is 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.

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag Shows that DMA transfer ends. 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 is cleared to 0. • DMA transfer is ended by clearing the DE bit and DME bit in the DMA operation register (DMAOR). To clear the TE bit, the TE bit should be written to 0 after reading 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been interrupted [Clearing condition] Writing 0 after TE = 1 read 1: DMA transfer ends by the specified count (DMATCR = 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this time, all of the bits TE, NMIF, and AE in DMAOR must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0, which is the same as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

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

[Notice]

About the TE bit (Transfer End Flag) of DMA Channel Control Registers (CHCR) in DMAC;

Just when a flag is set to 1, if the flag is read, the read data will be 0, but the internal state will be the same as reading 1.

In that case, if the flag is written 0, the flag will be cleared as 0, because the internal state is the same as reading 1.

[Workaround]

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

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

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

If the flag is not used, it is no problem to write 0 to flag (in the case of intended bit clear, write 0 after reading 1 to 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 shows the DMA transfer status.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	CMS1	0	R/W	Cycle Steal Mode Select 1, 0
12	CMS0	0	R/W	Select either normal mode or intermittent mode in cycle steal mode. It is necessary that all channel's bus modes are set to cycle steal mode to make valid intermittent mode. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clocks of an external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clocks of an external bus clock.
11, 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 1, 0
8	PR0	0	R/W	Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: Setting prohibited 11: Round-robin mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred during DMA transfer. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error [Clearing condition] Writing AE = 0 after AE = 1 read 1: DMAC address error occurs
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. 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. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input. 0: No NMI interrupt [Clearing condition] Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in CHCR are set to 1, transfer is enabled. In this time, all of the bits TE in CHCR, NMIF, and AE in DMAOR must be 0. If this bit is cleared during transfer, transfers in all channels are terminated. 0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels

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

[Notice]

About the AE bit (Address Error Flag) and the NMIF bit (NMIF Flag) of DMA Operation Register (DMAOR) in DMAC;

Just when a flag is set to 1, if the flag is read, the read data will be 0, but the internal state will be the same as reading 1.

In that case, if the flag is written 0, the flag will be cleared as 0, because the internal state is the same as reading 1.

[Workaround]

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

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

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

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

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

13.3.6 DMA Extended Resource Selectors 0 and 1 (DMARS0 and DMARS1)

DMARS are 16-bit readable/writable registers that specify the DMA transfer request sources from peripheral modules in each channel. DMARS0 specifies the sources for channels 0 and 1, and DMARS1 specifies the sources for channels 2 and 3. This register can set the transfer request of SCIF0, SCIF1, SCIF2, and SIOF0.

When MID/RID other than the values listed in table 13.2 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) has been set to B'1000 for CHCR_0 to CHCR_3 registers. Otherwise, even if DMARS has been set, transfer request source is not accepted.

- DMARS0

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

- DMARS1

Bit	Bit Name	Initial Value	R/W	Description
15	C3MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 3 (MID) See table 13.2.
14	C3MID4	0	R/W	
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 register ID1 and ID0 for DMA channel 3 (RID) See table 13.2.
8	C3RID0	0	R/W	
7	C2MID5	0	R/W	Transfer request module ID5 to ID0 for DMA channel 2 (MID) See table 13.2.
6	C2MID4	0	R/W	
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request register ID1 and ID0 for DMA channel 2 (RID) See table 13.2.
0	C2RID0	0	R/W	

Table 13.2 Transfer Request Sources

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

13.4 Operation

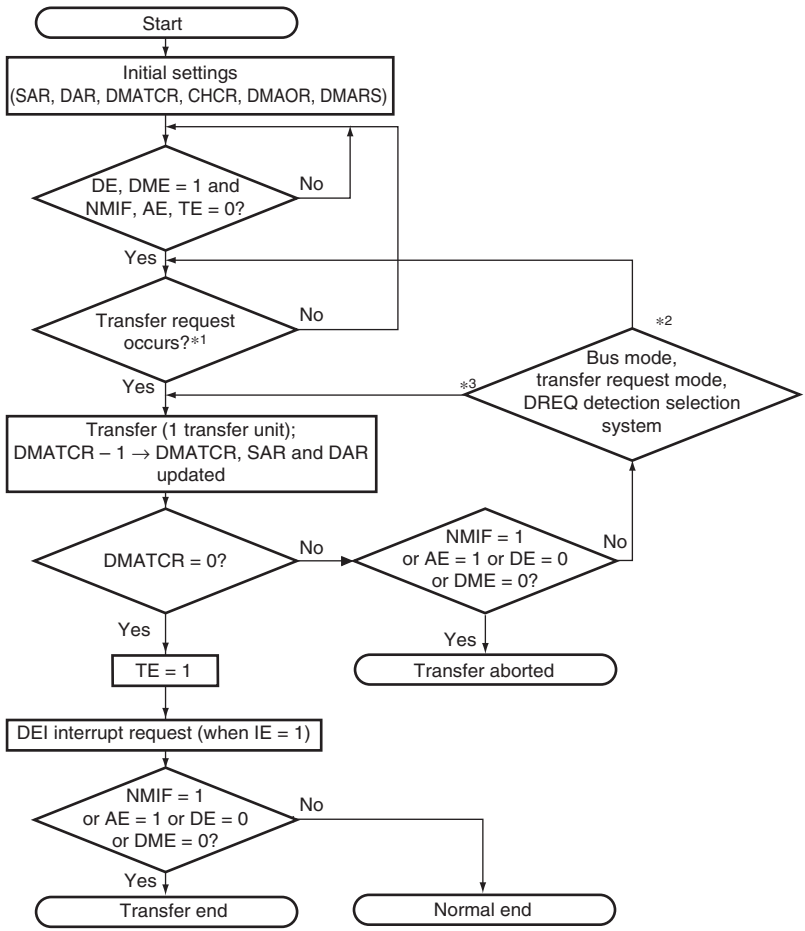
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or 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 extended resource selectors (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). In auto request mode, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a 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 in CHCR or the DME bit in DMAOR is changed to 0.

Figure 13.2 shows a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits 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 external devices or 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 DMARS0 and DMARS 1.

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, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR and the DME bit in DMAOR are set to 1, the transfer begins so long as the AE and NMIF bits in DMAOR are all 0.

External Request Mode: In this mode, a transfer is performed at the request signals (DREQ0 and DREQ1) of an external device. This mode is valid only in channel 0 and channel 1. 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
		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

Choose to detect DREQ by either the edge or level of the signal input with the DL bit and 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_0 or CHCR_1		
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. 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_0 or CHCR_1	
DO	**External Request**
0	Overrun 0
1	Overrun 1

On-Chip Peripheral Module Request Mode: In this mode, a transfer is performed at the transfer request signal of an on-chip peripheral module. Transfer request signals comprise the transmit data empty transfer request and receive data full transfer request from the SCIF0, SCIF1, SCIF2, and SIOF0 set by DMARS0 and DMARS 1.

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

When a transmit data empty transfer request of the SCIF0 is set as the transfer request, the transfer destination must be the SCIF0's transmit data register. Likewise, when receive data full transfer request of the SCIF0 is set as the transfer request, the transfer source must be the SCIF0's receive data register. These conditions also apply to the SCIF1, SCIF2, and SIOF0.

The number of the receive FIFO triggers can be set as a transfer request depending on an on-chip peripheral module. Data needs to be read after the DMA transfer is ended, because data may be remained in the receive FIFO when the receive FIFO trigger condition is not satisfied.

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	TXI0 (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
		10	SCIF0 receiver	RXI0 (receive FIFO data full interrupt)	SCFRDR0	Any	Cycle steal
	001001	01	SCIF1 transmitter	TXI1 (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
		10	SCIF1 receiver	RXI1 (receive FIFO data full interrupt)	SCFRDR1	Any	Cycle steal
	001010	01	SCIF2 transmitter	TXI2 (transmit FIFO data empty interrupt)	Any	SCFTDR2	Cycle steal
		10	SCIF2 receiver	RXI2 (receive FIFO data full interrupt)	SCFRDR2	Any	Cycle steal
	010100	01	SIOF0 transmitter	TXI0 (transmit FIFO data empty interrupt)	Any	SITDR0	Cycle steal
		10	SIOF0 receiver	RXI0 (receive FIFO data full interrupt)	SIRDR0	Any	Cycle steal

13.4.3 Channel Priority

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

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

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

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

Round-Robin Mode: In round-robin mode each time data of one transfer unit (word, byte, longword, or 16-byte unit) is transferred on one channel, the priority is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority. The round-robin mode operation is shown in figure 13.3. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after a reset. When round-robin mode is specified, the same bus mode, either cycle steal mode or burst mode, must be specified for all of the channels.

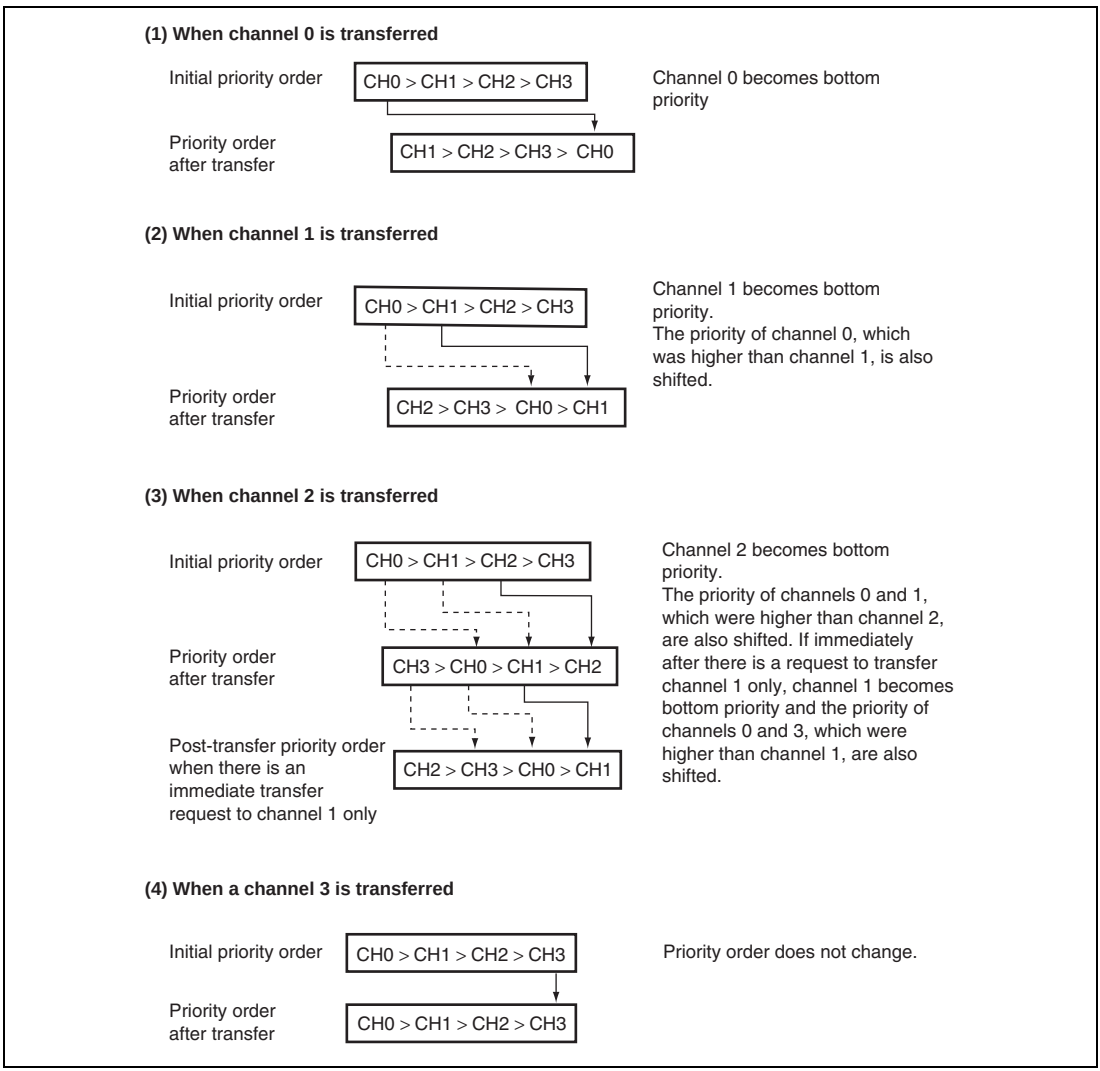


Figure 13.3 Round-Robin Mode

Figure 13.4 shows how the priority changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.

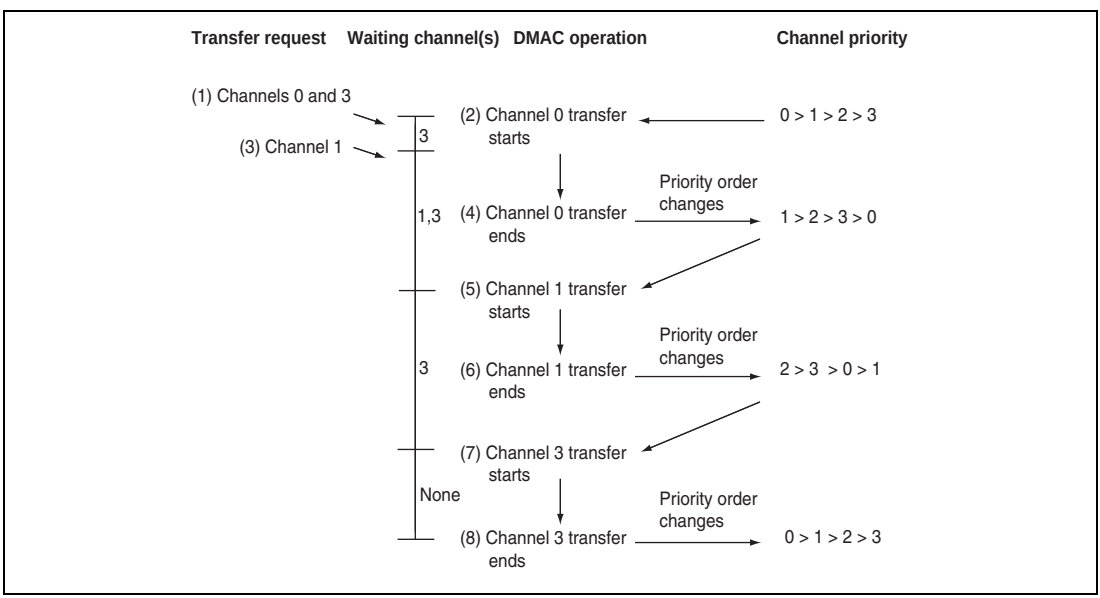


Figure 13.4 Changes in Channel Priority in Round-Robin Mode

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 U Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. For on-chip peripheral modules, 16-byte transfer is available only by registers which can be accessed in longword units.

Address Modes:

- Dual Address Mode

  In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

  DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 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.

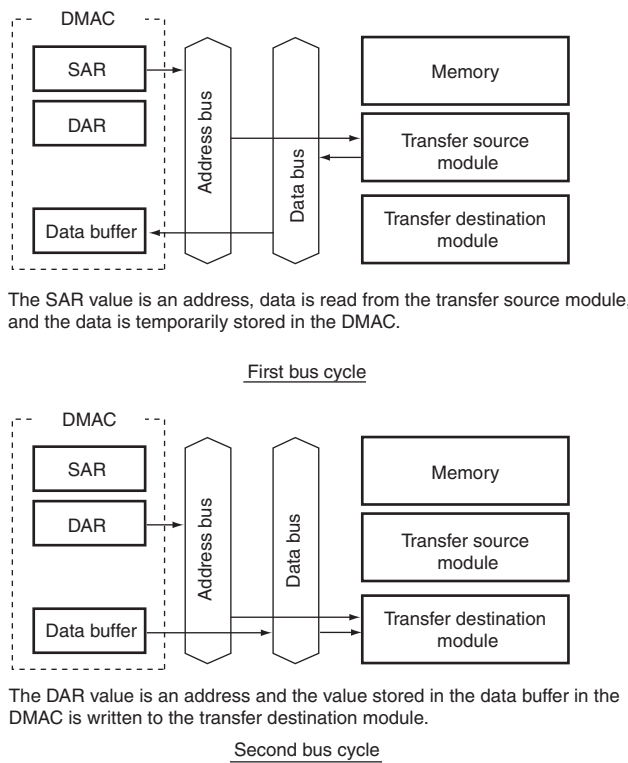
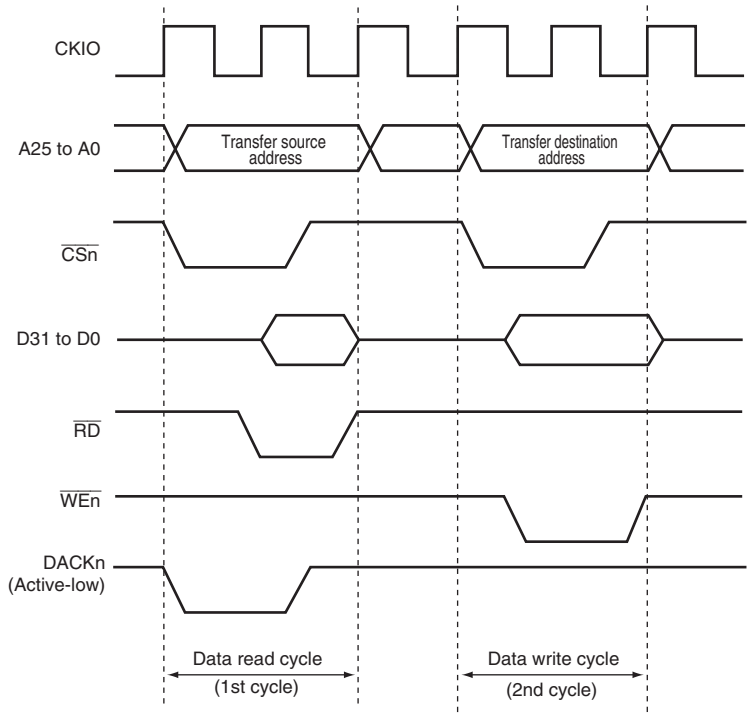


Figure 13.5 Data Flow of Dual Address Mode

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 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.



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

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

- Single Address Mode

  In single address mode, either the transfer source or transfer destination peripheral device is accessed (selected) by means of the DACK signal, and the other device is accessed by an 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.

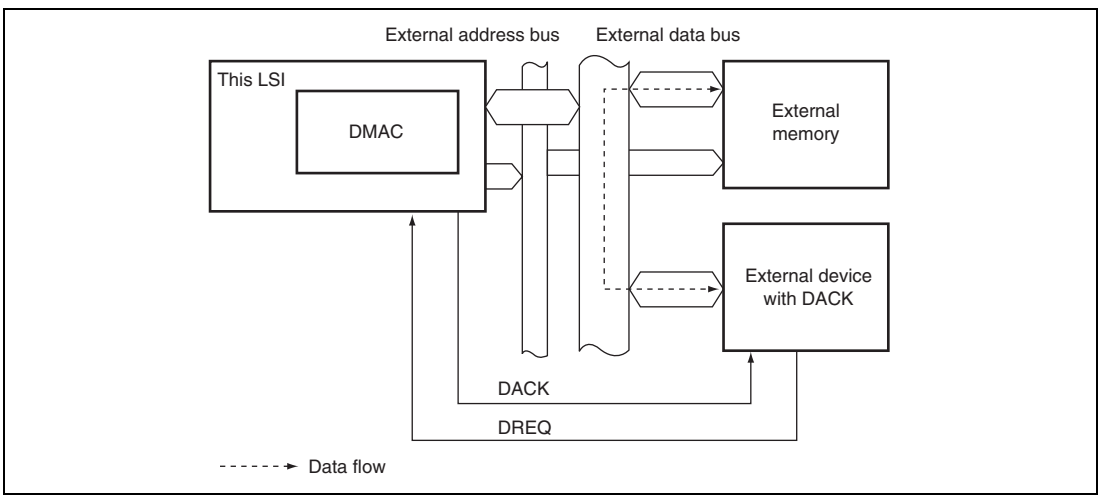


Figure 13.7 Data Flow in Single Address Mode

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.

Figure 13.8 shows an example of DMA transfer timing in single address mode.

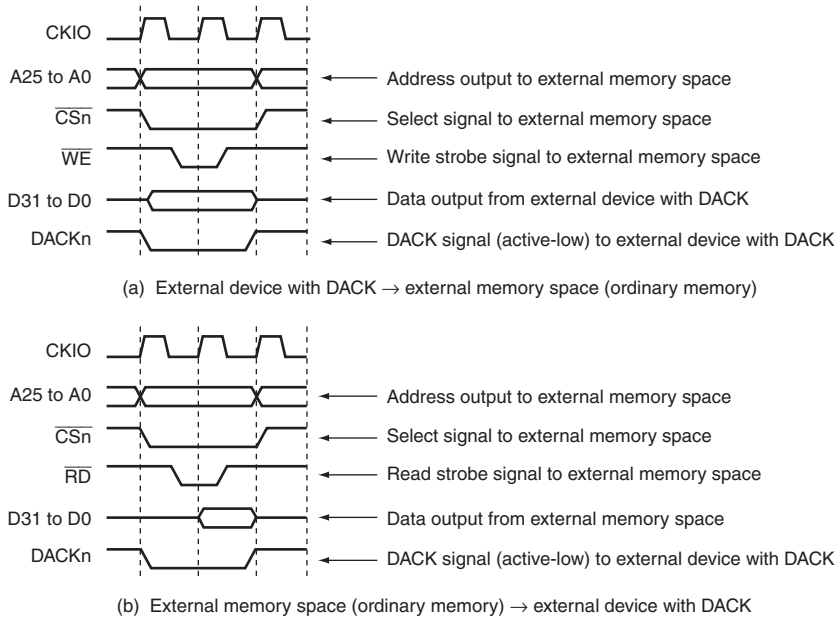


Figure 13.8 Example of DMA Transfer Timing in Single Address Mode

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

- Cycle-Steal Mode
  - — Normal mode

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

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

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

  — Dual address mode

  — DREQ low level detection

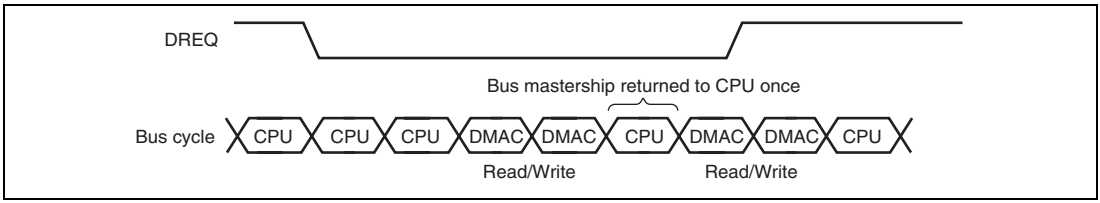


Figure 13.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

— Intermittent mode 16 and intermittent mode 64

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

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

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

Figure 13.10 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:

— Dual address mode

— DREQ low level detection

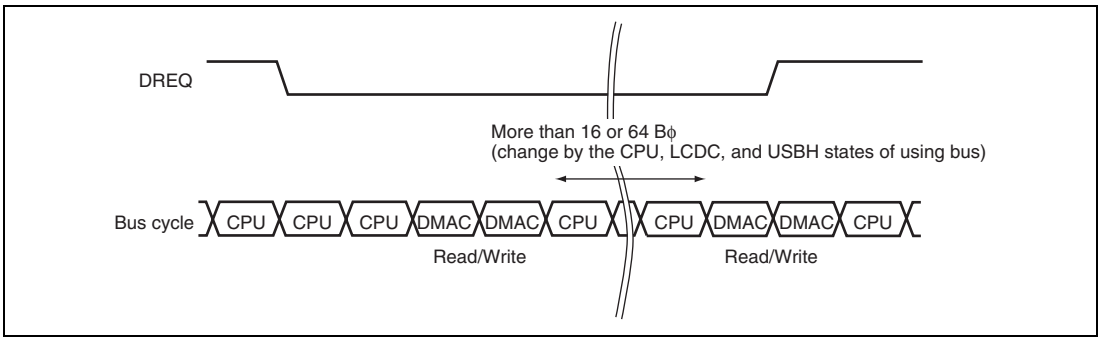


Figure 13.10 Example of DMA Transfer in Cycle Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

- Burst Mode

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

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

Figure 13.11 shows DMA transfer timing in burst mode.

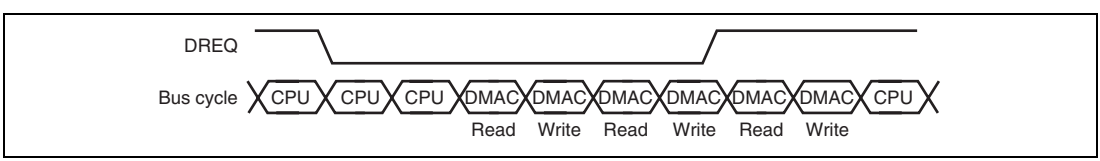


Figure 13.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

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 between 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	All*[1]	B/C	8/16/32/128	0 to 5*[4]
	External memory and memory-mapped external device	All*[1]	B/C	8/16/32/128	0 to 5*[4]
	Memory-mapped external device and memory-mapped external device	All*[1]	B/C	8/16/32/128	0 to 5*[4]
	External memory and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0 to 5*[4]
	Memory-mapped external device and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0 to 5*[4]
	On-chip peripheral module and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0 to 5*[4]
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

B: Burst mode, C: Cycle steal mode

Notes:
1. External requests and auto requests are all available.
2. External requests, auto requests, and on-chip peripheral module requests are all available. However, for on-chip peripheral module requests, the request source register must be designated as the transfer source or the transfer destination.
3. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.
4. If the transfer request is an external request, channels 0 and 1 are only available.

Bus Mode and Channel Priority: When the priority is set in fixed mode (CH0 > CH1), even though channel 1 is transferring in burst mode, if there is a transfer request to channel 0 which has a higher priority, the transfer of channel 0 will begin immediately.

At this time, if channel 0 is also operating in burst mode, the channel 1 transfer will continue when the channel 0 transfer with a higher priority has completely finished.

If channel 0 is operating in cycle steal mode, immediately after channel 0 with a higher priority completes the transfer of one transfer unit, the channel 1 transfer will begin again without releasing the bus mastership. Transfer will then switch between the two in the order of channel 0, channel 1, channel 0, and channel 1. For the bus state, the CPU cycle after cycle steal mode transfer finishes is replaced with a burst mode transfer cycle (hereafter referred to as burst mode high-priority execution).

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

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

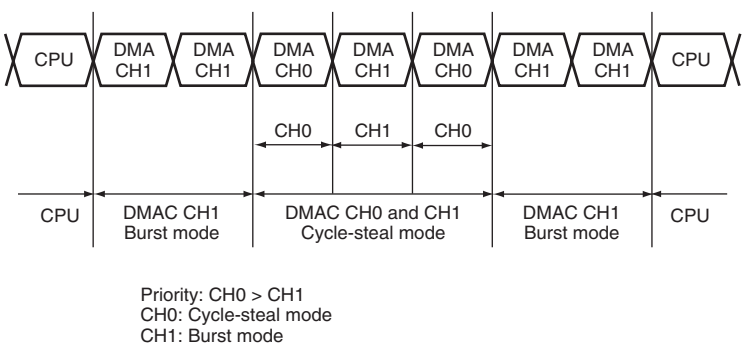


Figure 13.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes according to the specifications shown in figure 13.3. Note that a channel operating in cycle steal mode cannot be handled together with a channel operating in burst mode.

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 7, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing: Figures 13.13, 13.14, 13.15, and 13.16 show the sample timing of the DREQ input in each bus mode, respectively.

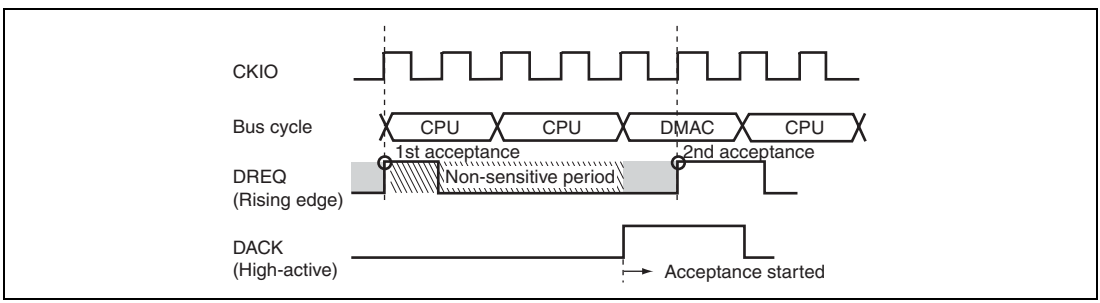


Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

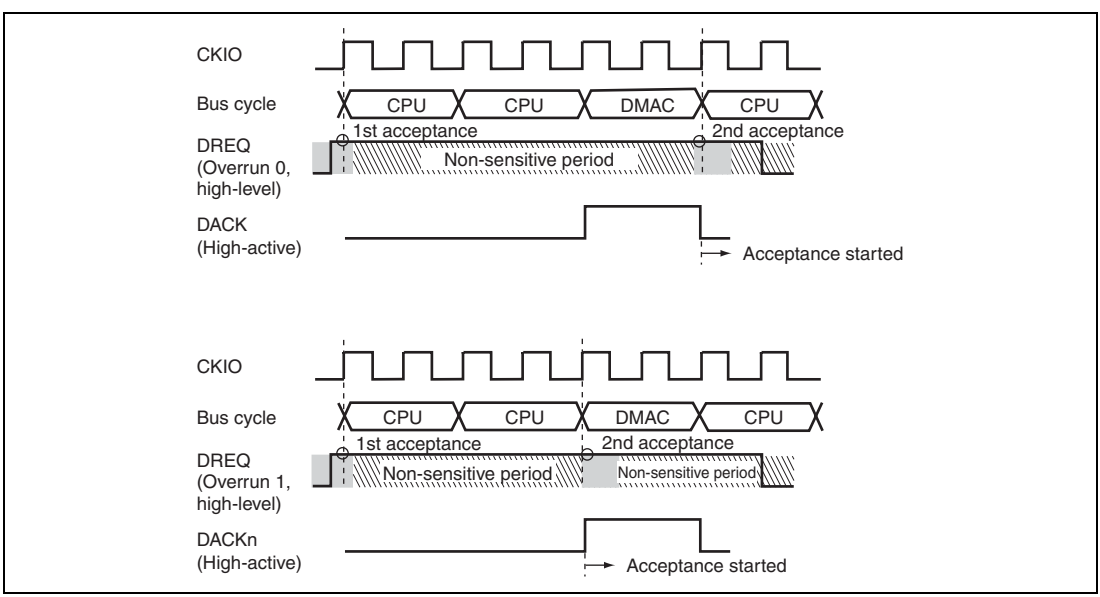


Figure 13.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

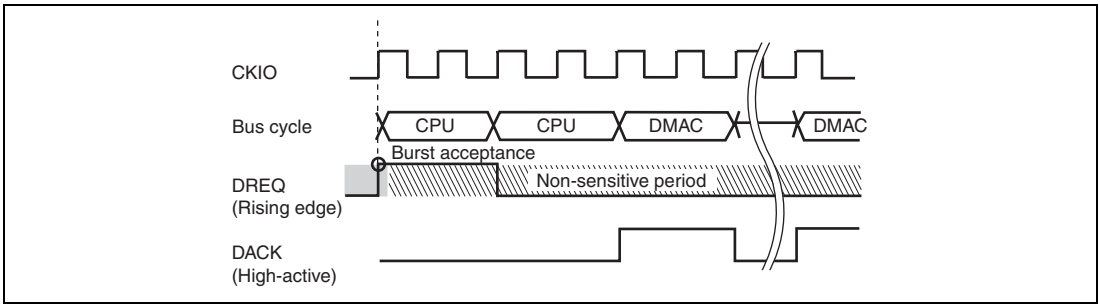


Figure 13.15 Example of DREQ Input Detection in Burst Mode Edge Detection

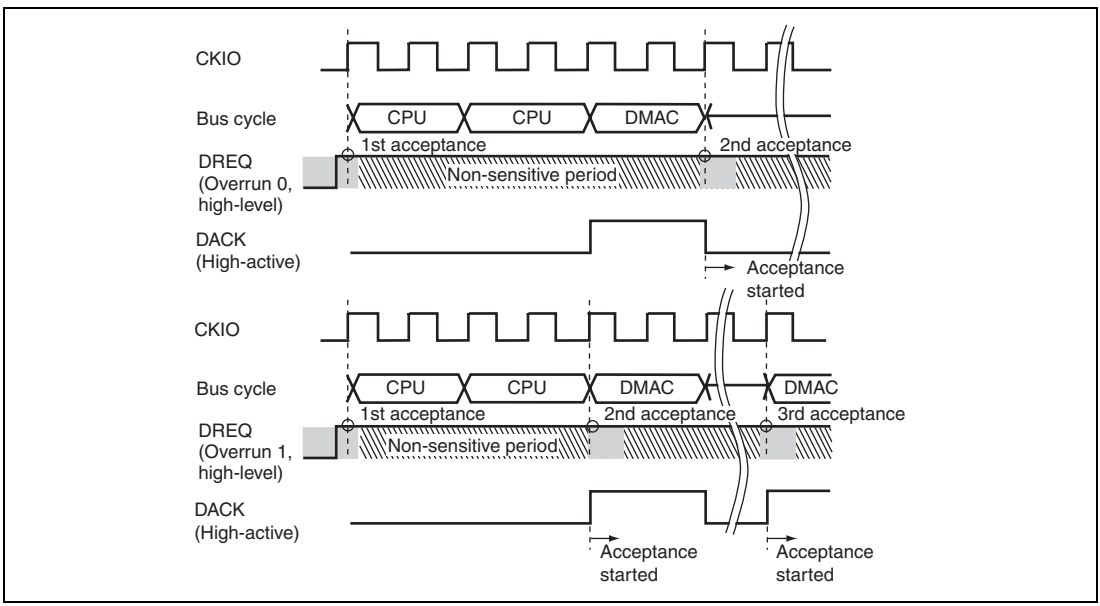


Figure 13.16 Example of DREQ Input Detection in Burst Mode Level Detection

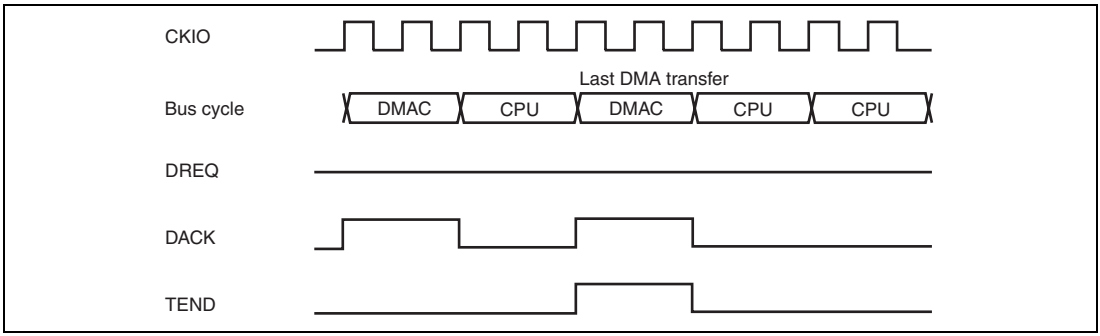
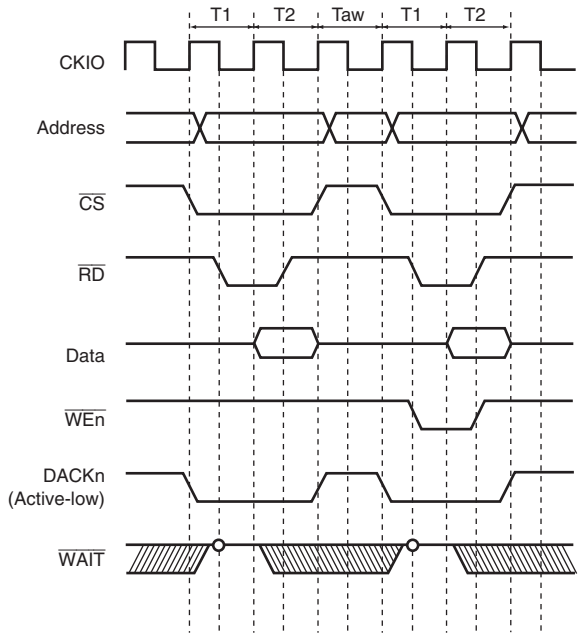


Figure 13.17 Example of DMA Transfer End in Cycle Steal Mode Level Detection

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



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

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

13.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

13.5.1 Notes on DACK Pin Output

When burst mode and cycle steal mode are simultaneously set in two or more channels, an additional DACK may be asserted at the end of burst transfer. This phenomenon will occur when all of the conditions described below are satisfied.

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

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

- Measure 1

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

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

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

13.5.2 Notes On DREQ Sampling When DACK is Divided in External Access

(1) Error Phenomenon

When the DACK output is divided in an external access, DREQ may be sampled twice at maximum in the external access.

(2) Error Conditions and Phenomenon

Conditions: The DACK output is divided in an external access when:

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

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

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

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

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

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

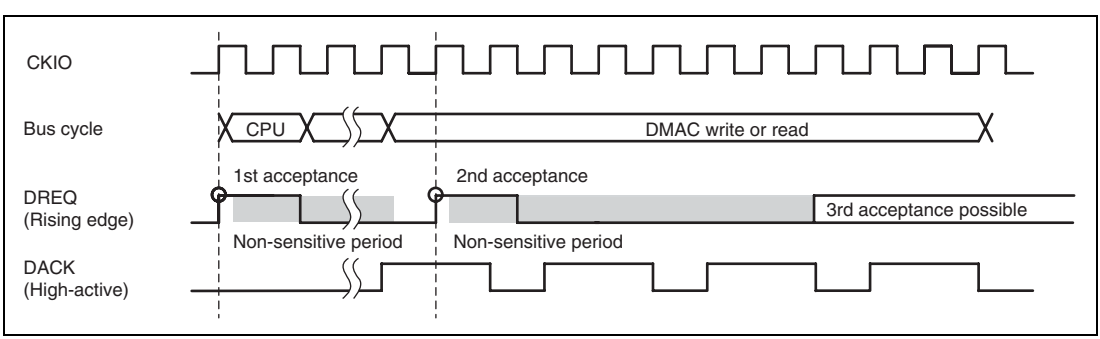


Figure 13.19 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 4 by Idle Cycles

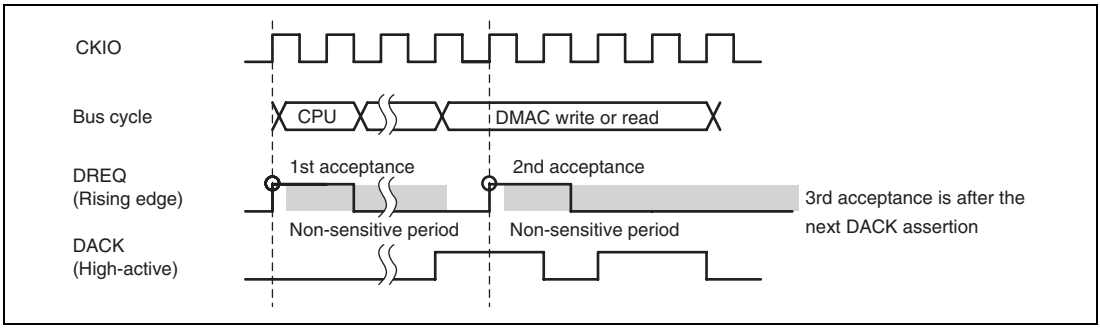


Figure 13.20 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection When DACK is Divided to 2 by Idle Cycles

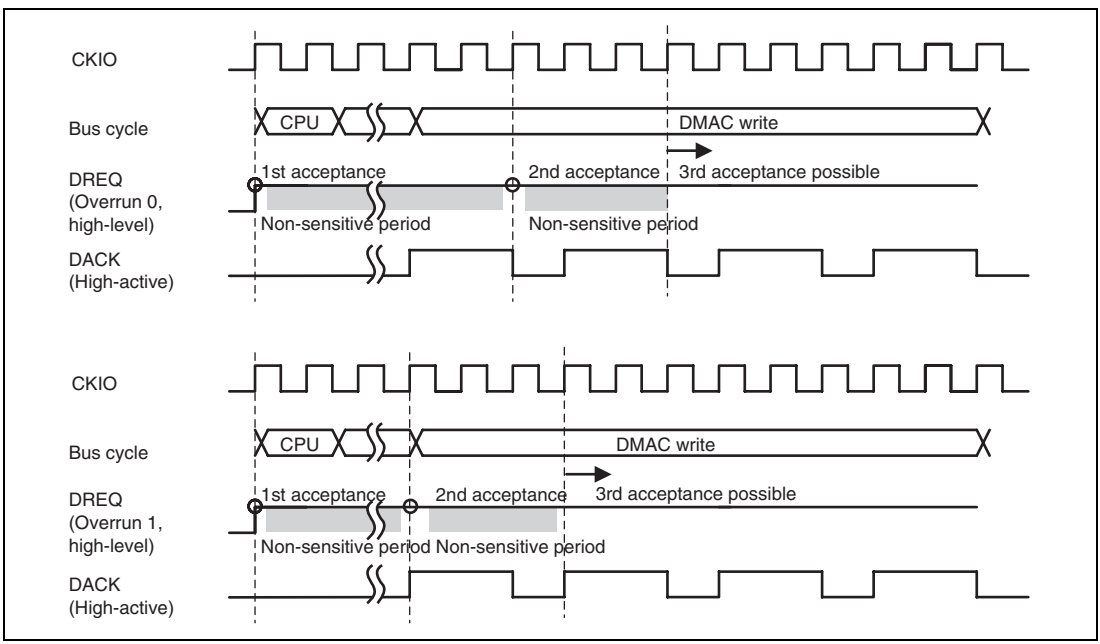


Figure 13.21 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 4 by Idle Cycles

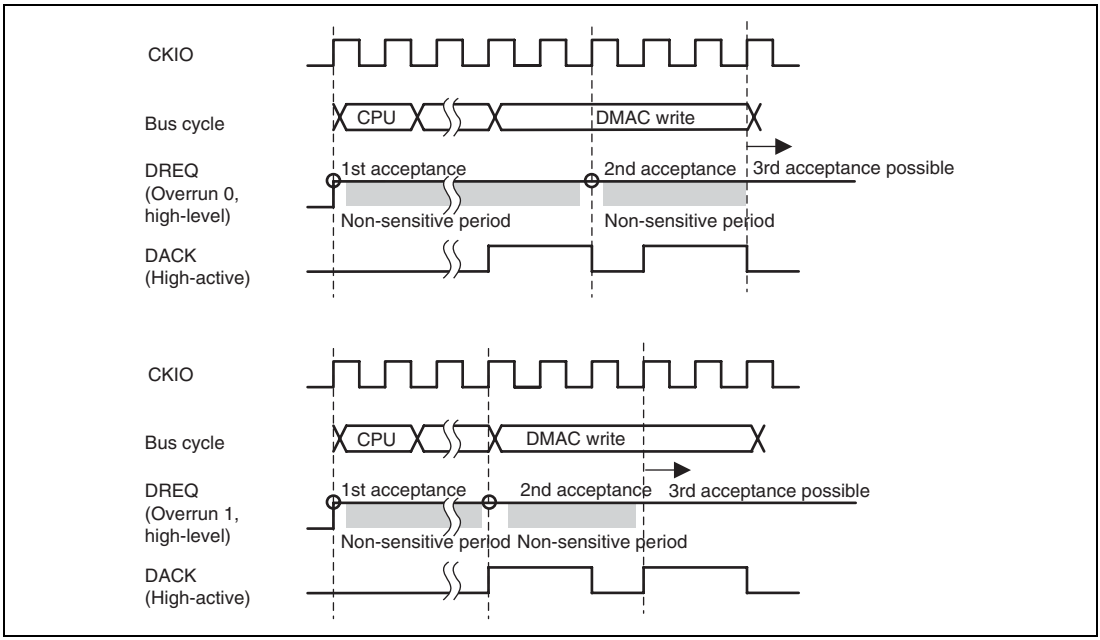


Figure 13.22 Example of DREQ Input Detection in Cycle Steal Mode Level Detection When DACK is Divided to 2 by Idle Cycles

(3) Notes

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

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

13.5.3 Other Notes

1. Before making a transition to standby mode, either wait until DMA transfer finishes or suspend DMA transfer.
2. If an on-chip peripheral module whose clock supply is to be stopped by the module standby function is performing DMA transfer, either wait until DMA transfer finishes or suspend DMA transfer before making a transition to module standby mode.
3. Do not write to SAR, DAR, DMATCR, or DMARS during DMA transfer.

Concerning Above Notes 1 and 2:

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

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

Section 14 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a 2-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

14.1 Features

CMT has the following features.

- Selection of four counter input clocks
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected independently for each channel.
- Interrupt request on compare match
- When not in use, CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 14.1 shows a block diagram of CMT.

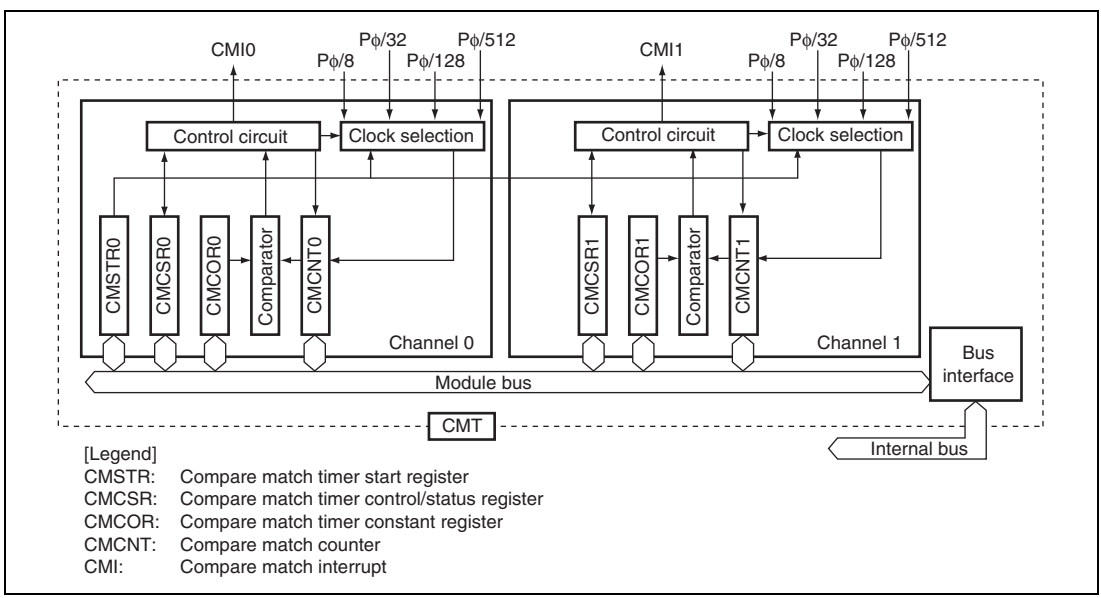


Figure 14.1 Block Diagram of Compare Match Timer

14.2 Register Descriptions

The CMT has the following registers.

- Compare match timer start register (CMSTR)
- Compare match timer control/status register_0 (CMCSR_0)
- Compare match counter_0 (CMCNT_0)
- Compare match constant register_0 (CMCOR_0)
- Compare match timer start register_1 (CMSTR_1)
- Compare match timer control/status register_1 (CMCSR_1)
- Compare match counter_1 (CMCNT_1)
- Compare match constant register_1 (CMCOR_1)

14.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

CMSTR is initialized to H'0000 by a power-on reset and a transition to standby mode.

Bit	Bit Name	Initial value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter 1 operates or is stopped. 0: CMCNT_1 count is stopped 1: CMCNT_1 count is started
0	STR0	0	R/W	Count Start 0 Specifies whether compare match counter 0 operates or is stopped. 0: CMCNT_0 count is stopped 1: CMCNT_0 count is started

14.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables interrupts and selects the counter input clock.

CMCSR is initialized to H'0000 by a power-on reset and a transition to standby mode.

Bit	Bit Name	Initial value	R/W	Description
15 to 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 whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] When 0 is written to this bit 1: CMCNT and CMCOR values match
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF=1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	Select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral operating clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

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

14.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset and a transition to standby mode.

14.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset and is initialized to H'FFFF in standby mode.

14.3 Operation

14.3.1 Interval Count Operation

When an internal clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CMIE bit in CMCSR is set to 1, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 14.2 shows the operation of the compare match counter.

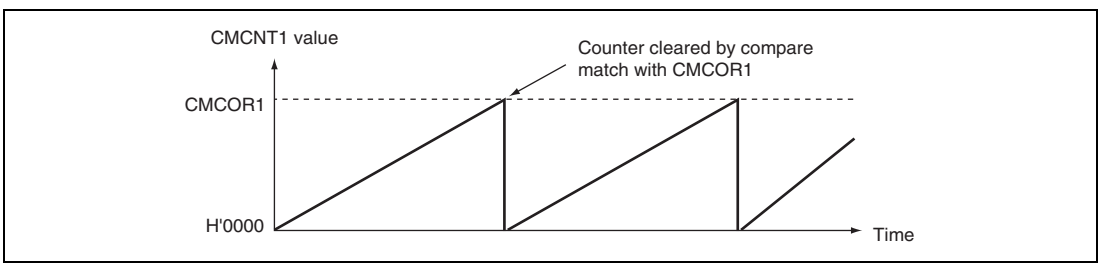


Figure 14.2 Counter Operation

14.3.2 CMCNT Count Timing

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

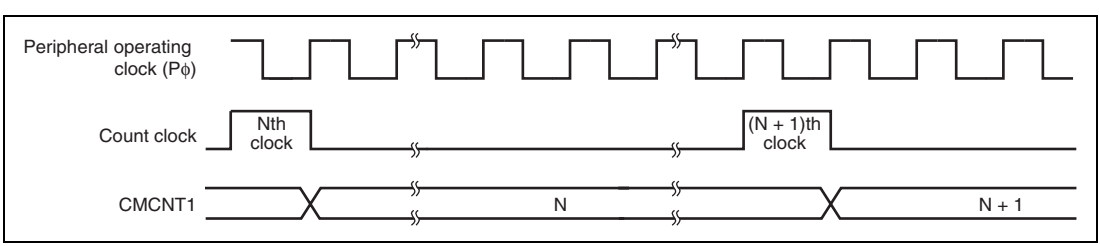


Figure 14.3 Count Timing

14.4 Interrupts

14.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has compare match interrupt. When both the interrupt request flag (CMF) and interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 6, Interrupt Controller (INTC).

14.4.2 Timing of Setting Compare Match Flag

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMCSR is set to 1. The compare match signal is generated in the last cycle in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 14.4 shows the timing of CMF bit setting.

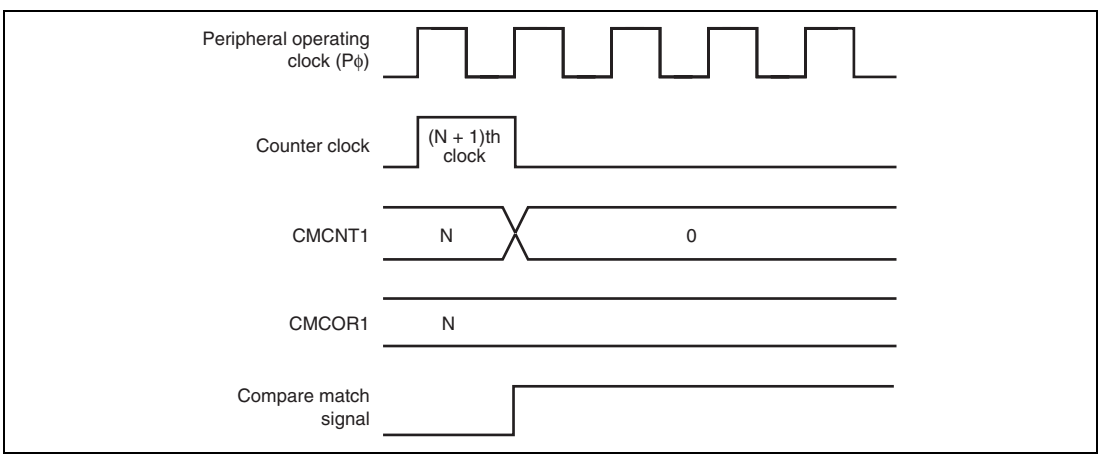


Figure 14.4 Timing of CMF Setting

14.4.3 Timing of Clearing Compare Match Flag

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

14.5 Usage Notes

14.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 14.5 shows the timing to clear the CMCNT counter.

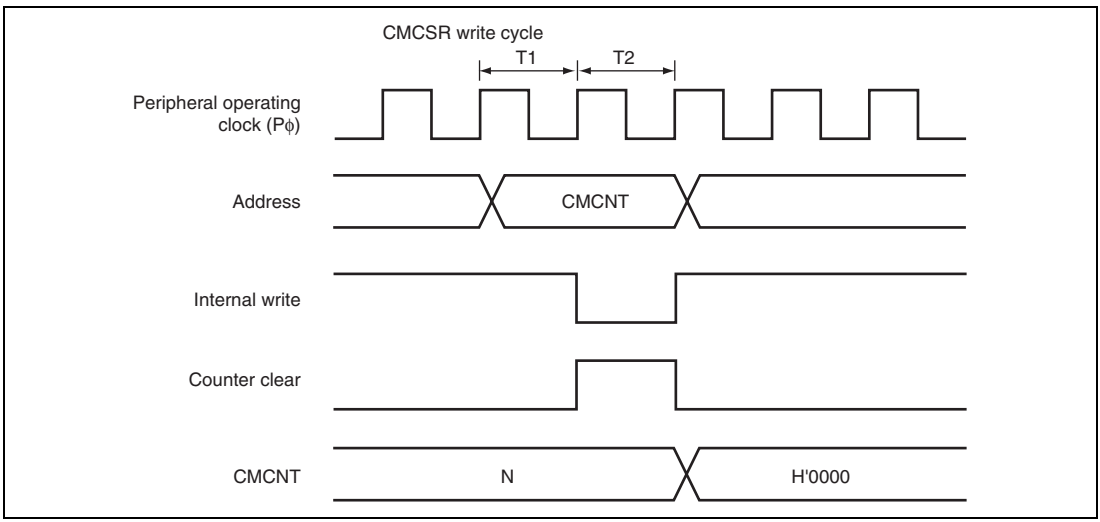


Figure 14.5 Conflict between Write and Compare-Match Processes of CMCNT

14.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 14.6 shows the timing to write to CMCNT in words.

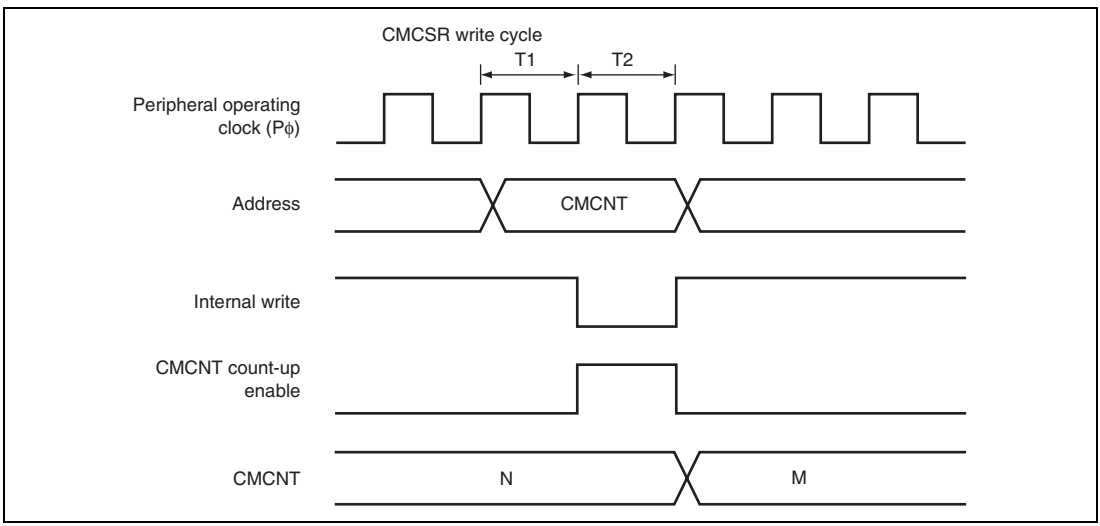


Figure 14.6 Conflict between Word-Write and Count-Up Processes of CMCNT

14.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the byte-writing has priority over the count-up. In this case, the count-up is not performed. The byte data on another side, which is not written to, is also not counted and the previous contents remain.
Figure 14.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNT in bytes.

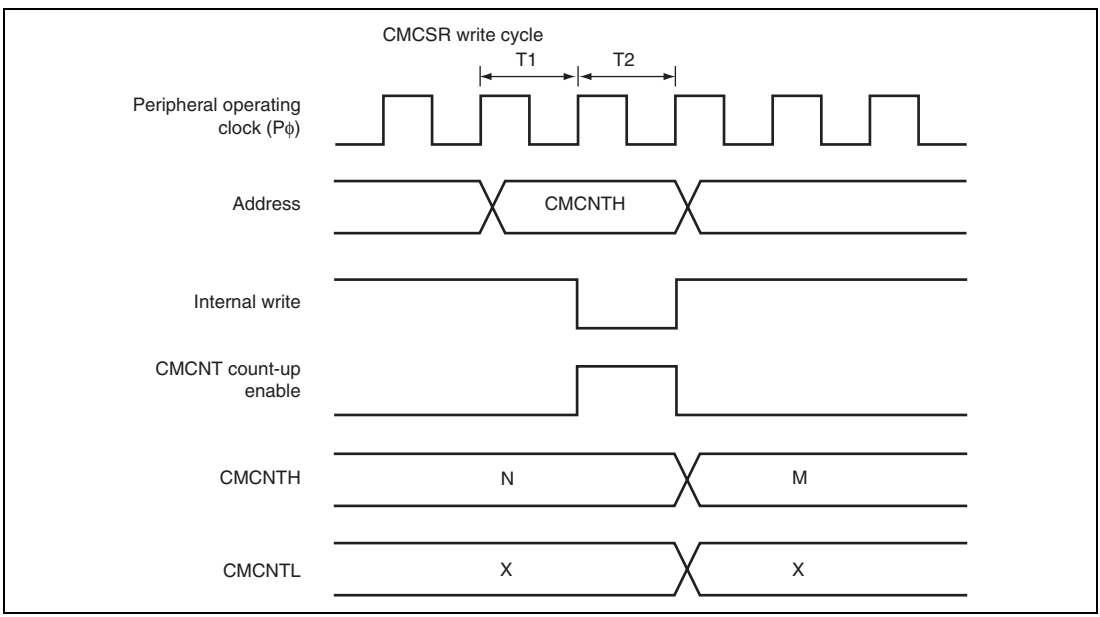


Figure 14.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

14.5.4 Conflict between Write Processes to CMCNT with the Counting Stopped and CMCOR

Writing the same value to CMCNT with the counting stopped and CMCOR is prohibited. If written, the CMF flag in CMCSR is set to 1 and CMCNT is cleared to H'0000.

Section 15 Serial Communication Interface with FIFO (SCIF)

15.1 Overview

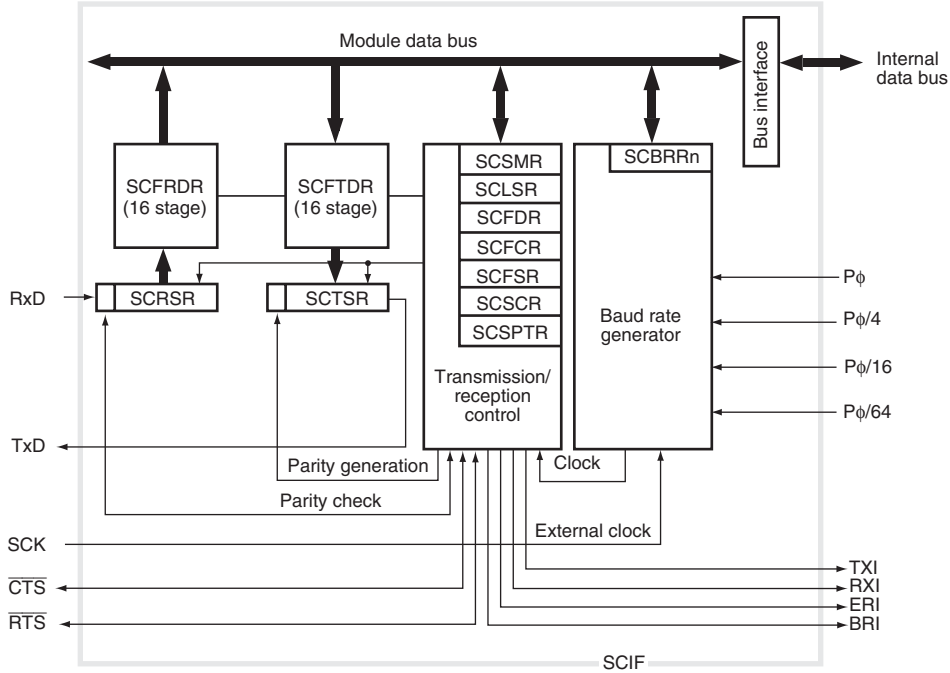
This LSI has a three-channel serial communication interface with FIFO (SCIF) that supports both asynchronous and clock synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

15.1.1 Features

- Asynchronous serial communication:
  - — Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - — Data length: 7 or 8 bits
  - — Stop bit length: 1 or 2 bits
  - — Parity: Even, odd, or none
  - — Receive error detection: Parity, framing, and overrun errors
  - — Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RxD level directly from the port data register when a framing error occurs.
- Synchronous mode:
  - — Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a synchronous communication function. There is one serial data communication format.
  - — Data length: 8 bits
  - — Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interrupts are requested independently.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous, on-chip modem control functions ($\overline{RTS}$ and $\overline{CTS}$) (only for channel 1 and channel 0).
- The number of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 15.1 shows a block diagram of the SCIF for each channel.



[Legend]

SCRSR:	Receive shift register	SCFSR:	Serial status register
SCFRDR:	Receive FIFO data register	SCBRR:	Bit rate register
SCTSR:	Transmit shift register	SCSPTR:	Serial port register
SCFTDR:	Transmit FIFO data register	SCFCR:	FIFO control register
SCSMR:	Serial mode register	SCFDR:	FIFO data count register
SCSCR:	Serial control register	SCLSR:	Line status register

Figure 15.1 Block Diagram of SCIF

15.2 Pin Configuration

The SCIF has the serial pins summarized in table 15.1.

Table 15.1 SCIF Pins

Channel	Pin Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK0	I/O	Clock I/O
	Receive data pin	RxD0	Input	Receive data input
	Transmit data pin	TxD0	Output	Transmit data output
	Request to send pin	$\overline{\text{RTS0}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS0}}$	I/O	Clear to send
1	Serial clock pin	SCK1	I/O	Clock I/O
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data output
	Request to send	$\overline{\text{RTS1}}$	Output	Request to send
	Clear to send pin	$\overline{\text{CTS1}}$	Input	Clear to send
2	Serial clock pin	SCK2	I/O	Clock I/O
	Receive data pin	RxD2	Input	Receive data input
	Transmit data pin	TxD2	Output	Transmit data output

15.3 Register Description

The SCIF has the following registers. These registers specify the data format and bit rate, and control the transmitter and receiver sections.

- Receive FIFO data register_0 (SCFRDR_0)
- Transmit FIFO data register_0 (SCFTDR_0)
- Serial mode register_0 (SCSMR_0)
- Serial control register_0 (SCSCR_0)
- Serial status register_0 (SCFSR_0)
- Bit rate register_0 (SCBRR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Serial port register_0 (SCSPTR_0)
- Line status register_0 (SCLSR_0)
- Receive FIFO data register_1 (SCFRDR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial mode register_1 (SCSMR_1)
- Serial control register_1 (SCSCR_1)
- Serial status register_1 (SCFSR_1)
- Bit rate register_1 (SCBRR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Serial port register_1 (SCSPTR_1)
- Line status register_1 (SCLSR_1)
- Receive FIFO data register_2 (SCFRDR_2)
- Transmit FIFO data register_2 (SCFTDR_2)
- Serial mode register_2 (SCSMR_2)
- Serial control register_2 (SCSCR_2)
- Serial status register_2 (SCFSR_2)
- Bit rate register_2 (SCBRR_2)
- FIFO control register_2 (SCFCR_2)
- FIFO data count register_2 (SCFDR_2)
- Serial port register_2 (SCSPTR_2)
- Line status register_2 (SCLSR_2)

15.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCFRDR, the receive FIFO data register. The CPU cannot read or write to SCRSR directly.

15.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-stage 8-bit FIFO register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored.

The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined. When this register is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	R	FIFO for transmits serial data

15.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again. The CPU cannot read or write to SCTSR directly.

15.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-stage 8-bit FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into SCTSR and starts serial transmission. Continuous serial transmission is performed until there is no transmit data left in SCFTDR. SCFTDR can always be written to by the CPU.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

SCFTDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	W	FIFO for transmits serial data

15.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or synchronous mode. 0: Asynchronous mode 1: Synchronous mode

Bit	Bit Name	Initial value	R/W	Description
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the synchronous mode, the data length is always eight bits, 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 is not transmitted.
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*[1] 1: Odd parity*[2] Note: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in synchronous mode because no stop bits are added. When receiving, 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, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted 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 internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 15.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

15.3.6 Serial Control Register (SCSCR)

SCSCR is a 16-bit register that operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI). Serial transmit data in the transmit FIFO data register (SCFTDR) is send to the transmit shift register (SCTSR). Then, the TDFE flag in the serial status register (SCFSR) is set to1 when the number of data in SCFTDR becomes less than the number of transmission triggers. At this time, a TXI is requested. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled* 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled Note: * The TXI interrupt request can be cleared by writing a greater number of transmit data than the specified transmission trigger number to SCFTDR and by clearing the TDFE bit to 0 after reading 1 from the TDFE bit, or can be cleared by clearing this bit to 0.

Bit	Bit Name	Initial value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-data-full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. 0: Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled* 1: Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIF serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

Bit	Bit Name	Initial value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the SCIF serial receiver. 0:Receiver disabled*[1] 1: Receiver enabled*[2] Note: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled* 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. The CKE0 setting is valid only when the SCIF is operating on the internal clock (CKE1 = 0). The CKE0 setting is ignored when an external clock source is selected (CKE1 = 1). In synchronous mode, select the SCIF operating mode in the serial mode register (SCSMR), then set CKE1 and CKE0. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (The input signal is ignored. The state of the SCK pin depends on both the SCKIO and SCKDT bits.) 01: Internal clock, SCK pin used for clock output (The output clock frequency is 16 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate.) 11: Setting prohibited • Synchronous mode 00: Internal clock, SCK pin used for serial clock output 01: Internal clock, SCK pin used for serial clock output 10: External clock, SCK pin used for serial clock input 11: Setting prohibited

15.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receives errors in the SCFRDR data, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written. SCFSR is initialized to H'0060 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15	PER3	0	R	Number of Parity Errors
14	PER2	0	R	Indicate the number of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER3 to PER0 show 0.
13	PER1	0	R	
12	PER0	0	R	
11	FER3	0	R	Number of Framing Errors
10	FER2	0	R	Indicate the number of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER3 to FER0 show 0.
9	FER1	0	R	
8	FER0	0	R	

Bit	Bit Name	Initial value	R/W	Description
7	ER	0	R/(W)*	Receive Error Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity. *1 0: Receiving is in progress or has ended normally [Clearing conditions] • ER is cleared to 0 a power-on reset • ER is cleared to 0 when the chip is when 0 is written after 1 is read from ER 1: A framing error or parity error has occurred. [Setting conditions] • ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*2 • ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/$\overline{E}$ bit in SCSMR Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCRDR includes a receive error can be detected by the FER and PER bits in SCFSR. 2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial value	R/W	Description
6	TEND	0	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing conditions] • TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR 1: End of transmission [Setting conditions] • TEND is set to 1 when the chip is a power-on reset • TEND is set to 1 when TE is cleared to 0 in the serial control register (SCSCR) • TEND is set to 1 when SCFTDR does not contain receive data when the last bit of a one-byte serial character is transmitted

Bit	Bit Name	Initial value	R/W	Description
5	TDFE	0	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the number of data in SCFTDR has become less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The number of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from the TDFE bit and then 0 is written • TDFE is cleared to 0 when DMAC write data exceeding the specified transmission trigger number to SCFTDR 1: The number of transmit data in SCFTDR is equal to or less than the specified transmission trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the number of transmit data in SCFTDR has become equal to or less than the specified transmission trigger number as a result of transmission Note: * Since SCFTDR is a 16-byte FIFO register, the maximum number of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write additional data, the data is ignored. The number of data in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit	Bit Name	Initial value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal received [Clearing conditions] • BRK is cleared to 0 when the chip is a power-on reset • BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a framing error is received, and a framing error occurs with space 0 in the subsequent receive data Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.
3	FER	0	R	Framing Error Indicates a framing error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive framing error occurred in the next data read from SCFRDR [Clearing conditions] • FER is cleared to 0 when the chip undergoes a power-on reset • FER is cleared to 0 when no framing error is present in the next data read from SCFRDR 1: A receive framing error occurred in the next data read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is present in the next data read from SCFRDR

Bit	Bit Name	Initial value	R/W	Description
2	PER	0	R	Parity Error Indicates a parity error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive parity error occurred in the next data read from SCFRDR [Clearing conditions] • PER is cleared to 0 when the chip undergoes a power-on reset • PER is cleared to 0 when no parity error is present in the next data read from SCFRDR 1: A receive parity error occurred in the data read from SCFRDR [Setting condition] • PER is set to 1 when a parity error is present in the next data read from SCFRDR

Bit	Bit Name	Initial value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the number of data in SCFRDR has become more than the receive trigger number specified by the RTRG1 and RTRG0 bits in the FIFO control register (SCFCR). 0: The number of transmit data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset • RDF is cleared to 0 when the SCFRDR is read until the number of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written 1: The number of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a number of receive data more than the specified receive trigger number is stored in SCFRDR* Note: * SCFTDR is a 16-byte FIFO register. When RDF is 1, the specified receive trigger number of data can be read at the maximum. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The number of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

Bit	Bit Name	Initial value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the number of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1. 0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally [Clearing conditions] • DR is cleared to 0 when the chip undergoes a power-on reset • DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written 1: Next receive data has not been received [Setting conditions] • DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

Note: * The only value that can be written is 0 to clear the flag.

15.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset. Each channel has independent baud rate generator control, so different values can be set in three channels.

The SCBRR setting is calculated as follows:

- Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- Synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 15.2.)

Table 15.2 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64^{2n-1} \times 2} - 1 \right\} \times 100$$

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

Table 15.3 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pϕ (MHz)								
	5			6			6.144		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	88	–0.25	2	106	–0.44	2	108	0.08
150	2	64	0.16	2	77	0.16	2	79	0.00
300	1	129	0.16	1	155	0.16	1	159	0.00
600	1	64	0.16	1	77	0.16	1	79	0.00
1200	0	129	0.16	0	155	0.16	0	159	0.00
2400	0	64	0.16	0	77	0.16	0	79	0.00
4800	0	32	–1.36	0	38	0.16	0	39	0.00
9600	0	15	1.73	0	19	–2.34	0	19	0.00
19200	0	7	1.73	0	9	–2.34	0	9	0.00
31250	0	4	0.00	0	5	0.00	0	5	2.40
38400	0	3	1.73	0	4	–2.34	0	4	0.00

	Pϕ (MHz)								
	7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
19200	0	11	0.00	0	12	0.16	0	15	0.00
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

	Pϕ (MHz)											
	10			12			12.288			14.7456		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	0.03	2	217	0.08	3	64	0.70
150	2	129	0.16	2	155	0.16	2	159	0.00	2	191	0.00
300	2	64	0.16	2	77	0.16	2	79	0.00	2	95	0.00
600	1	129	0.16	1	155	0.16	1	159	0.00	1	191	0.00
1200	1	64	0.16	1	77	0.16	1	79	0.00	1	95	0.00
2400	0	129	0.16	0	155	0.16	0	159	0.00	0	191	0.00
4800	0	64	0.16	0	77	0.16	0	79	0.00	0	95	0.00
9600	0	32	–1.36	0	38	0.16	0	39	0.00	0	47	0.00
19200	0	15	1.73	0	19	0.16	0	19	0.00	0	23	0.00
31250	0	9	0.00	0	11	0.00	0	11	2.40	0	14	–1.70
38400	0	7	1.73	0	9	–2.34	0	9	0.00	0	11	0.00

	Pϕ (MHz)											
	16			19.6608			20			24		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	86	0.31	3	88	–0.25	3	106	–0.44
150	2	207	0.16	2	255	0.00	3	64	0.16	3	77	0.16
300	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
600	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
1200	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
2400	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
4800	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
9600	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
19200	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
31250	0	15	0.00	0	19	–1.70	0	19	0.00	0	23	0.00
38400	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34

	Pϕ (MHz)								
	24.576			28.7			30		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	108	0.08	3	126	0.31	3	132	0.13
150	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	0.00	0	22	1.55	0	23	1.73

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

Table 15.4 Bit Rates and SCBRR Settings in Synchronous Mode

Bit Rate (bits/s)	Pϕ (MHz)									
	5		8		16		28.7		30	
	n	N	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—	—	—
250	3	77	3	124	3	249	—	—	—	—
500	3	38	2	249	3	124	3	223	3	233
1k	2	77	2	124	2	249	3	111	3	116
2.5k	1	124	1	199	2	99	2	178	2	187
5k	0	249	1	99	1	199	2	89	2	93
10k	0	124	0	199	1	99	1	178	1	187
25k	0	49	0	79	0	159	1	71	1	74
50k	0	24	0	39	0	79	0	143	0	149
100k	—	—	0	19	0	39	0	71	0	74
250k	0	4	0	7	0	15	—	—	0	29
500k	—	—	0	3	0	7	—	—	0	14
1M	—	—	0	1	0	3	—	—	—	—
2M			0	0*	0	1	—	—	—	—

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

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

Table 15.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 15.6 and 15.7 list the maximum rates for external clock input.

Table 15.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
5	156250	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 15.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	1.2500	78125
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

Table 15.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	0.8333	833333.3
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0

15.3.9 FIFO Control Register (SCFCR)

SCFCR is a 16-bit register that resets the number of data in the transmit and receive FIFO registers, sets the trigger data number, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU. It is initialized to H'0000 by a power-on reset.

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 When the number of receive data in the receive FIFO register (SCFRDR) becomes more than the number shown below, the $\overline{\text{RTS}}$ signal is set to high. These bits are available only in SCFCR_0 and SCFCR_1. In SCFCR_2, these bits are reserved. The initial value is 0 and the write value should always be 0. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
9	RSTRG1	0	R/W	
8	RSTRG0	0	R/W	

Bit	Bit Name	Initial value	R/W	Description
7	RTRG1	0	R/W	Receive FIFO Data Trigger
6	RTRG0	0	R/W	Set the specified receive trigger number. The receive data full (RDF) flag in the serial status register (SCFSR) is set when the number of receive data stored in the receive FIFO register (SCFRDR) exceeds the specified trigger number shown below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Synchronous mode 00: 1 01: 2 10: 8 11: 14
5	TTRG1	0	R/W	Transmit FIFO Data Trigger 1 and 0
4	TTRG0	0	R/W	Set the specified transmit trigger number. The transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR) is set when the number of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the specified trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of remaining bytes in SCFTDR when the TDFE flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. In synchronous mode, clear this bit to 0. This bit is available only in SCFCR_0 and SCFCR_1. In SCFCR_2, this bit is reserved. The initial value is 0 and the write value should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * The $\overline{\text{CTS}}$ pin state has no effect on transmit operations regardless of the input value. The $\overline{\text{RTS}}$ pin state has no effect on receive operations, either.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TxD) and receive input pin (RxD) and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

15.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the number of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR). It indicates the number of transmit data in SCFTDR with the upper eight bits, and the number of receive data in SCFRDR with the lower eight bits. SCFDR can always be read from by the CPU. SCFDR is initialized to H'0000 by a power on reset.

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	Indicate the number of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and 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	Indicate the number of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

15.3.11 Serial Port Register (SCSPTR)

SCSPTR is a 16-bit register that controls input/output and data for the pins multiplexed to the SCIF function. Bits 7 and 6 can control the $\overline{\text{RTS}}$ pin, bits 5 and 4 can control the $\overline{\text{CTS}}$ pin, and bits 3 and 2 can control the SCK pin. Bits 1 and 0 can be used to read the input data from the RxD pin and to output data to the TxD pin, so they control break of serial transfer. In addition to descriptions of individual bits shown below, see section 15.6, Serial Port Register (SCSPTR) and SCIF Pins.

SCSPTR can always be read from or written to by the CPU. Bits 7, 5, 3, and 1 in SCSPTR are initialized by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO	0	R/W	RTS Port Input/Output Control Controls the $\overline{\text{RTS}}$ pin in combination with the RTSDT bit in this register and the MCE bit in SCFCR. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
6	RTSDT	—*	R/W	RTS Port Data Controls the $\overline{\text{RTS}}$ pin in combination with the RTSIO bit in this register and the MCE bit in SCFCR. Select the $\overline{\text{RTS}}$ pin function in the PFC (pin function controller) beforehand. MCE RTSIO RTSDT: $\overline{\text{RTS}}$ pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 1 × ×: Sequence output according to modem control logic ×: Don't care The $\overline{\text{RTS}}$ pin state is read from this bit instead of the set value. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.

Bit	Bit Name	Initial value	R/W	Description
5	CTSIO	0	R/W	CTS Port Input/Output Control Controls the $\overline{\text{CTS}}$ pin in combination with the CTSDT bit in this register and the MCE bit in SCFCR. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
4	CTSDT	—*	R/W	CTS Port Data Controls the $\overline{\text{CTS}}$ pin in combination with the CTSIO bit in this register and the MCE bit in SCFCR. Select the $\overline{\text{CTS}}$ pin function in the PFC (pin function controller) beforehand. MCE CTSIO CTSDT: $\overline{\text{CTS}}$ pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 1 × ×: Input to modem control logic ×: Don't care The $\overline{\text{CTS}}$ pin state is read from this bit instead of the set value. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
3	SCKIO	0	R/W	SCK Port Input/Output Control Controls the SCK pin in combination with the SCKDT bit in this register, the C/A bit in SCSMR, and bits CKE1 and CKE0 in SCSCR.

Bit	Bit Name	Initial value	R/W	Description
2	SCKDT	—*	R/W	SCK Port Data Controls the SCK pin in combination with the SCKIO bit in this register, the C/$\overline{A}$ bit in SCSMR, and bits CKE1 and CKE0 in SCSCR. Select the SCK pin function in the PFC (pin function controller) beforehand. C/$\overline{A}$ CKE1 CKE0 SCKIO SCKDT: SCK pin state 0 0 0 0 ×: Input (initial state) 0 0 0 0 0: Low level output 0 0 0 1 1: High level output 0 0 1 × ×: Internal clock output according to serial core logic 0 1 0 × ×: External clock input to serial core logic 0 1 1 × ×: Setting prohibited 1 0 0 × ×: Internal clock output according to serial core logic 1 0 1 × ×: Internal clock output according to serial core logic 1 1 0 × ×: External clock input to serial core logic 1 1 1 × ×: Setting prohibited ×: Don't care The SCK pin state is read from this bit instead of the set value.
1	SPBIO	0	R/W	Serial Port Break Input/Output Control Controls the TxD pin in combination with the SPBDT bit in this register and the TE bit in SCSCR.

Bit	Bit Name	Initial value	R/W	Description
0	SPBDT	—*	R/W	Serial Port Break Data Controls the TxD pin in combination with the SPBIO bit in this register and the TE bit in SCSCR. The RxD pin state can also be monitored. Select the TxD or RxD pin function in the PFC (pin function controller) beforehand. TE SPBIO SPBDT: TxD pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 0 × ×: Transmit data output according to serial core logic ×: Don't care The RxD pin state is read from this bit instead of the set value.

Note: * This bit is read as an undefined value and the setting value is 0.

15.3.12 Line Status Register (SCLSR)

SCLSR is a 16-bit readable/writable register which can always be read from and written to by the CPU. However, a 1 cannot be written to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1). SCLSR is initialized to H'0000 by a power-on reset.

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 the occurrence of an overrun error. 0: Receiving is in progress or has ended normally *1 [Clearing conditions] • ORER is cleared to 0 when the chip is a power-on reset • ORER is cleared to 0 when 0 is written after 1 is read from ORER. 1: An overrun error has occurred *2 [Setting condition] • ORER is set to 1 when the next serial receiving is finished while receive FIFO data are full. Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value. 2. The receive FIFO data register (SCFRDR) hold the data before an overrun error is occurred, and the next receive data is extinguished. When ORER is set to 1, SCIF can not continue the next serial receiving.

Note: * The only value that can be written is 0 to clear the flag.

15.4 Operation

15.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. The SCIF has a 16-byte FIFO buffer for both transmit and receive operations, reducing the overhead of the CPU, and enabling continuous high-speed communication. Moreover, it has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals as modem control signals (for channels 0 and 1). The transmission format is selected in the serial mode register (SCSMR), which is shown in table 15.8. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), which is shown in table 15.9.

Asynchronous Mode:

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode:

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCIF operates on the input serial clock. The on-chip baud rate generator is not used.

Table 15.8 SCSMR Settings and SCIF Communication Formats

SCSMR Settings					SCIF Communication 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	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	*	*	*	Synchronous	8 bits	Not set	None

Note: * : Don't care

Table 15.9 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR Settings			SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin. The state of the SCK pin depends on both the SCKIO and SCKDT bits.
		1			Clock with a frequency 16 times the bit rate is output.
	1	0		External	Input a clock with frequency 16 times the bit rate.
		1		—	Setting prohibited.
1	0	*	Synchronous	Internal	Serial clock is output.
	1	0		External	Input the serial clock.
		1		—	Setting prohibited.

Note: * : Don't care

15.4.2 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.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 15.2 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 character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

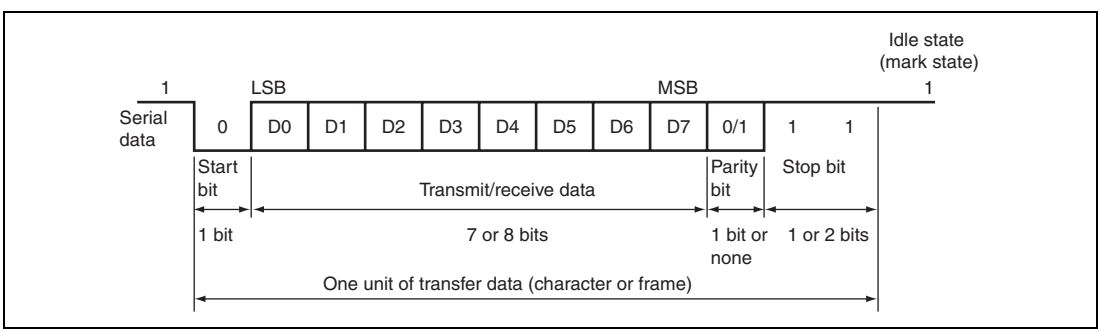


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

Transmit/Receive Formats: Table 15.10 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 15.10 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

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

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

Transmitting and Receiving Data: SCIF Initialization (Asynchronous Mode):

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 15.3 shows a sample flowchart for initializing the SCIF.

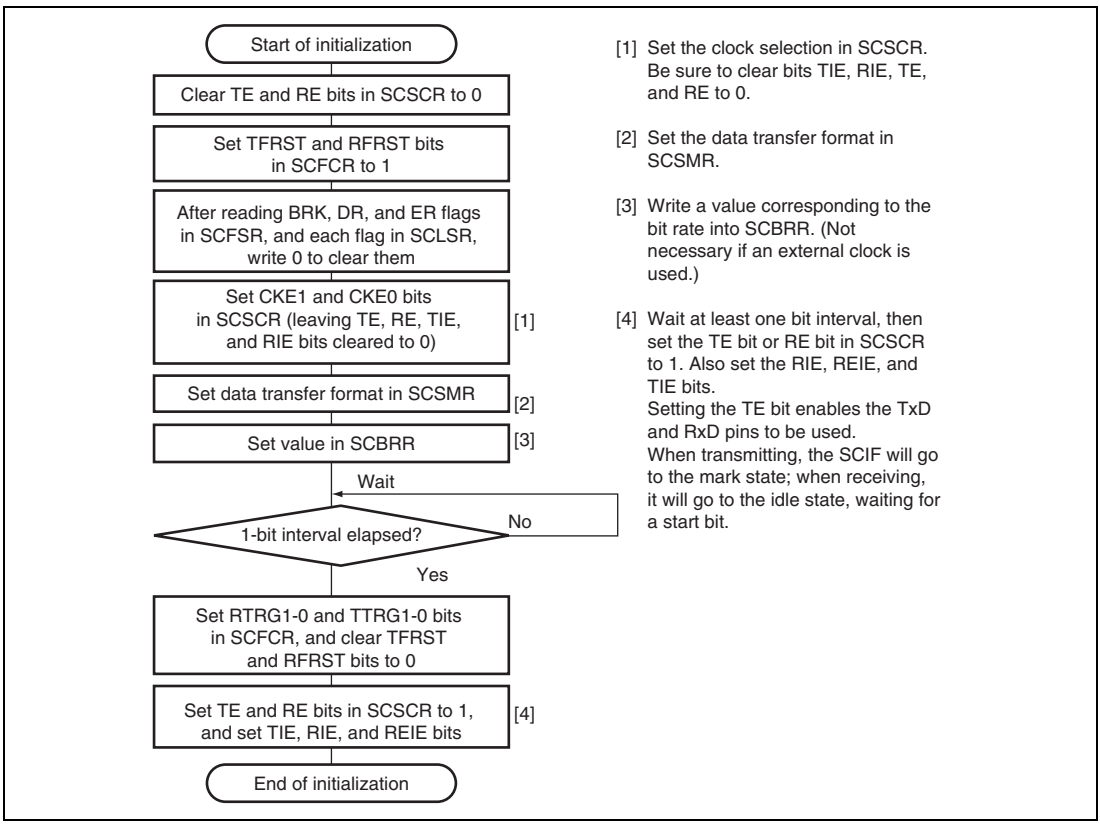
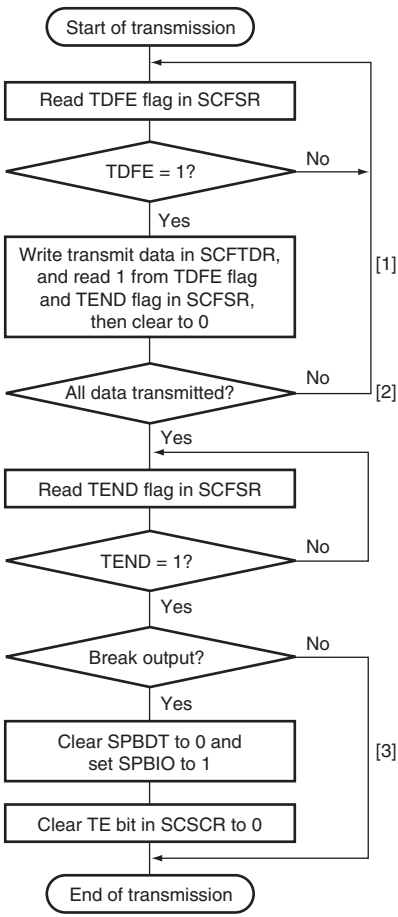


Figure 15.3 Sample Flowchart for SCIF Initialization

Transmitting Serial Data (Asynchronous Mode)

Figure 15.4 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 SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and read 1 from the TDFE and TEND flags, then clear to 0.

The number of transmit data bytes that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output at the end of serial transmission:

To output a break in serial transmission, clear the SPBDT bit to 0 and set the SPBIO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 15.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) 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-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a 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 continuously.

Figure 15.5 shows an example of the operation for transmission.

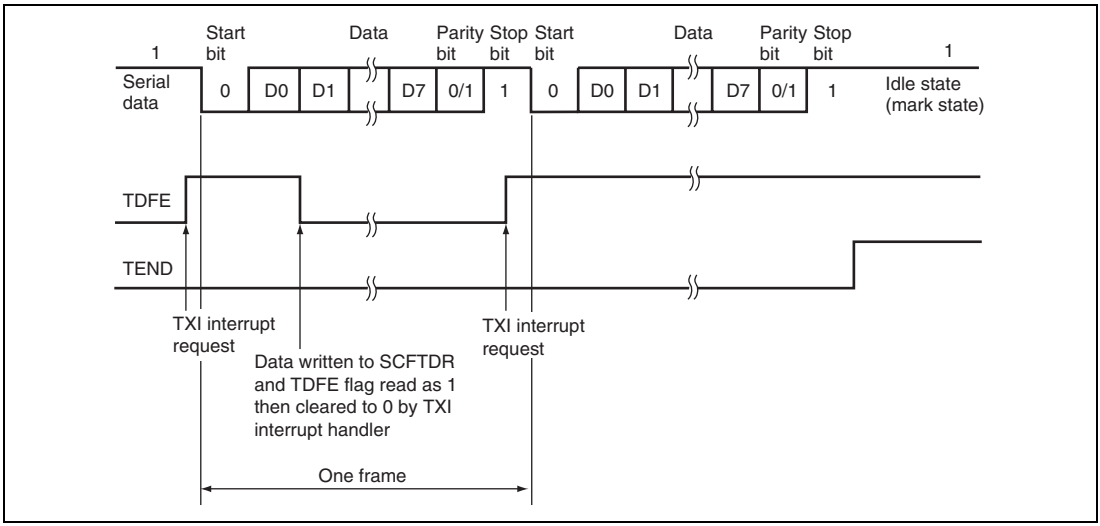


Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

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

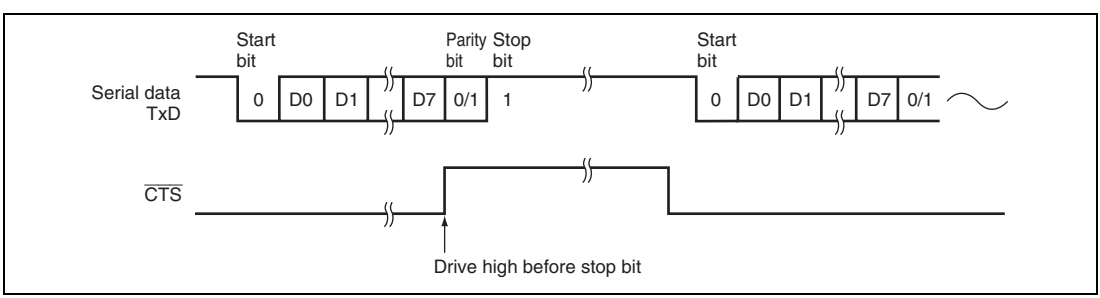
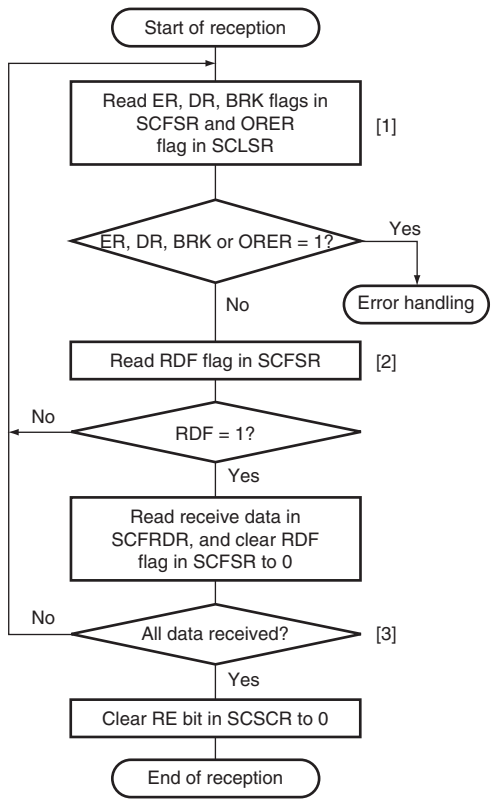


Figure 15.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

Receiving Serial Data (Asynchronous Mode):

Figures 15.7 and 15.8 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 the 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.

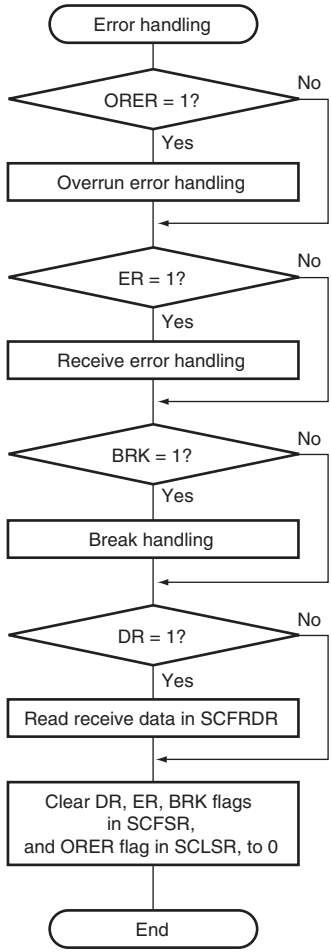
[2] SCIF status check and receive data read:

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 receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 15.7 Sample Flowchart for Receiving Serial Data (1)



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 15.8 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIF carries out the following checks.
   A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.
   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: When a parity error or a framing error occurs, reception is not suspended.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the 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 the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 15.9 shows an example of the operation for reception.

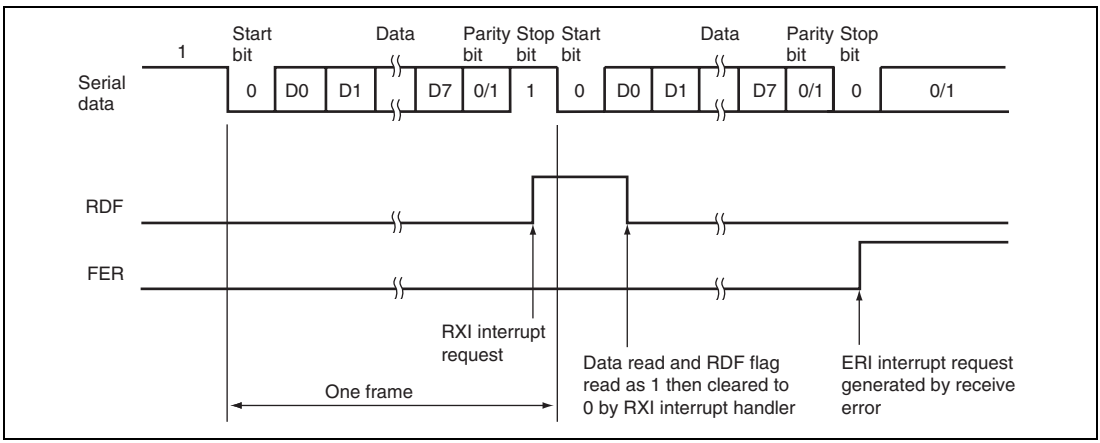


Figure 15.9 Example of SCIF Receive Operation (8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output depending on the empty status of SCFRDR. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that the SCFRDR is full and no extra data can be received. (Only for channel 0 and channel 1)

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

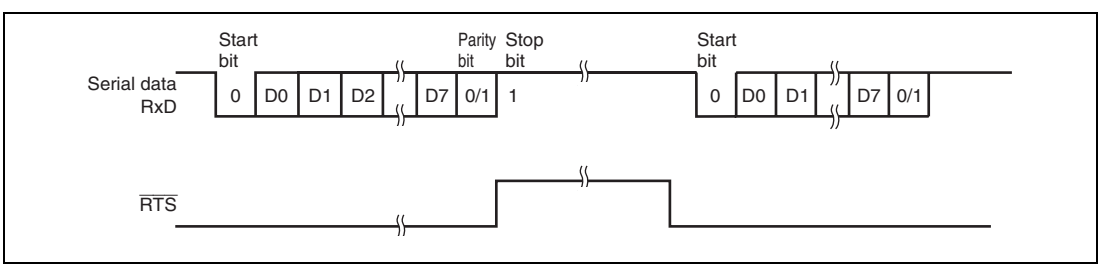


Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

15.4.3 Synchronous Mode

In synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 15.11 shows the general format in synchronous serial communication.

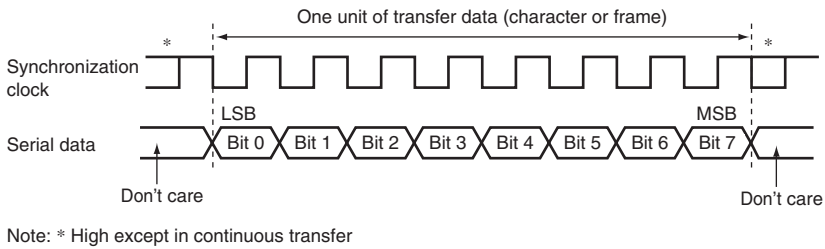


Figure 15.11 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode, the SCIF transmits data by synchronizing with the falling edge of the serial clock, and receives data by synchronizing with the rising edge of the serial clock.

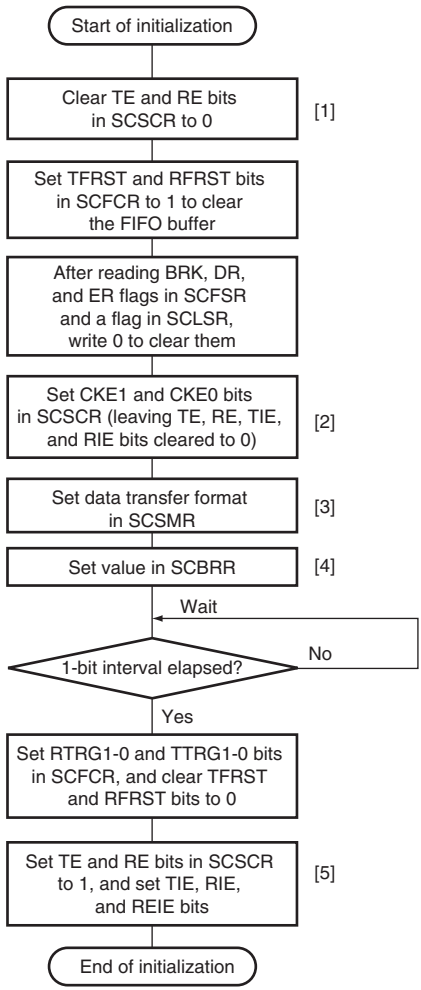
Communication Format: The data length is fixed at eight bits. No parity bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

When the SCIF operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is less than the receive FIFO data trigger number. In this case, $8 \times (16 + 1) = 136$ pulses of synchronous clock are output. To perform reception of n characters of data, select an external clock as the clock source. If an internal clock should be used, set RE = 1 and TE = 1 and receive n characters of data simultaneously with the transmission of n characters of dummy data.

Transmitting and Receiving Data SCIF Initialization (Synchronous Mode): Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 15.12 shows a sample flowchart for initializing the SCIF.



[1] Leave the TE and RE bits cleared to 0 until the initialization almost ends. Be sure to clear the TIE, RIE, TE, and RE bits to 0.

[2] Set the CKE1 and CKE0 bits.

[3] Set the data transfer format in SCSMR.

[4] Write a value corresponding to the bit rate into SCBRR. This is not necessary if an external clock is used. Wait at least one bit interval after this write before moving to the next step.

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

Figure 15.12 Sample Flowchart for SCIF Initialization

Transmitting Serial Data (Synchronous Mode): Figure 15.13 shows a sample flowchart for transmitting serial data.

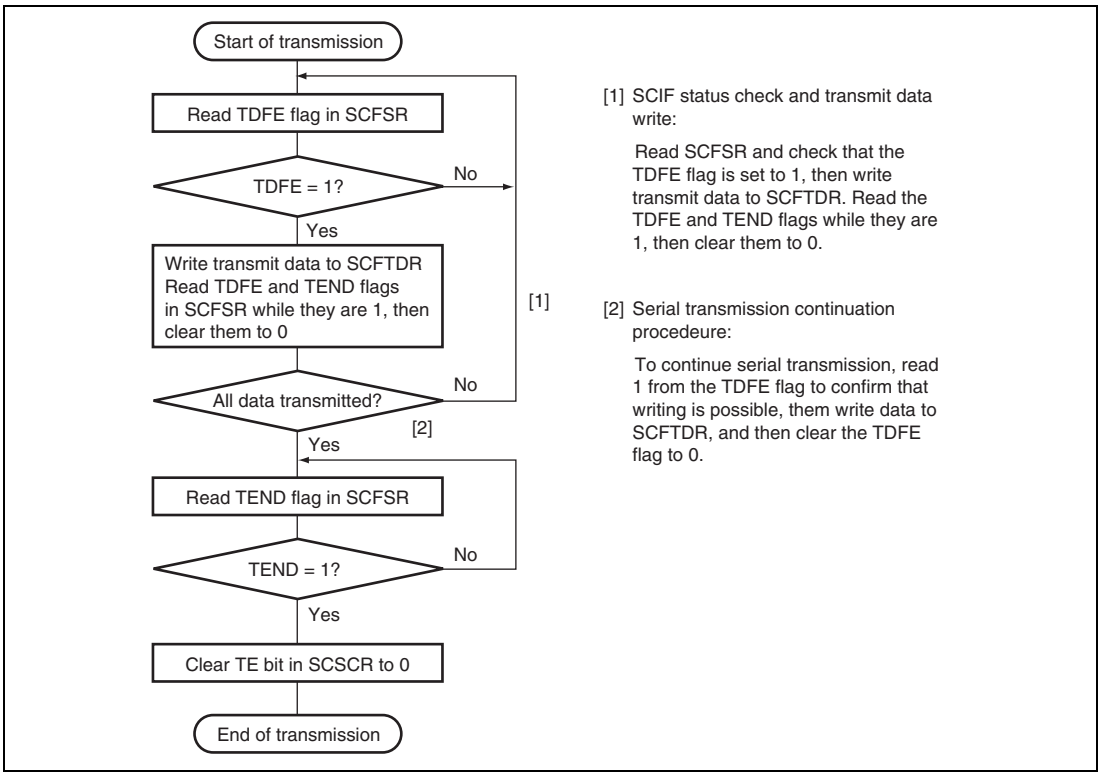


Figure 15.13 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCIF operates as follows:

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, the MSB (bit 7) 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 MSB (bit 7) is sent, and then the TxD pin holds the states.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 15.14 shows an example of SCIF transmit operation.

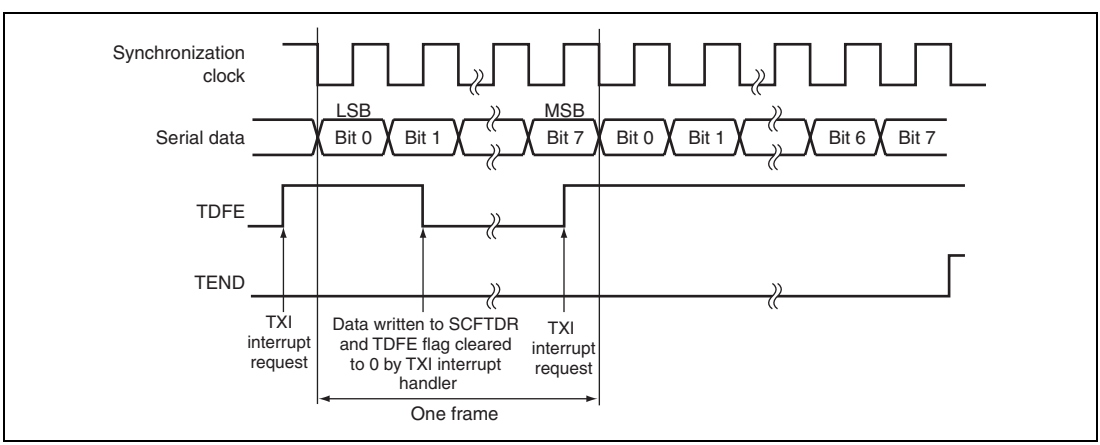
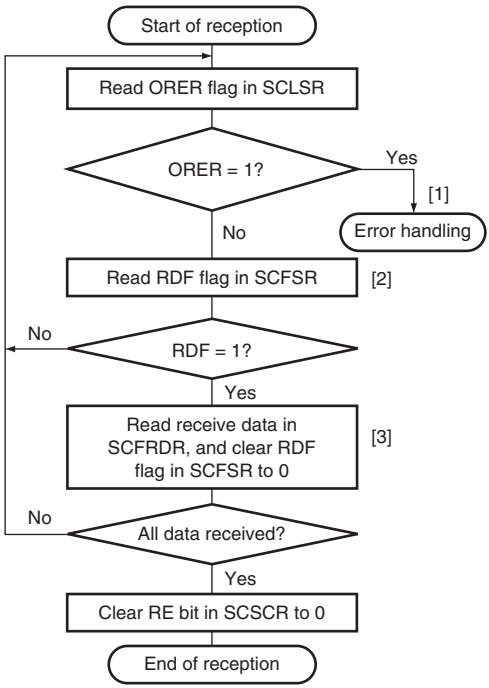


Figure 15.14 Example of SCIF Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 15.15 and 15.16 show a sample flowchart for receiving serial data. When switching from asynchronous mode to synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.



[1] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Transmission/reception cannot be resumed while the ORER flag is set to 1.

[2] SCIF status check and receive data read:

Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading SCFRDR.

Figure 15.15 Sample Flowchart for Receiving Serial Data (1)

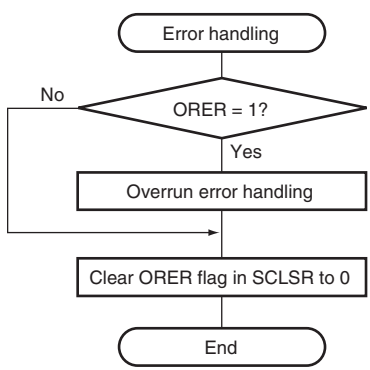


Figure 15.16 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCIF operates as follows:

1. The SCIF synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the SCIF stores the received data in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE bit or REIE bit in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 15.17 shows an example of SCIF receive operation.

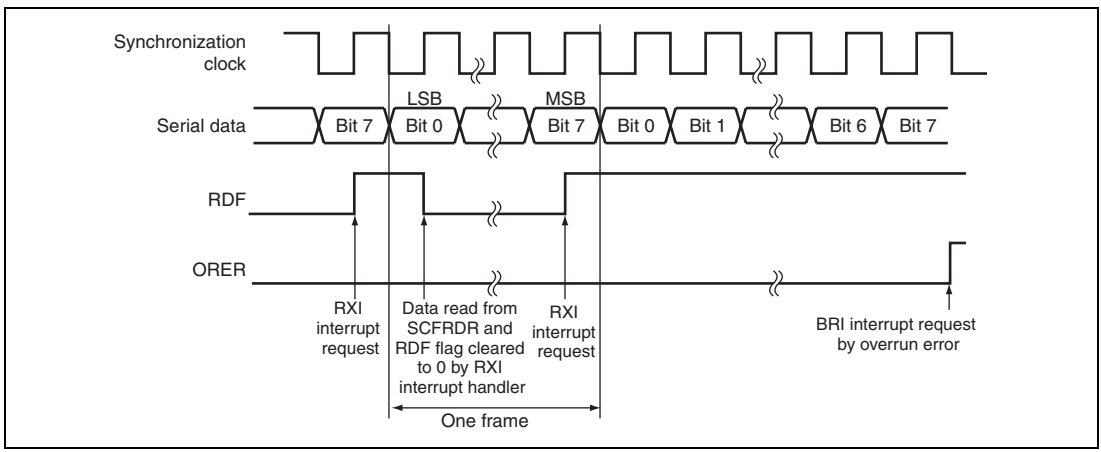
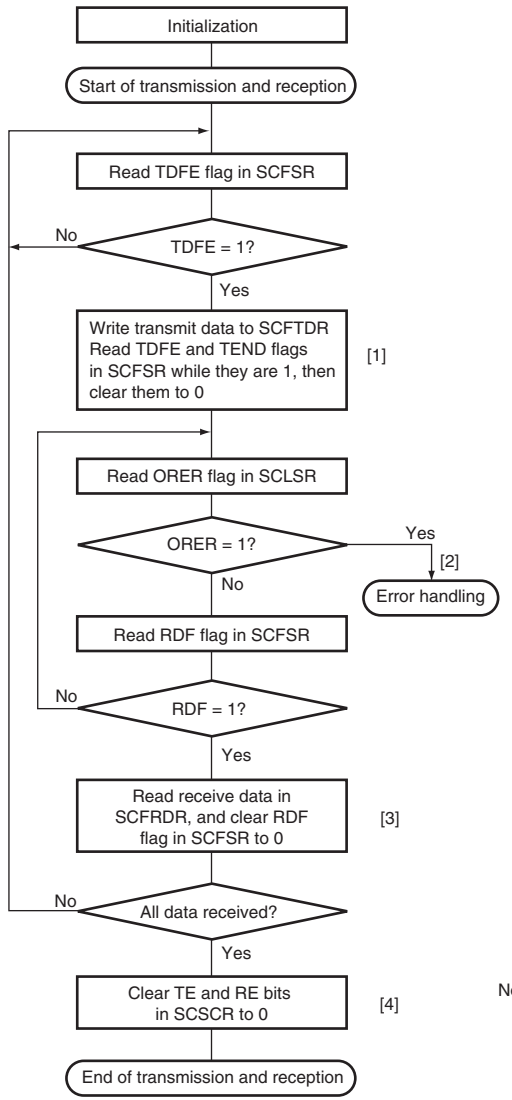


Figure 15.17 Example of SCIF Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 15.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.



[1] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR. Read the TDFE and TEND flags while they are 1, then clear them to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:

Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 15.18 Sample Flowchart for Transmitting/Receiving Serial Data

15.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 15.11 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When TXI request is enabled by TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated.

When RXI request is enabled by RIE bit and the RDF or DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The RXI interrupt request caused by DR flag is generated only in asynchronous mode.

When BRI request is enabled by RIE bit or REIE bit and the BRK flag in SCFSR or ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

When ERI request is enabled by RIE bit or REIE bit and the ER flag in SCFCR is set to 1, an ERI interrupt request is generated.

When the RIE bit is set to 0 and the REIE bit is set to 1, SCIF request ERI interrupt and BRI interrupt without requesting RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 15.11 SCIF Interrupt Sources

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

15.6 Serial Port Register (SCSPTR) and SCIF Pins

The relationship between SCSPTR and the SCIF pins is shown in figures 15.19 to 15.23.

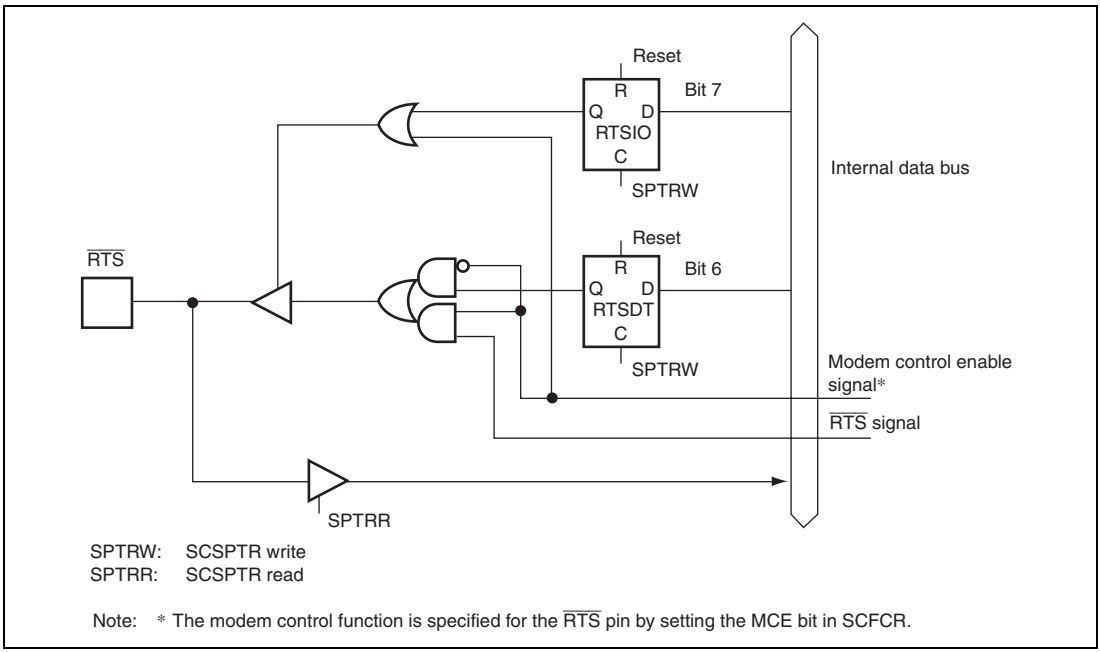


Figure 15.19 RTSIO Bit, RTSDT Bit, and $\overline{\text{RTS}}$ Pin

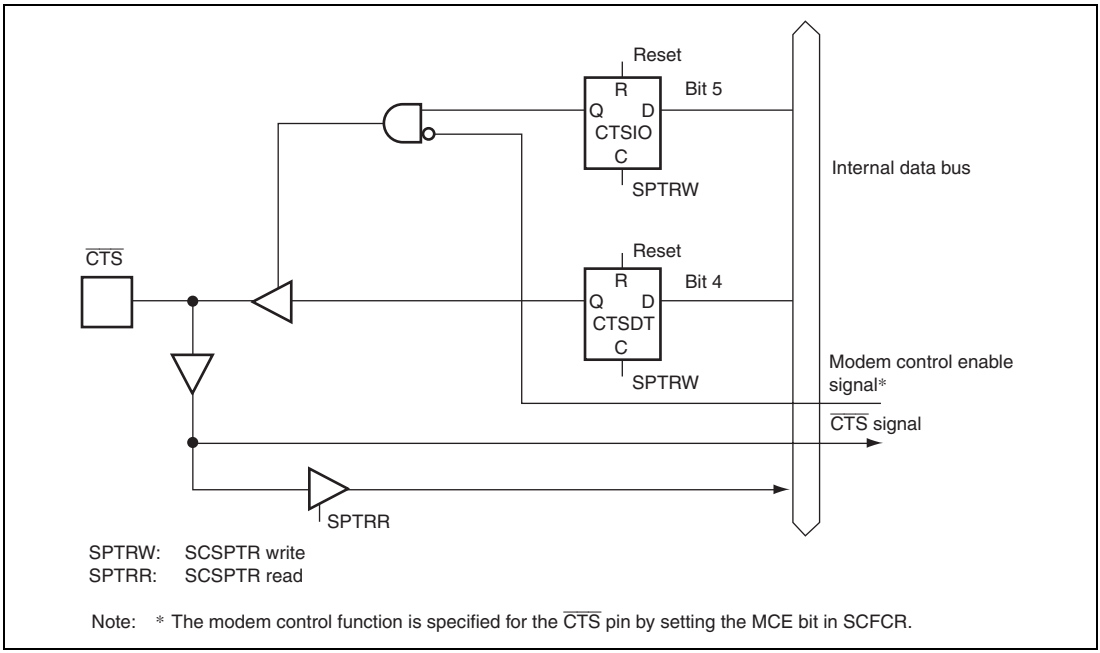


Figure 15.20 CTSIO Bit, CTSDT Bit, and $\overline{\text{CTS}}$ Pin

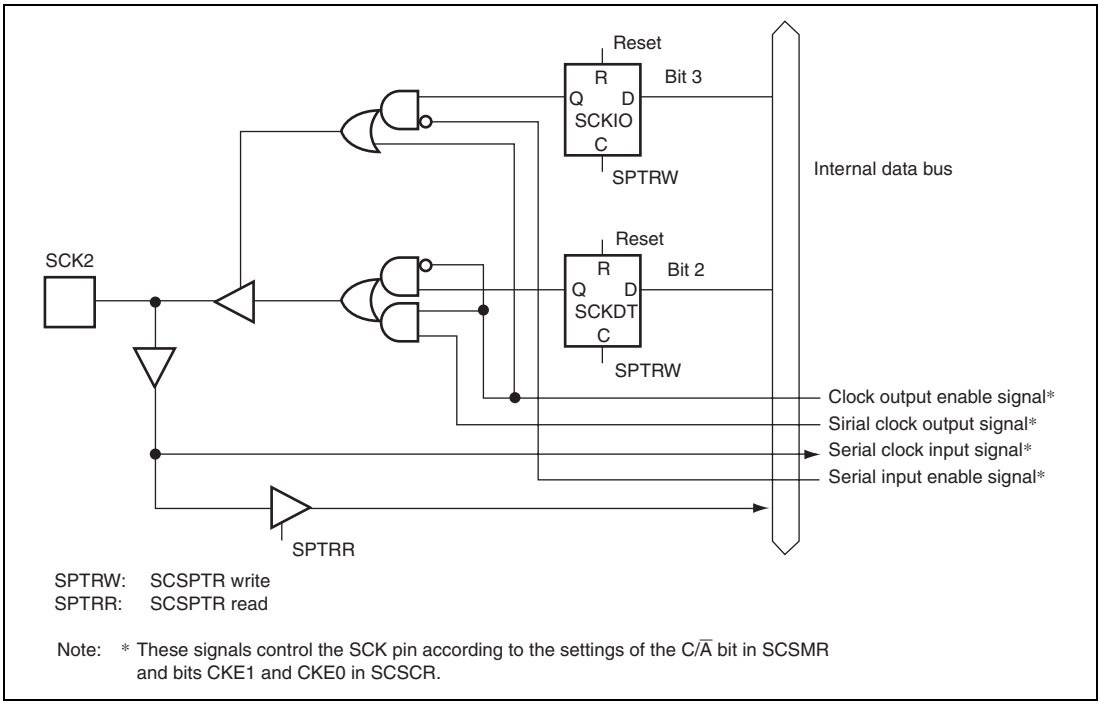


Figure 15.21 SCKIO Bit, SCKDT Bit, and SCK Pin

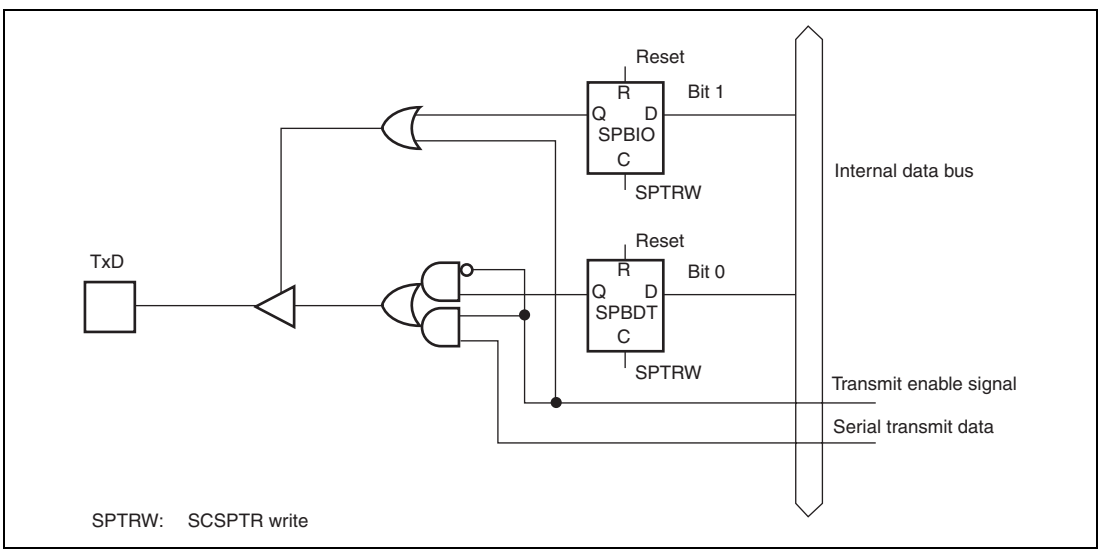


Figure 15.22 SPBIO Bit, SPBDT Bit, and TxD Pin

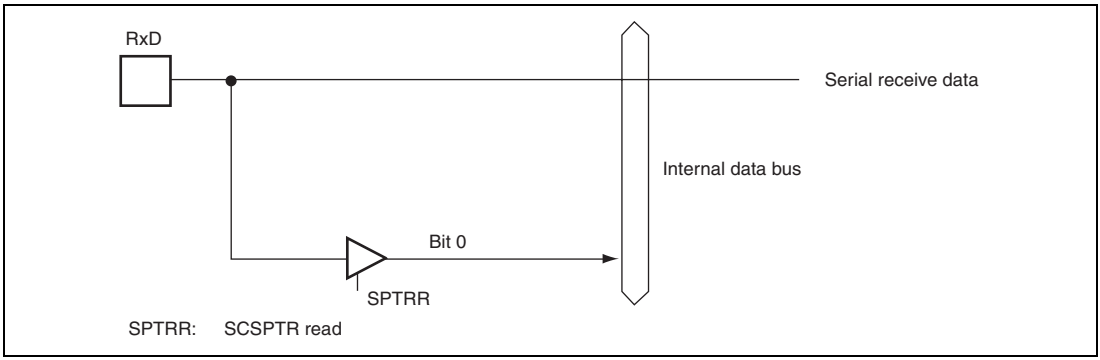


Figure 15.23 SPBDT Bit and RxD Pin

15.7 Usage Notes

Note the following when using the SCIF.

1. SCFTDR Writing and TDFE Flag

   The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.
   However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again 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 the upper 8 bits of the FIFO data count register (SCFDR).

2. SCFRDR Reading and RDF Flag

   The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.
   However, if the number of data bytes in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after all the receive data has been read.
   The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

3. 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. Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

4. Sending a Break Signal

   The I/O condition and level of the TxD pin are determined by the SPBIO and SPBDT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

   Until TE bit is set to 1 (enabling transmission) after initializing, TxD pin does not work. During the period, mark status is performed by SPBDT bit. Therefore, the SPBIO and SPBDT bits should be set to 1 (high level output).

   To send a break signal during serial transmission, clear the SPBDT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD pin.

5. Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

   The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In 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 15.24.

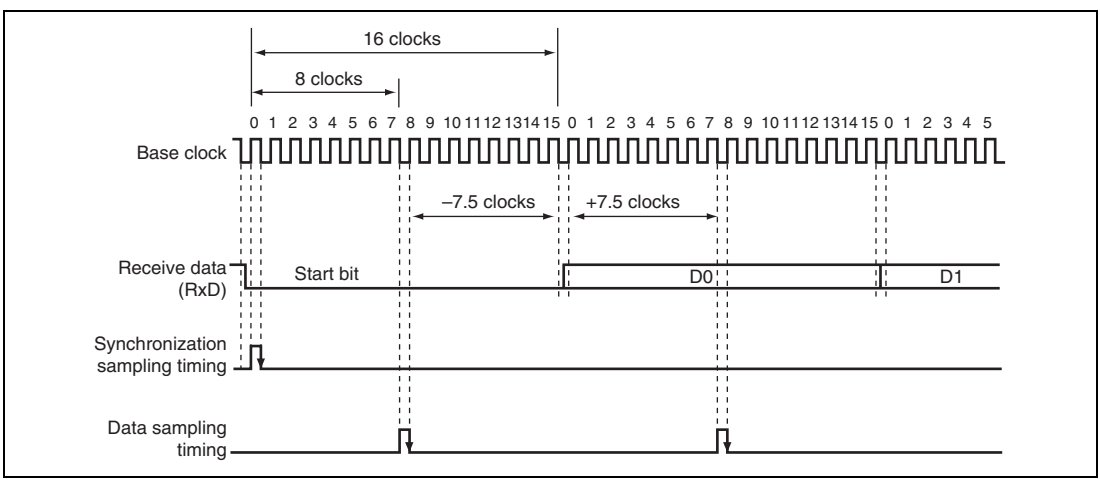


Figure 15.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) = (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of 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.

Equation 2:

When D = 0.5 and F = 0:

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

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

6. Prohibited Multiple Pin Allocation for Channel 1

   Although signals SCK1, RxD1, and TxD1 can be respectively assigned to multiple pins of PD4 or PE20, PD3 or PE19, and PD2 or PE18, either of them must be selected. For example, if signal SCK1 is assigned to both pins PD4 and PE20, correct operation of the SCIF is not guaranteed.

7. Status of the TxD and RTS Pins When the TE Bit is Cleared

   The TxDi (i = 0, 1, 2) and RTSj (j = 0, 1) pins usually function as output pins during serial communication. However, even if these functions are selected by the pin function controller (PFC), the internal weak keeper drives the pins to unstable levels as long as the TE bit in SCSCRi (i = 0, 1, 2) is cleared. To make these pins always function as output pins (regardless of the value of the TE bit), set SCSPTRi (i = 0, 1, 2) and PFC in the following order.

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

8. Interval from when the TE bit in SCSCR is Set to 1 until a Start Bit is Transmitted in Asynchronous Mode

   In the SCIF included in former products, a start bit is transmitted after the internal equivalent to one frame. In the SCIF included in this product, however, a start bit is transmitted directly after the TE bit is set to 1.

9. Clear Timing of the FER or PER Bits when the DMAC Saves the Receive Data in Asynchronous Mode

   The FER or PER bits in SCFSR are set when data including a framing error or parity error is received in asynchronous mode, whereas cleared when the corresponding data is read from SCFRDR. Therefore, when data with an error is received while the DMAC is set to save the receive data automatically, the receive-error interrupt is accepted after the DMAC reads the corresponding data. As a result, the CPU cannot check the FER or PER bits.

   To prevent this defect, the RTRG[1:0] bits in SCFCR should be set to the higher number to delay the DMAC call timing. This enables the CPU to check the FER or PER bits in the receive-error interrupt routine, prior to the DMAC to read the error data.

Section 16 Serial I/O with FIFO (SIOF)

This LSI includes a clock-synchronized serial I/O module with FIFO (SIOF) that comprises one channel. The SIOF can perform serial communication with a serial peripheral interface bus (SPI).

16.1 Features

- Serial transfer
  - 16-stage 32-bit FIFOs (independent transmission and reception)
  - Supports 8-bit data/16-bit data/16-bit stereo audio input and output
  - MSB first for data transmission
  - Supports a maximum of 48-kHz sampling rate
  - Synchronization by either frame synchronization pulse or left/right channel switch
  - Supports CODEC control data interface
  - Connectable to linear, audio, or A-Law or μ-Law CODEC chip
  - Supports both master and slave modes
- Serial clock
  - An external pin input or internal clock (Pϕ) can be selected as the clock source.
- Interrupts: One type
- DMA transfer
  - Supports DMA transmission and reception by a transfer request for transmission and reception
- SPI mode
  - Fixed master mode can perform the full-duplex communication with the SPI slave devices continuously.
  - Selects the falling/rising edge of the SCK as data sampling.
  - Selects the clock phase of the SCK as a transmit timing.
  - Selects one slave device.
  - The length of transmit/receive data is fixed to 8 bits.

Figure 16.1 shows a block diagram of the SIOF.

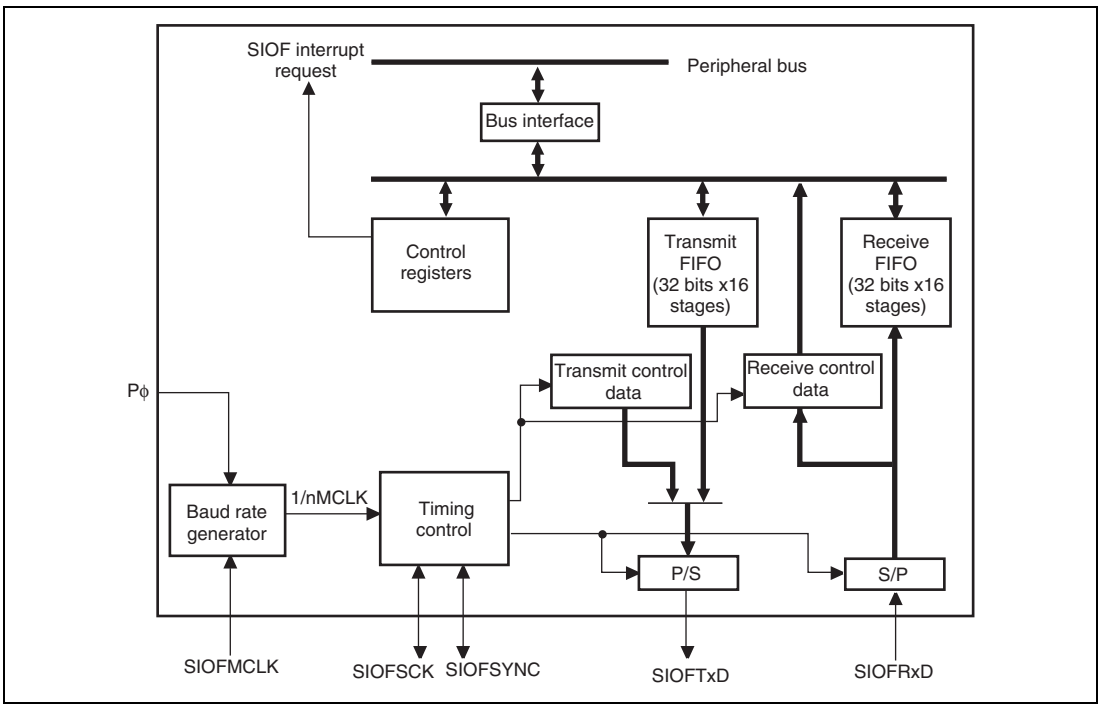


Figure 16.1 Block Diagram of SIOF

16.2 Input/Output Pins

The pin configuration in this module is shown in table 16.1.

Table 16.1 Pin Configuration

Channel	Pin Name	Abbreviation*	I/O	Function
0	SIOF0_MCLK	SIOFMCLK	Input	Master clock input
	SIOF0_SCK (SCK0)	SIOFSCK (SCK)	I/O	Serial clock (common to transmission/reception) In SPI mode, fixed to output.
	SIOF0_SYNC ($\overline{\text{SS00}}$)	SIOFSYNC ($\overline{\text{SS0}}$)	I/O	Frame synchronous signal (common to transmission/reception) In SPI mode, fixed to output, and selects slave device 0.
	SIOF0_TxD (MOSI0)	SIOFTxD (MOSI)	Output	Transmit data
	SIOF0_RxD (MISO0)	SIOFRxD (MISO)	Input	Receive data

Note: * The pins are abbreviated as SIOFMCLK, SIOFSCK, SIOFSYNC, SIOFTxD, and SIOFRxD in the following descriptions. In SPI mode, the pins are called SCK, $\overline{\text{SS0}}$, MOSI, and MISO.

16.3 Register Descriptions

The SIOF has the following registers. For the addresses of these registers and the register states in each operating state, refer to section 24, List of Registers. In the register descriptions following this section, channel numbers are omitted.

Channel 0:

- Mode register_0 (SIMDR_0)
- Control register_0 (SICTR_0)
- Transmit data register_0 (SITDR_0)
- Receive data register_0 (SIRDR_0)
- Transmit control data register_0 (SITCR_0)
- Receive control data register_0 (SIRCR_0)
- Status register_0 (SISTR_0)
- Interrupt enable register_0 (SIIER_0)
- FIFO control register_0 (SIFCTR_0)
- Clock select register_0 (SISCR_0)
- Transmit data assign register_0 (SITDAR_0)
- Receive data assign register_0 (SIRDAR_0)
- Control data assign register_0 (SICDAR_0)
- SPI control register_0 (SPICR_0)

16.3.1 Mode Register (SIMDR)

SIMDR is a 16-bit readable/writable register that sets the SIOF operating mode.

Bit	Bit Name	Initial Value	R/W	Description
15	TRMD1	1	R/W	Transfer Mode 1, 0
14	TRMD0	0	R/W	Select transfer mode. For details, see table 16.2. 00: Slave mode 1 01: Slave mode 2 10: Master mode 1 11: Master mode 2
13	SYNCAT	0	R/W	SIOFSYNC Pin Valid Timing Indicates the position of the SIOFSYNC signal to be output as a synchronization pulse. 0: At the start-bit data of frame 1: At the last-bit data of slot
12	REDG	0	R/W	Receive Data Sampling Edge 0: The SIOFRxD signal is sampled at the falling edge of SIOFSCK (The SIOFTxD signal is transmitted at the rising edge of SIOFSCK.) 1: The SIOFRxD signal is sampled at the rising edge of SIOFSCK (The SIOFTxD signal is transmitted at the falling edge of SIOFSCK.) Note: This bit is valid only in master mode.

Bit	Bit Name	Initial Value	R/W	Description
11	FL3	0	R/W	Frame Length 3 to 0
10	FL2	0	R/W	00xx: Data length is 8 bits and frame length is 8 bits.
9	FL1	0	R/W	0100: Data length is 8 bits and frame length is 16 bits.
8	FL0	0	R/W	0101: Data length is 8 bits and frame length is 32 bits.
				0110: Data length is 8 bits and frame length is 64 bits.
				0111: Data length is 8 bits and frame length is 128 bits.
				10xx: Data length is 16 bits and frame length is 16 bits.
				1100: Data length is 16 bits and frame length is 32 bits.
				1101: Data length is 16 bits and frame length is 64 bits.
				1110: Data length is 16 bits and frame length is 128 bits.
				1111: Data length is 16 bits and frame length is 256 bits.
				Note: When data length is specified as 8 bits, control data cannot be transmitted or received.
				x: Don't care
7	TXDIZ	0	R/W	SIOFTxD Pin Output when Transmission is Invalid*
				0: High output (1 output) when invalid
				1: High-impedance state when invalid
				Note: Invalid means when disabled, and when a slot that is not assigned as transmit data or control data is being transmitted.
6	RCIM	0	R/W	Receive Control Data Interrupt Mode
				0: Sets the RCRDY bit in SISTR when the contents of SIRCR change.
				1: Sets the RCRDY bit in SISTR each time when the SIRCR receives the control data.
5	—	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
4	SYNCDL	0	R/W	Data Pin Bit Delay for SIOFSYNC Pin Valid when the SIOFSYNC signal is output as synchronous pulse. Only one-bit delay is valid for transmission in slave mode. 0: No bit delay 1: 1-bit delay
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 16.2 Operation in Each Transfer Mode

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

Note: * The control data method is valid only when the FL3 to FL0 bits are specified as 1xxx. (x: Don't care.)

16.3.2 Control Register (SICTR)

SICTR is a 16-bit readable/writable register that sets the SIOF operating state.

Bit	Bit Name	Initial Value	R/W	Description
15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 0: Disables the SIOFSCK output (outputs 0) 1: Enables the SIOFSCK output • If this bit is set to 1, the SIOF initializes the baud rate generator and initiates the operation. At the same time, the SIOF outputs the clock generated by the baud rate generator to the SIOFSCK pin. This bit is initialized in module stop mode.
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output • If this bit is set to 1, the SIOF initializes the frame counter and initiates the operation. This bit is initialized in module stop mode.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	TXE	0	R/W	Transmit Enable 0: Disables data transmission from the SIOFTxD pin 1: Enables data transmission from the SIOFTxD pin • 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 issues a transmit transfer request according to the setting of the TFWM bit in SIFCTR. When transmit data is stored in the transmit FIFO, transmission of data from the SIOFTxD pin begins. • This bit is initialized upon a transmit reset. This bit is initialized in module stop mode.
8	RXE	0	R/W	Receive Enable 0: Disables data reception from SIOFRxD 1: Enables data reception from SIOFRxD • This bit setting becomes valid at the start 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 SIOFRxD pin. When receive data is stored in the receive FIFO, the SIOF issues a reception transfer request according to the setting of the RFWM bit in SIFCTR. • This bit is initialized upon receive reset. This bit is initialized in module stop mode.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	TXRST	0	R/W	Transmit Reset 0: Does not reset transmit operation 1: Resets transmit operation • This bit setting becomes valid immediately. This bit should be cleared to 0 before setting the register to be initialized. • When the 1 setting for this bit becomes valid, the SIOF immediately sets transmit data from the SIOFTxD pin to 1, and initializes the transmit data register and transmit-related status. The following are initialized. — SITDR — SITCR — Transmit FIFO write pointer and read pointer — TCRDY, TFEMP, and TDREQ bits in SISTR — TXE bit
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. This bit should be cleared to 0 before setting the register to be initialized. • When the 1 setting for this bit becomes valid, the SIOF immediately disables reception from the SIOFRxD pin, and initializes the receive data register and receive-related status. The following are initialized. — SIRDR — SIRCR — Receive FIFO write pointer and read pointer — RCRDY, RFFUL, and RDREQ bits in SISTR — RXE bit

16.3.3 Transmit Data Register (SITDR)

SITDR is a 32-bit write-only register that specifies the SIOF transmit data.

SITDR is initialized by the conditions specified in section 24, List of Registers, or by a transmit reset caused by the TXRST bit in SICTR.

SITDR is initialized in module stop mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL 15 to 0	All 0	W	Left-Channel Transmit Data Specify data to be output from the SIOFTxD pin as left-channel data. The position of the left-channel data in the transmit frame is specified by the TDLA bit in SITDAR. • These bits are valid only when the TDLE bit in SITDAR is set to 1.
15 to 0	SITDR 15 to 0	All 0	W	Right-Channel Transmit Data Specify data to be output from the SIOFTxD pin as right-channel data. The position of the right-channel data in the transmit frame is specified by the TDRA bit in SITDAR. • These bits are valid only when the TDRE bit and TLREP bit in SITDAR are set to 1 and cleared to 0, respectively.

16.3.4 Receive Data Register (SIRDR)

SIRDR is a 32-bit read-only register that reads receive data of the SIOF. SIRDR stores data in the receive FIFO and is initialized to undefined value by the conditions specified in section 24, List of Registers, or by a receive reset caused by the RXRST bit in SICTR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL 15 to 0	Undefined	R	Left-Channel Receive Data Store data received from the SIOFRxD pin as left-channel data. The position of the left-channel data in the receive frame is specified by the RDLA bit in SIRDAR. • These bits are valid only when the RDLE bit in SIRDAR is set to 1.
15 to 0	SIRDR 15 to 0	Undefined	R	Right-Channel Receive Data Store data received from the SIOFRxD pin as right-channel data. The position of the right-channel data in the receive frame is specified by the RDRA bit in SIRDAR. • These bits are valid only when the RDRE bit in SIRDAR is set to 1.

16.3.5 Transmit Control Data Register (SITCR)

SITCR is a 32-bit readable/writable register that specifies transmit control data of the SIOF. SITCR can be specified only when the FL3 to FL0 bits in SIMDR are specified as 1xxx (x: Don't care.).

SITCR is initialized in module stop mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC0 15 to 0	All 0	R/W	Control Channel 0 Transmit Data Specify data to be output from the SIOFTxD pin as control channel 0 transmit data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SITC1 15 to 0	All 0	R/W	Control Channel 1 Transmit Data Specify data to be output from the SIOFTxD pin as control channel 1 transmit data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

16.3.6 Receive Control Data Register (SIRCR)

SIRCR is a 32-bit readable/writable register that stores receive control data of the SIOF. SIRCR can be specified only when the FL3 to FL0 bits in SIMDR are specified as 1xxx (x: Don't care.).

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRC0 15 to 0	All 0	R	Control Channel 0 Receive Data Store data received from the SIOFRxD pin as control channel 0 receive data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SIRC1 15 to 0	All 0	R	Control Channel 1 Receive Data Store data received from the SIOFRxD pin as control channel 1 receive data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

16.3.7 Status Register (SISTR)

SISTR is a 16-bit read-only register that shows the SIOF state. Each bit in this register becomes an SIOF interrupt source when the corresponding bit in SIIER is set to 1.

SISTR is initialized in module stop mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared to 0, SITCR is over-written and the previous contents of SITCR are not output from the SIOFTxD pin. • This bit is valid when the TXE bit in SITCR is set to 1. • This bit indicates a state of the SIOF. If SITCR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if SITDR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
12	TDREQ	0	R	Transmit Data Transfer Request 0: Indicates that the size of empty space in the transmit FIFO does not exceed the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. When using transmit data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if the size of empty space in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	RCRDY	0	R	Receive Control Data Ready 0: Indicates that the SIRCR stores no valid data. 1: Indicates that the SIRCR stores valid data. • If SIRCR is written when this bit is set to 1, SIRCR is modified by the latest data. • This bit is valid when the RXE bit in SICTR is set to 1. • This bit indicates a state of the SIOF. If SIRCR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
9	RFFUL	0	R	Receive FIFO Full 0: Receive FIFO not full 1: Receive FIFO full • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if SIRDR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid space in the receive FIFO does not exceed the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. When using receive data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if the size of valid space in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SAERR	0	R/W	Slot Assign Error 0: Indicates that no slot assign error occurs 1: Indicates that a slot assign error occurs A slot assign error occurs when the specifications in SITDAR, SIRDAR, and SICDAR overlap. If a slot assign error occurs, the SIOF does not transmit data to the SIOFTxD pin and does not receive data from the SIOFRxD pin. Note that the SIOF does not clear the TXE bit or RXE bit in SICTR at a slot assign error. • This bit is valid when the TXE bit or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs A frame synchronization error occurs when the next frame synchronization timing appears before the previous data or control data transfers have been completed. If a frame synchronization error occurs, the SIOF performs transmission or reception for slots that can be transferred. • This bit is valid when the TXE or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
3	TFOVF	0	R/W	Transmit FIFO Overflow 0: No transmit FIFO overflow 1: Transmit FIFO overflow A transmit FIFO overflow means that there has been an attempt to write to SITDR when the transmit FIFO is full. When a transmit FIFO overflow occurs, the SIOF indicates overflow, and writing is invalid. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit FIFO underflow occurs, the SIOF repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
1	RFUDF	0	R/W	Receive FIFO Underflow 0: No receive FIFO underflow 1: Receive FIFO underflow A receive FIFO underflow means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive FIFO underflow occurs, the value of data read from SIRDR is not guaranteed. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF indicates overflow, and receive data is lost. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

16.3.8 Interrupt Enable Register (SIIER)

SIIER is a 16-bit readable/writable register that enables the issue of SIOF interrupts. When each bit in this register is set to 1 and the corresponding bit in SISTR is set to 1, the SIOF issues an interrupt.

Bit	Bit Name	Initial Value	R/W	Description
15	TDMAE	0	R/W	Transmit Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The TDREQE bit can be set as transmit interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC
14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control data ready 1: Enables interrupts due to transmit control data ready
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO empty 1: Enables interrupts due to transmit FIFO empty
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data transfer requests 1: Enables interrupts due to transmit data transfer requests
11	RDMAE	0	R/W	Receive Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The RDREQE bit can be set as receive interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC
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

Bit	Bit Name	Initial Value	R/W	Description
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive Data Transfer Request Enable 0: Disables interrupts due to receive data transfer requests 1: Enables interrupts due to receive data transfer requests
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SAERRE	0	R/W	Slot Assign Error Enable 0: Disables interrupts due to slot assign error 1: Enables interrupts due to slot assign error
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

16.3.9 FIFO Control Register (SIFCTR)

SIFCTR is a 16-bit readable/writable register that indicates the area available for the transmit/receive FIFO transfer.

Bit	Bit Name	Initial Value	R/W	Description
15	TFWM2	0	R/W	Transmit FIFO Watermark
14	TFWM1	0	R/W	000: Issue a transfer request when 16 stages of the transmit FIFO are empty.
13	TFWM0	0	R/W	001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or more stages of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages of the transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages of the transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages of transmit FIFO are empty. • A transfer request to the transmit FIFO is issued by the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
12	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 (full) to B'10000 (empty).
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	000: Issue a transfer request when 1 stage or more of the receive FIFO are valid.
5	RFWM0	0	R/W	001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 4 or more stages of the receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of the receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of the receive FIFO are valid. 111: Issue a transfer request when 16 stages of the receive FIFO are valid. • A transfer request to the receive FIFO is issued by the RDREQE bit in SISTR. • The receive FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
4	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 (empty) to B'10000 (full).
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

16.3.10 Clock Select Register (SISCR)

SISCR is a 16-bit readable/writable register that sets the serial clock generation conditions for the master clock. SISCR can be specified when the TRMD1 and TRMD0 bits in SIMDR are specified as B'10 or B'11.

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection 0: Uses the input signal of the SIOFMCLK pin as the master clock 1: Uses Pϕ as the master clock The master clock is the clock input to the baud rate generator.
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate generator as the serial clock 1: Uses the master clock itself as the serial clock
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	BRPS4	0	R/W	Prescalar Setting Set the master clock division ratio according to the count value of the prescalar of the baud rate generator. The range of settings is from B'00000 (× 1/1) to B'11111 (× 1/32).
11	BRPS3	0	R/W	
10	BRPS2	0	R/W	
9	BRPS1	0	R/W	
8	BRPS0	0	R/W	
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write 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
1	BRDV1	0	R/W	Set the frequency division ratio for the output stage of the baud rate generator.
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
				101: Setting prohibited
				110: Setting prohibited
				111: Prescalar output × 1/1*
				The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (maximum 1/1024).
				Note: *This setting is valid only when the BRPS4 to BRPS0 bits are set to B'00000.

16.3.11 Transmit Data Assign Register (SITDAR)

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

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

Bit	Bit Name	Initial Value	R/W	Description
11	TDLA3	0	R/W	Transmit Left-Channel Data Assigns 3 to 0
10	TDLA2	0	R/W	Specify the position of left-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
9	TDLA1	0	R/W	
8	TDLA0	0	R/W	1111: Setting prohibited • Transmit data for the left channel is specified in the SITDL bit in SITDR.
7	TDRE	0	R/W	Transmit Right-Channel Data Enable 0: Disables right-channel data transmission 1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit in SITDR as right-channel data 1: Repeatedly transmits data specified in the SITDL bit in SITDR as right-channel data • This bit setting is valid when the TDRE bit is set to 1. • When this bit is set to 1, the SITDR settings are ignored.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TDRA3	0	R/W	Transmit Right-Channel Data Assigns 3 to 0
2	TDRA2	0	R/W	Specify the position of right-channel data in a transmit frame as B'0000 (0) to B'1110 (14).
1	TDRA1	0	R/W	
0	TDRA0	0	R/W	1111: Setting prohibited • Transmit data for the right channel is specified in the SITDR bit in SITDR.

16.3.12 Receive Data Assign Register (SIRDAR)

SIRDAR is a 16-bit readable/writable register that specifies the position of the receive data in a frame (slot number).

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

16.3.13 Control Data Assign Register (SICDAR)

SICDAR is a 16-bit readable/writable register that specifies the position of the control data in a frame (slot number). SICDAR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: Don't care.).

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.
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 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the control channel 0 data is specified in the SITD0 bit in SITCR. • Receive data for the control channel 0 data is stored in the SIRD0 bit in SIRCR.
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.

Bit	Bit Name	Initial Value	R/W	Description
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 (0) to B'1110 (14).
1	CD1A1	0	R/W	
0	CD1A0	0	R/W	1111: Setting prohibited • Transmit data for the control channel 1 data is specified in the SITD1 bit in SITCR. • Receive data for the control channel 1 data is stored in the SIRD1 bit in SIRCR.

16.3.14 SPI Control Register (SPICR)

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

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

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SS0E	0	R/W	Slave Device 0 ($\overline{SS0}$) Enable 0: Not select slave device 0. 1: Selects slave device 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SSAST1	0	R/W	Setting of SS Assert
4	SSAST0	0	R/W	Set the setup timing of the SS for the SCK. • CPHA = 0 (Unit: SCK clock)

SSAST[1:0]	SS Setup	SS Hold
00	0.5 clock	0 clock
01	1 clock	0.5 clock
10	1.5 clock	1 clock
11	2 clock	1.5 clock

• CPHA = 1

(Unit: SCK clock)

SSAST[1:0]	SS Setup	SS Hold
00	0 clock	0.5 clock
01	0.5 clock	1 clock
10	1 clock	1.5 clock
11	1.5 clock	2 clock

3, 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	FLD1	0	R/W	Frame Delay
0	FLD0	0	R/W	Specify the minimum time of the idle state between frames in terms of the clock number of the SCK. 00: Not delay. The continuous communication of the SPI is performed when the $\overline{SS}$ pin asserts low continuously. 01: Delay for 1 clock of the SCK. 10: Delay for 2 clocks of the SCK. 11: Delay for 3 clocks of the SCK.

16.4 Operation

16.4.1 Serial Clocks

Master/Slave Modes: The following two modes are available as the SIOF clock mode.

- Slave mode: SIOFSCK, SIOFSYNC input
- Master mode: SIOFSCK, SIOFSYNC output

Baud Rate Generator: In SIOF master mode, the baud rate generator (BRG) is used to generate the serial clock. The division ratio is from 1/1 to 1/1024.

Figure 16.2 shows connections for supply of the serial clock.

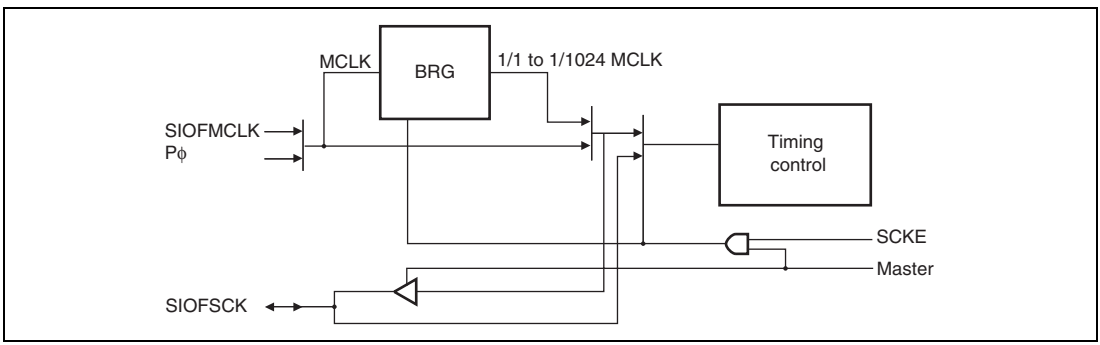


Figure 16.2 Serial Clock Supply

Table 16.3 shows an example of serial clock frequency.

Table 16.3 SIOF Serial Clock Frequency

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

16.4.2 Serial Timing

SIOFSYNC: The SIOFSYNC is a frame synchronous signal. Depending on the transfer mode, it has the following 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 16.3 shows the SIOFSYNC synchronization timing.

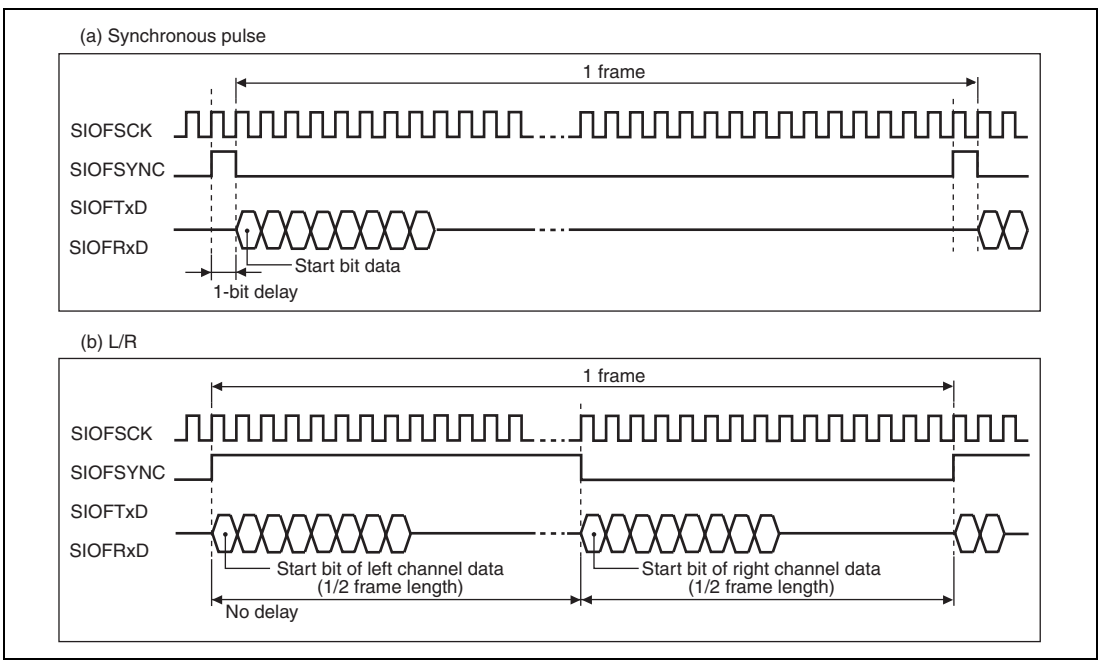


Figure 16.3 Serial Data Synchronization Timing

Transmit/Receive Timing: The SIOFTxD transmit timing and SIOFRxD receive timing relative to the SIOFSCK can be set as the sampling timing in the following ways. The transmit/receive timing is set using the REDG bit in SIMDR.

- Falling-edge sampling
- Rising-edge sampling

Figure 16.4 shows the transmit/receive timing.

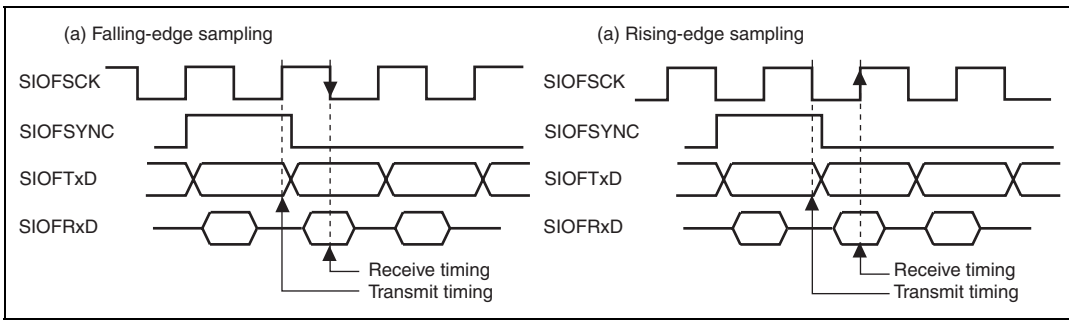


Figure 16.4 SIOF Transmit/Receive Timing

16.4.3 Transfer Data Format

The SIOF performs the following transfer.

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

Transfer Mode: The SIOF supports the following four transfer modes as listed in table 16.4. The transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 16.4 Serial Transfer Modes

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

Frame Length: The length of the frame to be transferred by the SIOF is specified by the FL3 to FL0 bits in SIMDR. Table 16.5 shows the relationship between the FL3 to FL0 bit settings and frame length.

Table 16.5 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

Note: x: Don't care.

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

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

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

16.4.4 Register Allocation of Transfer Data

Transmit/Receive Data: Writing and reading of transmit/receive data is performed for the following registers.

- Transmit data writing: SITDR (32-bit access)
- Receive data reading: SIRDR (32-bit access)

Figure 16.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

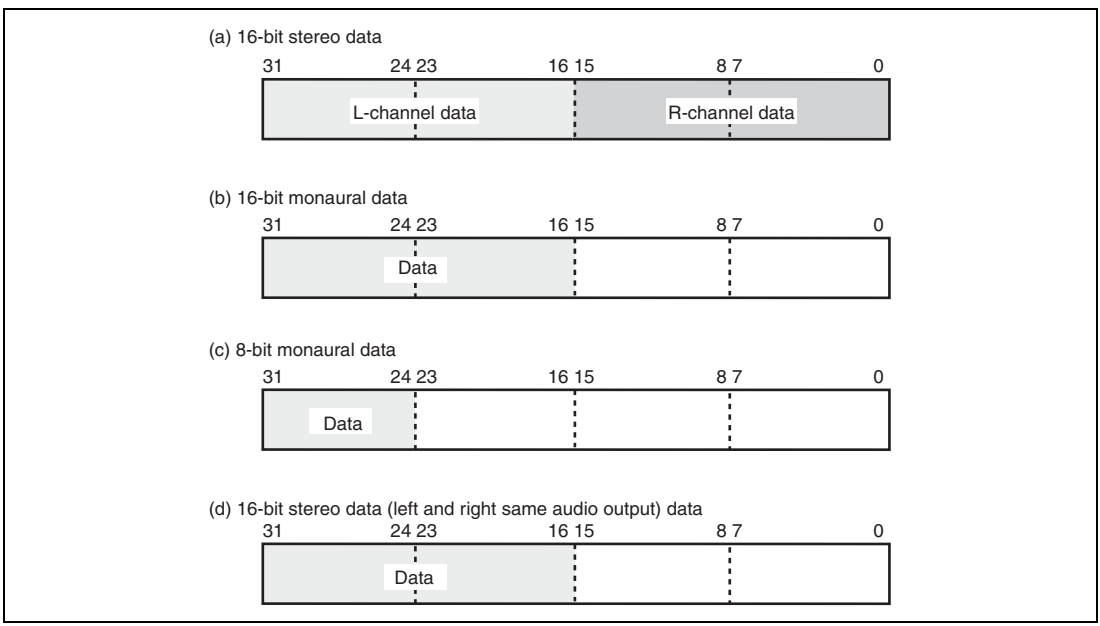


Figure 16.5 Transmit/Receive Data Bit Alignment

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 transmit data, specify the TLREP bit in SITDAR. Tables 16.6 and 16.7 show the audio mode specification for transmit data and that for receive data, respectively.

Table 16.6 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 16.7 Audio Mode Specification for Receive Data

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

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

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

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

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

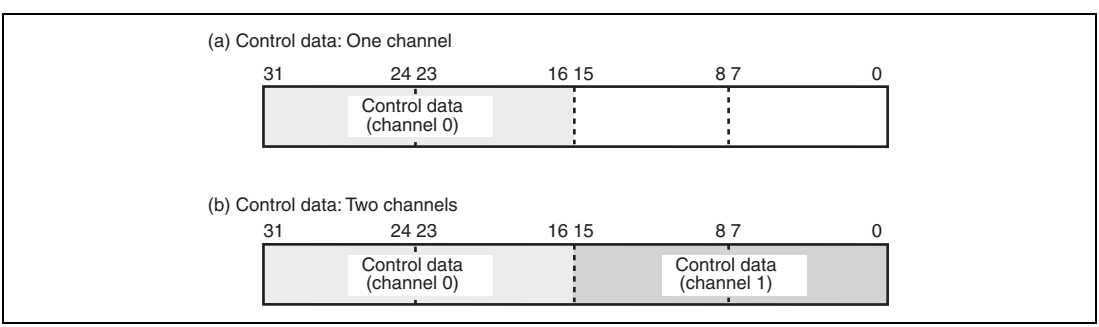


Figure 16.6 Control Data Bit Alignment

The number of channels in control data is specified by the CD0E and CD1E bits in SICDAR. Table 16.8 shows the relationship between the number of channels in control data and bit settings.

Table 16.8 Setting Number of Channels in Control Data

	Bit	
Number of Channels	**CD0E**	**CD1E**
1	1	0
2	1	1

Note: To use only one channel in control data, use channel 0.

16.4.5 Control Data Interface

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

- Control by slot position
- Control by secondary FS

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

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

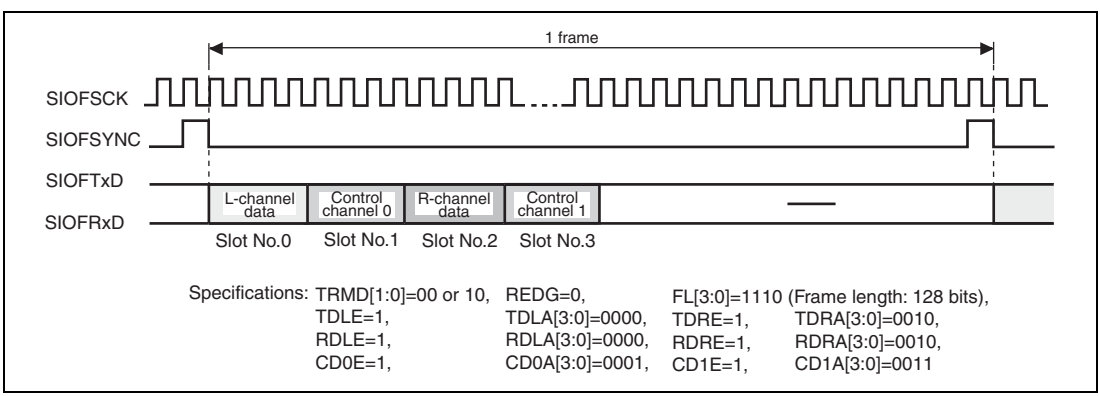


Figure 16.7 Control Data Interface (Slot Position)

Control by Secondary FS (Slave Mode 2): The CODEC normally outputs the SIOFSYNC signal as synchronization pulse (FS). In this method, the CODEC outputs the secondary FS specific to the control data transfer after 1/2 frame time has been passed (not the normal FS output timing) to transmit or receive control data. This method is valid for SIOF slave mode. The following summarizes the control data interface procedure by the secondary FS.

- Transmit normal transmit data of LSB = 0 (the SIOF forcibly clears 0).
- To execute control data transmission, send transmit data of LSB = 1 (the SIOF forcibly set to 1 by writing SITCDR).
- The CODEC outputs the secondary FS.
- The SIOF transmits or receives (stores in SIRCDR) control data (data specified by SITCDR) synchronously with the secondary FS.

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

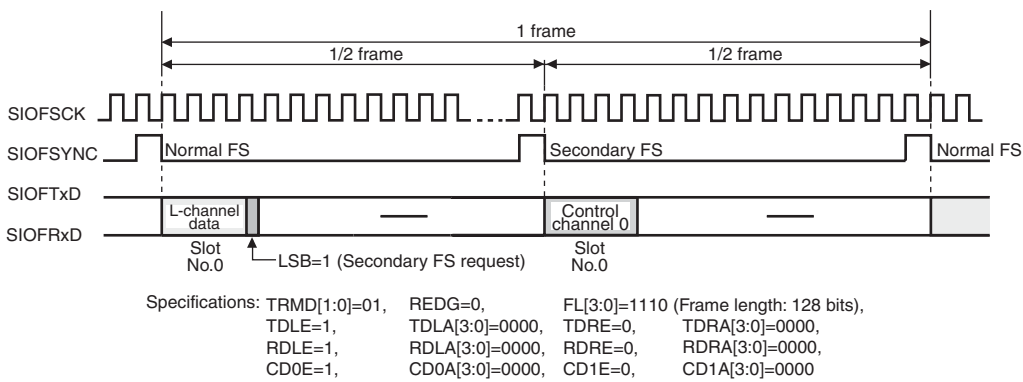


Figure 16.8 Control Data Interface (Secondary FS)

16.4.6 FIFO

Overview: The transmit and receive FIFOs of the SIOF have the following features.

- 16-stage 32-bit FIFOs for transmission and reception
- The FIFO pointer can be updated in one read or write cycle regardless of access size of the CPU and DMAC. (One-stage 32-bit FIFO access cannot be divided into multiple accesses.)

Transfer Request: The transfer request of the FIFO can be issued to the CPU or DMAC as the following interrupt sources.

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

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

Table 16.9 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 16.10 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 FIFO size (the number of FIFOs). Accordingly, an overflow error or underflow error occurs if data area or empty area exceeds sixteen FIFO stages. The FIFO transmit or receive request is canceled when the above condition is not satisfied even if the FIFO is not empty or full.

Number of FIFOs: The number of FIFO stages used in transmission and reception is indicated by the following register.

- Transmit FIFO: The number of empty FIFO stages is indicated by the TFUA4 to TFUA0 bits in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the RFUA4 to RFUA0 bits in SIFCTR.

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

16.4.7 Transmit and Receive Procedures

Transmission/Reception and Transmission in Master Mode: Figure 16.9 (1) shows an example of settings and operation for master mode transmission/reception. Figure 16.9 (2) shows an example of settings and operation for master mode transmission.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit/receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	Note: Serial clock will not be output from the pin until communication is actually started.
3	Set the FSE bit in SICTR to 1	Set the start for frame synchronous signal output	
4	Set the TXE and RXE bit in SICTR to 1	Enable transmission and reception	Note: Communication is actually started after SITDR has been written.
5	TDREQ = 1? No / Yes		
6	Set the SITDR register		
7	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC	Set transmit data	Transmission and reception are carried out simultaneously (Even when transmission is not necessary, dummy transmission must be performed. The output of dummy transmission can be masked by setting the pin function).
8	RDREQ = 1? No / Yes		
9	Read SIRDR register	Read receive data	
10	Transfer complete? No / Yes		Transmission and reception are carried out simultaneously, please check the TFEMP bit in SISTR (the transmit FIFO is empty?) and build a waiting loop to check the communication finished.
11	Clear the TXE bit and RXE bit in SICTR to 0	Disable transmission	End transmission
12	Clear the FSE bit in SICTR to 0	To be prepared for the transmission/ reception that is resumed later, set the FSE bit to '0' to synchronize the frame in this LSI.	
13	Set the MSSEL bit in SISCR to 1 Set the BPRS to 00000 and the BRDV to 111 in SISCR Apply a pulse to bits TxRST and RxRST in the SICTR (input 0→1→0) Set the SISCR to set the baud rate and the master clock source again	To be prepared for the transmission/ reception that is resumed later, initialize inside the baud rate generator.	
14	Change communication mode ? No / Yes END	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
15	With FSE = 0, TXE = 0, and RXE = 0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

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

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

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit/receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit to 1 in SICTR	Set operation start for baud rate generator	Note: Serial clock will not be output from the pin until communication is actually started.
3	Set the FSE bit in SICTR to 1	Set the start for frame synchronous signal output	
4	Set the TXE, and RXE bit in SICTR to 1	Enable transmission and reception	Note: Communication is actually started after SITDR has been written.
5	TDREQ = 1? No / Yes		
6	Set SITDR register	Set transmit data	
7	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC		Transmission
8	Transfer complete? No / Yes	* Please check the TFEMP bit in SISTR (the transmit FIFO is empty?) and build a waiting loop to check the communication finished.	
9	Clear the TXE bit in SICTR to 0	Disable transmission	End of transmission
10	Clear the FSE bit in SICTR to 0	To be prepared for the transmission/ reception that is resumed later, set the FSE bit to '0' to synchronize the frame in this LSI.	
11	Set the MSSEL bit in SISCR to 1 Set the BPRS to 00000 and the BRDV to 111 in SISCR Apply a pulse to bits TxRST and RxRST in the SICTR (input 0→1→0) Set the SISCR register to set the baud rate and the master clock source again	To be prepared for the transmission/ reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode ? No / Yes END	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
13	With FSE = 0, TXE = 0, and RXE = 0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

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

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

Reception in Master Mode: Figure 16.10 shows an example of settings and operation for master mode reception.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit/receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	
3	Start SIOFSCK output		Output serial clock
4	Set the FSE and RXE bits in SICTR to 1	Set the start for frame synchronous signal output and enable reception	Output frame synchronous signal
5	Store SIOFRXD receive data in SIRDR synchronously with SIOFSYNC		Issue receive transfer request according to the receive FIFO threshold value
6	RDREQ = 1? No / Yes		Reception
7	Read SIRDR	Read receive data	
8	Transfer ended? No / Yes Clear the RXE bit in SICTR to 0 End	Set to disable reception	End reception

Figure 16.10 Example of Receive Operation in Master Mode

Transmission in Slave Mode: Figure 16.11 shows an example of settings and operation for slave mode transmission.

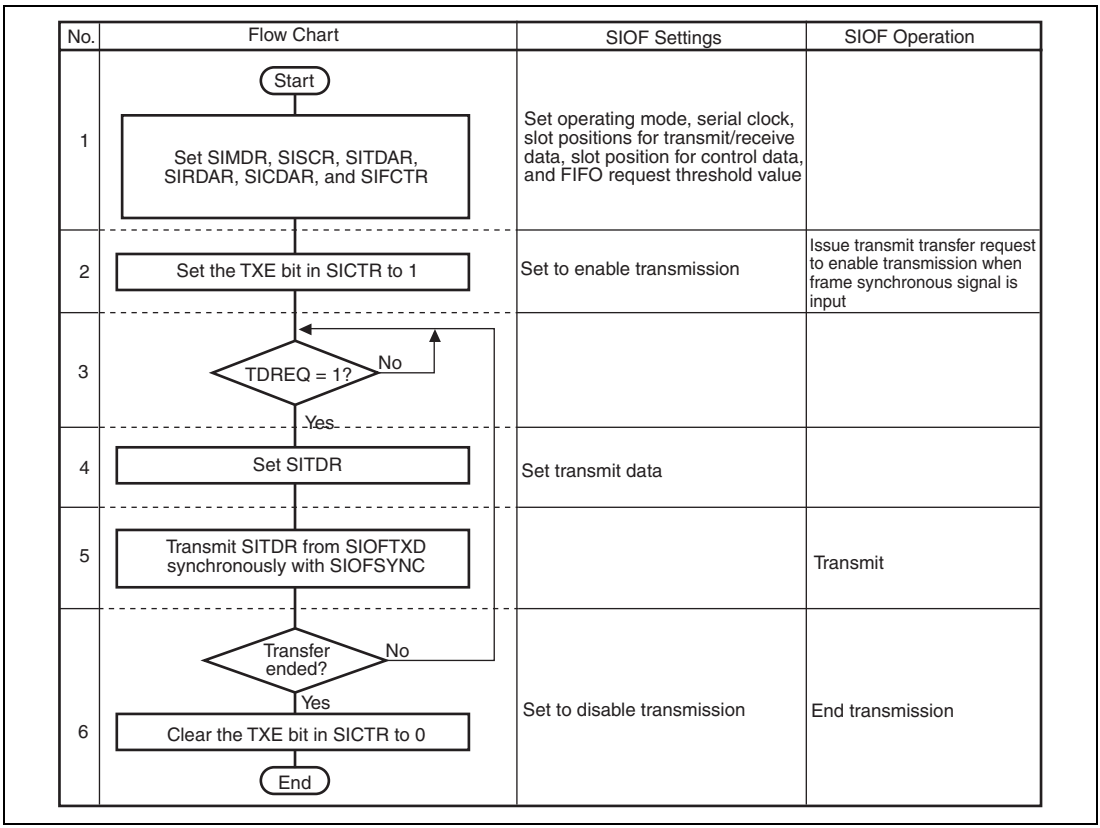


Figure 16.11 Example of Transmit Operation in Slave Mode

Reception in Slave Mode: Figure 16.12 shows an example of settings and operation for slave mode reception.

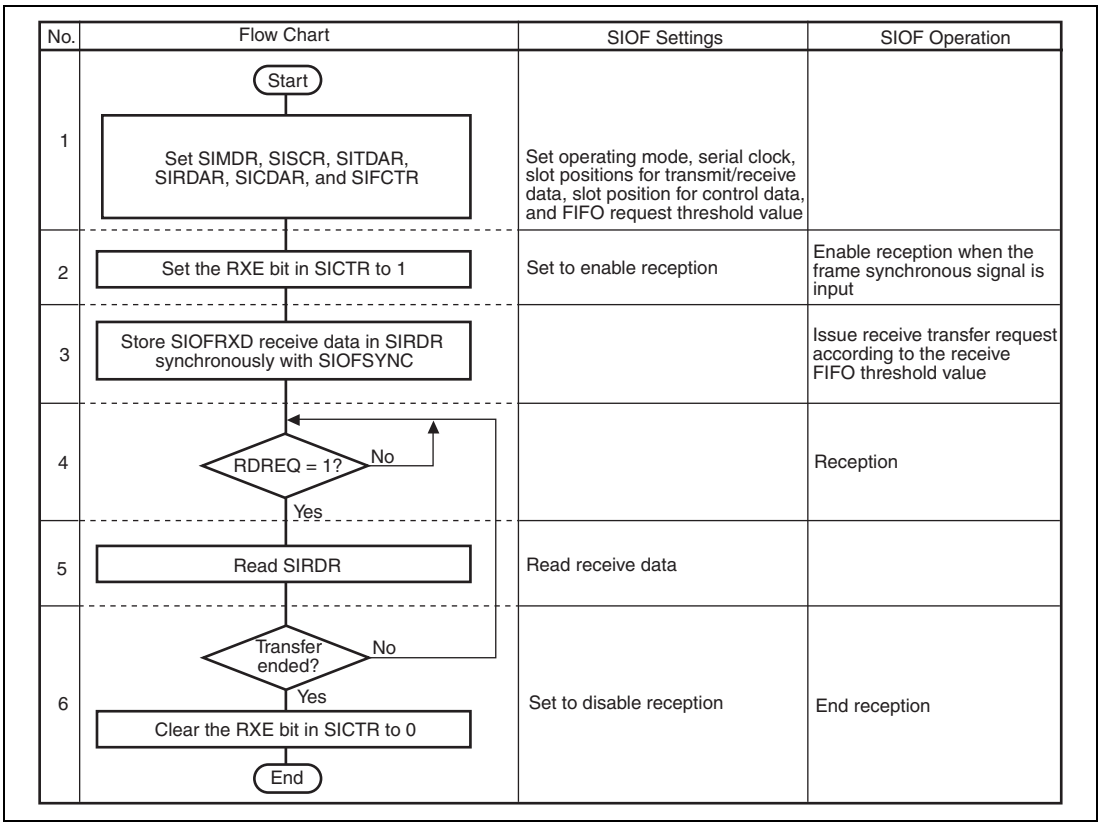


Figure 16.12 Example of Receive Operation in Slave Mode

Transmit/Receive Reset: The SIOF can separately reset the transmit and receive units by setting the following bits to 1.

- Transmit reset: TXRST bit in SICTR
- Receive reset: RXRST bit in SICTR

Table 16.11 shows the details of initialization upon transmit or receive reset.

Table 16.11 Transmit and Receive Reset

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

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

16.4.8 Interrupts

The SIOF has one type of interrupt.

Interrupt Sources: Interrupts can be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 16.12 lists the SIOF interrupt sources.

Table 16.12 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register is ready to be written.
6		RCRDY	Receive control data ready	The receive control data register stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in both serial data and control data.

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

Regarding Transmit and Receive Classification: The transmit sources and receive sources are signals indicating the state; after being set, if the state changes, they are automatically cleared by the SIOF.

When the DMA transfer is used, a DMA transfer request is pulled low (0 level) for one cycle at the end of DMA transfer.

Processing when Errors Occur: On occurrence of each of the errors indicated as a status in SISTR, the SIOF performs the following operations.

- Transmit FIFO underflow (TFUDF)

  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)

  The contents of the transmit FIFO are protected, and the write operation causing the overflow is ignored.
- Receive FIFO overflow (RFOVF)

  Data causing the overflow is discarded and lost.
- Receive FIFO underflow (RFUDF)

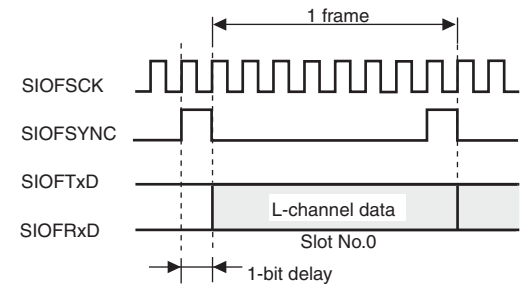
  An undefined value is output on the bus.
- FS error (FSERR)

  The internal counter is reset according to the FSYN signal in which an error occurs.
- Assign error (SAERR)
  - If the same slot is assigned to both serial data and control data, the slot is assigned to serial data.
  - If the same slot is assigned to two control data items, data cannot be transferred correctly.

16.4.9 Transmit and Receive Timing

Examples of the SIOF serial transmission and reception are shown in figures 16.13 to 16.19.

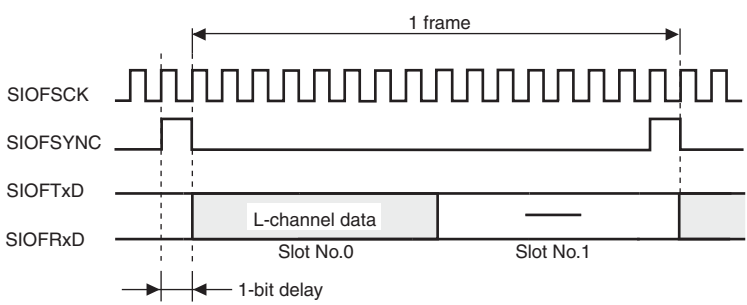
8-bit Monaural Data (1): Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, an frame length = 8 bits



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

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

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



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

Figure 16.14 Transmit and Receive Timing (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, and frame length = 64 bits

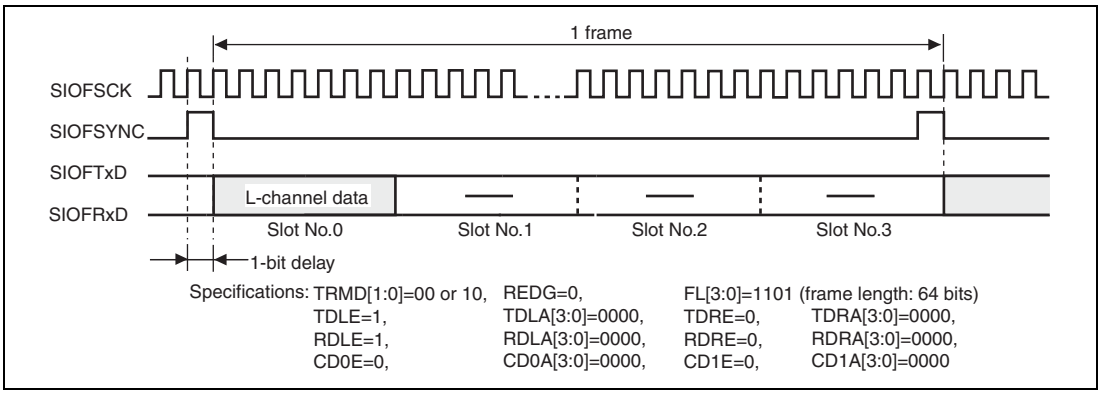


Figure 16.15 Transmit and Receive Timing (16-Bit Monaural Data (1))

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

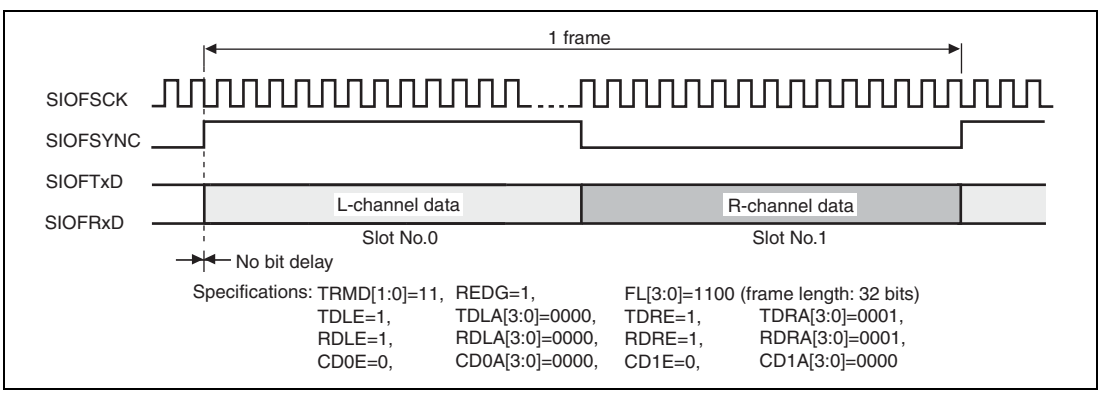


Figure 16.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

16-bit Stereo Data (2): L/R method, rising edge sampling, slot No.0 used for left-channel transmit data, slot No.1 used for left-channel receive data, slot No.2 used for right-channel transmit data, slot No.3 used for right-channel receive data, and frame length = 64 bits

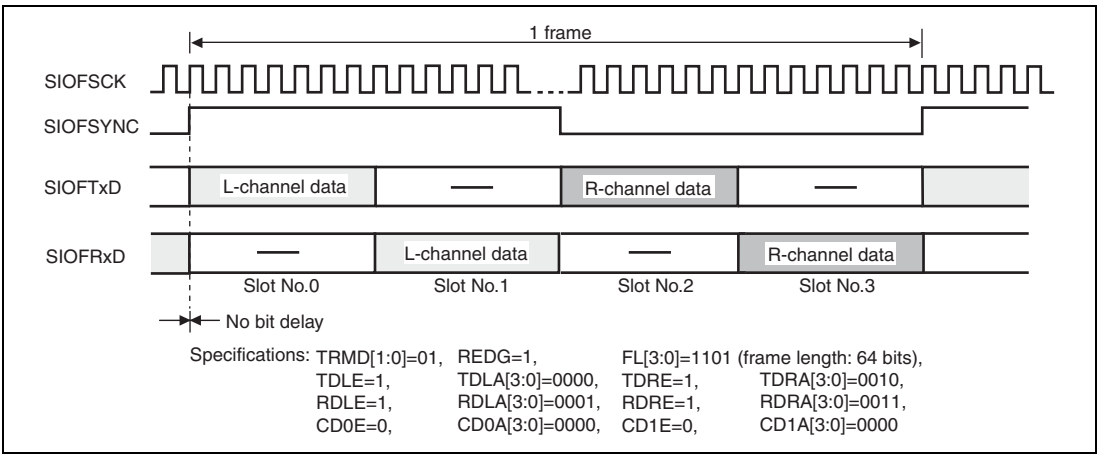


Figure 16.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

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

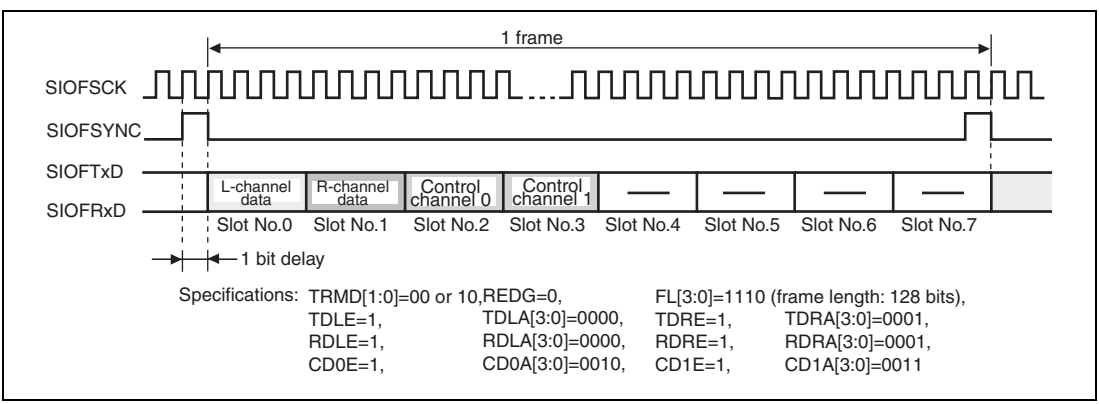


Figure 16.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

16-bit Stereo Data (4): Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.2 used for right-channel data, slot No.1 used for control channel 0 data, slot No.3 used for control channel 1 data, and frame length = 128 bits

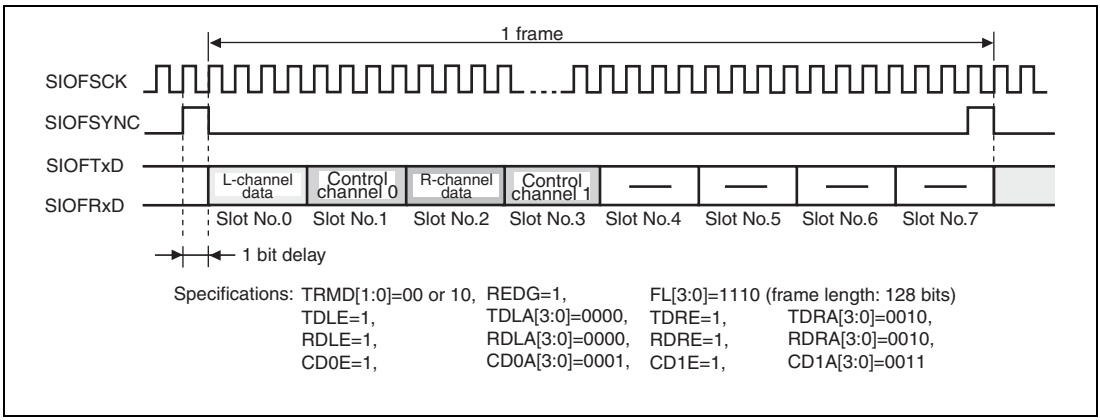


Figure 16.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

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

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

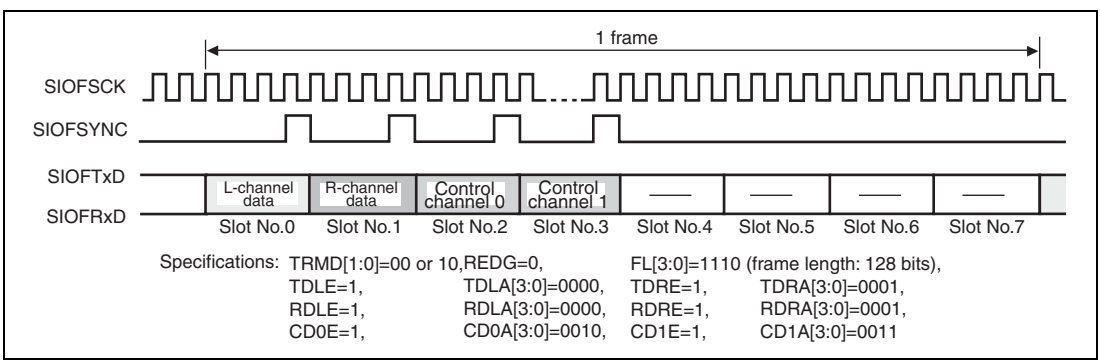


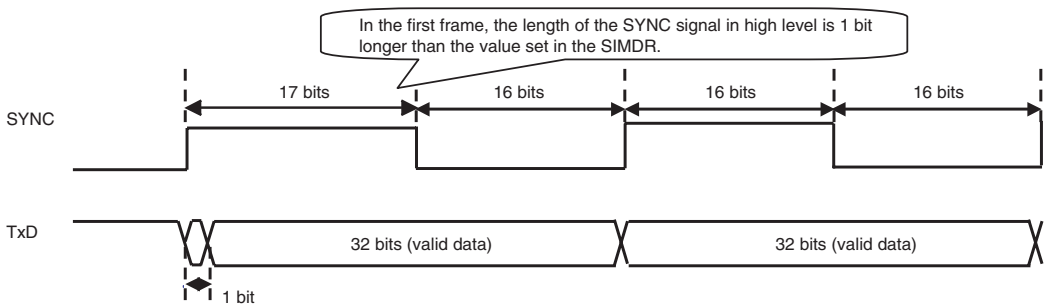
Figure 16.20 Transmit and Receive Timing (16-Bit Stereo Data)

[Notes on Usage]

1. Defect in Frame Synchronous Signal (SYNC) Length

(a) Phenomena

With the SIOF Master Mode 2, in some case, the length of the SYNC signal in high level error occurs by changing the SYNC signal output condition from disabled (the FSE bit = 0) to enabled (the FSE bit = 1). In the above case, the SYNC signal rises before the correct timing and the length of the signal in high level will be 1 bit longer in the first frame than the value set in the SIMDR, whereas the error DOES NOT occur after second frame.



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

(b) Defect prevention

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

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

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

2. Resume Data Transmission with the SIOF Master Mode

(a) Defect data transmission

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

(b) Cause of the Defect

After the transmission, the SIOF will be reset and initialized by setting the TXE bit in the SICTR to ‘0’ with the SIOF master mode 1/2. However, if that SIOF fails to be reset, the transmission will resume without the transmission module initialized and the defect mentioned above will occur.

(c) Defect prevention

Please change the setting of SCK temporarily to make reset the modules using the CK clock surely, specifically, whenever the TXE bit or the RXE bit in SICTR is set to ‘0’, please take the procedures as follows;

(i) Set the Pϕ as the master clock source.

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

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

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

(iii) Set the frequency division ratio for the output stage of the baud rate generator as ×1/1.

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

(iv) Reset the transmission/reception operation.

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

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

16.4.10 SPI Mode

SPI-mode operation is selected for the SIOF by the setting in SPICR.

Example of Configuration: Figure 16.21 shows an example of the configuration for SPI-mode communications.

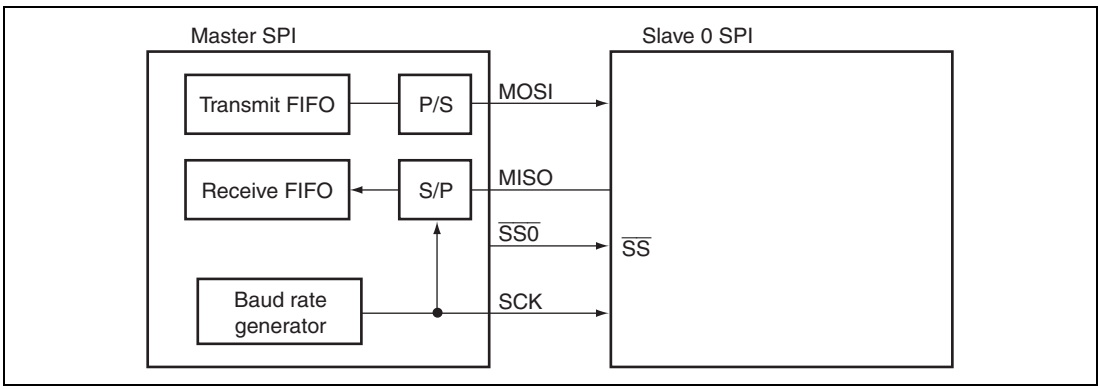
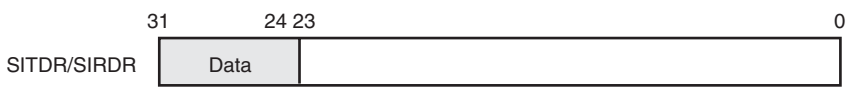


Figure 16.21 Example of Configuration in SPI Mode

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



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

The only sources of interrupts that should be enabled in SPI-mode transfer are transmit data transfer request (TDREQ), transmit FIFO empty (TFEMP), receive-data transfer request (RDREQ), receive-FIFO full (RFFUL), and receive-FIFO overflow (RFOVF). Enabled or disabled states are selectable by the interrupt enable register (SIIER). Interrupts from other sources must be disabled at all times.

For the DMA transfer requests, the enabled sources are transmit-data DMA transfer request (TDMA) and receive-data DMA transfer request (RDMA). Enabled or disabled states are selectable by the interrupt enable register (SIIER).

In SPI mode, the baud rate is set by SISCR.

Table 16.13 States of Transmit and Receive Operations in SPI Mode

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

TXE	RXE	TDMAE	RDMAE	SPI Transmit/Receive Operation
1	1	0	0	Full-Duplex Communication The transmit and receive FIFOs operate at the same time. Data in the transmit FIFO are transmitted or received. When there is no data left in the transmit FIFO, the transmit and receive operations end. Note that even if only reception is to be done, dummy transmission is necessary because only master mode is allowed in the SPI mode.

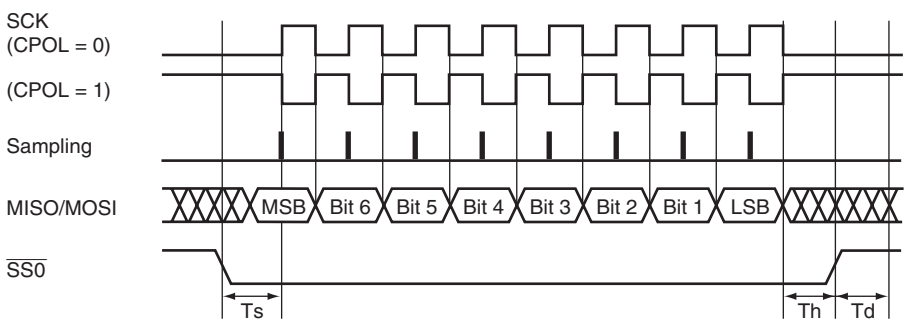
Note: In SPI mode, settings other than the above are prohibited.

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

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

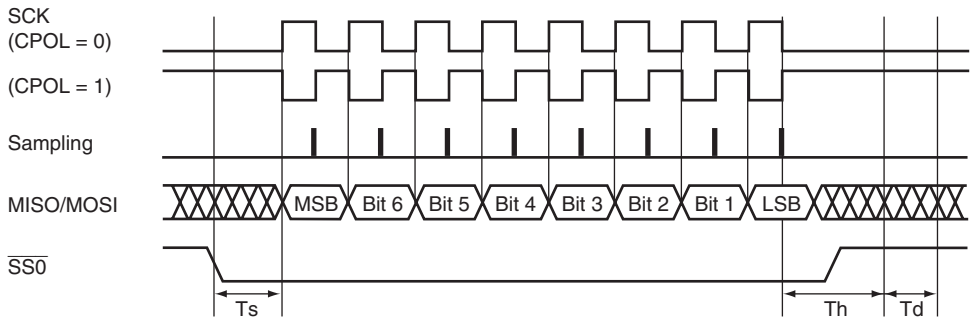
TXDIZ = 1: Transmission is disabled, the MOSI is in the high-impedance state.

Serial Clock Timing: Timing on the data and clock lines in SPI mode is shown in figures 16.22 and 16.23. The user can select from four serial transfer formats, which differ according to the phase and polarity of the serial clock.



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

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



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

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

Procedures for Transmission/Reception: Shown in figures 16.24 to 16.27 are examples of settings for SPI transmission/reception along with the corresponding operations.

No.	Time chart	SIOF Setting	SIOF Operation
1	Start Set SISCR, SIFCTR, and SPICR.	Set the serial clock and threshold values for FIFO.	[Note] In SPI mode, registers SIMDR, SITDAR, SIRDAR, and SICDAR should be set to their initial values.
2	Set the SCKE bit in SICTR to 1.	Start baud rate generator operation.	[Note] Serial clock will not be output form the pin until communication is actually started.
3	Set the FSE bit in SICTR to 1. Set the TXE and RXE bits in SICTR to 1.*	Initialize the frame in the SIOF (ie, initialize the state of signal $\overline{SS0}$), and enable transmission and reception.	[Note] Communication is actually started after SITDR has been written.
4	TDREQ=1? No / Yes		
5	Set the SITDR register.	Set the data for transmission.	
6	Synchronously to $\overline{SS0}$, output the contents of SITDR from MOSI and receive data from MISO.		Executes transmission and reception simultaneously. (Even when transmission is not necessary, dummy transmission must be performed. The output of dummy transmission can be masked by setting the pin function.)
7	RDREQ=1? No / Yes		
8	Read the SIRDR register.	Read the received data.	
9	Transfer complete? No / Yes	Check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop or other means. (Checking SISTR.TFEMP is enough to confirm the completion of simultaneous transmission and reception.)	
10	Clear the TXE and RXE bits in SICTR to 0.	Disable transmission and reception.	Transmission/reception end.
11	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
12	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bits TxRST and RxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
13	Change communication mode? No / Yes End	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
14	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Note: * For the case when interrupt generation on transmit FIFO underflow is enabled, set the TXE bit to 1 after setting data for transmission at step No.5.

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

No.	Time chart	SIOF Setting	SIOF Operation
1	Start Set SISCR, SIFCTR, and SPICR.	Set the serial clock and threshold values for FIFO.	[Note] In SPI mode, registers SIMDR, SITDAR, SIRDAR, and SICDAR should be set to their initial values.
2	Set the SCKE bit in SICTR to 1.	Start baud rate generator operation.	[Note] Serial clock will not be output form the pin until communication is actually started.
3	Set the FSE bit in SICTR to 1. Set the TXE bit in SICTR to 1.*	Initialize the frame in the SIOF (ie, initialize the state of signal $\overline{SS0}$), and enable transmission.	[Note] Communication is actually started after SITDR has been written.
4	TDREQ=1? No / Yes		
5	Set the SITDR register.	Set the data for transmission.	
6	Synchronously to $\overline{SS0}$, output the contents of SITDR from MOSI.		Executes transmission.
7	Transfer complete? No / Yes	Check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop or other means.	
8	Clear the TXE bit in SICTR to 0.	Disable transmission.	Transmission ends.
9	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
10	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit TxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
11	Change communication mode? No / Yes End	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
12	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Note: * For the case when interrupt generation on transmit FIFO underflow is enabled, set the TXE bit to 1 after setting data for transmission at step No.5.

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

No.	Time Chart	SIOF/DMA Setting	SIOF/DMA Operation
1	Start Make settings for DMA.	Complete setting for DMA before making settings for the SIOF.	
2	Set SISCR, SIFCTR, SPICR, and SIIER.	Set the serial clock, threshold values for FIFO, and TDMAE = 1	[Note] In SPI mode, registers SIMDR, SITDAR, SIRDAR, and SICDAR should be set to their initial values.
3	Set the SCKE bit in SICTR to 1.	Start baud rate generator operation.	[Note] Serial clock will not be output form the pin until communication is actually started.
4	Set the FSE bit in SICTR to 1. Set the TXE bit in SICTR to 1.	Initialize the frame in the SIOF (ie, initialize the state of signal $\overline{SS0}$), and enable transmission.	[Note] Communication is actually started after SITDR has been written.
5	TDREQ=1? No / Yes		
6	DMA transfer (Set the SITDR register.)	Set the data for transmission.	
7	Synchronously to $\overline{SS0}$, output the contents of SITDR from MOSI.		Executes transmission.
8	Transfer complete? No / Yes	For example, in the DMA transfer end interrupt service routine, check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop.	When the DMA transfr end interrupt is used, clear the IE bit in DMA.CHCRn before the execution returns from the interrupt service routine.
9	Clear the TXE bit in SICTR to 0.	Disable transmission.	Transmission ends.
10	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
11	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit TxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode? No / Yes End	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 4 of this flowchart.	
13	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

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

No.	Time Chart	SIOF/DMA Setting	SIOF/DMA Operation
1	Start Make settings for DMA.	Complete setting for DMA before making settings for the SIOF.	
2	Set SISCR, SIFCTR, SPICR, and SIIER.	Set the serial clock, threshold values for FIFO, and RDMAE = 1	[Note] In SPI mode, registers SIMDR, SITDAR, SIRDAR, and SICDAR should be set to their initial values.
3	Set the SCKE bit in SICTR to 1.	Start baud rate generator operation.	[Note] Serial clock will not be output form the pin until communication is actually started.
4	Set the FSE bit in SICTR to 1. Set the RXE bit in SICTR to 1.	Initialize the frame in the SIOF (ie, initialize the state of signal $\overline{SS0}$), and enable reception.	[Note] Communication is actually started after RXE = 1 has been set.
5	RDREQ=1? No / Yes		
6	DMA transfer (Read from the SIRDR register.)	Read data from the SIRDR register.	
7	Synchronously to $\overline{SS0}$, receive data from MISO.		Executes reception.
8	Transfer complete? No / Yes	At the point when a DEI interrupt is generated, reception has already proceeded excessively. Therefore, read the necessary amount of data from FIFO and skip the remainder, or cancel by RxRST.	When the DMA transfr end interrupt is used, clear the IE bit in DMA.CHCRn before the execution returns from the interrupt service routine.
9	Clear the RXE bit in SICTR to 0.	Disable reception.	Reception ends.
10	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
11	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit RxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode? No / Yes End	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 4 of this flowchart.	
13	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

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

Section 17 Host Interface (HIF)

This LSI incorporates a host interface (HIF) for use in high-speed transfer of data between external devices which cannot utilize the system bus.

The HIF allows external devices to read from and write to 2 kbytes (1 kbyte × 2 banks) of the on-chip RAM exclusively for HIF use (HIFRAM) within this LSI, in 32-bit units. Interrupts issued to this LSI by an external device, interrupts sent from this LSI to the external device, and DMA transfer requests sent from this LSI to the external device are also supported. By using HIFRAM and these interrupt functions, software-based data transfer between external devices and this LSI becomes possible, and connection to external devices not releasing bus mastership is enabled.

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

17.1 Features

The HIF has the following features.

- An external device can read from or write to HIFRAM in 32-bit units via the HIF pins (access in 8-bit or 16-bit units not allowed). The on-chip CPU can read from or write to HIFRAM in 8-bit, 16-bit, or 32-bit units, via the internal peripheral bus. The HIFRAM access mode can be specified as bank mode or non-bank mode.
- When an external device accesses HIFRAM via the HIF pins, automatic increment of addresses and the endian can be specified with the HIF internal registers.
- By writing to specific bits in the HIF internal registers from an external device, or by accessing the end address of HIFRAM from the external device, interrupts (internal interrupts) can be issued to the on-chip CPU. Conversely, by writing to specific bits in the HIF internal registers from the on-chip CPU, interrupts (external interrupts) or DMAC transfer requests can be sent from the on-chip CPU to the external device.
- There are seven interrupt source bits each for internal interrupts and external interrupts. Accordingly, software control of 128 different interrupts is possible, enabling high-speed data transfer using interrupts.
- In HIF boot mode, this LSI can be booted from HIFRAM by an external device storing the instruction code in HIFRAM.

Figure 17.1 shows a block diagram of the HIF.

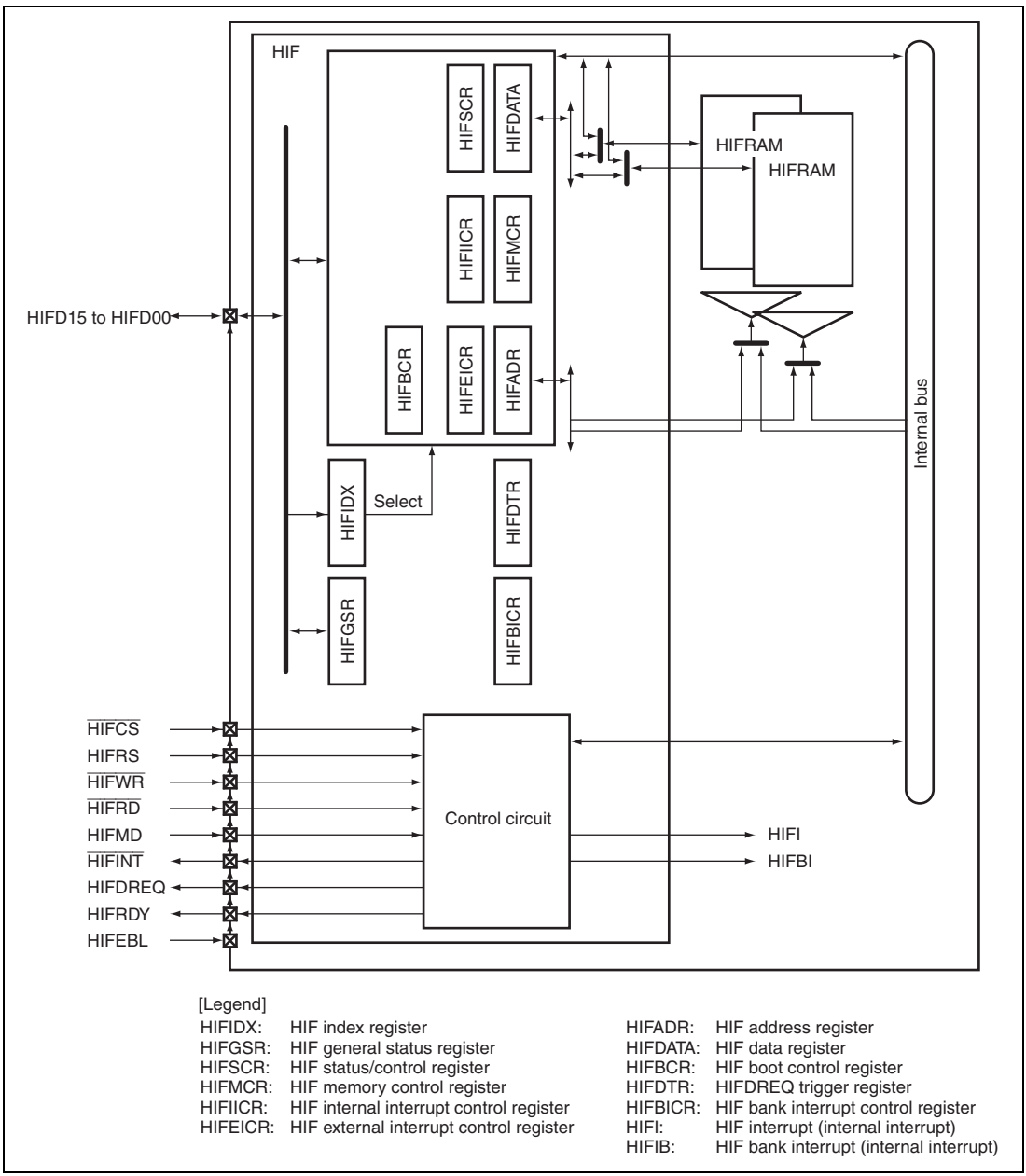


Figure 17.1 Block Diagram of HIF

17.2 Input/Output Pins

Table 17.1 shows the HIF pin configuration.

Table 17.1 Pin Configuration

Name	Abbreviation	I/O	Description
HIF data pins	HIFD15 to HIFD00	I/O	Address, data, or command input/output to the HIF
HIF chip select	$\overline{\text{HIFCS}}$	Input	Chip select input to the HIF
HIF register select	HIFRS	Input	Switching between HIF access types 0: Normal access (other than below) 1: Index register write or status register read
HIF write	$\overline{\text{HIFWR}}$	Input	Write strobe signal. Low level is input when an external device writes data to the HIF.
HIF read	$\overline{\text{HIFRD}}$	Input	Read strobe signal. Low level is input when an external device reads data from the HIF.
HIF interrupt	$\overline{\text{HIFINT}}$	Output	Interrupt request to an external device from the HIF
HIF mode	HIFMD	Input	Selects whether or not this LSI is started up in HIF boot mode. If a power-on reset is canceled when high level is input, this LSI is started up in HIF boot mode.
HIFDMAC transfer request	HIFDREQ	Output	To an external device, DMAC transfer request with HIFRAM as the destination
HIF boot ready	HIFRDY	Output	Indicates that the HIF reset is canceled in this LSI and access from an external device to the HIF can be accepted. After 10 clock cycles (max.) of the peripheral clock following negate of the reset input pin of this LSI, this pin is asserted.
HIF pin enable	HIFEBL	Input	All HIF pins other than this pin are asserted by high-level input.

17.3 Parallel Access

17.3.1 Operation

The HIF can be accessed by combining the $\overline{\text{HIFCS}}$, HIFRS, $\overline{\text{HIFWR}}$, and $\overline{\text{HIFRD}}$ pins. Table 17.2 shows the correspondence between combinations of these signals and HIF operations.

Table 17.2 HIF Operations

$\overline{\text{HIFCS}}$	HIFRS	$\overline{\text{HIFWR}}$	$\overline{\text{HIFRD}}$	Operation
1	*	*	*	No operation (NOP)
0	0	1	0	Read from register specified by HIFIDX[7:0]
0	0	0	1	Write to register specified by HIFIDX[7:0]
0	1	1	0	Read from status register (HIFGSR[7:0])
0	1	0	1	Write to index register (HIFIDX[7:0])
0	*	1	1	No operation (NOP)
0	*	0	0	Setting prohibited

[Legend]
*: Don't care

17.3.2 Connection Method

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

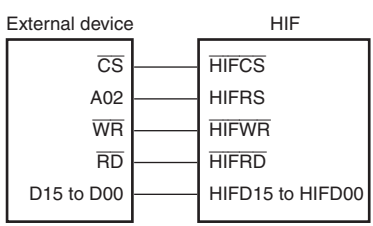


Figure 17.2 HIF Connection Example

17.4 Register Descriptions

The HIF has the following registers.

- HIF index register (HIFIDX)
- HIF general status register (HIFGSR)
- HIF status/control register (HIFSCR)
- HIF memory control register (HIFMCR)
- HIF internal interrupt control register (HIFIICR)
- HIF external interrupt control register (HIFEICR)
- HIF address register (HIFADR)
- HIF data register (HIFDATA)
- HIF boot control register (HIFBCR)
- HIFDREQ trigger register (HIFDTR)
- HIF bank interrupt control register (HIFBICR)

17.4.1 HIF Index Register (HIFIDX)

HIFIDX is a 32-bit register used to specify the register read from or written to by an external device when the HIFRS pin is held low. HIFIDX can be only read by the on-chip CPU. HIFIDX can be only written to by an external device while the HIFRS pin is driven high.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	REG5	0	R/W*	HIF Internal Register Select
6	REG4	0	R/W*	These bits specify which register among HIFGSR, HIFSCR, HIFMCR, HIFIICR, HIFEICR, HIFADR, HIFDATA, and HIFBCR is accessed by an external device.
5	REG3	0	R/W*	
4	REG2	0	R/W*	
3	REG1	0	R/W*	000000: HIFGSR
2	REG0	0	R/W*	000001: HIFSCR
				000010: HIFMCR
				000011: HIFIICR
				000100: HIFEICR
				000101: HIFADR
				000110: HIFDATA
				001111: HIFBCR
				Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
1	BYTE1	0	R/W*	Internal Register Byte Specification
0	BYTE0	0	R/W*	These bits specify in advance the target word location before the external device accesses a register among HIFGSR, HIFSCR, HIFMCR, HIFIICR, HIFEICR, HIFADR, HIFDATA, and HIFBCR. See also section 17.9, Alignment Control. • When HIFSCR.BO = 0 00: Bits 31 to 16 in register 01: Setting prohibited 10: Bits 15 to 0 in register 11: Setting prohibited • When HIFSCR.BO = 1 00: Bits 15 to 0 in register 01: Setting prohibited 10: Bits 31 to 16 in register 11: Setting prohibited However, when HIFDATA is selected using bits REG5 to REG0, each time reading or writing of HIFDATA occurs, these bits change according to the following rule. 00 → 10 → 00 → 10... repeated

Note: * This bit can be only written to by an external device while the HIFRS pin is held high. It cannot be written to by the on-chip CPU.

17.4.2 HIF General Status Register (HIFGSR)

HIFGSR is a 32-bit register, which can be freely used for handshaking between an external device connected to the HIF and the software of this LSI. HIFGSR can be read from and written to by the on-chip CPU. Reading from HIFGSR by an external device should be performed with the HIFRS pin high, or HIFGSR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low. Writing to HIFGSR by an external device should be performed with HIFGSR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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	STATUS15 to STATUS0	All 0	R/W	General Status This register can be read from and written to by an external device connected to the HIF, and by the on-chip CPU. These bits are initialized only at a power-on reset.

17.4.3 HIF Status/Control Register (HIFSCR)

HIFSCR is a 32-bit register used to control the HIFRAM access mode and endian setting. HIFSCR can be read from and written to by the on-chip CPU. Access to HIFSCR by an external device should be performed with HIFSCR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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

Bit	Bit Name	Initial Value	R/W	Description
11	DMD	0	R/W	DREQ Mode
10	DPOL	0	R/W	DREQ Polarity Controls the assert mode for the HIFDREQ pin. For details on the negate timing, see section 17.8, External DMAC Interface. 00: For a DMAC transfer request to an external device, low level is generated at the HIFDREQ pin. The default for the HIFDREQ pin is high-level output. 01: For a DMAC transfer request to an external device, high level is generated at the HIFDREQ pin. The default for the HIFDREQ pin is low-level output. 10: For a DMAC transfer request to an external device, falling edge is generated at the HIFDREQ pin. The default for the HIFDREQ pin is high-level output. 11: For a DMAC transfer request to an external device, rising edge is generated at the HIFDREQ pin. The default for the HIFDREQ pin is low-level output.
9	BMD	0	R/W	HIFRAM Bank Mode
8	BSEL	0	R/W	HIFRAM Bank Select Controls the HIFRAM access mode. 00: Both an external device and the on-chip CPU can access bank 0. When access by both of these conflict, even though the access addresses differ, access by the external device is processed before access by the on-chip CPU. Bank 1 cannot be accessed. 01: Both an external device and the on-chip CPU can access bank 1. When access by both of these conflict, even though the access addresses differ, access by the external device is processed before access by the on-chip CPU. Bank 0 cannot be accessed. 10: An external device can access only bank 0 while the on-chip CPU can access only bank 1. 11: An external device can access only bank 1 while the on-chip CPU can access only bank 0.

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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	MD1	0/1	R	HIF Mode 1 Indicates whether this LSI was started up in HIF boot mode or non-HIF boot mode. This bit stores the value of the HIFMD pin sampled at a power-on reset 0: Started up in non-HIF boot mode (booted from the memory connected to area 0) 1: Started up in HIF boot mode (booted from HIFRAM)
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	WBSWP	0	R/W	Byte Order for Access of HIFDATA Specifies the byte order when an external device accesses HIFDATA. See also section 17.9, Alignment Control. 0: Aligned according to the BO bit. 1: Swapped in word units from the big endian order and then swapped in byte units within each word. The setting of the BO bit is ignored.
1	EDN	0	R/W	Endian for HIFRAM Access Specifies the byte order when HIFRAM is accessed by the on-chip CPU. 0: Big endian (MSB first) 1: Little endian (LSB first)

Bit	Bit Name	Initial Value	R/W	Description
0	BO	0	R/W	Byte Order for Access of All HIF Registers Including HIFDATA Specifies the byte order when an external device accesses all HIF registers including HIFDATA. However, for the HIFDATA alignment, this bit is referred to only when WBSWP = 0 and ignored when WBSWP = 1. See also section 17.9, Alignment Control. 0: Big endian (MSB first) 1: Little endian (LSB first)

17.4.4 HIF Memory Control Register (HIFMCR)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	LOCK	0	R/W*	Lock This bit is used to lock the access direction (read or write) for consecutive access of HIFRAM by an external device via HIFDATA. When this bit is set to 1, the values of the RD and WT bits set at the same time are held until this bit is next cleared to 0. When the RD bit and this bit are simultaneously set to 1, consecutive read mode is entered. When the WT bit and this bit are simultaneously set to 1, consecutive write mode is entered. Both the RD and WT bits should not be set to 1 simultaneously.
6	—	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
5	WT	0	R/W*	Write When this bit is set to 1, the HIFDATA value is written to the HIFRAM position corresponding to HIFADR. If this bit and the LOCK bit are set to 1 simultaneously, HIFRAM consecutive write mode is entered, and high-speed data transfer becomes possible. This mode is maintained until this bit is next cleared to 0, or until the LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 with this bit, writing to HIFRAM is performed only once. Thereafter, the value of this bit is automatically cleared to 0.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	RD	0	R/W*	Read When this bit is set to 1, the HIFRAM data corresponding to HIFADR is fetched to HIFDATA. If this bit and the LOCK bit are set to 1 simultaneously, HIFRAM consecutive read mode is entered, and high-speed data transfer becomes possible. This mode is maintained until this bit is next cleared to 0, or until the LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 with this bit, reading of HIFRAM is performed only once. Thereafter, the value of this bit is automatically cleared to 0.
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AI/AD	0	R/W*	Address Auto-Increment/Decrement This bit is valid only when the LOCK bit is 1. The value of HIFADR is automatically incremented by 4 or decremented by 4 according to the setting of this bit each time reading or writing of HIFRAM is performed. 0: Auto-increment mode (+4) 1: Auto-decrement mode (–4)

Note: * This bit can be only written to by an external device when the HIFRS pin is low. It cannot be written to by the on-chip CPU. Changing the HIFRAM banks accessible from an external device by setting the BMD and BSEL bits in HIFSCR does not affect the setting of this bit.

17.4.5 HIF Internal Interrupt Control Register (HIFIICR)

HIFIICR is a 32-bit register used to issue interrupts from an external device connected to the HIF to the on-chip CPU. Access to HIFIICR by an external device should be performed with HIFIICR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IIC6	0	R/W	Internal Interrupt Source These bits specify the source for interrupts generated by the IIR bit. These bits can be written to from both an external device and the on-chip CPU. By using these bits, fast execution of interrupt exception handling is possible. These bits are completely under software control, and their values have no effect on the operation of this LSI.
6	IIC5	0	R/W	
5	IIC4	0	R/W	
4	IIC3	0	R/W	
3	IIC2	0	R/W	
2	IIC1	0	R/W	
1	IIC0	0	R/W	
0	IIR	0	R/W	Internal Interrupt Request While this bit is 1, an interrupt request (HIFI) is issued to the on-chip CPU.

17.4.6 HIF External Interrupt Control Register (HIFEICR)

HIFEICR is a 32-bit register used to issue interrupts to an external device connected to the HIF from this LSI. Access to HIFEICR by an external device should be performed with HIFEICR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	EIC6	0	R/W	External Interrupt Source These bits specify the source for interrupts generated by the EIR bit. These bits can be written to from both an external device and the on-chip CPU. By using these bits, fast execution of interrupt exception handling is possible. These bits are completely under software control, and their values have no effect on the operation of this LSI.
6	EIC5	0	R/W	
5	EIC4	0	R/W	
4	EIC3	0	R/W	
3	EIC2	0	R/W	
2	EIC1	0	R/W	
1	EIC0	0	R/W	
0	EIR	0	R/W	External Interrupt Request While this bit is 1, the $\overline{\text{HIFINT}}$ pin is asserted to issue an interrupt request to an external device from this LSI.

17.4.7 HIF Address Register (HIFADR)

HIFADR is a 32-bit register which indicates the address in HIFRAM to be accessed by an external device. When using the LOCK bit setting in HIFMCR to specify consecutive access of HIFRAM, auto-increment (+4) or auto-decrement (-4) of the address, according to the AI/AD bit setting in HIFMCR, is performed automatically, and HIFADR is updated. HIFADR can be only read by the on-chip CPU. Access to HIFADR by an external device should be performed with HIFADR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	A9 to A2	All 0	R/W*	HIFRAM Address Specification These bits specify the address of HIFRAM to be accessed by an external device, with 32-bit boundary.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit can be only written to by an external device when the HIFRS pin is low. It cannot be written to by the on-chip CPU.

17.4.8 HIF Data Register (HIFDATA)

HIFDATA is a 32-bit register used to hold data to be written to HIFRAM and data read from HIFRAM for external device accesses. If HIFDATA is not used when accessing HIFRAM, it can be used for data transfer between an external device connected to the HIF and the on-chip CPU. HIFDATA can be read from and written to by the on-chip CPU. Access to HIFDATA by an external device should be performed with HIFDATA specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	D31 to D0	All 0	R/W	32-bit Data

17.4.9 HIF Boot Control Register (HIFBCR)

HIFBCR is a 32-bit register for exclusive control of an external device and the on-chip CPU regarding access of HIFRAM. HIFBCR can be only read by the on-chip CPU. Access to HIFBCR by an external device should be performed with HIFBCR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 1	—	All 0	R/W	AC-Bit Writing Assistance These bits should be used to write the bit pattern (H'A5) needed to set the AC bit to 1. These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
0	AC	0/1	R/W	HIFRAM Access Exclusive Control Controls accessing of HIFRAM by the on-chip CPU for the HIFRAM bank selected by the BMD and BSEL bits in HIFSCR as the bank allowed to be accessed by this LSI. 0: The on-chip CPU can perform reading/writing of HIFRAM. 1: When an HIFRAM read/write operation by the on-chip CPU occurs, the CPU enters the wait state, and execution of the instruction is halted until this bit is cleared to 0. When booted in non-HIF boot mode, the initial value of this bit is 0. When booted in HIF boot mode, the initial value of this bit is 1. After an external device writes a boot program to HIFRAM via the HIF, clearing this bit to 0 boots the on-chip CPU from HIFRAM. When 1 is written to this bit by an external device, H'A5 should be written to bits 7 to 0 to prevent erroneous writing.

17.4.10 HIFDREQ Trigger Register (HIFDTR)

HIFDTR is a 32-bit register. Writing to HIFDTR by the on-chip CPU asserts the HIFDREQ pin. HIFDTR cannot be accessed by an external device.

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R*[1]	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	DTRG	0	R/W*1 *2	HIFDREQ Trigger When 1 is written to this bit, the HIFDREQ pin is asserted according to the setting of the DMD and DPOL bits in HIFSCR. This bit is automatically cleared to 0 in synchronization with negate of the HIFDREQ pin. Though this bit can be set to 1 by the on-chip CPU, it cannot be cleared to 0. To avoid conflict between clearing of this bit by negate of the HIFDREQ pin and setting of this bit by the on-chip CPU, make sure this bit is cleared to 0 before setting this bit to 1 by the on-chip CPU.

Notes: 1. This bit cannot be accessed by an external device. It can be accessed only by the on-chip CPU.
2. Writing 0 to this bit by the on-chip CPU is ignored.

17.4.11 HIF Bank Interrupt Control Register (HIFBICR)

HIFBICR is a 32-bit register that controls HIF bank interrupts. HIFBICR cannot be accessed by an external device.

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R*1	Reserved These bits are always read as 0. The write value should always be 0.
1	BIE	0	R/W*1	Bank Interrupt Enable Enables or disables a bank interrupt request (HIFBI) issued to the on-chip CPU. 0: HIFBI disabled 1: HIFBI enabled

Bit	Bit Name	Initial Value	R/W	Description
0	BIF	0	R/W*1 *2	Bank Interrupt Request Flag While this bit is 1, a bank interrupt request (HIFBI) is issued to the on-chip CPU according to the setting of the BIE bit. In auto-increment mode (AI/AD bit in HIFMCR is 0), this bit is automatically set to 1 when an external device has completed access to the 32-bit data in the end address of HIFRAM and the $\overline{\text{HIFCS}}$ pin has been negated. In auto-decrement mode (AI/AD bit in HIFMCR is 1), this bit is automatically set to 1 when an external device has completed access to the 32-bit data in the start address of HIFRAM and the $\overline{\text{HIFCS}}$ pin has been negated. Though this bit can be cleared to 0 by the on-chip CPU, it cannot be set to 1. Make sure setting of this bit by HIFRAM access from an external device and clearing of this bit by the on-chip CPU do not conflict using software.

Notes: 1. This bit cannot be accessed by an external device. It can only be accessed by the on-chip CPU.
2. Writing 1 to this bit by the on-chip CPU is ignored.

17.5 Memory Map

Table 17.3 shows the memory map of HIFRAM.

Table 17.3 Memory Map

Classification	Start Address	End Address	Memory Size
Map from external device*[1]	H'0000	H'03FF	1 kbyte
Map from on-chip CPU*[1] *[2]	H'F84E0000	H'F84E03FF	1 kbyte

Notes: 1. Map for a single HIFRAM bank. Which bank is to be accessed by an external device or the on-chip CPU depends on the BMD and BSEL bits in HIFSCR. The mapping addresses are common between the banks.

2. Note that in HIF boot mode, bank 0 is selected, and the first 1 kbyte in each of the following address ranges are also mapped: H'00000000 to H'01FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'20000000 to H'21FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'40000000 to H'41FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'60000000 to H'61FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'80000000 to H'81FFFFFF (first-half 32 Mbytes of area 0 in the P1 area), H'A0000000 to H'A1FFFFFF (first-half 32 Mbytes of area 0 in the P2 area), and H'C0000000 to H'C1FFFFFF (first-half 32 Mbytes of area 0 in the P3 area).
If an external device modifies HIFRAM when HIFRAM is accessed from the P0, P1, or P3 area with the cache enabled, coherency may not be ensured. When the cache is enabled, accessing HIFRAM from the P2 area is recommended.
In HIF boot mode, among the first-half 32 Mbytes of each area 0, access to the areas to which HIFRAM is not mapped is inhibited.
Even in HIF boot mode, the second-half 32 Mbytes of area 0, area 3, area 4, area 5B, area 5, area 6B, and area 6 are mapped to the external memory as normally.

17.6 Interface (Basic)

Figure 17.3 shows the basic read/write sequence. HIF read is defined by the overlap period of the $\overline{\text{HIFRD}}$ low-level period and $\overline{\text{HIFCS}}$ low-level period, and HIF write is defined by the overlap period of the $\overline{\text{HIFWR}}$ low-level period and $\overline{\text{HIFCS}}$ low-level period. The HIFRS signal indicates whether this is normal access or index/status register access; low level indicates normal access and high level indicates index/status register access.

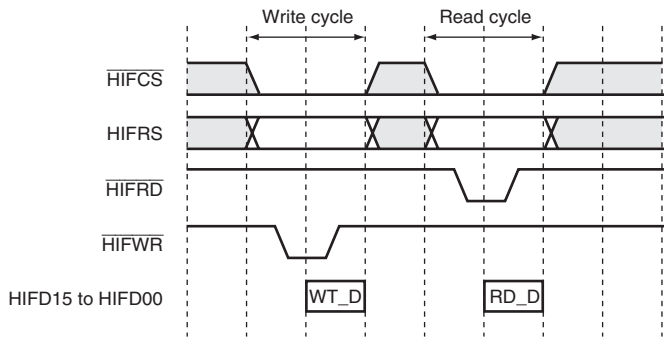


Figure 17.3 Basic Timing for HIF Interface

17.7 Interface (Details)

17.7.1 HIFIDX Write/HIFGSR Read

Writing of HIFIDX and reading of HIFGSR are shown in figure 17.4.

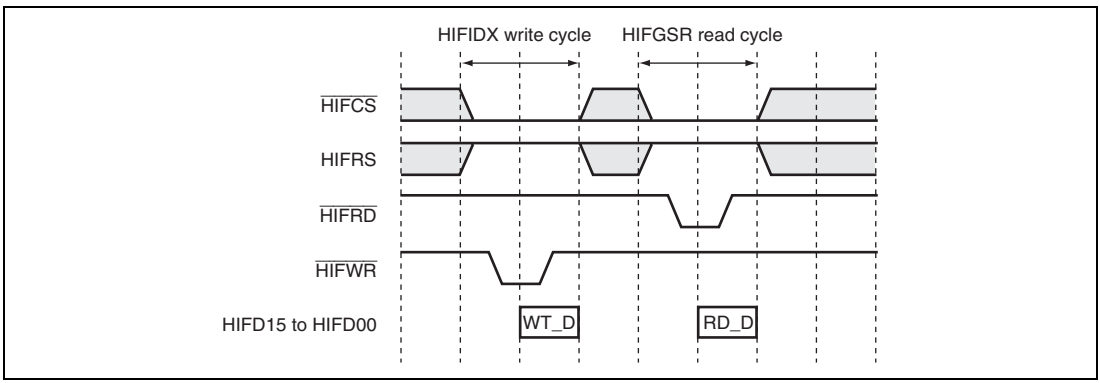


Figure 17.4 HIFIDX Write and HIFGSR Read

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

As shown in figure 17.5, in reading and writing of HIF internal registers other than HIFIDX and HIFGSR, first HIFRS is held high and HIFIDX is written to in order to select the register to be accessed and the byte location. Then HIFRS is held low, and reading or writing of the register selected by HIFIDX is performed.

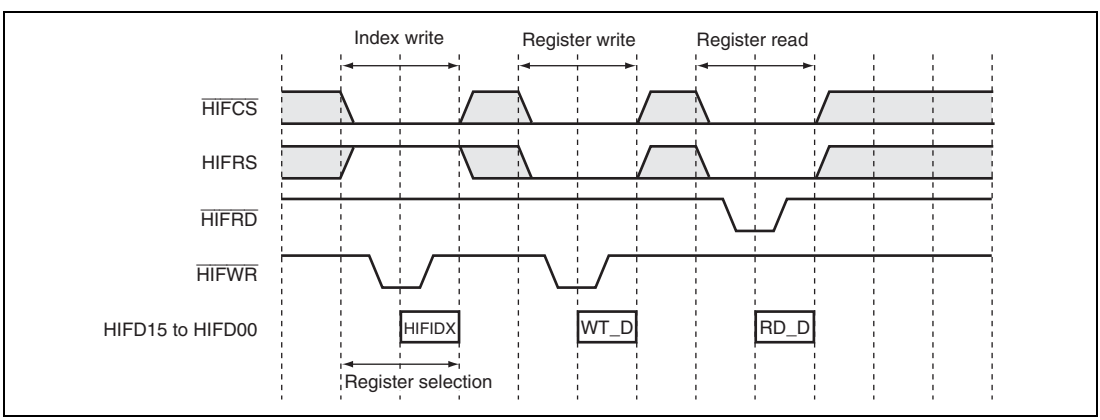


Figure 17.5 HIF Register Settings

17.7.3 Consecutive Data Writing to HIFRAM by External Device

Figure 17.6 shows the timing chart for consecutive data transfer from an external device to HIFRAM. As shown in this timing chart, by setting the start address and the data to be written first, consecutive data transfer can subsequently be performed.

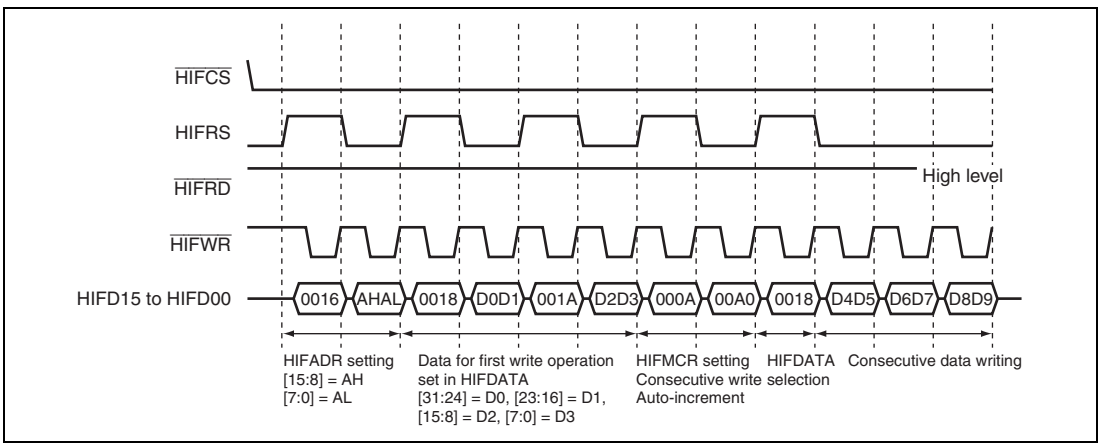


Figure 17.6 Consecutive Data Writing to HIFRAM

17.7.4 Consecutive Data Reading from HIFRAM to External Device

Figure 17.7 shows the timing chart for consecutive data reading from HIFRAM to an external device. As this timing chart indicates, by setting the start address, data can subsequently be read out consecutively.

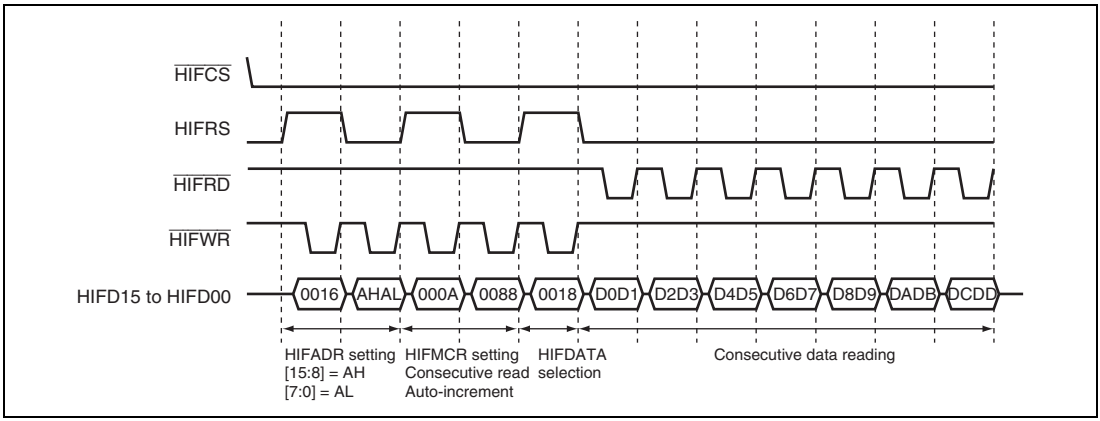


Figure 17.7 Consecutive Data Reading from HIFRAM

17.8 External DMAC Interface

Figures 17.8 to 17.11 show the HIFDREQ output timing. The start of the HIFDREQ assert synchronizes with the DTRG bit in HIFDTR being set to 1. The HIFDREQ negate timing and assert level are determined by the DMD and DPOL bits in HIFSCR, respectively.

When the external DMAC is specified to detect low level of the HIFDREQ signal, set DMD = 0 and DPOL = 0. After writing 1 to the DTRG bit, the HIFDREQ signal remains low until low level is detected for both the $\overline{\text{HIFCS}}$ and HIFRS signals.

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

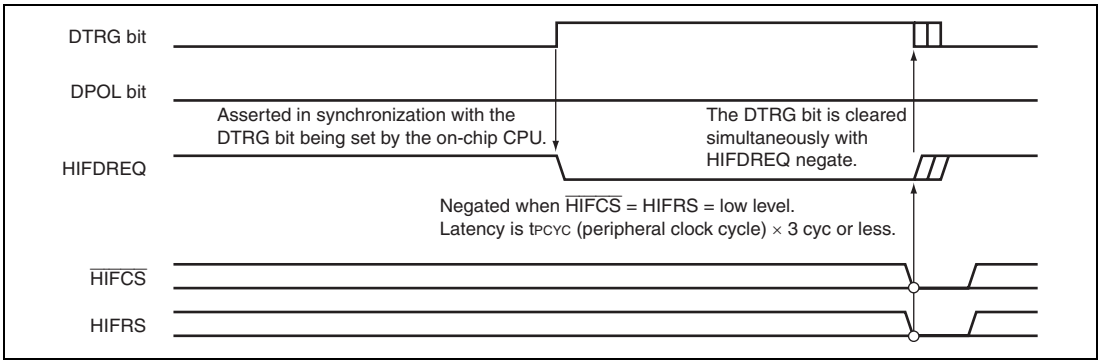


Figure 17.8 HIFDREQ Timing (When DMD = 0 and DPOL = 0)

When the external DMAC is specified to detect high level of the HIFDREQ signal, set DMD = 0 and DPOL = 1. At the time the DPOL bit is set to 1, HIFDREQ becomes low. Then after writing 1 to the DTRG bit, HIFDREQ remains high until low level is detected for both the $\overline{\text{HIFCS}}$ and HIFRS signals.

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

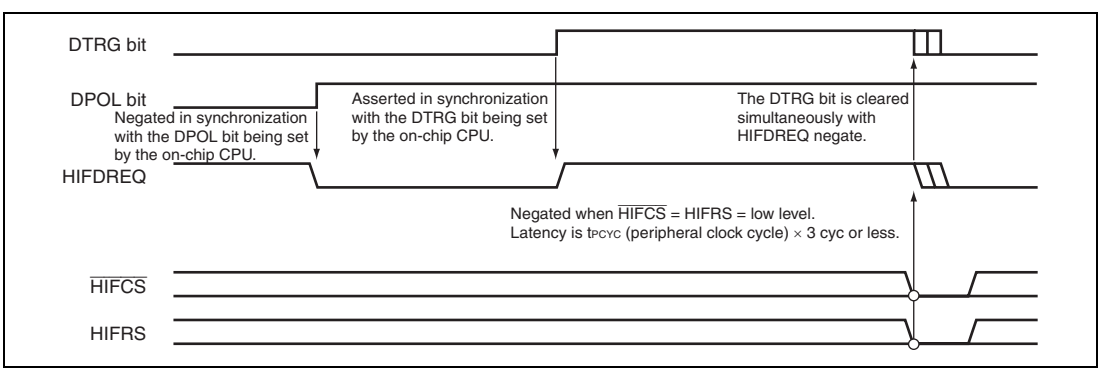


Figure 17.9 HIFDREQ Timing (When DMD = 0 and DPOL = 1)

When the external DMAC is specified to detect the falling edge of the HIFDREQ signal, set DMD = 1 and DPOL = 0. After writing 1 to the DTRG bit, a low pulse of 32 peripheral clock cycles is generated at the HIFDREQ pin.

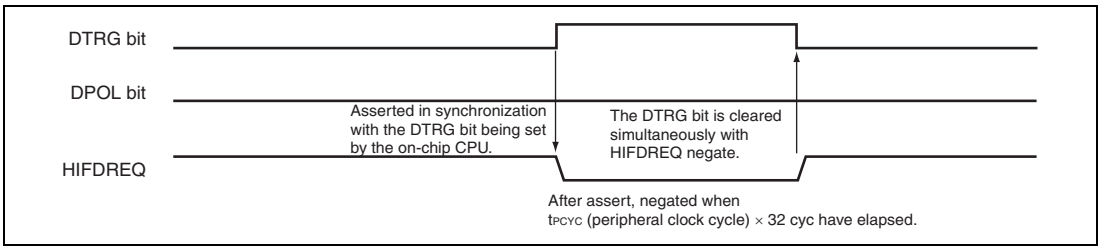


Figure 17.10 HIFDREQ Timing (When DMD = 1 and DPOL = 0)

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

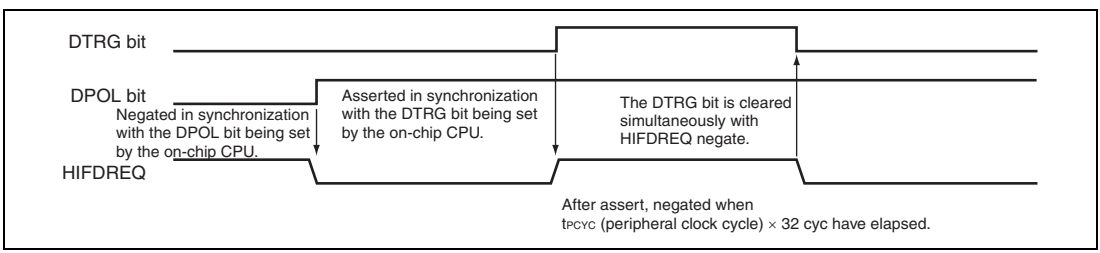


Figure 17.11 HIFDREQ Timing (When DMD = 1 and DPOL = 1)

When the external DMAC supports intermittent operating mode (block transfer mode), efficient data transfer can be implemented by using the HIFRAM consecutive access and bank functions.

Table 17.4 Consecutive Write Procedure to HIFRAM by External DMAC

	External Device			This LSI		
No.	**CPU**		**DMAC**	**HIF**		**CPU**
1	HIF initial setting					HIF initial setting
2	DMAC initial setting					
3	Set HIFADR to HIFRAM end address – 8					
4	Select HIFDATA and write dummy data (4 bytes) to HIFDATA					
5	Set HIFRAM consecutive write with address increment in HIFMCR					
6	Select HIFDATA and write dummy data (4 bytes) to HIFDATA	→		→ HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
7			Activate DMAC	← Assert HIFDREQ	←	Set DTRG bit to 1
8			Consecutive data write to bank 1 in HIFRAM			
9			Write to end address of bank 1 in HIFRAM completes and operation halts	→ HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 0 and on-chip CPU accesses bank 1)
10			Re-activate DMAC	← Assert HIFDREQ	←	Set DTRG bit to 1
11			Consecutive data write to bank 0 in HIFRAM			Read data from bank 1 in HIFRAM

	External Device			This LSI		
No.	**CPU**	**DMAC**		**HIF**		**CPU**
12		Write to end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
13		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1
Hereafter No. 11 to 13 are repeated. When a register other than HIFDATA is accessed (except that HIFGSR read with HIFRS = low), HIFRAM consecutive write is interrupted, and No. 3 to 6 need to be done again.						

Table 17.5 Consecutive Read Procedure from HIFRAM by External DMAC

	External Device			This LSI		
No.	**CPU**	**DMAC**		**HIF**		**CPU**
1	HIF initial setting					HIF initial setting
2	DMAC initial setting					
3	Set HIFADR to HIFRAM start address					
4	Set HIFRAM consecutive read with address increment in HIFMCR					
5	Select HIFDATA					
6						Write data to bank 1 in HIFRAM
7						After writing data to end address of bank 1 in HIFRAM, perform HIFRAM bank switching (external device accesses bank 1 and on-chip CPU accesses bank 0)
8		Activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1

	External Device			This LSI		
No.	**CPU**	**DMAC**		**HIF**		**CPU**
9		Consecutive data read from bank 1 in HIFRAM				Write data to bank 0 in HIFRAM
10		Read from end address of bank 1 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 0 and on-chip CPU accesses bank 1)
11		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1
12		Consecutive data read from bank 0 in HIFRAM				Write data to bank 1 in HIFRAM
13		Read from end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
14		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1

Hereafter No. 12 to 14 are repeated. When a register other than HIFDATA is accessed (except that HIFGSR read with HIFRS = low), HIFRAM consecutive read is interrupted, and No. 3 to 5 need to be done again.

17.9 Alignment Control

Tables 17.6 and 17.7 show the alignment control when an external device accesses the HIFDATA register, and the HIF registers other than the HIFDATA register, respectively.

Table 17.6 HIFDATA Register Alignment for Access by an External Device

Data in HIFDATA	WBSWP Bit	BO Bit	BYTE[1:0] Bits	Alignment in HIFD[15:0] Pins
H'76543210	0	0	B'00	H'7654
			B'10	H'3210
		1	B'00	H'3210
			B'10	H'7654
	1	0	B'00	H'1032
			B'10	H'5476
		1	B'00	H'5476
			B'10	H'1032

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

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

17.10 Interface When External Device Power is Cut Off

When the power supply of an external device interfacing with the HIF is cut off, intermediate levels may be applied to the HIF input pins or the HIF output pins may drive an external device not powered, thus causing the device to be damaged. The HIFEBL pin is provided to prevent this from happening. The system power monitor block controls the HIFEBL pin in synchronization with the cutoff of the external device power so that all HIF pins can be set to the high-impedance state. Figure 17.12 shows an image of high-impedance control of the HIF pins. Table 17.8 lists the input/output control for the HIF pins.

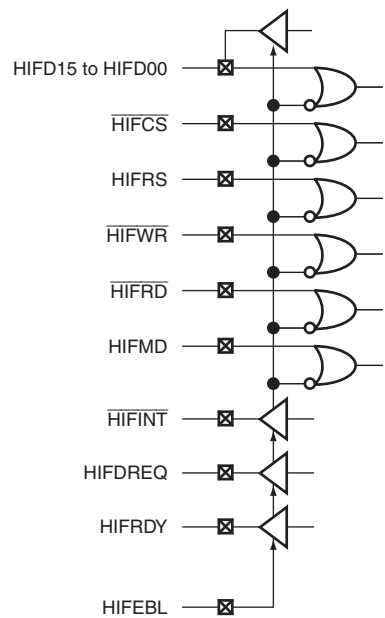


Figure 17.12 Image of High-Impedance Control of HIF Pins by HIFEBL Pin

Table 17.8 Input/Output Control for HIF Pins

LSI Status	Reset State by $\overline{\text{RES}}$ Pin			Reset Canceled by $\overline{\text{RES}}$ Pin		
HIFMD input level	**High (Boot setting)**		**Low (Non-boot setting)**	**High (After the reset canceled by boot setting)**		**Low (After the reset canceled by non-boot setting)**
HIFEBL input level	**Low**	**High**	**The HIFEBL pin is a general input port and the HIF is not controlled by the signal input on this pin.**	**Low**	**High**	**General input port at the initial state *1**
HIFRDY output control	Output buffer: On (Low output)	Output buffer: On (Low output)	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
$\overline{\text{HIFINT}}$ output control	Output buffer: Off	Output buffer: Off	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
HIFDREQ output control	Output buffer: Off	Output buffer: Off	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
HIFD 15 to HIFD0 I/O control	I/O buffer: Off	I/O buffer: Off	General input port	I/O buffer: Off	I/O buffer controlled according to states of $\overline{\text{HIFCS}}$, $\overline{\text{HIFWR}}$, and $\overline{\text{HIFRD}}$	General input port at the initial state*2
$\overline{\text{HIFCS}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2
HIFRS input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2

LSI Status	**Reset State by $\overline{\text{RES}}$ Pin**			**Reset Canceled by $\overline{\text{RES}}$ Pin**		
HIFMD input level	**High (Boot setting)**		**Low (Non-boot setting)**	**High (After the reset canceled by boot setting)**		**Low (After the reset canceled by non-boot setting)**
HIFEBL input level	**Low**	**High**	**The HIFEBL pin is a general input port and the HIF is not controlled by the signal input on this pin.**	**Low**	**High**	**General input port at the initial state *1**
$\overline{\text{HIFWR}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2
$\overline{\text{HIFRD}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2

Notes: 1. The pin also functions as an HIFEBL pin by setting the PFC registers.

2. The pin also functions as an HIF pin by setting the PFC registers.

When the HIF pin function is selected for the HIFEBL pin and this pin by setting the PFC registers, the input and/or output buffers are controlled according to the HIFEBL pin state.

When the HIF pin function is not selected for the HIFEBL pin and is selected for this pin by setting the PFC registers, the input and/or output buffers are always turned off. This setting is prohibited.

Section 18 Pin Function Controller (PFC)

The pin function controller (PFC) consists of registers that select multiplexed pin functions and input/output directions. Tables 18.1 to 18.5 show the multiplexed pins in this LSI. Table 18.6 shows the pin functions in each operating mode.

Table 18.1 List of Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA16 input/output (port)	A16 output (BSC)	—	—
	PA17 input/output (port)	A17 output (BSC)	—	—
	PA18 input/output (port)	A18 output (BSC)	—	—
	PA19 input/output (port)	A19 output (BSC)	—	—
	PA20 input/output (port)	A20 output (BSC)	—	—
	PA21 input/output (port)	A21 output (BSC)	SCK_SIO0 input/output (SIOF)	—
	PA22 input/output (port)	A22 output (BSC)	SIOMCLK0 input (SIOF)	—
	PA23 input/output (port)	A23 output (BSC)	RXD_SIO0 input (SIOF)	—
	PA24 input/output (port)	A24 output (BSC)	TXD_SIO0 output (SIOF)	—
	PA25 input/output (port)	A25 output (BSC)	SIOFSYNC0 input/output (SIOF)	—

Table 18.2 List of Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)			Function 3 (Related Module)	Function 4 (Related Module)
B	PB00 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)			—	—
	PB01 input/output (port)			$\overline{\text{IOIS16}}$ input (BSC)	—	—
	PB02 input/output (port)		CKE output (BSC)		—	—

Port	Function 1 (Related Module)	Function 2 (Related Module)			Function 3 (Related Module)	Function 4 (Related Module)
B	PB03 input/output (port)		$\overline{\text{CAS}}$ output (BSC)		—	—
	PB04 input/output (port)		$\overline{\text{RAS}}$ output (BSC)		—	—
	PB05 input/output (port)	$\overline{\text{WE2(BE2)}}$ output (BSC)	DQMUL output (BSC)	$\overline{\text{ICIORD}}$ output (BSC)	—	—
	PB06 input/output (port)	$\overline{\text{WE3(BE3)}}$ output (BSC)	DQMUU output (BSC)	$\overline{\text{ICIOWR}}$ output (BSC)	—	—
	PB07 input/output (port)			$\overline{\text{CE2B}}$ output (BSC)	—	—
	PB08 input/output (port)	$\overline{\text{CS6B}}$ output (BSC)		$\overline{\text{CE1B}}$ output (BSC)	—	—
	PB09 input/output (port)			$\overline{\text{CE2A}}$ output (BSC)	—	—
	PB10 input/output (port)	$\overline{\text{CS5B}}$ output (BSC)		$\overline{\text{CE1A}}$ output (BSC)	—	—
	PB11 input/output (port)	$\overline{\text{CS4}}$ output (BSC)			—	—
	PB12 input/output (port)	$\overline{\text{CS3}}$ output (BSC)			—	—
	PB13 input/output (port)	$\overline{\text{BS}}$ output (BSC)			—	—

Table 18.3 List of Multiplexed Pins (Port C)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
C	PC00 input/output (port)	MII_RXD0 input (EtherC)	—	—
	PC01 input/output (port)	MII_RXD1 input (EtherC)	—	—
	PC02 input/output (port)	MII_RXD2 input (EtherC)	—	—
	PC03 input/output (port)	MII_RXD3 input (EtherC)	—	—
	PC04 input/output (port)	MII_TXD0 output (EtherC)	—	$\overline{\text{SPEED100}}$ output (PHY)
	PC05 input/output (port)	MII_TXD1 output (EtherC)	—	$\overline{\text{LINK}}$ output (PHY)
	PC06 input/output (port)	MII_TXD2 output (EtherC)	—	$\overline{\text{CRS}}$ output (PHY)
	PC07 input/output (port)	MII_TXD3 output (EtherC)	—	$\overline{\text{DUPLEX}}$ output (PHY)
	PC08 input/output (port)	RX_DV input (EtherC)	—	—
	PC09 input/output (port)	RX_ER input (EtherC)	—	—
	PC10 input/output (port)	RX_CLK input (EtherC)	—	—
	PC11 input/output (port)	TX_ER output (EtherC)	—	—
	PC12 input/output (port)	TX_EN output (EtherC)	—	—
	PC13 input/output (port)	TX_CLK input (EtherC)	—	—
	PC14 input/output (port)	COL input (EtherC)	—	—
	PC15 input/output (port)	CRS input (EtherC)	—	—
	PC16 input/output (port)	MDIO input/output (EtherC)	—	—
	PC17 input/output (port)	MDC output (EtherC)	—	—
	PC18 input/output (port)	LNKSTA input (EtherC)	—	—
	PC19 input/output (port)	EXOUT output (EtherC)	—	—
	PC20 input/output (port)	WOL output (EtherC)	—	—

Table 18.4 List of Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD0 input/output (port)	IRQ0 input (INTC)	—	TEND0 output (DMAC)
	PD1 input/output (port)	IRQ1 input (INTC)	—	TEND1 output (DMAC)
	PD2 input/output (port)	IRQ2 input (INTC)	TxD1 output (SCIF)	DREQ0 input (DMAC)
	PD3 input/output (port)	IRQ3 input (INTC)	RxD1 input (SCIF)	DACK0 output (DMAC)
	PD4 input/output (port)	IRQ4 input (INTC)	SCK1 input/output (SCIF)	—
	PD5 input/output (port)	IRQ5 input (INTC)	TxD2 output (SCIF)	DREQ1 input (DMAC)
	PD6 input/output (port)	IRQ6 input (INTC)	RxD2 input (SCIF)	DACK1 output (DMAC)
	PD7 input/output (port)	IRQ7 input (INTC)	SCK2 input/output (SCIF)	—

Table 18.5 List of Multiplexed Pins (Port E)

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

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE13 input/output (port)	HIFD04 input/output (HIF)	—	D20 input/output (BSC)
	PE14 input/output (port)	HIFD05 input/output (HIF)	—	D21 input/output (BSC)
	PE15 input/output (port)	HIFD06 input/output (HIF)	TxD0 output (SCIF)	D22 input/output (BSC)
	PE16 input/output (port)	HIFD07 input/output (HIF)	RxD0 input (SCIF)	D23 input/output (BSC)
	PE17 input/output (port)	HIFD08 input/output (HIF)	SCK0 input/output (SCIF)	D24 input/output (BSC)
	PE18 input/output (port)	HIFD09 input/output (HIF)	TxD1 output (SCIF)	D25 input/output (BSC)
	PE19 input/output (port)	HIFD10 input/output (HIF)	RxD1 input (SCIF)	D26 input/output (BSC)
	PE20 input/output (port)	HIFD11 input/output (HIF)	SCK1 input/output (SCIF)	D27 input/output (BSC)
	PE21 input/output (port)	HIFD12 input/output (HIF)	RTS0 output (SCIF)	D28 input/output (BSC)
	PE22 input/output (port)	HIFD13 input/output (HIF)	CTS0 input (SCIF)	D29 input/output (BSC)
	PE23 input/output (port)	HIFD14 input/output (HIF)	RTS1 output (SCIF)	D30 input/output (BSC)
	PE24 input/output (port)	HIFD15 input/output (HIF)	CTS1 input (SCIF)	D31 input/output (BSC)

Table 18.6 Pin Functions in Each Operating Mode

Pin No.	Not HIF Boot Mode Initial Function	Not HIF Boot Mode Function Settable by PFC	HIF Boot Mode Initial Function	HIF Boot Mode Function Settable by PFC
C13	A00	—	A00	—
A14	A01	—	A01	—
B13	A02	—	A02	—
A13	A03	—	A03	—
C12	A04	—	A04	—
B12	A05	—	A05	—
D11	A06	—	A06	—
A12	A07	—	A07	—
C11	A08	—	A08	—
B11	A09	—	A09	—
D10	A10	—	A10	—
A11	A11	—	A11	—
C10	A12	—	A12	—
A10	A13	—	A13	—
D9	A14	—	A14	—
B10	A15	—	A15	—
A5	PA16	PA16/A16	PA16	PA16/A16
B5	PA17	PA17/A17	PA17	PA17/A17
A4	PA18	PA18/A18	PA18	PA18/A18
D5	PA19	PA19/A19	PA19	PA19/A19
B4	PA20	PA20/A20	PA20	PA20/A20
C4	PA21	PA21/A21/SCK_SIO0	PA21	PA21/A21/SCK_SIO0
A3	PA22	PA22/A22/SIOMCLK0	PA22	PA22/A22/SIOMCLK0
D4	PA23	PA23/A23/RXD_SIO0	PA23	PA23/A23/RXD_SIO0
B3	PA24	PA24/A24/TXD_SIO0	PA24	PA24/A24/TXD_SIO0
A2	PA25	PA25/A25/SIOFSYNC0	PA25	PA25/A25/SIOFSYNC0
B8	PB00	PB00/$\overline{\text{WAIT}}$	PB00	PB00/$\overline{\text{WAIT}}$
D6	PB01	PB01/$\overline{\text{IOIS16}}$	PB01	PB01/$\overline{\text{IOIS16}}$
C15	PB02	PB02/CKE	PB02	PB02/CKE

Pin No.	Not HIF Boot Mode Initial Function	Not HIF Boot Mode Function Settable by PFC	HIF Boot Mode Initial Function	HIF Boot Mode Function Settable by PFC
D12	PB03	PB03/$\overline{\text{CAS}}$	PB03	PB03/$\overline{\text{CAS}}$
C14	PB04	PB04/$\overline{\text{RAS}}$	PB04	PB04/$\overline{\text{RAS}}$
D15	($\overline{\text{WE0}}$/DQMLL)	—	($\overline{\text{WE0}}$/DQMLL)	—
D14	($\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$)	—	($\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$)	—
D7	PB05	PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/ $\overline{\text{ICIORD}}$	PB05	PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/ $\overline{\text{ICIORD}}$
C7	PB06	PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/ $\overline{\text{ICIOWR}}$	PB06	PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/ $\overline{\text{ICIOWR}}$
A8	$\overline{\text{RD}}$	—	$\overline{\text{RD}}$	—
D13	RDWR	—	RDWR	—
B6	PB07	PB07/$\overline{\text{CE2B}}$	PB07	PB07/$\overline{\text{CE2B}}$
C5	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$)	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$)
A6	PB09	PB09/$\overline{\text{CE2A}}$	PB09	PB09/$\overline{\text{CE2A}}$
C6	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)
C8	PB11	PB11/$\overline{\text{CS4}}$	PB11	PB11/$\overline{\text{CS4}}$
A15	PB12	PB12/$\overline{\text{CS3}}$	PB12	PB12/$\overline{\text{CS3}}$
D8	$\overline{\text{CS0}}$	—	$\overline{\text{CS0}}$	—
C9	PB13	PB13/$\overline{\text{BS}}$	PB13	PB13/$\overline{\text{BS}}$
R6	PC00	PC00/MII_RXD0	PC00	PC00/MII_RXD0
M7	PC01	PC01/MII_RXD1	PC01	PC01/MII_RXD1
P6	PC02	PC02/MII_RXD2	PC02	PC02/MII_RXD2
N7	PC03	PC03/MII_RXD3	PC03	PC03/MII_RXD3
P8	PC04	PC04/MII_TXD0/ $\overline{\text{SPEED100}}$	PC04	PC04/MII_TXD0/ $\overline{\text{SPEED100}}$
M9	PC05	PC05/MII_TXD1/$\overline{\text{LINK}}$	PC05	PC05/MII_TXD1/$\overline{\text{LINK}}$
R9	PC06	PC06/MII_TXD2/$\overline{\text{CRS}}$	PC06	PC06/MII_TXD2/$\overline{\text{CRS}}$
N9	PC07	PC07/MII_TXD3/$\overline{\text{DUPLEX}}$	PC07	PC07/MII_TXD3/$\overline{\text{DUPLEX}}$
N6	PC08	PC08/RX_DV	PC08	PC08/RX_DV
M6	PC09	PC09/RX_ER	PC09	PC09/RX_ER
R8	PC10	PC10/RX_CLK	PC10	PC10/RX_CLK
N8	PC11	PC11/TX_ER	PC11	PC11/TX_ER

Pin No.	Not HIF Boot Mode		HIF Boot Mode	
	Initial Function	**Function Settable by PFC**	**Initial Function**	**Function Settable by PFC**
P9	PC12	PC12/TX_EN	PC12	PC12/TX_EN
M8	PC13	PC13/TX_CLK	PC13	PC13/TX_CLK
R10	PC14	PC14/COL	PC14	PC14/COL
P1	PC15	PC15/CRS	PC15	PC15/CRS
N2	PC16	PC16/MDIO	PC16	PC16/MDIO
M4	PC17	PC17/MDC	PC17	PC17/MDC
P2	PC18	PC18/LNKSTA	PC18	PC18/LNKSTA
N11	PC19	PC19/EXOUT	PC19	PC19/EXOUT
P10	PC20	PC20/WOL	PC20	PC20/WOL
D1	PD0	PD0/IRQ0/TEND0	PD0	PD0/IRQ0/TEND0
E4	PD1	PD1/IRQ1/TEND1	PD1	PD1/IRQ1/TEND1
D2	PD2	PD2/IRQ2/TxD1/DREQ0	PD2	PD2/IRQ2/TxD1/DREQ0
D3	PD3	PD3/IRQ3/RxD1/DACK0	PD3	PD3/IRQ3/RxD1DACK0
C1	PD4	PD4/IRQ4/SCK1	PD4	PD4/IRQ4/SCK1
C2	PD5	PD5/IRQ5/TxD2/DREQ1	PD5	PD5/IRQ5/TxD2/DREQ1
C3	PD6	PD6/IRQ6/RxD2/DACK1	PD6	PD6/IRQ6/RxD2/DACK1
B2	PD7	PD7/IRQ7/SCK2	PD7	PD7/IRQ7/SCK2
N1	PE00	PE00/HIFEBL/SCK_SIO0	HIFEBL	PE00/HIFEBL/SCK_SIO0
M3	PE01	PE01/HIFRDY/SIOMCLK0	HIFRDY	PE01/HIFRDY/SIOMCLK0
M2	PE02	PE02/HIFDREQ/ RXD_SIO0	HIFDREQ	PE02/HIFDREQ/ RXD_SIO0
L4	HIFMD	PE03/HIFMD	HIFMD	PE03/HIFMD
M1	PE04	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0	$\overline{\text{HIFINT}}$	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0
L2	PE05	PE05/$\overline{\text{HIFRD}}$	$\overline{\text{HIFRD}}$	PE05/$\overline{\text{HIFRD}}$
L1	PE06	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0	$\overline{\text{HIFWR}}$	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0
L3	PE07	PE07/HIFRS	HIFRS	PE07/HIFRS
E3	PE08	PE08/$\overline{\text{HIFCS}}$	$\overline{\text{HIFCS}}$	PE08/$\overline{\text{HIFCS}}$
K3	PE09	PE09/HIFD00/D16	HIFD00	PE09/HIFD00/D16
K4	PE10	PE10/HIFD01/D17	HIFD01	PE10/HIFD01/D17
J2	PE11	PE11/HIFD02/D18	HIFD02	PE11/HIFD02/D18
J3	PE12	PE12/HIFD03/D19	HIFD03	PE12/HIFD03/D19

Pin No.	Not HIF Boot Mode Initial Function	Not HIF Boot Mode Function Settable by PFC	HIF Boot Mode Initial Function	HIF Boot Mode Function Settable by PFC
J1	PE13	PE13/HIFD04/D20	HIFD04	PE13/HIFD04//D20
J4	PE14	PE14/HIFD05/D21	HIFD05	PE14/HIFD05/D21
H2	PE15	PE15/HIFD06/TxD0/D22	HIFD06	PE15/HIFD06/TxD0/D22
H1	PE16	PE16/HIFD07/RxD0/D23	HIFD07	PE16/HIFD07/RxD0/D23
G2	PE17	PE17/HIFD08/SCK0/D24	HIFD08	PE17/HIFD08/SCK0/D24
G1	PE18	PE18/HIFD09/TxD1/D25	HIFD09	PE18/HIFD09/TxD1/D25
G3	PE19	PE19/HIFD10/RxD1/D26	HIFD10	PE19/HIFD10/RxD1/D26
F2	PE20	PE20/HIFD11/SCK1/D27	HIFD11	PE20/HIFD11/SCK1/D27
G4	PE21	PE21/HIFD12/RTS0/D28	HIFD12	PE21/HIFD12/RTS0/D28
F1	PE22	PE22/HIFD13/CTS0/D29	HIFD13	PE22/HIFD13/CTS0/D29
F3	PE23	PE23/HIFD14/RTS1/D30	HIFD14	PE23/HIFD14/RTS1/D30
F4	PE24	PE24/HIFD15/CTS1/D31	HIFD15	PE24/HIFD15/CTS1/D31
K14	D00	—	D00	—
J13	D01	—	D01	—
J15	D02	—	D02	—
H12	D03	—	D03	—
J14	D04	—	D04	—
H13	D05	—	D05	—
G12	D06	—	D06	—
G15	D07	—	D07	—
E15	D08	—	D08	—
E14	D09	—	D09	—
F14	D10	—	D10	—
F13	D11	—	D11	—
F15	D12	—	D12	—
F12	D13	—	D13	—
G14	D14	—	D14	—
G13	D15	—	D15	—
M14	$\overline{\text{TRST}}$ input	—	$\overline{\text{TRST}}$ input	—
N12	TDO output	—	TDO output	—

Pin No.	Not HIF Boot Mode		HIF Boot Mode	
	Initial Function	**Function Settable by PFC**	**Initial Function**	**Function Settable by PFC**
M12	TDI input	—	TDI input	—
M13	TMS input	—	TMS input	—
P12	TCK input	—	TCK input	—
R13	EXTAL input	—	EXTAL input	—
R14	XTAL output	—	XTAL output	—
K15	CKIO output	—	CKIO output	—
P11	CK_PHY input	—	CK_PHY input	—
L13	$\overline{\text{ASEMD}}$ input	—	$\overline{\text{ASEMD}}$ input	—
L14	$\overline{\text{TESTMD}}$ input	—	$\overline{\text{TESTMD}}$ input	—
R12	MD3 input	—	MD3 input	—
J12	MD2 input	—	MD2 input	—
L15	MD1 input	—	MD1 input	—
N13	MD0 input	—	MD0 input	—
M15	$\overline{\text{RES}}$ input	—	$\overline{\text{RES}}$ input	—
L12	NMI input	—	NMI input	—
M11	MD5 input	—	MD5 input	—
R11	$\overline{\text{TESTOUT}}$ output	—	$\overline{\text{TESTOUT}}$ output	—

18.1 Register Descriptions

The PFC has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Port A IO register H (PAIORH)
- Port A control register H1 (PACRH1)
- Port A control register H2 (PACRH2)
- Port B IO register L (PBIORL)
- Port B control register L1 (PBCRL1)
- Port B control register L2 (PBCRL2)
- Port C IO register H (PCIORH)
- Port C IO register L (PCIORL)
- Port C control register H2 (PCCRH2)
- Port C control register L1 (PCCRL1)
- Port C control register L2 (PCCRL2)
- Port D IO register L (PDIORL)
- Port D control register L2 (PDCRL2)
- Port E IO register H (PEIORH)
- Port E IO register L (PEIORL)
- Port E control register H1 (PECRH1)
- Port E control register H2 (PECRH2)
- Port E control register L1 (PECRL1)
- Port E control register L2 (PECRL2)

18.1.1 Port A IO Register H (PAIORH)

PAIORH is a 16-bit readable/writable register that selects the input/output directions of the port A pins. Bits PA25IOR to PA16IOR correspond to pins PA25 to PA16 (the pin name abbreviations for multiplexed functions are omitted). PAIORH is enabled when a port A pin functions as a general input/output (PA25 to PA16), otherwise, disabled.

Setting a bit in PAIORH to 1 makes the corresponding pin function as an output and clearing a bit in PAIORH to 0 makes the pin function as an input.

Bits 15 to 10 in PAIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PAIORH is H'0000.

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

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

- PACRH1

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

Bit	Bit Name	Initial Value	R/W	Description
1	PA24MD1	0	R/W	PA24 Mode
0	PA24MD0	0	R/W	Select the function of pin PA24/A24/TXD_SIO0. 00: PA24 input/output (port) 01: A24 output (BSC) 10: TXD_SIO0 output (BSC) 11: Setting prohibited

- PACRH2

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

Bit	Bit Name	Initial Value	R/W	Description
8	PA20MD0	0	R/W	PA20 Mode Selects the function of pin PA20/A20. 0: PA20 input/output (port) 1: A20 output (BSC)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PA19MD0	0	R/W	PA19 Mode Selects the function of pin PA19/A19. 0: PA19 input/output (port) 1: A19 output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PA18MD0	0	R/W	PA18 Mode Selects the function of pin PA18/A18. 0: PA18 input/output (port) 1: A18 output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PA17MD0	0	R/W	PA17 Mode Selects the function of pin PA17/A17. 0: PA17 input/output (port) 1: A17 output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PA16MD0	0	R/W	PA16 Mode Selects the function of pin PA16/A16. 0: PA16 input/output (port) 1: A16 output (BSC)

18.1.3 Port B IO Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that selects the input/output directions of the port B pins. Bits PB13IOR to PB0IOR correspond to pins PB13 to PB00 (the pin name abbreviations for multiplexed functions are omitted). PBIORL is enabled when a port B pin functions as a general input/output (PB13 to PB00), otherwise, disabled.

Setting a bit in PBIORL to 1 makes the corresponding pin function as an output and clearing a bit in PBIORL to 0 makes the pin function as an input.

Bits 15 and 14 in PBIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PAIBRL is H'0000.

18.1.4 Port B Control Register L1 and L2 (PBCRL1 and PBCRL2)

PBCRL1 and PBCRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port B pins.

- PBCRL1

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PB13MD0	0	R/W	PB13 Mode Selects the function of pin PB13/$\overline{\text{BS}}$. 0: PB13 input/output (port) 1: $\overline{\text{BS}}$ output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PB12MD0	0	R/W	PB12 Mode Selects the function of pin PB12/$\overline{\text{CS3}}$. 0: PB12 input/output (port) 1: $\overline{\text{CS3}}$ output (BSC)

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	PB11MD0	0	R/W	PB11 Mode Selects the function of pin PB11/$\overline{\text{CS4}}$. 0: PB11 input/output (port) 1: $\overline{\text{CS4}}$ output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PB10MD0	0	R/W	PB10 Mode Selects the function of pin PB10/$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$. 0: PB10 input/output (port) 1: $\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$ output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PB9MD0	0	R/W	PB9 Mode Selects the function of pin PB09/$\overline{\text{CE2A}}$. 0: PB09 input/output (port) 1: $\overline{\text{CE2A}}$ output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PB8MD0	0	R/W	PB8 Mode Selects the function of pin PB08/$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$. 0: PB08 input/output (port) 1: $\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$ output (BSC)

- PBCRL2

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	PB7MD0	0	R/W	PB7 Mode Selects the function of pin PB07/$\overline{\text{CE2B}}$. 0: PB07 input/output (port) 1: $\overline{\text{CE2B}}$ output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PB6MD0	0	R/W	PB6 Mode Selects the function of pin PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$. 0: PB06 input/output (port) 1: $\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PB5MD0	0	R/W	PB5 Mode Selects the function of pin PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$. 0: PB05 input/output (port) 1: $\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$ output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PB4MD0	0	R/W	PB4 Mode Selects the function of pin PB04/$\overline{\text{RAS}}$. 0: PB04 input/output (port) 1: $\overline{\text{RAS}}$ output (BSC)

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	PB3MD0	0	R/W	PB3 Mode Selects the function of pin PB03/$\overline{\text{CAS}}$. 0: PB03 input/output (port) 1: $\overline{\text{CAS}}$ output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PB2MD0	0	R/W	PB2 Mode Selects the function of pin PB02/CKE. 0: PB02 input/output (port) 1: CKE output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PB1MD0	0	R/W	PB1 Mode Selects the function of pin PB01/$\overline{\text{IOIS16}}$. 0: PB01 input/output (port) 1: $\overline{\text{IOIS16}}$ input (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PB0MD0	0	R/W	PB0 Mode Selects the function of pin PB00/$\overline{\text{WAIT}}$. 0: PB00 input/output (port) 1: $\overline{\text{WAIT}}$ input (BSC)

18.1.5 Port C IO Register H and L (PCIORH and PCIORL)

PCIORH and PCIORL are 16-bit readable/writable registers that select the input/output directions of the port C pins. Bits PC20IOR to PC0IOR correspond to pins PC20 to PC00 (the pin name abbreviations for multiplexed functions are omitted). PCIORH is enabled when a port C pin functions as a general input/output (PC20 to PC16), otherwise, disabled. PCIORL is enabled when a port C pin functions as a general input/output (PC15 to PC00), otherwise, disabled.

Setting a bit in PCIORH and PCIORL to 1 makes the corresponding pin function as an output and clearing a bit in PCIORH and PCIORL to 0 makes the pin function as an input.

Bits 15 to 5 in PCIORH are reserved. These bits are always read as 0. The write value should always be 0.

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

18.1.6 Port C Control Register H2, L1, and L2 (PCCRH2, PCCRL1, and PCCRL2)

PCCRH2, PCCRL1, and PCCRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port C pins.

- PCCRH2

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC20MD0	0	R/W	PC20 Mode Selects the function of pin PC20/WOL. 0: PC20 input/output (port) 1: WOL output (EtherC)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	PC19MD0	0	R/W	PC19 Mode Selects the function of pin PC19/EXOUT. 0: PC19 input/output (port) 1: EXOUT output (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC18MD0	0	R/W	PC18 Mode Selects the function of pin PC18/LNKSTA. 0: PC18 input/output (port) 1: LNKSTA input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC17MD0	0	R/W	PC17 Mode Selects the function of pin PC17/MDC. 0: PC17 input/output (port) 1: MDC output (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC16MD0	0	R/W	PC16 Mode Selects the function of pin PC16/MDIO. 0: PC16 input/output (port) 1: MDIO input/output (EtherC)

- PCCRL1

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	PC15MD0	0	R/W	PC15 Mode Selects the function of pin PC15/CRS. 0: PC15 input/output (port) 1: CRS input (EtherC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PC14MD0	0	R/W	PC14 Mode Selects the function of pin PC14/COL. 0: PC14 input/output (port) 1: COL input (EtherC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PC13MD0	0	R/W	PC13 Mode Selects the function of pin PC13/TX_CLK. 0: PC13 input/output (port) 1: TX_CLK input (EtherC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PC12MD0	0	R/W	PC12 Mode Selects the function of pin PC12/TX_EN. 0: PC12 input/output (port) 1: TX_EN output (EtherC)

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	PC11MD0	0	R/W	PC11 Mode Selects the function of pin PC11/TX_ER. 0: PC11 input/output (port) 1: TX_ER output (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC10MD0	0	R/W	PC10 Mode Selects the function of pin PC10/RX_CLK. 0: PC10 input/output (port) 1: RX_CLK input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC9MD0	0	R/W	PC9 Mode Selects the function of pin PC09/RX_ER. 0: PC09 input/output (port) 1: RX_ER input (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC8MD0	0	R/W	PC8 Mode Selects the function of pin PC08/RX_DV. 0: PC08 input/output (port) 1: RX_DV input (EtherC)

- PCCRL2

Bit	Bit Name	Initial Value	R/W	Description
15	PC7MD1	0	R/W	PC7 Mode
14	PC7MD0	0	R/W	Select the function of pin PC7/MII_TXD3/$\overline{\text{DUPLEX}}$. 00: PC07 input/output (port) 01: MII_TXD3 output (EtherC) 10: Setting prohibited 11: $\overline{\text{DUPLEX}}$ output (PHY)
13	PC6MD1	0	R/W	PC6 Mode
12	PC6MD0	0	R/W	Select the function of pin PC6/MII_TXD2/$\overline{\text{CRS}}$. 00: PC06 input/output (port) 01: MII_TXD2 output (EtherC) 10: Setting prohibited 11: $\overline{\text{CRS}}$ output (PHY)
11	PC5MD1	0	R/W	PC5 Mode
10	PC5MD0	0	R/W	Select the function of pin PC5/MII_TXD1/$\overline{\text{LINK}}$. 00: PC05 input/output (port) 01: MII_TXD1 output (EtherC) 10: Setting prohibited 11: $\overline{\text{LINK}}$ output (PHY)
9	PC4MD1	0	R/W	PC4 Mode
8	PC4MD0	0	R/W	Select the function of pin PC4/MII_TXD0/$\overline{\text{SPEED100}}$. 00: PC04 input/output (port) 01: MII_TXD0 output (EtherC) 10: Setting prohibited 11: $\overline{\text{SPEED100}}$ output (PHY)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	PC3MD0	0	R/W	PC3 Mode Selects the function of pin PC03/MII_RXD3. 0: PC03 input/output (port) 1: MII_RXD3 input (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC2MD0	0	R/W	PC2 Mode Selects the function of pin PC02/MII_RXD2. 0: PC02 input/output (port) 1: MII_RXD2 input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC1MD0	0	R/W	PC1 Mode Selects the function of pin PC01/MII_RXD1. 0: PC01 input/output (port) 1: MII_RXD1 input (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC0MD0	0	R/W	PC0 Mode Selects the function of pin PC00/MII_RXD0. 0: PC00 input/output (port) 1: MII_RXD0 input (EtherC)

18.1.7 Port D IO Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that selects the input/output directions of the port D pins. Bits PD7IOR to PD0IOR correspond to pins PD7 to PD0 (the pin name abbreviations for multiplexed functions are omitted). PDIORL is enabled when a port C pin functions as a general input/output (PD7 to PD0), otherwise, disabled.

Setting a bit in PDIORL to 1 makes the corresponding pin function as an output and clearing a bit in PDIORL to 0 makes the pin function as an input.

Bits 15 to 8 in PDIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PDIORL is H'0000.

18.1.8 Port D Control Register L2 (PDCRL2)

PDCRL2 is a 16-bit readable/writable register that selects the pin functions for the multiplexed port B pins.

- PDCRL2

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	0	R/W	PD7 Mode
14	PD7MD0	0	R/W	Select the function of pin PD7/IRQ7/SCK2. 00: PD7 input/output (port) 01: IRQ7 input (INTC) 10: SCK2 input/output (SCIF) 11: Setting prohibited
13	PD6MD1	0	R/W	PD6 Mode
12	PD6MD0	0	R/W	Select the function of pin PD6/IRQ6/RxD2/DACK1. 00: PD6 input/output (port) 01: IRQ6 input (INTC) 10: RxD2 input (SCIF) 11: DACK1 output (DMAC)
11	PD5MD1	0	R/W	PD5 Mode
10	PD5MD0	0	R/W	Select the function of pin PD5/IRQ5/TxD2/DREQ1. 00: PD5 input/output (port) 01: IRQ5 input (INTC) 10: TxD2 output (SCIF) 11: DREQ1 input (DMAC)

Bit	Bit Name	Initial Value	R/W	Description
9	PD4MD1	0	R/W	PD4 Mode
8	PD4MD0	0	R/W	Select the function of pin PD4/IRQ4/SCK1. 00: PD4 input/output (port) 01: IRQ4 input (INTC) 10: SCK1 input/output (SCIF) 11: Setting prohibited
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	Select the function of pin PD3/IRQ3/RxD1/DACK0. 00: PD3 input/output (port) 01: IRQ3 input (INTC) 10: RxD1 input (SCIF) 11: DACK0 output (DMAC)
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	Select the function of pin PD2/IRQ2/TxD1/DREQ0. 00: PD2 input/output (port) 01: IRQ2 input (INTC) 10: TxD1 output (SCIF) 11: DREQ0 input (DMAC)
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	Select the function of pin PD1/IRQ1/TEND1. 00: PD1 input/output (port) 01: IRQ1 input (INTC) 10: Setting prohibited 11: TEND1 output (DMAC)
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	Select the function of pin PD0/IRQ0/TEND0. 00: PD0 input/output (port) 01: IRQ0 input (INTC) 10: Setting prohibited 11: TEND0 output (DMAC)

18.1.9 Port E IO Register H and L (PEIORH and PEIORL)

PEIORH and PEIORL are 16-bit readable/writable registers that select the input/output directions of the port E pins. Bits PE24IOR to PE0IOR correspond to pins PE24 to PE00 (the pin name abbreviations for multiplexed functions are omitted). PEIORH is enabled when a port E pin functions as a general input/output (PE24 to PE16), otherwise, disabled. PEIORL is enabled when a port E pin functions as a general input/output (PE15 to PE00), otherwise, disabled.

Setting a bit in PEIORH and PEIORL to 1 makes the corresponding pin function as an output and clearing a bit in PEIORH and PEIORL to 0 makes the pin function as an input.

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

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

18.1.10 Port E Control Register H1, H2, L1, and L2 (PECRH1, PECRH2, PECRL1, and PECRL2)

PECRH1, PECRH2, PECRL1, and PECRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port E pins.

- PECRH1

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PE24MD1	0	R/W	PE24 Mode
0	PE24MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE24/HIFD15/CTS1/D31. 00: PE24 input/output (port) 01: HIFD15 input/output (HIF) 10: CTS1 input (SCIF) 11: D31 input/output (BSC)

- PECRH2

Bit	Bit Name	Initial Value	R/W	Description
15	PE23MD1	0	R/W	PE23 Mode
14	PE23MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE23/HIFD14/RTS1/D30. 00: PE23 input/output (port) 01: HIFD14 input/output (HIF) 10: RTS1 input (SCIF) 11: D30 input/output (BSC)
13	PE22MD1	0	R/W	PE22 Mode
12	PE22MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE22/HIFD13/CTS0/D29. 00: PE22 input/output (port) 01: HIFD13 input/output (HIF) 10: CTS0 input (SCIF) 11: D29 input/output (BSC)
11	PE21MD1	0	R/W	PE21 Mode
10	PE21MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE21/HIFD12/RTS0/D28. 00: PE21 input/output (port) 01: HIFD12 input/output (HIF) 10: RTS0 output (SCIF) 11: D28 input/output (BSC)

Bit	Bit Name	Initial Value	R/W	Description
9	PE20MD1	0	R/W	PE20 Mode
8	PE20MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE20/HIFD11/SCK1/D27. 00: PE20 input/output (port) 01: HIFD11 input/output (HIF) 10: SCK1 input/output (SCIF) 11: D27 input/output (BSC)
7	PE19MD1	0	R/W	PE19 Mode
6	PE19MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE19/HIFD10/RxD1/D26. 00: PE19 input/output (port) 01: HIFD10 input/output (HIF) 10: RxD1 output (SCIF) 11: D26 input/output (BSC)
5	PE18MD1	0	R/W	PE18 Mode
4	PE18MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE18/HIFD09/TxD1/D25. 00: PE18 input/output (port) 01: HIFD09 input/output (HIF) 10: TxD1 output (SCIF) 11: D25 input/output (BSC)
3	PE17MD1	0	R/W	PE17 Mode
2	PE17MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE17/HIFD08/SCK0/D24. 00: PE17 input/output (port) 01: HIFD08 input/output (HIF) 10: SCK0 input/output (SCIF) 11: D24 input/output (BSC)

Bit	Bit Name	Initial Value	R/W	Description
1	PE16MD1	0	R/W	PE16 Mode
0	PE16MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE16/HIFD07/RxD0/D23. 00: PE16 input/output (port) 01: HIFD07 input/output (HIF) 10: RxD0 input (SCIF) 11: D23 input/output (BSC)

- PECRL1

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

Bit	Bit Name	Initial Value	R/W	Description
11	PE13MD1	0	R/W	PE13 Mode
10	PE13MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE13/HIFD04/D20. 00: PE13 input/output (port) 01: HIFD04 input/output (HIF) 10: Setting prohibited 11: D20 input/output (BSC)
9	PE12MD1	0	R/W	PE12 Mode
8	PE12MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE12/HIFD03/D19. 00: PE12 input/output (port) 01: HIFD03 input/output (HIF) 10: Setting prohibited 11: D19 input/output (BSC)
7	PE11MD1	0	R/W	PE11 Mode
6	PE11MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE11/HIFD02/D18. 00: PE11 input/output (port) 01: HIFD02 input/output (HIF) 10: Setting prohibited 11: D18 input/output (BSC)
5	PE10MD1	0	R/W	PE10 Mode
4	PE10MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE10/HIFD01/D17. 00: PE10 input/output (port) 01: HIFD01 input/output (HIF) 10: Setting prohibited 11: D17 input/output (BSC)

Bit	Bit Name	Initial Value	R/W	Description
3	PE9MD1	0	R/W	PE9 Mode
2	PE9MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE09/HIFD00/D16. 00: PE09 input/output (port) 01: HIFD00 input/output (HIF) 10: Setting prohibited 11: D16 input/output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PE8MD0	0 (When in non-HIF boot mode) 1 (When in HIF boot mode)	R/W	PE8 Mode Selects the function of pin PE08/$\overline{\text{HIFCS}}$. 0: PE08 input/output (port) 1: $\overline{\text{HIFCS}}$ input (HIF)

- PECRL2

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

Bit	Bit Name	Initial Value	R/W	Description
13	PE6MD1	0	R/W	PE6 Mode
12	PE6MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0. 00: PE06 input/output (port) 01: $\overline{\text{HIFWR}}$ input (HIF) 10: SIOFSYNC0 input/output (SIOF) 11: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PE5MD0	0 (When in non-HIF boot mode) 1 (When in HIF boot mode)	R/W	PE5 Mode Selects the function of pin PE05/$\overline{\text{HIFRD}}$. 0: PE05 input/output (port) 1: $\overline{\text{HIFRD}}$ input (HIF)
9	PE4MD1	0	R/W	PE4 Mode
8	PE4MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE04/$\overline{\text{HIFINT}}$/TXD_SIO0. 00: PE04 input/output (port) 01: $\overline{\text{HIFINT}}$ input (HIF) 10: TXD_SIO0 output (SIOF) 11: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	PE3MD0	1	R/W	PE3 Mode Selects the function of pin PE03/HIFMD. 0: PE03 input/output (port) 1: HIFMD input (HIF)
5	PE2MD1	0	R/W	PE2 Mode
4	PE2MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE02/HIFDREQ/RXD_SIO0. 00: PE02 input/output (port) 01: HIFDREQ output (HIF) 10: RXD_SIO0 input (SIOF) 11: Setting prohibited
3	PE1MD0	0	R/W	PE1 Mode
2	PE1MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE01/HIFRDY/SIOMCLK0. 00: PE01 input/output (port) 01: HIFRDY output (HIF) 10: SIOMCLK0 input (SIOF) 11: Setting prohibited
1	PE0MD1	0	R/W	PE0 Mode
0	PE0MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE00/HIFEBL/SCK_SIO0. 00: PE00 input/output (port) 01: HIFEBL input (HIF) 10: SCK_SIO0 input/output (SIOF) 11: Setting prohibited

18.2 Notes on Usage

18.2.1 Restriction in Using

In the use of the pins listed in the following table, please set the data registers and also pin function controller (i.e. in the use of the output functions EXCEPT the function 1, please DO NOT change the initial value (= 0) in the data registers of that pins).

Table 18.7 Pins restricted in using in SH7619

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
PD2 input/output (port)	IRQ2 input (INTC)	TxD1 output (SCIF)	DREQ0 input (DMAC)
PD4 input/output (port)	IRQ4 input (INTC)	SCK1 input/output (SCIF)	DACK0 output (DMAC)
PD5 input/output (port)	IRQ5 input (INTC)	TxD2 input/output (SCIF)	DREQ1 input (DMAC)

18.2.2 Details of Restriction

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

Section 19 I/O Ports

This LSI has 26 ports (ports A, B, C, D, and E). Port A, port B, port C, port D, and port E are 10-bit, 14-bit, 21-bit, 8-bit, and 25-bit I/O port, respectively. The pins of each port are multiplexed with other functions. The pin function controller (PFC) handles the selection of multiplex pin functions. Each port has a data register to store data of pin.

19.1 Port A

Port A of this LSI is an I/O port with ten pins as shown in figure 19.1.

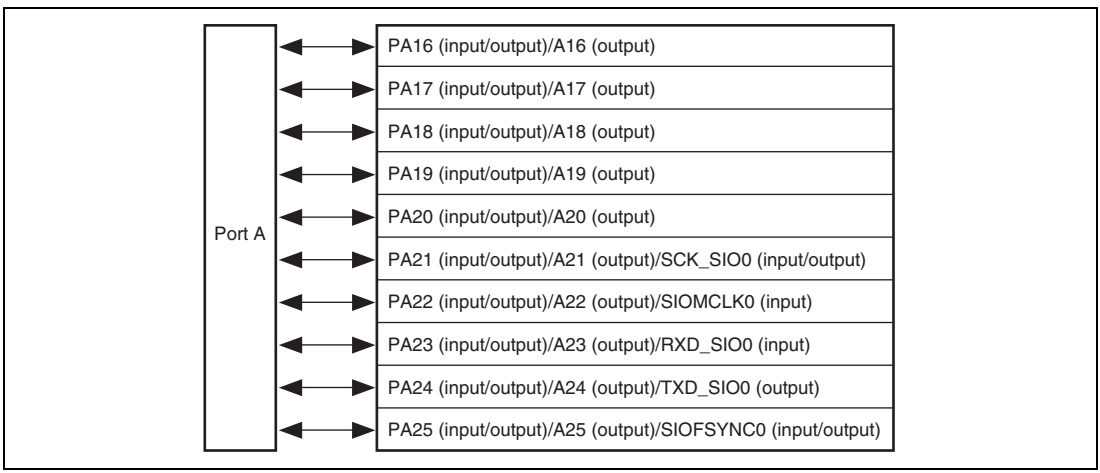


Figure 19.1 Port A

19.1.1 Register Description

Port A is a 10-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 24, List of Registers.

- Port A data register H (PADRH)

19.1.2 Port A Data Register H (PADRH)

PADRH is a 16-bit readable/writable register which stores data for port A. Bits PA25DR to PA16DR correspond to pins PA25 to PA16. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PADRH, the value is output from the pin; if PADRH is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PADRH is read. Data can be written to PADRH but no effect on the pin state. Table 19.1 shows the reading/writing function of the port A data register H.

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	PA25DR	0	R/W	See table 19.1.
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

Table 19.1 Port A Data Register H (PADRH) Read/Write Operation

- Bits 9 to 0 in PADRH

Pin Function	PAIORH	Read	Write
General input	0	Pin state	Data can be written to PADRH but no effect on the pin state.
General output	1	PADRH value	Written value is output from the pin.
Other functions	*	PADRH value	Data can be written to PADRH but no effect on the pin state.

19.2 Port B

Port B of this LSI is an I/O port with 14 pins as shown in figure 19.2.

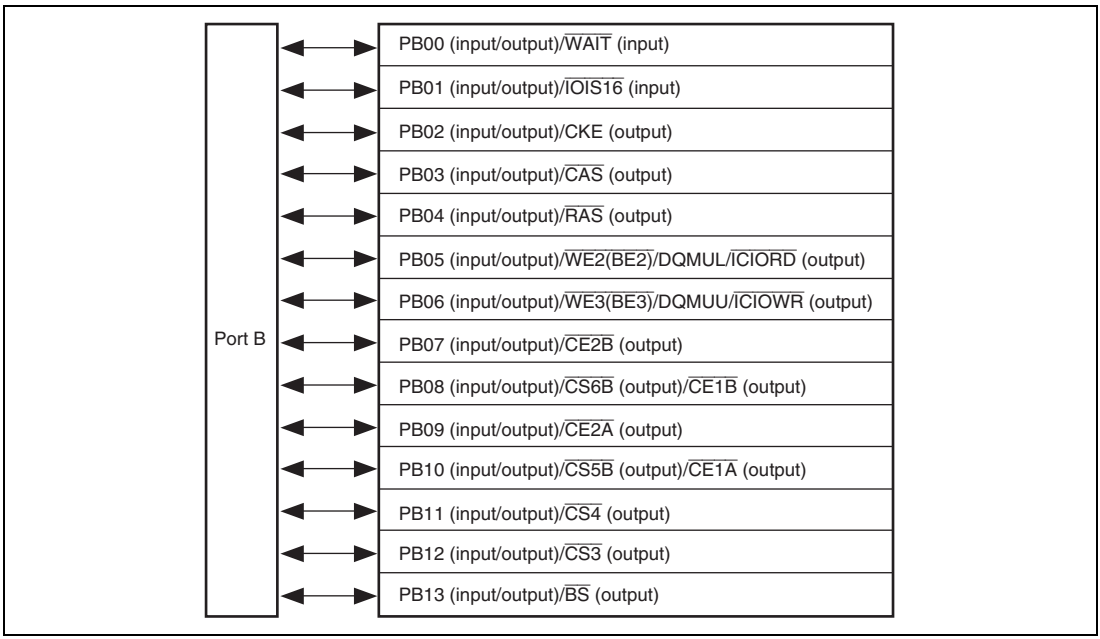


Figure 19.2 Port B

19.2.1 Register Description

Port B is a 14-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 24, List of Registers.

- Port B data register L (PBDRL)

19.2.2 Port B Data Register L (PBDRL)

PBDRL is a 16-bit readable/writable register which stores data for port B. Bits PB13DR to PB0DR correspond to pins PB13 to PB00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PBDRL, the value is output from the pin; if PBDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PBDRL is read. Data can be written to PBDRL but no effect on the pin state. Table 19.2 shows the reading/writing function of the port B data register L.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved
14	—	0	R	These bits are always read as 0. The write value should always be 0.
13	PB13DR	0	R/W	See table 19.2.
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 19.2 Port B Data Register L (PBDRL) Read/Write Operation

- Bits 13 to 0 in PBDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PBDRL but no effect on the pin state.
General output	1	PBDRL value	Written value is output from the pin.
Other functions	*	PBDRL value	Data can be written to PBDRL but no effect on the pin state.

19.3 Port C

Port C of this LSI is an I/O port with 21 pins as shown in figure 19.3.

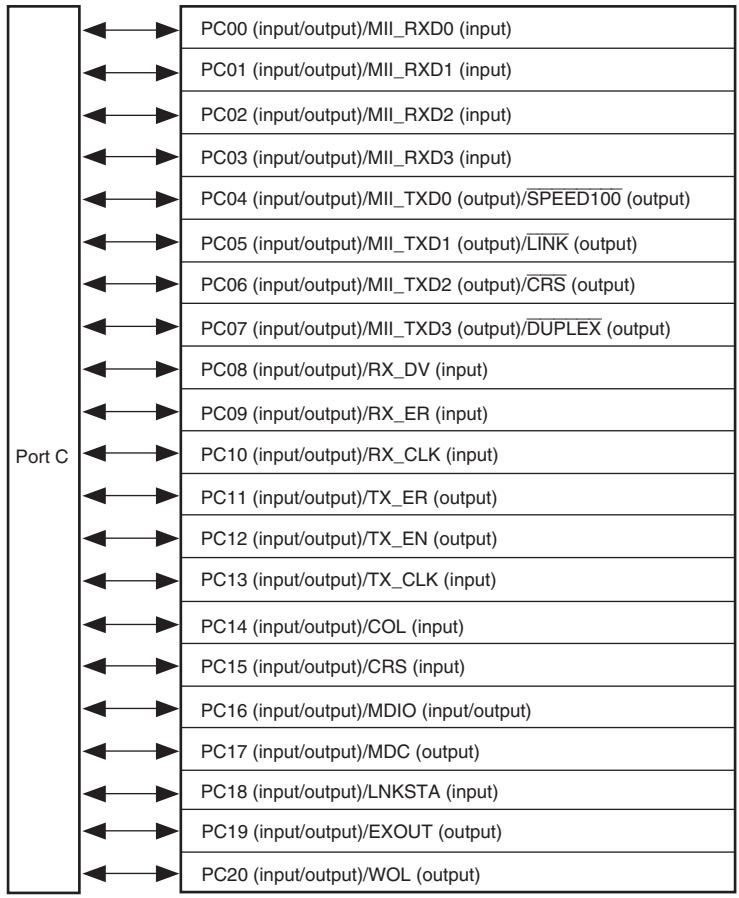


Figure 19.3 Port C

19.3.1 Register Description

Port C is a 21-bit I/O port that has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Port C data register H (PCDRH)
- Port C data register L (PCDRL)

19.3.2 Port C Data Registers H and L (PCDRH and PCDRL)

PCDRH and PCDRL are 16-bit readable/writable registers that stores data for port C. Bits PC20DR to PC0DR correspond to pins PC20 to PC00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PCDRH or PCDRL, the value is output from the pin; if PCDRH or PCDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PCDRH or PCDRL is read. Data can be written to PCDRH or PCDRL but no effect on the pin state. Table 19.3 shows the reading/writing function of the port C data registers H and L.

- PCDRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC20DR	0	R/W	See table 19.3.
3	PC19DR	0	R/W	
2	PC18DR	0	R/W	
1	PC17DR	0	R/W	
0	PC16DR	0	R/W	

- PCDRL

Bit	Bit Name	Initial Value	R/W	Description
15	PC15DR	0	R/W	See table 19.3.
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 19.3 Port C Data Registers H and L (PCDRH and PCDRL) Read/Write Operation

- Bits 4 to 0 in PCDRH and Bits 15 to 0 in PCDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PCDRH or PCDRL but no effect on the pin state.
General output	1	PCDRH or PCDRL value	Written value is output from the pin.
Other functions	*	PCDRH or PCDRL value	Data can be written to PCDRH or PCDRL but no effect on the pin state.

19.4 Port D

Port D of this LSI is an I/O port with eight pins as shown in figure 19.4.

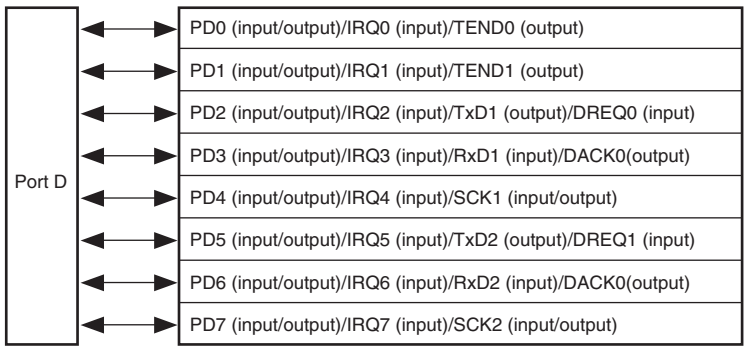


Figure 19.4 Port D

19.4.1 Register Description

Port D is an 8-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 24, List of Registers.

- Port D data register L (PDDRL)

19.4.2 Port D Data Register L (PDDRL)

PDDRL is a 16-bit readable/writable register which stores data for port D. Bits PD7DR to PD0DR correspond to pins PD7 to PD0. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PDDRL, the value is output from the pin; if PDDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PDDRL is read. Data can be written to PDDRL but no effect on the pin state. Table 19.4 shows the reading/writing function of the port D data register L.

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	PD7DR	0	R/W	See table 19.4.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

Table 19.4 Port D Data Register L (PDDRL) Read/Write Operation

- Bits 7 to 0 in PDDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PDDRL but no effect on the pin state.
General output	1	PDDRL value	Written value is output from the pin.
Other functions	*	PDDRL value	Data can be written to PDDRL but no effect on the pin state.

19.5 Port E

Port E of this LSI is an I/O port with 25 pins as shown in figure 19.5.

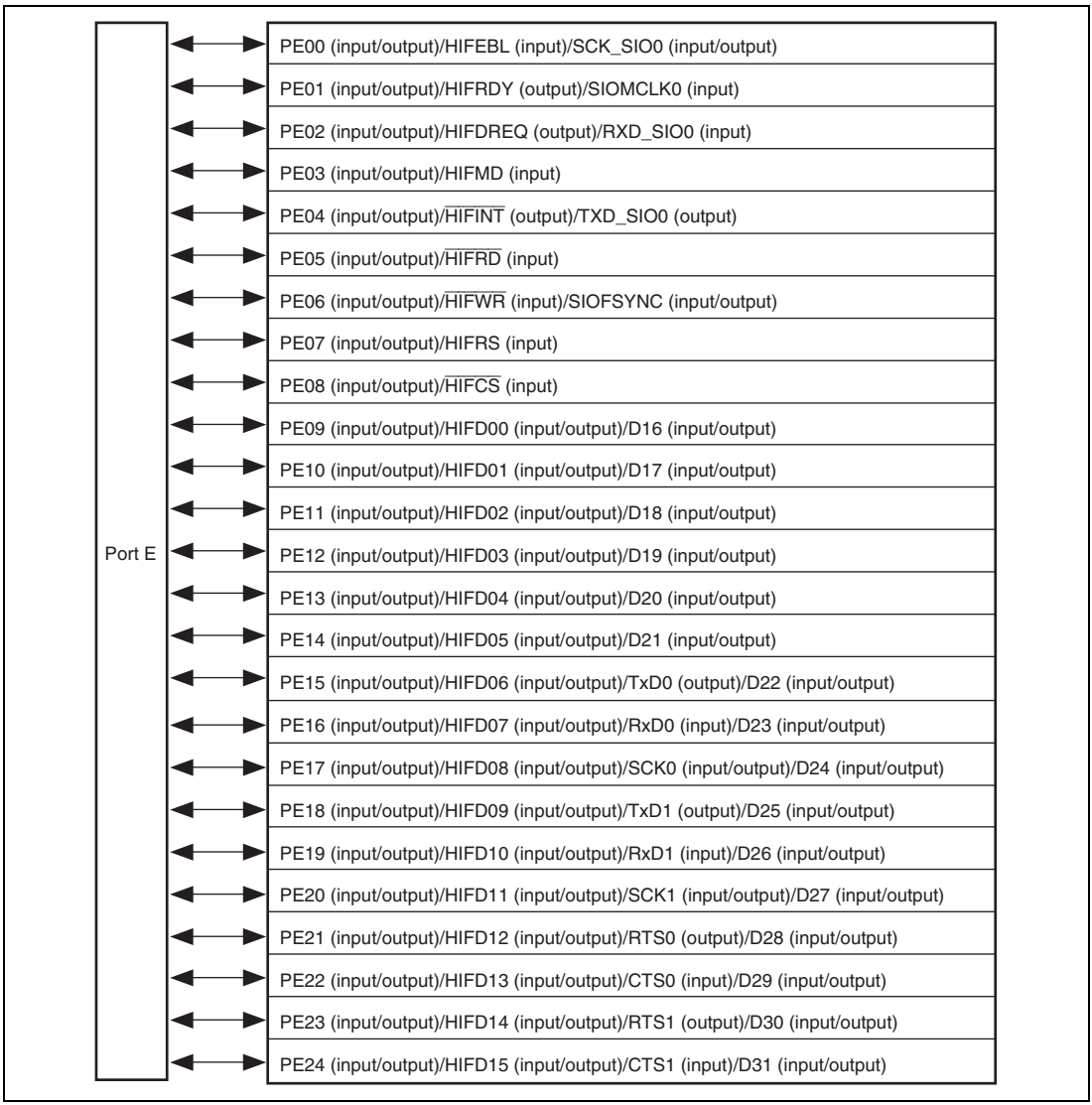


Figure 19.5 Port E

19.5.1 Register Description

Port E is a 25-bit I/O port that has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Port E data register H (PEDRH)
- Port E data register L (PEDRL)

19.5.2 Port E Data Registers H and L (PEDRH and PEDRL)

PEDRH and PEDRL are 16-bit readable/writable registers that store data for port E. Bits PE24DR to PE0DR correspond to pins PE24 to PE00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PEDRH or PEDRL, the value is output from the pin; if PEDRH or PEDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PEDRH or PEDRL is read. Data can be written to PEDRH or PEDRL but no effect on the pin state. Table 19.5 shows the reading/writing function of the port E data registers H and L.

- PEDRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PE24DR	0	R/W	See table 19.5.
7	PE23DR	0	R/W	
6	PE22DR	0	R/W	
5	PE21DR	0	R/W	
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

- PEDRL

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DR	0	R/W	See table 19.5.
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 19.5 Port E Data Registers H, L (PEDRH, PEDRL) Read/Write Operation

- Bits 8 to 0 in PEDRH and Bits 15 to 0 in PEDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PEDRH or PEDRL but no effect on the pin state.
General output	1	PEDRH or PEDRL value	Written value is output from the pin.
Other functions	*	PEDRH or PEDRL value	Data can be written to PEDRH or PEDRL but no effect on the pin state.

19.6 Usage Notes

1. When pins that are multiplexed with general I/O is used as output pins for other functions, these pins work as general output pins for the period of $1 \times t_{PCYC}$ synchronized with internal power-on reset by WDT overflow. For example, when the pin PB12/$\overline{CS3}$ works as $\overline{CS3}$ and the PB12DR bit in PBDRL is set to 0, the pin is driven low for the period of $1 \times t_{PCYC}$ and may lead to memory malfunction. To prevent this, port registers that correspond to pins used for the strobe output must be set to strobe non-active level. This case does not apply to the power-on reset from the $\overline{RES}$ pin.
2. The weak keeper circuit is included in all pins except MD5, MD3, MD2, MD1, MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, EXTAL, XTAL, TxP, TxM, RxP, RxM, EXRES1, and TSTBUSA. The weak keeper is a circuit, always operating while the power is on, that fixes the input in I/O pins to low or high when the pins are not driven from outside. Notes on processing the input pins are as follows:
   - When using pins having the weak keeper circuit as input pins and driving these pins to a certain level from outside, adjust the resistance of pull-up/pull-down resistors to let the weak keeper circuit keep the intended levels. (2 kΩ and 8 kΩ are recommended respectively.) The larger the resistance is, the longer the transition time is. In addition, a large resistance may fail to let the weak keeper circuit to keep the intended levels. Therefore, when the resistors adjusted comparatively large are used, ensure that any transition does not delay in the system.
   - While using the pins having the weak keeper circuit as input pins, if their levels do not matter, there is no need to deal with pins from outside.
   - MD5, MD3, MD2, MD1, MD0, $\overline{ASEMD}$, and $\overline{TESTMD}$.

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

     See section 8.6, Notes on Board Design in section 8, Clock Pulse Generator (CPG).
3. Since the HIFMD pin is not initially set to function as a general port pin, it must be pulled up or down externally to fix its state.
4. When using a multiplexed pin with a function not selected with its initial value (for example, using the PB12/$\overline{CS3}$ pin, the initial function of which is PB12, as the $\overline{CS3}$ pin), the pin must be pulled up or down externally at least after a reset until its pin function is selected by software to fix its state.

Section 20 User Break Controller (UBC)

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.

20.1 Features

The UBC has the following features:

- The following break comparison conditions can be set.

  Number of break channels: two channels (channels A and B)

  User break can be requested as either the independent or sequential condition on channels A and B (sequential break: when channel A and channel B match with break conditions in the different bus cycles in that order, a break condition is satisfied).

  — Address (Compares addresses 32 bits):
  Comparison bits are maskable in 1-bit units; user can mask addresses at lower 12 bits (4-k page), lower 10 bits (1-k page), or any size of page, etc.

  One of the two address buses (L-bus address (LAB) and I-bus address (IAB)) can be selected.

  — Data (only on channel B, 32-bit maskable)

  One of the two data buses (logic data bus (LDB) and internal data bus (IDB)) can be selected.

  — Bus cycle: Instruction fetch or data access

  — Read/write

  — Operand size: Byte, word, or longword
- User break interrupt is generated upon satisfying break conditions. A user-designed user-break condition interrupt exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.
- Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
- Four pairs of branch source/destination buffers.

Figure 20.1 shows a block diagram of the UBC.

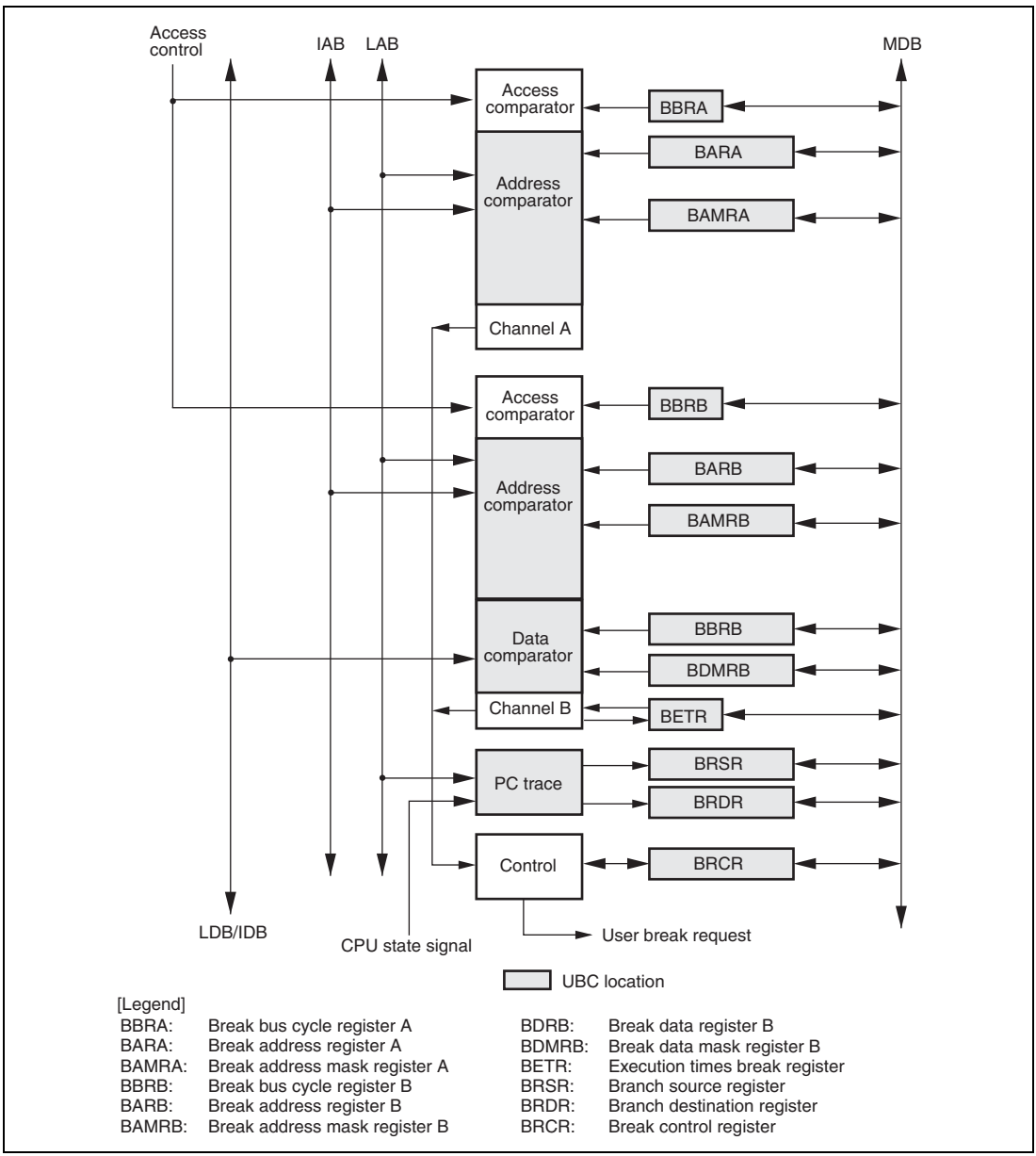


Figure 20.1 Block Diagram of UBC

20.2 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)

20.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used for 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.

20.2.2 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 to 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

20.2.3 Break Bus Cycle Register A (BBRA)

Break bus cycle register A (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

20.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used for a break condition in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB 0	All 0	R/W	Break Address B Store an address of LAB or IAB which specifies a break condition in channel B.

20.2.5 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 Specify 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

20.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. BDBR selects data used for a break condition in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB 0	All 0	R/W	Break Data Bit B Store data which specifies a break condition in channel B. BDRB specifies the break data on LDB or IDB.

Notes: 1. Specify an operated size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.

20.2.7 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 Specify 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 operated size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break mask data.

20.2.8 Break Bus Cycle Register B (BBRB)

Break bus cycle register B (BBRB) is a 16-bit readable/writable register, which 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 B.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
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
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

20.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

- Channels A and B are used in two independent channel conditions or under the sequential condition.
- A break is set before or after instruction execution.
- Specify whether to include the number of execution times on channel B in comparison conditions.
- Specify whether to include data bus on channel B in comparison conditions.
- Enable PC trace.

The break control register (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 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

Bit	Bit Name	Initial Value	R/W	Description
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
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, 8	—	All 0	R	Reserved 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

Bit	Bit Name	Initial Value	R/W	Description
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, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential conditions. 0: Channels A and B are compared under independent conditions 1: Channels A and B are compared under sequential conditions (channel A, then channel B)
2, 1	—	All 0	R	Reserved 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

20.2.10 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. Every time the break condition is satisfied, BETR is decremented by 1. 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

20.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch 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 four 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 or not the branch source address is stored. When a branch is made, this flag is set to 1. This flag is cleared to 0 by one of the following conditions: when this flag is read from this register, when PC trace is enabled, and when a power-on reset is generated. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid

Bit	Bit Name	Initial Value	R/W	Description
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	Undefined	R	Branch Source Address Store bits 27 to 0 of the branch source address.

20.2.12 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 four 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 or not the branch source address is stored. When a branch is made, this flag is set to 1. This flag is cleared to 0 by one of the following conditions: when this flag is read from this register, when PC trace is enabled, and when a power-on reset is generated. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BDA27 to BDA0	Undefined	R	Branch Destination Address Store bits 27 to 0 of the branch destination address.

20.3 Operation

20.3.1 Flow of User Break Operation

The flow from setting of break conditions to user break exception processing is described below:

1. The break addresses are set in the break address registers (BARA and BARB). The masked addresses are set in the break address mask registers (BAMRA and 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 and BBRB). There are three control bit combinations in both BBRA and BBRB: bits to select L-bus cycle or I-bus cycle, bits to select instruction fetch or data access, and bits to select read or write. No user break will be generated if one of these combinations is set to B'00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA/BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the 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.
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. Reset the flags by writing 0 before they are used again.
4. There is a chance that the data access break and its following instruction fetch break occur around the same time, there will be only one break request to the CPU, but these two break channel match flags could be both set.

20.3.2 Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBRA/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/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/BARB) to 0. A break cannot be generated as long as this bit is set to 1.
2. If the condition is matched while a break before execution is selected, a break is generated when it is confirmed that the instruction has been fetched and it 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 in the delay slot of a delayed branch instruction, the break is generated immediately before the execution of the instruction that first accepts the break. Meanwhile, a break before the execution of the instruction in a delay slot and a break after the execution of the SLEEP instruction are also prohibited.
3. When a break after execution is selected, the instruction that matches the break condition is executed and then the break is generated prior to the execution of the next instruction. As with a break before execution, this cannot be used with overrun fetch instructions. When this kind of break is set for a delayed branch instruction, a break is not generated until the first instruction at which breaks are accepted.
4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB) is ignored. There is thus no need to set break data for the break of the instruction fetch cycle.

20.3.3 Break on Data Access Cycle

- The bus cycles in which L bus data access breaks occur are from instructions.
- The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 20.1.

Table 20.1 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

- 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 registers (BBRA and BBRB). In this case, 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.

20.3.4 Sequential Break

- 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 break 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.
- In sequential break specification, the L- or I-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 is generated when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

20.3.5 Value of Saved Program Counter (PC)

When a break occurs, PC is saved onto the stack. The PC value saved is as follows depending on the type of break.

- When a break before execution is selected:

  The value of the program counter (PC) saved is the address of the instruction that matches the break condition. The fetched instruction is not executed, and a break occurs before it.

- When a break after execution is selected:

  The PC value saved is the address of the instruction to be executed following the instruction in which the break condition matches. The fetched instruction is executed, and a break occurs before the execution of the next instruction.
- When an address in a data access cycle is specified as a break condition:

  The PC value is the address of the instruction to be executed following the instruction that matched the break condition. The instruction that matched the condition is executed and the break occurs before the next instruction is executed.
- When an address and data in a data access cycle are specified as a break condition:

  The PC value is the start address of the instruction that follows the instruction already executed when break processing started. When a data value is added to the break conditions, the break will occur before the execution of an instruction that is within two instructions of the instruction that matched the break condition. Therefore, where the break will occur cannot be specified exactly.

20.3.6 PC Trace

- Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
- The branch source address has different values due to the kind of branch.
  - Branch instruction

    The branch instruction address.
  - Interrupt and exception

    The address of the instruction in which the interrupt or exception was accepted. This address is equal to the return address saved onto the stack.

    The start address of the interrupt or exception handling routine is stored in BRDR.

    The TRAPA instruction belongs to interrupt and exception above.
- BRSR and BRDR have four pairs of queue structures. The top of queues is read first 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.

20.3.7 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)

  — 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)

    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

  Specified conditions: Channel A/channel B sequential mode

  — Channel A

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

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

  — Channel B

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

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

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

  After address H'00037226 is executed, a user break occurs before an instruction of 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

    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

  Specified conditions: Channel A/channel B sequential mode

  — Channel A

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

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

  — Channel B

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

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

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

  Since instruction fetch is not a write cycle on channel A, a sequential condition does 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)

  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

  Specified conditions: Channel A/channel B independent mode

  — Channel A

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

    Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  — Channel B

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

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

    Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction of addresses H'00008000 to H'00008FFE is executed or before an instruction of addresses H'00008010 to H'00008016 is 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

  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

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

    Bus cycle: L bus/data access/write/word

  On channel A, a user break occurs with longword read from address H'00123454, 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 addresses H'000ABC00 to H'000ABCFE.

Break Condition Specified for I Bus Data Access Cycle:

- Register specifications:

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  — Channel A

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

    Bus cycle: I bus/instruction fetch/read (operand size is not included in the 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 address H'00314156 in the memory space.

  On channel B, a user break occurs when the I bus writes byte data H'7* in address H'00055555.

20.3.8 Notes

1. The CPU reads from or writes to the UBC registers via the I bus. A desired break may not occur until the instruction to rewrite the UBC registers are executed and the actual values are reflected. In order to know the timing the UBC register is changed, read 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 user breaks and other exceptions occur by the same instruction, they are handled according to the priority listed in table 5.1 of section 5, Exception Handling. When an exception with a higher priority is generated, no user break occurs.
   - A break before the execution of an instruction is accepted with a priority over other exceptions.
   - When a break after the execution of an instruction or a data access break occurs simultaneously with a re-execution-type exception with a higher priority (including a break before the execution of an instruction), the re-execution-type exception is accepted and the condition match flag is not set (however, there is an exception as explained in 5. of section 20.3.8, Notes). When the exception source of the re-execution type is cleared by exception handling and the same instruction is executed again and completed, the break is generated again and the flag is set.
   - When a break after the execution of an instruction or a data access break occurs simultaneously with a completion-type exception with a higher priority (TRAPA), no break occurs but the condition match flag is set.
5. Note on exception of 4. of section 20.3.8, Notes

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

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

Section 21 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) to provide a boundary scan function and emulator support.

This section describes the boundary scan function of the H-UDI. For details on emulator functions of the H-UDI, refer to the user's manual of the relevant emulator.

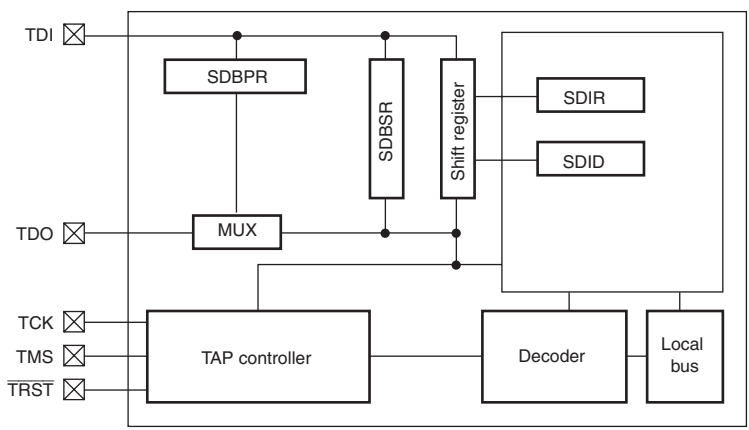
21.1 Features

The H-UDI 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 function, 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 21.1 shows a block diagram of the H-UDI.



[Legend]
SDBPR: Bypass register
SDBSR: Boundary scan register
SDIR: Instruction register
SDID: ID register

Figure 21.1 Block Diagram of H-UDI

21.2 Input/Output Pins

Table 21.1 shows the pin configuration of the H-UDI.

Table 21.1 Pin Configuration

Abbr.	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 the given period when the power is turned on regardless of using the H-UDI function. This is different from the JTAG standard. For details on resets, see section 21.4.2, Reset Configuration.
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 SDIR. For details, see section 21.3.2, Instruction Register (SDIR).
$\overline{\text{ASEMD}}$	Input	ASE Mode Select Pin When a low level is input to the $\overline{\text{ASEMD}}$ pin, ASE mode is entered; if a high level is input, normal mode is entered. In ASE mode, the emulator functions can be used. The input level on the $\overline{\text{ASEMD}}$ pin should be held except during the $\overline{\text{RES}}$ pin assertion period.

21.3 Register Descriptions

The H-UDI has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 24, List of Registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

21.3.1 Bypass Register (SDBPR)

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

21.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. This 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.
11 to 8	TI3 to TI0	All 1	R	For commands, see table 21.2.
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 21.2 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 above								Reserved

21.3.3 Boundary Scan Register (SDBSR)

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

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

Table 21.3 External pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
	from TDI		303	PE02/HIFDREQ/RXD_SIO0/-	IN
332	PD06/IRQ6/RxD2/DACK1	IN	302	PE01/HIFRDY/SIOMCLK0/-	IN
331	PD05/IRQ5/TxD2/DREQ1	IN	301	PE00/HIFEBL/SCK_SIO0/-	IN
330	PD04/IRQ4/SCK1/-	IN	300	PC17/MDC/-/-	IN
329	PD03/IRQ3/RxD1/DACK0	IN	299	PC16/MDIO/-/-	IN
328	PD02/IRQ2/TxD1/DREQ0	IN	298	PC15/CRS/-/-	IN
327	PD01/IRQ1/-/TEND1	IN	297	PC18/LNKSTA	IN
326	PD00/IRQ0/-/TEND0	IN	296	PD06/IRQ6/RxD2/DACK1	OUT
325	PE08/$\overline{\text{HIFCS}}$	IN	295	PD05/IRQ5/TxD2/DREQ1	OUT
324	PE24/HIFD15/CTS1/D31	IN	294	PD04/IRQ4/SCK1/-	OUT
323	PE23/HIFD14/RTS1/D30	IN	293	PD03/IRQ3/RxD1/DACK0	OUT
322	PE22/HIFD13/CTS0/D29	IN	292	PD02/IRQ2/TxD1/DREQ0	OUT
321	PE21/HIFD12/RTS0/D28	IN	291	PD01/IRQ1/-/TEND1	OUT
320	PE20/HIFD11/SCK1/D27	IN	290	PD00/IRQ0/-/TEND0	OUT
319	PE19/HIFD10/RxD1/D26	IN	289	PE08/$\overline{\text{HIFCS}}$	OUT
318	PE18/HIFD09/TxD1/D25	IN	288	PE24/HIFD15/CTS1/D31	OUT
317	PE17/HIFD08/SCK0/D24	IN	287	PE23/HIFD14/RTS1/D30	OUT
316	PE16/HIFD07/RxD0/D23	IN	286	PE22/HIFD13/CTS0/D29	OUT
315	PE15/HIFD06/TxD0/D22	IN	285	PE21/HIFD12/RTS0/D28	OUT
314	PE14/HIFD05/-/D21	IN	284	PE20/HIFD11/SCK1/D27	OUT
313	PE13/HIFD04/-/D20	IN	283	PE19/HIFD10/RxD1/D26	OUT
312	PE12/HIFD03/-/D19	IN	282	PE18/HIFD09/TxD1/D25	OUT
311	PE11/HIFD02/-/D18	IN	281	PE17/HIFD08/SCK0/D24	OUT
310	PE10/HIFD01/-/D17	IN	280	PE16/HIFD07/RxD0/D23	OUT
309	PE09/HIFD00/-/D16	IN	279	PE15/HIFD06/TxD0/D22	OUT
308	PE07/HIFRS	IN	278	PE14/HIFD05/-/D21	OUT
307	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0/-	IN	277	PE13/HIFD04/-/D20	OUT
306	PE05/$\overline{\text{HIFRD}}$	IN	276	PE12/HIFD03/-/D19	OUT
305	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0/-	IN	275	PE11/HIFD02/-/D18	OUT
304	PE03/HIFMD	IN	274	PE10/HIFD01/-/D17	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
273	PE09/HIFD00/-/D16	OUT	241	PE13/HIFD04/-/D20	Control
272	PE07/HIFRS	OUT	240	PE12/HIFD03/-/D19	Control
271	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0/-	OUT	239	PE11/HIFD02/-/D18	Control
270	PE05/$\overline{\text{HIFRD}}$	OUT	238	PE10/HIFD01/-/D17	Control
269	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0/-	OUT	237	PE09/HIFD00/-/D16	Control
268	PE03/HIFMD	OUT	236	PE07/HIFRS	Control
267	PE02/HIFDREQ/RXD_SIO0/-	OUT	235	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0/-	Control
266	PE01/HIFRDY/SIOMCLK0/-	OUT	234	PE05/$\overline{\text{HIFRD}}$	Control
265	PE00/HIFEBL/SCK_SIO0/-	OUT	233	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0/-	Control
264	PC17/MDC/-/-	OUT	232	PE03/HIFMD	Control
263	PC16/MDIO/-/-	OUT	231	PE02/HIFDREQ/RXD_SIO0/-	Control
262	PC15/CRS/-/-	OUT	230	PE01/HIFRDY/SIOMCLK0/-	Control
261	PC18/LNKSTA	OUT	229	PE00/HIFEBL/SCK_SIO0/-	Control
260	PD06/IRQ6/RxD2/DACK1	Control	228	PC17/MDC/-/-	Control
259	PD05/IRQ5/TxD2/DREQ1	Control	227	PC16/MDIO/-/-	Control
258	PD04/IRQ4/SCK1/-	Control	226	PC15/CRS/	Control
257	PD03/IRQ3/RxD1/DACK0	Control	225	PC18/LNKSTA	Control
256	PD02/IRQ2/TxD1/DREQ0	Control	224	PC09/RX_ER/-/-	IN
255	PD01/IRQ1/-/TEND1	Control	223	PC08/RX_DV/-/-	IN
254	PD00/IRQ0/-/TEND0	Control	222	PC00/MIIRXD0/-/-	IN
253	PE08/$\overline{\text{HIFCS}}$	Control	221	PC01/MIIRXD1/-/-	IN
252	PE24/HIFD15/CTS1/D31	Control	220	PC02/MIIRXD2/-/-	IN
251	PE23/HIFD14/RTS1/D30	Control	219	PC03/MIIRXD3/-/-	IN
250	PE22/HIFD13/CTS0/D29	Control	218	PC10/RX_CLK/-/-	IN
249	PE21/HIFD12/RTS0/D28	Control	217	PC11/TX_ER/-/-	IN
248	PE20/HIFD11/SCK1/D27	Control	216	PC13/TX_CLK/-/-	IN
247	PE19/HIFD10/RxD1/D26	Control	215	PC04/MIITXD0/-/$\overline{\text{SPEED100}}$	IN
246	PE18/HIFD09/TxD1/D25	Control	214	PC05/MIITXD1/-/$\overline{\text{LINK}}$	IN
245	PE17/HIFD08/SCK0/D24	Control	213	PC06/MIITXD2/-/$\overline{\text{CRS}}$	IN
244	PE16/HIFD07/RxD0/D23	Control	212	PC07/MIITXD3/-/$\overline{\text{DUPLEX}}$	IN
243	PE15/HIFD06/TxD0/D22	Control	211	PC12/TX_EN/-/-	IN
242	PE14/HIFD05/-/D21	Control	210	PC14/COL/-/-	IN

Bit	Pin Name	I/O
209	PC20/WOL	IN
208	PC19/EXOUT	IN
207	MD3	IN
206	MD5	IN
205	$\overline{\text{TESTMD}}$	IN
204	PC09/RX_ER/-/-	OUT
203	PC08/RX_DV/-/-	OUT
202	PC00/MIIRXD0/-/-	OUT
201	PC01/MIIRXD1/-/-	OUT
200	PC02/MIIRXD2/-/-	OUT
199	PC03/MIIRXD3/-/-	OUT
198	PC10/RX_CLK/-/-	OUT
197	PC11/TX_ER/-/-	OUT
196	PC13/TX_CLK/-/-	OUT
195	PC04/MIITXD0/-/$\overline{\text{SPEED100}}$	OUT
194	PC05/MIITXD1/-/$\overline{\text{LINK}}$	OUT
193	PC06/MIITXD2/-/$\overline{\text{CRS}}$	OUT
192	PC07/MIITXD3/-/$\overline{\text{DUPLEX}}$	OUT
191	PC12/TX_EN/-/-	OUT
190	PC14/COL/-/-	OUT
189	PC20/WOL	OUT
188	PC19/EXOUT	OUT
187	TESTOUT	OUT
186	PC09/RX_ER/-/-	Control
185	PC08/RX_DV/-/-	Control
184	PC00/MIIRXD0/-/-	Control
183	PC01/MIIRXD1/-/-	Control
182	PC02/MIIRXD2/-/-	Control
181	PC03/MIIRXD3/-/-	Control
180	PC10/RX_CLK/-/-	Control
179	PC11/TX_ER/-/-	Control
178	PC13/TX_CLK/-/-	Control

Bit	Pin Name	I/O
177	PC04/MIITXD0/-/$\overline{\text{SPEED100}}$	Control
176	PC05/MIITXD1/-/$\overline{\text{LINK}}$	Control
175	PC06/MIITXD2/-/$\overline{\text{CRS}}$	Control
174	PC07/MIITXD3/-/$\overline{\text{DUPLEX}}$	Control
173	PC12/TX_EN/-/-	Control
172	PC14/COL/-/-	Control
171	PC20/WOL	Control
170	PC19/EXOUT	Control
169	TESTOUT	Control
168	MD0	IN
167	NMI	IN
166	MD1	IN
165	MD2	IN
164	D00	IN
163	D01	IN
162	D02	IN
161	D03	IN
160	D04	IN
159	D05	IN
158	D06	IN
157	D07	IN
156	D15	IN
155	D14	IN
154	D13	IN
153	D12	IN
152	D11	IN
151	D10	IN
150	D09	IN
149	D08	IN
148	PB02/CKE	IN
147	PB03/$\overline{\text{CAS}}$	IN
146	PB04/$\overline{\text{RAS}}$	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
145	D00	OUT	115	D15	Control
144	D01	OUT	114	D14	Control
143	D02	OUT	113	D13	Control
142	D03	OUT	112	D12	Control
141	D04	OUT	111	D11	Control
140	D05	OUT	110	D10	Control
139	D06	OUT	109	D09	Control
138	D07	OUT	108	D08	Control
137	D15	OUT	107	$\overline{WE0}$, DQMLL	Control
136	D14	OUT	106	$\overline{WE1}$, DQMLU, $\overline{WE}$	Control
135	D13	OUT	105	RDWR	Control
134	D12	OUT	104	PB02/CKE	Control
133	D11	OUT	103	PB03/$\overline{CAS}$	Control
132	D10	OUT	102	PB04/$\overline{RAS}$	Control
131	D09	OUT	101	PB12/$\overline{CS3}$	IN
130	D08	OUT	100	PB13/$\overline{BS}$	IN
129	$\overline{WE0}$, DQMLL	OUT	99	PB11/$\overline{CS4}$	IN
128	$\overline{WE1}$, DQMLU, $\overline{WE}$	OUT	98	PB00/$\overline{WAIT}$	IN
127	RDWR	OUT	97	PB05/$\overline{WE2(BE2)}$/DQMUL/ $\overline{ICIORD}$	IN
126	PB02/CKE	OUT	96	PB06/$\overline{WE3(BE3)}$/DQMUU/ $\overline{ICIOWR}$	IN
125	PB03/$\overline{CAS}$	OUT	95	PB01/$\overline{IOIS16}$	IN
124	PB04/$\overline{RAS}$	OUT	94	PB09/$\overline{CE2A}$	IN
123	D00	Control	93	PB10/$\overline{CS5B}$, $\overline{CE1A}$	IN
122	D01	Control	92	PB07/$\overline{CE2B}$	IN
121	D02	Control	91	PB08/$\overline{CS6B}$, $\overline{CE1B}$	IN
120	D03	Control	90	PA16/A16	IN
119	D04	Control	89	PA17/A17	IN
118	D05	Control	88	PA18/A18	IN
117	D06	Control	87	PA19/A19	IN
116	D07	Control	86	PA20/A20	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
85	PA21/A21/SCK_SIO0/-	IN	56	PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/ $\overline{\text{ICIOWR}}$	OUT
84	PA22/A22/SIOMCLK0/-	IN	55	PB01/$\overline{\text{IOIS16}}$	OUT
83	PA23/A23/RXD_SIO0/-	IN	54	PB09/$\overline{\text{CE2A}}$	OUT
82	PA24/A24/TXD_SIO0/-	IN	53	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$	OUT
81	PA25/A25/SIOFSYNC0/-	IN	52	PB07/$\overline{\text{CE2B}}$	OUT
80	PD07/IRQ7/SCK2/-	IN	51	PB08/$\overline{\text{CS6B}}$, $\overline{\text{CE1B}}$	OUT
79	PB12/$\overline{\text{CS3}}$	OUT	50	PA16/A16	OUT
78	A00	OUT	49	PA17/A17	OUT
77	A01	OUT	48	PA18/A18	OUT
76	A02	OUT	47	PA19/A19	OUT
75	A03	OUT	46	PA20/A20	OUT
74	A04	OUT	45	PA21/A21/SCK_SIO0/-	OUT
73	A05	OUT	44	PA22/A22/SIOMCLK0/-	OUT
72	A06	OUT	43	PA23/A23/RXD_SIO0/-	OUT
71	A07	OUT	42	PA24/A24/TXD_SIO0/-	OUT
70	A08	OUT	41	PA25/A25/SIOFSYNC0/-	OUT
69	A09	OUT	40	PD07/IRQ7/SCK2/-	OUT
68	A10	OUT	39	PB12/$\overline{\text{CS3}}$	Control
67	A11	OUT	38	A00	Control
66	A12	OUT	37	A01	Control
65	A13	OUT	36	A02	Control
64	A14	OUT	35	A03	Control
63	A15	OUT	34	A04	Control
62	PB13/$\overline{\text{BS}}$	OUT	33	A05	Control
61	$\overline{\text{CS0}}$	OUT	32	A06	Control
60	PB11/$\overline{\text{CS4}}$	OUT	31	A07	Control
59	$\overline{\text{RD}}$	OUT	30	A08	Control
58	PB00/$\overline{\text{WAIT}}$	OUT	29	A09	Control
57	PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/ $\overline{\text{ICIORD}}$	OUT	28	A10	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
27	A11	Control	12	PB07/$\overline{\text{CE2B}}$	Control
26	A12	Control	11	PB08/$\overline{\text{CS6B}}$, $\overline{\text{CE1B}}$	Control
25	A13	Control	10	PA16/A16	Control
24	A14	Control	9	PA17/A17	Control
23	A15	Control	8	PA18/A18	Control
22	PB13/$\overline{\text{BS}}$	Control	7	PA19/A19	Control
21	$\overline{\text{CS0}}$	Control	6	PA20/A20	Control
20	PB11/$\overline{\text{CS4}}$	Control	5	PA21/A21/SCK_SIO0/-	Control
19	$\overline{\text{RD}}$	Control	4	PA22/A22/SIOMCLK0/-	Control
18	PB00/$\overline{\text{WAIT}}$	Control	3	PA23/A23/RXD_SIO0/-	Control
17	PB05/$\overline{\text{WE2(BE2)}}$/ DQMUL/$\overline{\text{ICIORD}}$	Control	2	PA24/A24/TXD_SIO0/-	Control
16	PB06/$\overline{\text{WE3(BE3)}}$/ DQMUU/$\overline{\text{ICIOWR}}$	Control	1	PA25/A25/SIOFSYNC0/-	Control
15	PB01/$\overline{\text{IOIS16}}$	Control	0	PD07/IRQ7/SCK2/-	Control
14	PB09/$\overline{\text{CE2A}}$	Control		to TDO	
13	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$	Control			

Note: * Control means a low active signal.
The corresponding pin is driven with an OUT value when the Control is driven low.

21.3.4 ID Register (SDID)

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 the CPU.

To read this register by the H-UDI side, the contents can be read via the TDO pin when the IDCODE command is set and 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 ID 31 to Device ID 0 ID register that is stipulated by JTAG. H'0800C447 (initial value) for this LSI. Upper four bits may be changed according to the LSI version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

21.4 Operation

21.4.1 TAP Controller

Figure 21.2 shows the internal states of the TAP controller. State transitions basically conform to the JTAG standard.

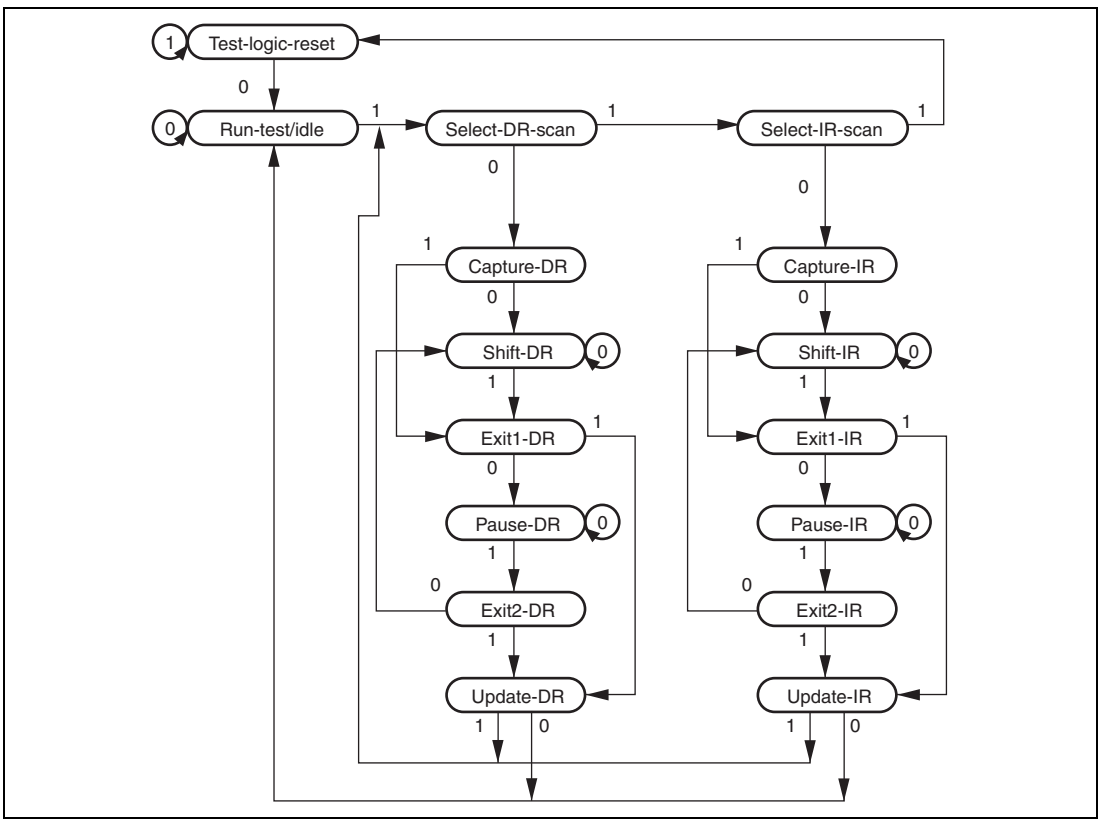


Figure 21.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of the TCK signal. The TDI value is sampled at the rising edge of the TCK signal and is shifted at the falling edge of the TCK signal. For details on change timing of the TDO value, see section 21.4.3, TDO Output Timing. The TDO pin is high impedance, except in the shift-DR and shift-IR states. A transition to the Test-Logic-Reset state is made asynchronously with TCK by driving the $\overline{\text{TRST}}$ signal 0.

21.4.2 Reset Configuration

Table 21.4 Reset Configuration

ASEMD*1	RES	TRST	LSI State
High	Low	Low	Normal reset and H-UDI reset
		High	Normal reset
	High	Low	H-UDI reset only
		High	Normal operation
Low	Low	Low	Reset hold*2
		High	Normal reset
	High	Low	H-UDI reset only
		High	Normal operation

Notes: 1. Selects to normal mode or ASE mode.
$\overline{\text{ASEMD0}}$ = high: normal mode
$\overline{\text{ASEMD0}}$ = low: ASE mode
2. In ASE mode, the reset hold state is entered by driving the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins low for the given time. In this state, the CPU does not start up, even if the $\overline{\text{RES}}$ pin is driven high. After that, when the $\overline{\text{TRST}}$ pin is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is canceled by the following: another $\overline{\text{RES}}$ assert (power-on reset) or $\overline{\text{TRST}}$ reassert.

21.4.3 TDO Output Timing

The timing of data output from the TDO differs according to the command type set in 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, the TDO signal is output at the TCK rising edge earlier than the JTAG standard by a half cycle.

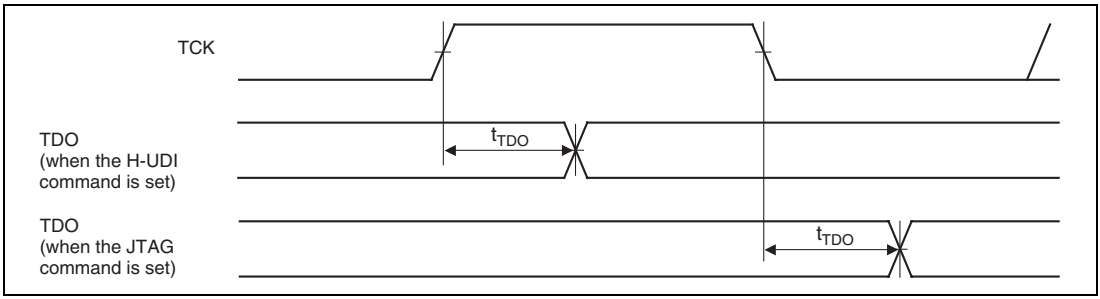


Figure 21.3 H-UDI Data Transfer Timing

21.4.4 H-UDI Reset

An H-UDI reset is generated by setting the 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 the 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.

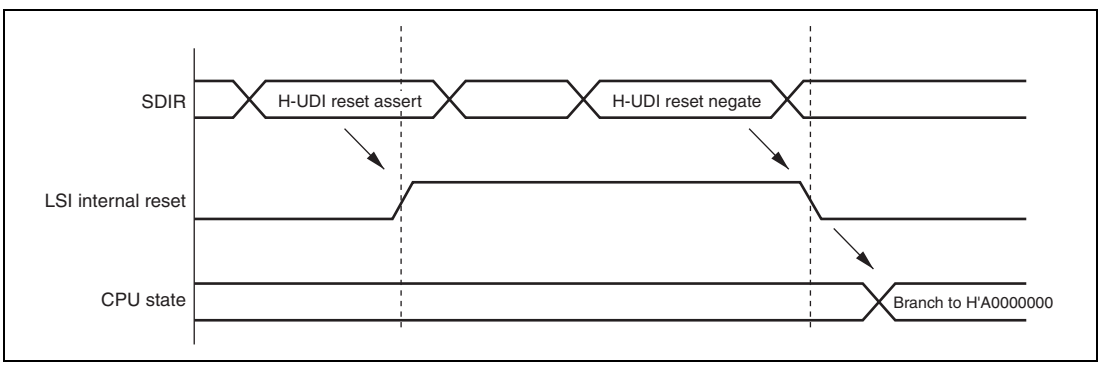


Figure 21.4 H-UDI Reset

21.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting an H-UDI command in SDIR. An H-UDI interrupt is an interrupt of general exceptions, 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, but not in standby mode.

21.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in boundary scan mode stipulated by JTAG.

21.5.1 Supported Instructions

This LSI supports the three mandatory instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

BYPASS: The BYPASS instruction is a mandatory 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 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs data from this LSI's internal circuitry to the boundary scan register, outputs data from the scan path, and loads data onto the scan path. While this instruction is executed, signals input to this LSI pins are transmitted directly to the internal circuitry, and internal circuit outputs 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 rising edge of the TCK signal 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 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 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.

21.5.2 Points for Attention

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

21.6 Usage Notes

- An H-UDI command, once set, will not be modified as long as another command is 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 LSI operations is once set.
- Because LSI operations are suspended in standby mode, H-UDI commands are not accepted. To hold the state of the TAP before and after standby mode, the TCK signal must be high during standby mode transition.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

Section 22 Ethernet Physical Layer Transceiver (PHY)

This LSI has an on-chip PHY module.

22.1 Features

- Fully-integrated IEEE 802.3/802.3u compliant 10/100Mbps Ethernet PHY
- PHY clock = 25 MHz, 3.3 V Analog power supply.
- Integrated DSP with adaptive-equalizer and Baseline Wander (BLW) correction High immunity to crosstalk
- Link-configuration automatically determined by Auto-negotiation / parallel detection; manual configuration also available
- Low power consumption
- Half- and Full-duplex capable for both 10 and 100 Mbps links
- Automatic Polarity Correction in 10Base-T
- Extended cable length option in 10Base-T
- MII interface to the CPU core of this LSI.
- Serial Management Interface (SMI)
- Link, Activity, Duplex and Speed LED outputs

Figure 22.1 shows the block diagram around the PHY module.

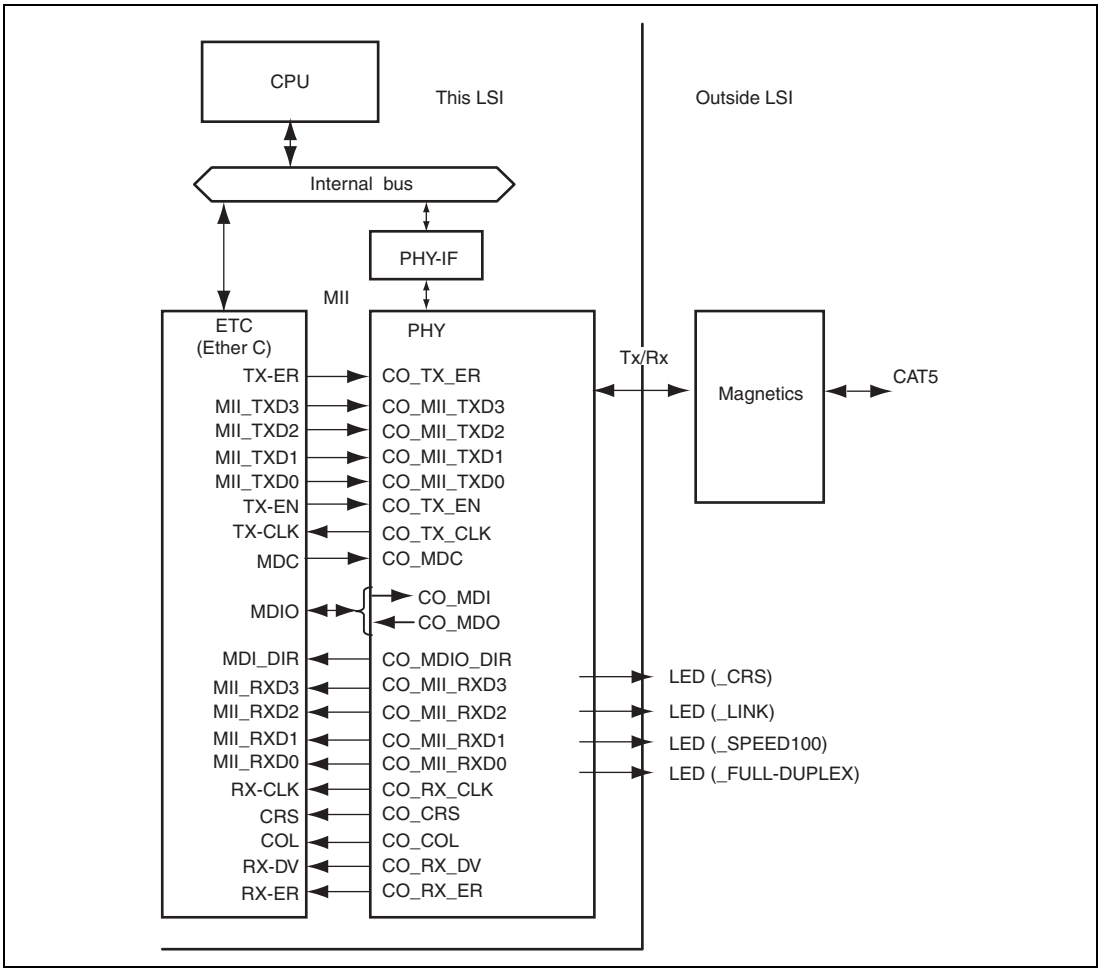


Figure 22.1 The Block Diagram around PHY Module

22.2 Pin Configuration

PHY module has below pins.

Table 22.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply 1 for PHY	Vcc1A	Input	Analog power supply for PHY
Analog power supply 2 for PHY	Vcc2A	Input	Analog power supply for PHY
Analog power supply 3 for PHY	Vcc3A	Input	Analog power supply for PHY
Analog ground 1 for PHY	Vss1A	Input	Analog ground for PHY
Analog ground 2 for PHY	Vss2A	Input	Analog ground for PHY
PHY clock	CK_PHY	Input	For providing the external clock for PHY. Of course you can provide a clock from internal clock pulse generator (CPG), but you have to pull up or down this pin in that case.
Differential transmit output (+)	TxP	Output	The differential transmit output (+) from PHY to Ethernet network
Differential transmit output (-)	TxM	Output	The differential transmit output (-) from PHY to Ethernet network
Differential receive input (+)	RxP	Input	The differential receive input (+) from Ethernet network to PHY
Differential receive input (-)	RxM	Input	The differential receive input (-) from Ethernet network to PHY
SPEED100 signal	SPEED100	Output	SPEED100 Output Low shows that the operating speed is 100 Mbit/s or during Auto negotiation
$\overline{\text{LINK}}$ signal	$\overline{\text{LINK}}$	Output	$\overline{\text{LINK}}$ Output (Low indicates that link is on.)
CRS signal	CRS	Output	CRS Output (Low indicates that there is CRS (carrier sense), keeps low after inactivation of CRS about 128 ms.)
DUPLEX signal	DUPLEX	Output	DUPLEX Output (Low indicates FULL DUPLEX)

Pin Name	Abbreviation	I/O	Function
Reference resistor	EXRES1	Input	Connect to the ground through a resistor of value 12.4Kohm, 1%.
Test input/output	TSTBUSA	I/O	I/O for test. Do not connect anything to this pin.

22.3 Top Level Functional Architecture

Functionally, this PHY module can be divided into the following sections:

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

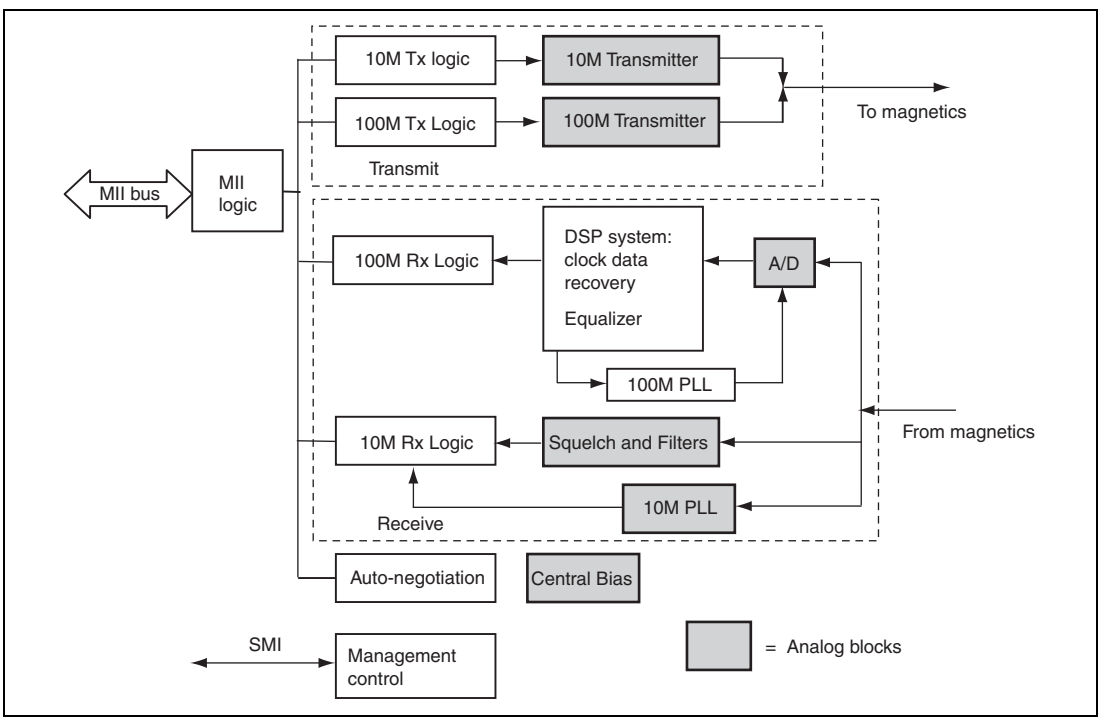


Figure 22.2 Architectural Overview

22.4 PHY Management Control

The Management Control module includes 2 blocks:

- Serial Management Interface (SMI)
- Management Registers Set

22.4.1 Serial Management Interface (SMI)

The Serial Management Interface is used to control this PHY core and obtain its status. This interface supports registers 0 through 6 as required by Clause 22 of the IEEE802.3 standard. Non-supported registers (7 to 15) will be read as hexadecimal "FFFF".

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

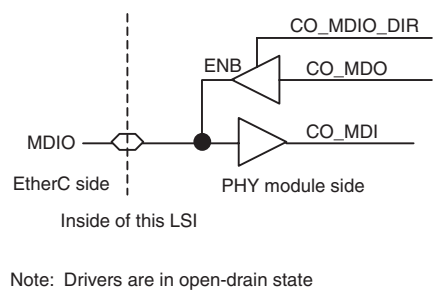


Figure 22.3 How to Derive MDIO Signal from Core Signals

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

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

The data on the CO_MDO and CO_MDI lines is latched on the rising edge of the CO_MDC. The frame structure and timing of the data is shown in figure 22.4 and figure 22.5.

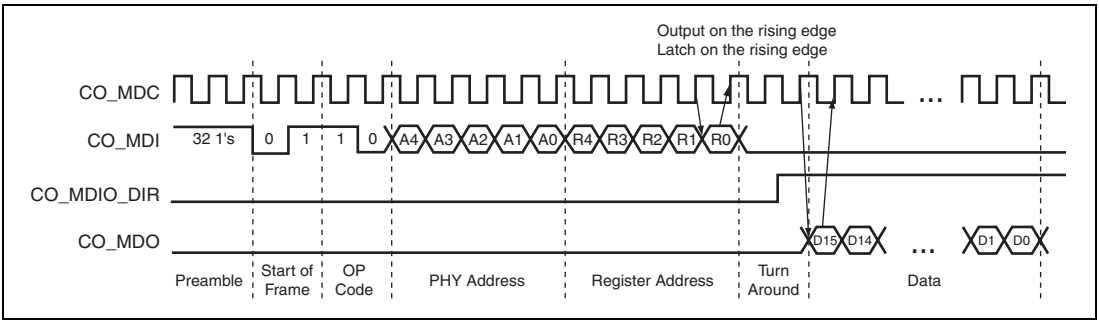


Figure 22.4 MDIO Timing and Frame Structure (READ Cycle)

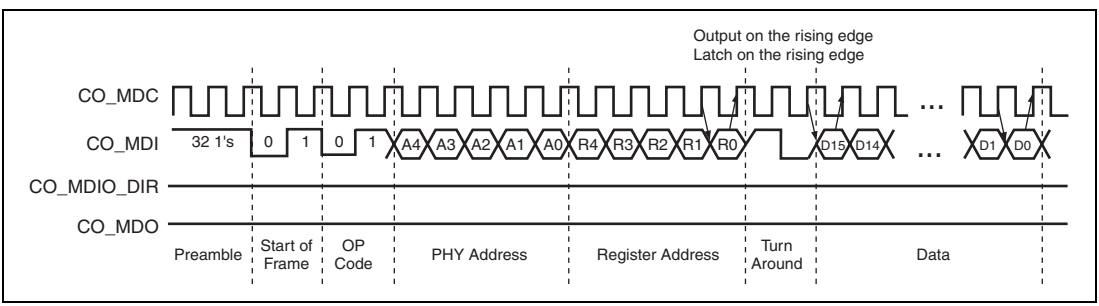


Figure 22.5 MDIO Timing and Frame Structure (WRITE Cycle)

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

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

```
/* SMI register read */
unsigned short        ether_reg_read( unsigned short reg_addr )
{
     unsigned short        data;

     phy_preamble();
     phy_reg_set( reg_addr, PHY_READ );
     phy_ta_z0();
     phy_reg_read( &data );
     mii_idle();
     return( data );
}

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

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

     i = 32;
     while( i > 0 )
     {
          mii_write_1();
          i--;
     }
```

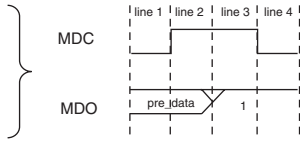
```
}

void        phy_reg_set( unsigned short reg_addr, long option )
{
     long        i;
     unsigned short        data;

     data = 0;
     data = (PHY_ST << 14);   /* ST code        */
     if( option == PHY_READ )
     {
          data |= (PHY_READ << 12);   /* OP code(RD)      */
     }
     else
     {
          data |= (PHY_WRITE << 12);  /* OP code(WT)      */
     }
     data |= (PHY_ADDR << 7); /* PHY Address    */
     data |= (reg_addr << 2); /* Reg Address    */

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

```
#define QuatA 6 // =25cyc/4  (Please define to keep, “MDC cycle > 400ns”)
void        phy_reg_read( unsigned short *data )
{
     long        i;
     long        j;
     unsigned short        reg_data;

     reg_data = 0;
     i = 16;

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

void        phy_reg_write( unsigned short data )
{
     long        i;

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

```
            }
            else
            {
                    mii_write_1();
            }
            i--;
            data <<= 1;
        }
}

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

void        phy_ta_10( void )
{
    mii_write_1();
    mii_write_0();
}

/* Output 1 */
void        mii_write_1( void )
{
  int j;
  unsigned short  pre_data;
```

```
  pre_data = REG_PIR&0x00000006; /* MDO,MMD */
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000 | pre_data; //line 1
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001 | pre_data; //line 2
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000007; //line 3
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000006; //line 4
}
/* Output 0 */
void        mii_write_0( void )
{
  int j;
  unsigned short  pre_data;

  pre_data = REG_PIR&0x00000006; /* MDO,MMD */
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000003;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000002;
}
/* Idle cycle */
void        mii_idle( void )
{
  int j;
  unsigned short  pre_data;

  pre_data = REG_PIR&0x00000006; /* MDO,MMD */
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000;
}
```

Example of corresponding waveforms



22.4.2 SMI Register Mapping

The following registers are supported (register numbers are in decimal):

Register #	Description	Group
0	Basic Control Register	Basic
1	Basic Status Register	Basic
2	PHY Identifier 1	Extended
3	PHY Identifier 2	Extended
4	Auto-Negotiation Advertisement Register	Extended
5	Auto-Negotiation Link Partner Ability Register	Extended
6	Auto-Negotiation Expansion Register	Extended

- SMI Register Format

  The mode key is as follows:

  RW = read/write, SC = self clearing, WO = write only, RO = read only

  LH = latch high, clear on read of register

  LL = latch low, clear on read of register

  NASR = Not Affected by Software Reset

(n,m) = register n, bit m

- Register 0 (Basic Control)

Address	Name	Description	Mode	Default
0.15	Reset	1 = software reset. This bit is self-clearing. With this LSI, do not use this bit.	RW/SC	0
0.14	Loopback	1 = loopback mode, 0 = normal operation	RW	0
0.13	Speed Select	1 = 100Mbps, 0 = 10Mbps. Ignored if Auto Negotiation is enabled (0.12 = 1).	RW	Set by co_st_mode [2:0] of PHYIFCR
0.12	Auto-Negotiation Enable	1 = enable auto-negotiate process (overrides 0.13 and 0.8), 0 = disable auto-negotiate process	RW	Set by co_st_mode [2:0] of PHYIFCR
0.11	Power Down	1 = General power down mode, 0 = normal operation	RW	0
0.10	Isolate	Reserved. (0= normal operation)The write value should always be 0.	RW	Set by co_st_mode [2:0] of PHYIFCR
0.9	Restart Auto-Negotiate	1 = restart auto-negotiate process, 0 = normal operation. Bit is self-clearing.	RW/SC	0
0.8	Duplex Mode	1 = full duplex, 0 = half duplex . Ignored if Auto Negotiation is enabled (0.12 = 1).	RW	Set by co_st_mode [2:0] of PHYIFCR
0.7	Collision Test	1 = enable COL test, 0 = disable COL test	RW	0
0.6:0	Reserved	The write value should always be 0.	RO	0

- Register 1 (Basic Status)

Address	Name	Description	Mode	Default
1.15	100Base-T4	This bit indicates that T4 is available or not, but this module has no capability of T4 and it is fixed to 0.The write value should always be 0.	RO	0
1.14	100Base-TX Full Duplex	1 = TX with full duplex, 0 = no TX full duplex ability	RO	1
1.13	100Base-TX Half Duplex	1 = TX with half duplex, 0 = no TX half duplex ability	RO	1
1.12	10Base-T Full Duplex	1 = 10Mbps with full duplex, 0 = no 10Mbps with full duplex ability	RO	1
1.11	10Base-T Half Duplex	1 = 10Mbps with half duplex, 0 = no 10Mbps with half duplex ability	RO	1
1.10:6	Reserved	The write value should always be 0.	RO	0
1.5	Auto-Negotiate Complete	1 = auto-negotiate process completed, 0 = auto-negotiate process not completed	RO	0
1.4	Remote Fault	1 = remote fault condition detected, 0 = no remote fault	RO/LH	0
1.3	Auto-Negotiate Ability	1 = able to perform auto-negotiation function, 0 = unable to perform auto-negotiation function	RO	1
1.2	Link Status	1 = link is up, 0 = link is down	RO/LL	0
1.1	Jabber Detect	1 = jabber condition detected, 0 = no jabber condition detected	RO/LH	0
1.0	Extended Capabilities	1 = supports extended capabilities registers, 0 = does not support extended capabilities registers	RO	1

- Register 2 (PHY Identifier 1)

Address	Name	Description	Mode	Default
2.15:0	PHY ID Number	Assigned to the 3rd through 18th bits of the Organizationally Unique Identifier (OUI), respectively.	RW	co_reg2_oui_in [15:0] of PHYIFSMIR2

- Register 3 (PHY Identifier 2)

Address	Name	Description	Mode	Default
3.15:10	PHY ID Number b	Assigned to the 19th through 24th bits of the OUI.	RW	co_reg3_oui_in[15:0] of PHYIFSMIR3
3.9:4	Model Number	Six bit manufacturer's model number.	RW	
3.3:0	Revision Number	Four bit manufacturer's revision number.	RW	

- Register 4 (Auto Negotiation Advertisement)

Address	Name	Description	Mode	Default
4.15	Next Page	This bit indicates next page is available or not, but this core does not support next page ability and it is fixed to 0.The write value should always be 0.	RO	0
4.14	Reserved	The write value should always be 0.	RO	0
4.13	Remote Fault	1 = remote fault detected, 0 = no remote fault	RW	0
4.12	Reserved	The write value should always be 0.	R/W	0
4.11:10	Pause Operation	00 No PAUSE, 01 Asymmetric PAUSE toward link partner, 10 Symmetric PAUSE, 11 Both Symmetric PAUSE and Asymmetric PAUSE toward local device	R/W	00
4.9	100Base-T4	Reserved. The write value should always be 0.	RO	0
4.8	100Base-TX Full Duplex	1 = TX with full duplex, 0 = no TX full duplex ability	RW	Set by co_st_mode [2:0] of PHYIFCR
4.7	100Base-TX	1 = TX able, 0 = no TX ability	RW	1
4.6	10Base-T Full Duplex	1 = 10Mbps with full duplex, 0 = no 10Mbps with full duplex ability	RW	Set by co_st_mode [2:0] of PHYIFCR

Address	Name	Description	Mode	Default
4.5	10Base-T	1 = 10Mbps able, 0 = no 10Mbps ability	RW	Set by co_st_mode [2:0] of PHYIFCR
4.4:0	Selector Field	[00001] = IEEE 802.3	RW	00001

- Register 5 (Auto Negotiation Link Partner Ability)

Address	Name	Description	Mode	Default
5.15	Next Page	1 = next page capable 0 = no next page ability. This part does not support next page ability.	RO	0
5.14	Acknowledge	1 = link code word received from partner 0 = link code word not yet received	RO	0
5.13	Remote Fault	1 = remote fault detected 0 = no remote fault	RO	0
5.12:11	Reserved	The write value should always be 0.	RO	0
5.10	Pause Operation	1 = Pause Operation is supported by remote MAC 0 = Pause Operation is not supported by remote MAC	RO	0
5.9	100Base-T4	1 = T4 able, 0 = no T4 ability	RO	0
5.8	100Base-TX Full Duplex	1 = TX with full duplex 0 = no TX full duplex ability	RO	0
5.7	100Base-TX	1 = TX able, 0 = no TX ability	RO	0
5.6	10Base-T Full Duplex	1 = 10Mbps with full duplex 0 = no 10Mbps with full duplex ability	RO	0
5.5	10Base-T	1 = 10Mbps able 0 = no 10Mbps ability	RO	0
5.4:0	Selector Field	[00001] = IEEE 802.3	RO	00001

- Register 6 (Auto Negotiation Expansion)

Address	Name	Description	Mode	Default
6.15:5	Reserved	The write value should always be 0.	RO	0
6.4	Parallel Detection Fault	1 = fault detected by parallel detection logic 0 = no fault detected by parallel detection logic	RO/LH	0
6.3	Link Partner Next Page Able	1 = link partner has next page ability 0 = link partner does not have next page ability	RO	0
6.2	Next Page Able	1 = local device has next page ability 0 = local device does not have next page ability	RO	0
6.1	Page Received	1 = new page received 0 = new page not yet received	RO/LH	0
6.0	Link Partner Auto-Negotiation Able	1 = link partner has auto-negotiation ability, 0 = link partner does not have auto-negotiation ability	RO	0

22.5 100Base-TX Transmit

The data path of the 100Base-TX is shown in figure 22.6. Each major block is explained below.

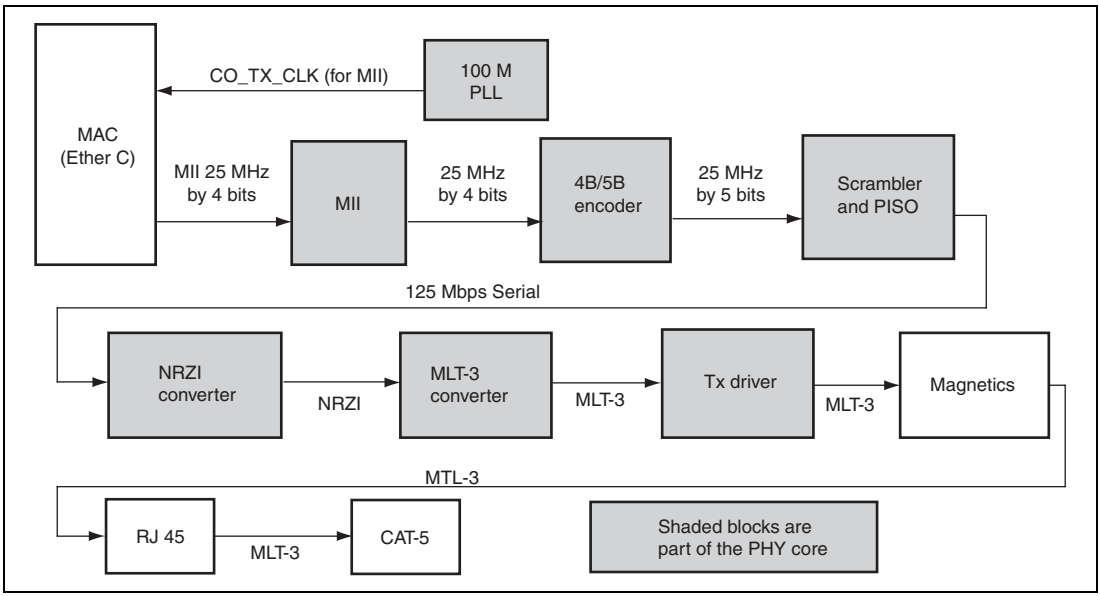


Figure 22.6 100Base-TX Data Path

(1) 100M Transmit Data across the MII

The MAC controller drives the transmit data onto the CO_MII_TXD bus and asserts the internal signal (CO_TX_EN) to indicate valid data. The data is latched by the PHY's MII block on the rising edge of CO_TX_CLK. The data is in the form of 4-bit wide 25MHz data.

(2) 4B/5B Encoding

The transmit data passes from the MII block to the 4B/5B encoder. This block encodes the data from 4-bit nibbles to 5-bit symbols (known as "code-groups") according to table 22.2. Each 4-bit data-nibble is mapped to 16 of the 32 possible code-groups. The remaining 16 code-groups are either used for control information or are not valid.

The first 16 code-groups are referred to by the hexadecimal values of their corresponding data nibbles, 0 through F. The remaining code-groups are given letter designations with slashes on either side. For example, an IDLE code-group is /I/, a transmit error code-group is /H/, etc.

Table 22.2 4B/5B Code Table

CODE GROUP	SYM	RECEIVER INTERPRETATION (CO_MII_RXD)			TRANSMITTER INTERPRETATION (CO_MII_TXD)
11110	0	0	0000	DATA 0	
01001	1	1	0001	DATA 1	
10100	2	2	0010	DATA 2	
10101	3	3	0011	DATA 3	
01010	4	4	0100	DATA 4	
01011	5	5	0101	DATA 5	
01110	6	6	0110	DATA 6	
01111	7	7	0111	DATA 7	
10010	8	8	1000	DATA 8	
10011	9	9	1001	DATA 9	
10110	A	A	1010	DATA A	
10111	B	B	1011	DATA B	
11010	C	C	1100	DATA C	
11011	D	D	1101	DATA D	
11100	E	E	1110	DATA E	
11101	F	F	1111	DATA F	
11111	I	IDLE			Sent after /T/R/ until CO_TX_EN
11000	J	First nibble of SSD, translated to "0101" following IDLE, else CO_RX_ER			Sent for rising CO_TX_EN
10001	K	Second nibble of SSD, translated to "0101" following J, else CO_RX_ER			Sent for rising CO_TX_EN
01101	T	First nibble of ESD, causes de-assertion of CRS if followed by /R/, else assertion of CO_RX_ER			Sent for falling CO_TX_EN
00111	R	Second nibble of ESD, causes deassertion of CRS if following /T/, else assertion of CO_RX_ER			Sent for falling CO_TX_EN
00100	H	Transmit Error Symbol			Sent for rising CO_TX_ER
00110	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID
11001	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID
00000	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID

CODE GROUP	SYM	RECEIVER INTERPRETATION (CO_MII_RXD)	TRANSMITTER INTERPRETATION (CO_MII_TXD)
00001	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
00010	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
00011	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
00101	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
01000	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
01100	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID
10000	V	INVALID, CO_RX_ER if during CO_RX_DV	INVALID

(3) Scrambling

Repeated data patterns (especially the IDLE code-group) can have power spectral densities with large narrow-band peaks. Scrambling the data helps eliminate these peaks and spread the signal power more uniformly over the entire channel bandwidth. This uniform spectral density is required by FCC regulations to prevent excessive EMI from being radiated by the physical routing.

The seed for the scrambler is generated from the PHY address. The scrambler also performs the Parallel In Serial Out conversion (PISO) of the data.

(4) NRZI and MLT3 Encoding

The scrambler block passes the 5-bit wide parallel data to the NRZI converter where it becomes a serial 125MHz NRZI data stream. The NRZI is encoded to MLT-3. MLT3 is a tri-level code where a change in the logic level represents a code bit "1" and the logic output remaining at the same level represents a code bit "0".

(5) 100M Transmit Driver

The MLT3 data is then passed to the analog transmitter, which launches the differential MLT-3 signal, on outputs TXP and TXM, to the twisted pair media via a 1:1 ratio isolation transformer. The 10Base-T and 100Base-TX signals pass through the same transformer so that common "magnetics" can be used for both. The transmitter drives into the 100 ohm impedance of the CAT-5 cable. Cable termination and impedance matching require external components.

(6) 100M Phase Lock Loop (PLL)

The 100M PLL locks onto reference clock and generates the 125MHz clock used to drive the 125 MHz logic and the 100Base-Tx Transmitter.

22.6 100Base-TX Receive

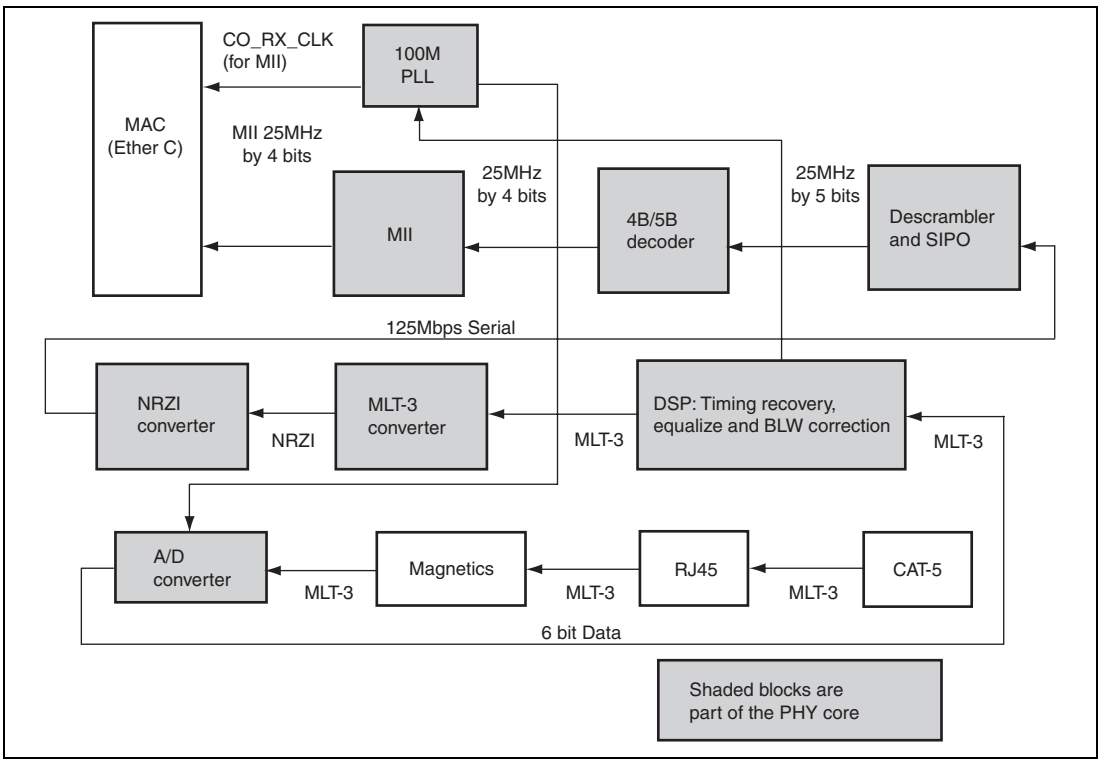


Figure 22.7 Receive Data Path

The receive data path is shown in figure 22.7. Detailed descriptions are given below.

(1) 100M Receive Input

The MLT-3 from the cable is fed into the Core PHY (on inputs RXP and RXM) via a 1:1 ratio transformer. The ADC samples the incoming differential signal at a rate of 125M samples per second. Using a 64-level quantizer it generates 6 digital bits to represent each sample. The DSP adjusts the gain of the ADC according to the observed signal levels such that the full dynamic range of the ADC can be used.

(2) Equalizer, Baseline Wander Correction and Clock and Data Recovery

The 6 bits from the ADC are fed into the DSP block. The equalizer in the DSP section compensates for phase and amplitude distortion caused by the physical channel consisting of magnetics, connectors, and CAT- 5 cable. The equalizer can restore the signal for any good-quality CAT-5 cable between 1m and 150m.

If the DC content of the signal is such that the low-frequency components fall below the low frequency pole of the isolation transformer, then the droop characteristics of the transformer will become significant and Baseline Wander (BLW) on the received signal will result. To prevent corruption of the received data, the Core PHY corrects for BLW and can receive the ANSI X3.263-1995 FDDI TP-PMD defined "killer packet" with no bit errors.

The 100M PLL generates multiple phases of the 125MHz clock. A multiplexer, controlled by the timing unit of the DSP, selects the optimum phase for sampling the data. This is used as the received recovered clock. This clock is used to extract the serial data from the received signal.

(3) NRZI and MLT-3 Decoding

The DSP generates the MLT-3 recovered levels that are fed to the MLT-3 converter. The MLT-3 is then converted to an NRZI data stream.

(4) Descrambling

The descrambler performs an inverse function to the scrambler in the transmitter and also performs the Serial In Parallel Out (SIPO) conversion of the data.

During reception of IDLE (/I/) symbols. the descrambler synchronizes its descrambler key to the incoming stream. Once synchronization is achieved, the descrambler locks on this key and is able to descramble incoming data.

Special logic in the descrambler ensures synchronization with the remote PHY by searching for IDLE symbols within a window of 4000 bytes. This window ensures that a maximum packet size of 1514 bytes, allowed by the IEEE 802.3 standard, can be received with no interference.

If no IDLE-symbols are detected within this time-period, receive operation is aborted and the descrambler re-starts the synchronization process.

(5) Alignment

The de-scrambled signal is then aligned into 5-bit code-groups by recognizing the /J/K/ Start-of-Stream Delimiter (SSD) pair at the start of a packet. Once the code-word alignment is determined, it is stored and utilized until the next start of frame.

(6) 5B/4B Decoding

The 5-bit code-groups are translated into 4-bit data nibbles according to the 4B/5B table. The translated data is presented on the CO_MII_RXD[3:0] signal lines. The SSD, /J/K/, is translated to "0101 0101" as the first 2 nibbles of the MAC preamble. Reception of the SSD causes the PHY to assert the CO_RX_DV signal, indicating that valid data is available on the CO_MII_RXD bus. Successive valid code-groups are translated to data nibbles. Reception of either the End of Stream Delimiter (ESD) consisting of the /T/R/ symbols, or at least two /I/ symbols causes the PHY to de-assert carrier sense and CO_RX_DV.

These symbols are not translated into data.

(7) Receive Data Valid Signal

The Receive Data Valid signal (CO_RX_DV) indicates that recovered and decoded nibbles are being presented on the CO_MII_RXD[3:0] outputs synchronous to CO_RX_CLK. CO_RX_DV becomes active after the /J/K/ delimiter has been recognized and CO_MII_RXD is aligned to nibble boundaries. It remains active until either the /T/R/ delimiter is recognized or link test indicates failure, etc.

CO_RX_DV is asserted when the first nibble of translated /J/K/ is ready for transfer over the Media Independent Interface (MII).

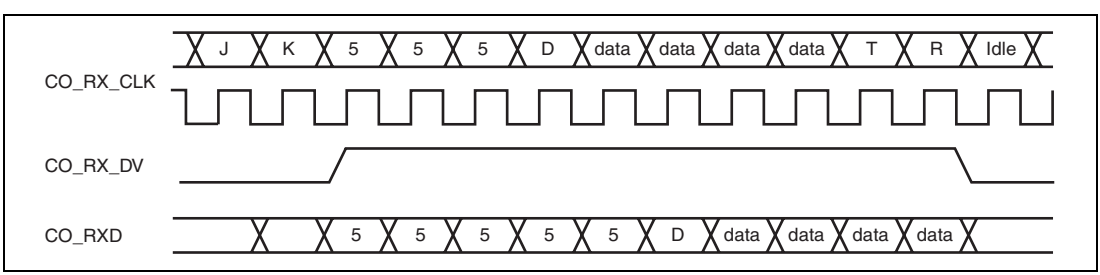


Figure 22.8 Relationship between Received Data and Some MII Signals

(8) Receiver Errors

During a frame, unexpected code-groups are considered receive errors. Expected code groups are the DATA set (0 through F), and the /T/R/ (ESD) symbol pair. When a receive error occurs, the CO_RX_ER signal is asserted and arbitrary data is driven onto the CO_MII_RXD lines. Should an error be detected during the time that the /J/K/ delimiter is being decoded (bad SSD error), CO_RX_ER is asserted true and the value '1110' is driven onto the CO_MII_RXD lines. Note that the Valid Data signal is not yet asserted when the bad SSD error occurs.

(9) 100M Receive Data across the MII

The 4-bit data nibbles are sent to the MII block. These data nibbles are clocked to the controller at a rate of 25MHz. The controller samples the data on the rising edge of CO_RX_CLK. CO_RX_CLK is the 25MHz output clock for the MII bus. It is recovered from the received data to clock the CO_MII_RXD bus. If there is no received signal, it is derived from the system reference clock (CO_CLKIN).

When tracking the received data, CO_RX_CLK has a maximum jitter of 0.8ns (provided that the jitter of the input clock, CO_CLKIN, is below 100ps).

22.7 10Base-T Transmit

Data to be transmitted comes from the MAC layer controller. The 10Base-T transmitter receives 4-bit nibbles from the MII at a rate of 2.5MHz and converts them to a 10Mbps serial data stream. The data stream is then Manchester-encoded and sent to the analog transmitter which drives a signal onto the twisted pair via the external magnetics.

The 10M transmitter uses the following blocks:

- MII (digital)
- TX 10M (digital)
- 10M Transmitter (analog)
- 10M PLL (analog)

(1) 10M Transmit Data across the MII

The MAC controller (EtherC) drives the transmit data onto the CO_MII_TXD BUS. When the controller (EtherC) has driven CO_TX_EN high to indicate valid data, the data is latched by the MII block on the rising edge of CO_TX_CLK. The data is in the form of 4-bit wide 2.5 MHz data.

In order to comply with legacy 10Base-T MAC/Controllers, in Half-duplex mode the PHY loops back the transmitted data, on the receive path. This does not confuse the MAC/Controller since the CO_COL signal is not asserted during this time. The PHY also support the SQE (Heart beat) signal.

(2) Manchester Encoding

The 4-bit wide data is sent to the TX10M block. The nibbles are converted to a 10Mbps serial NRZI data stream. The 10M PLL locks onto the external clock or internal oscillator and produces a 20MHz clock. This is used to Manchester encode the NRZ data stream. When no data is being transmitted (CO_TX_EN is low, the TX10M block outputs Normal Link Pulses (NLPs) to maintain communications with the remote link partner.

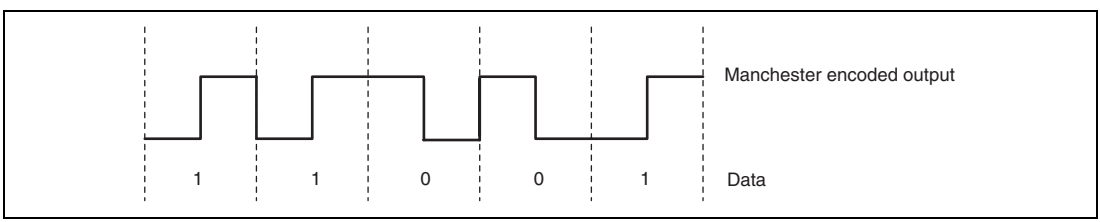


Figure 22.9 Manchester Encoded Output

(3) 10M Transmit Drivers

The Manchester encoded data is sent to the analog transmitter where it is shaped and filtered before being driven out as a differential signal across the TXP and TXM outputs.

22.8 10Base-T Receive

The 10Base-T receiver gets the Manchester- encoded analog signal from the cable via the magnetics. It recovers the receive clock from the signal and uses this clock to recover the NRZI data stream. This 10M serial data is converted to 4-bit data nibbles which are passed to the controller (EtherC) across the MII at a rate of 2.5 MHz.

This 10M receiver uses the following blocks:

- Filter and SQUELCH (analog)
- 10M PLL (analog)
- RX 10M (digital)
- MII (digital)

(1) 10M Receive Input and Squelch

The Manchester signal from the cable is fed into the core PHY (on inputs RXP and RXM) via 1:1 ratio magnetics. It is first filtered to reduce any out-of-band noise. It then passes through a SQUELCH circuit. The SQUELCH is a set of amplitude and timing comparators that normally reject differential voltage levels below 300mV and detect and recognize differential voltages above 585mV.

(2) Manchester Decoding

The output of the SQUELCH goes to the RX10M block where it is validated as Manchester encoded data. The polarity of the signal is also checked. If the polarity is reversed (local RXP is connected to RXM of the remote partner and vice versa), then this is identified and corrected. The 10M PLL is locked onto the received Manchester signal and from this, generates the received 20MHz clock. Using this clock, the Manchester encoded data is extracted and converted to a 10MHz NRZI data stream. It is then converted from serial to 4-bit wide parallel data.

The RX10M block also detects valid 10Base-T IDLE signals, Normal Link Pulses (NLPs), to maintain the link.

(3) 10M Receive Data across the MII

The 4 bit data nibbles are sent to the MII block. These data nibbles are valid on the rising edge of the 2.5 MHz CO_RX_CLK.

(4) Jabber detection

Jabber is a condition in which a station transmits for a period of time longer than the maximum permissible packet length, usually due to a fault condition, that results in holding the CO_TX_EN input for a long period. Special logic is used to detect the jabber state and abort the transmission to the line, within 45ms. Once CO_TX_EN is deasserted, the logic resets the jabber condition.

Bit 1.1 indicates that a jabber condition was detected.

22.9 MAC Interface

The MII (Media Independent Interface) block is responsible for the communication with the controller (EtherC). Special sets of hand-shake signals are used to indicate that valid received/transmitted data is present on the 4 bit receive/transmit bus.

(1) The MII includes 16 interface signals:

- transmit data: CO_MII_TXD[3:0]
- transmit strobe: CO_TX_EN
- transmit: CO_TX_CLK
- transmit error: CO_TX_ER
- receive data: CO_MII_RXD[3:0]
- receive strobe: CO_RX_DV
- receive clock: CO_RX_CLK
- receive error: CO_RX_ER
- collision indication: CO_COL
- carrier sense: CO_CRS

On the transmit path, the PHY drives the transmit clock, CO_TX_CLK, to the controller (EtherC). The controller (EtherC) synchronizes the transmit data to the rising edge of CO_TX_CLK. The controller (EtherC) drives CO_TX_EN high to indicate valid transmit data. The controller (EtherC) drives CO_TX_ER high when a transmit error is detected.

On the receive path, the PHY drives both the receive data, CO_RXD, and the CO_RX_CLK signal. The controller (EtherC) clocks in the receive data on the rising edge of CO_RX_CLK when the PHY drives CO_RX_DV high. The PHY drives CO_RX_ER high when a receive error is detected.

(2) Auto-negotiation

The purpose of the Auto-negotiation function is to automatically configure the PHY to the optimum link parameters based on the capabilities of its link partner. Auto-negotiation is a mechanism for exchanging configuration information between two link-partners and automatically selecting the highest performance mode of operation supported by both sides. Auto-negotiation is fully defined in clause 28 of the IEEE 802.3 specification.

Once auto-negotiation has completed, information about the resolved link can be passed back to the controller (EtherC) via the Serial Management Interface (SMI). The results of the negotiation process are reflected in the Link Partner Ability Register (Register 5).

The auto-negotiation protocol is a purely physical layer activity and proceeds independently of the MAC controller (EtherC).

The advertised capabilities of the PHY are stored in register 4 of the SMI registers. The default advertised by the core PHY is determined by user-defined on-chip signal options. (i.e. the configuration of PHY-IF)

The following blocks are activated during an Auto-negotiation session:

- Auto-negotiation (digital)
- 100M ADC (analog)
- 100M PLL (analog)
- 100M equalizer/BLW/clock recovery (DSP)
- 10M SQUELCH (analog)
- 10M PLL (analog)
- 10M Transmitter (analog)

When enabled, auto-negotiation is started by the occurrence of one of the following events:

- Module reset (co_resetb of PHY-IF)
- PHY power on reset
- Software reset
- Power-down reset
- Link status down
- Setting register 0, bit 9 high (auto-negotiation restart)

On detection of one of these events, the PHY begins auto-negotiation by transmitting bursts of Fast Link Pulses (FLP). These are bursts of link pulses from the 10M transmitter. They are shaped as Normal Link Pulses and can pass uncorrupted down CAT-3 or CAT-5 cable. A Fast Link Pulse Burst consists of up to 33 pulses. The 17 odd-numbered pulses, which are always present, frame the FLP burst. The 16 even-numbered pulses, which may be present or absent, contain the data word being transmitted. Presence of a data pulse represents a "1", while absence represents a "0".

The data transmitted by an FLP burst is known as a "Link Code Word." These are defined fully in IEEE 802.3 clause 28. In summary, the Core PHY advertises 802.3 compliance in its selector field (the first 5 bits of the Link Code Word). It advertises its technology ability according to the bits set in register 4 of the SMI registers.

There are 4 possible matches of the technology abilities. In the order of priority these are:

- 100M Full Duplex (Highest priority)
- 100M Half Duplex
- 10M Full Duplex
- 10M Half Duplex

If the full capabilities of the core PHY are advertised (100M, Full Duplex), and if the link partner is capable of 10M and 100M, then auto-negotiation selects 100M as the highest performance mode. If the link partner is capable of Half and Full duplex modes, then auto-negotiation selects Full Duplex as the highest performance operation.

Once a capability match has been determined, the link code words are repeated with the acknowledge bit set. Any difference in the main content of the link code words at this time will cause auto-negotiation to re-start. Auto-negotiation will also re-start if not all of the required FLP bursts are received.

The capabilities advertised during auto-negotiation by the core PHY are initially determined the co_st_mode[2:0] bits (PHYIFCR in the PHY-IF) latched after Module reset or PHY power on reset completes. This bit can also be used to disable auto-negotiation on power-up.

Writing register 4 bits [8:5] allows software control of the capabilities advertised by the core PHY. Writing register 4 does not automatically re-start auto-negotiation. Register 0, bit 9 must be set before the new abilities will be advertised. Auto-negotiation can also be disabled via software by clearing register 0, bit 12.

The PHY module does not support the Next Page capability.

(3) Parallel Detection

If the PHY module is connected to a device lacking the ability to auto-negotiate (i.e. no FLPs are detected), it is able to determine the speed of the link based on either 100M MLT-3 symbols or 10M Normal Link Pulses. In this case the link is presumed to be Half Duplex per the IEEE standard. This ability is known as "Parallel Detection. This feature ensures interoperability with legacy link partners. If a link is formed via parallel detection, then bit 0 in register 6 is cleared to indicate that the Link Partner is not capable of auto-negotiation. The controller (EtherC) has access to this information via the management interface (SMI). If a fault occurs during parallel detection, bit 4 of register 6 is set.

Register 5 is used to store the Link Partner Ability information, which is coded in the received FLPs. If the Link Partner is not auto-negotiation capable, then register 5 is updated after completion of parallel detection to reflect the speed capability of the Link Partner.

(4) Re-starting Auto-negotiation

Auto-negotiation can be re-started at any time by setting register 0, bit 9. Auto-negotiation will also re-start if the link is broken at any time. A broken link is caused by signal loss. This may occur because of a cable break, or because of an interruption in the signal transmitted by the Link Partner. Auto-negotiation resumes in an attempt to determine the new link configuration.

If the management entity re-starts Auto-negotiation by writing to bit 9 of the control register, the PHY module will respond by stopping all transmission/receiving operations. Once the break_link_timer is done, in the Auto-negotiation state-machine (approximately 1200ms) the auto-negotiation will re-start. The Link Partner will have also dropped the link due to lack of a received signal, so it too will resume auto-negotiation detection is disabled.

(5) Auto-negotiation Disabling

Auto-negotiation is disabled by setting the bit 12 in the register 0 to 0. The device forcibly reflects the information in the bit 13 (SPEED) and bit 8 (Duplex) in the register 0 to the operation speed. Information in the bit 13 (SPEED) and bit 8 (Duplex) in the register 0 is ignored while auto-negotiation is enabled.

(6) Half-duplex and Full-duplex

Half-duplex operation conforms to CSMA/CD (Carrier Sense Multiple Access/Collision Detect) protocol that deals with the network traffic and collision. In this mode, the carrier signal (CRS) supports either of transmit/receive operation. Receiving data during PHY transmission causes a collision.

In full-duplex mode, the PHY performs transmit and receive simultaneously. In this mode, the CRS supports only receive. The CSMA/CD protocol is not applied and the collision detection is disabled.

22.10 Miscellaneous Functions

(1) Carrier Sense

The carrier sense is output on CRS (to EtherC). CRS is a signal defined by the MII specification in the IEEE 802.3u standard. The PHY asserts CRS based only on receive activity whenever the PHY is either in repeater mode or full-duplex mode. Otherwise the PHY asserts CRS based on either transmit or receive activity.

The carrier sense logic uses the encoded, unscrambled data to determine carrier activity status. It activates carrier sense with the detection of 2 non-contiguous zeros within any 10 bit span. Carrier sense terminates if a span of 10 consecutive ones is detected before a /J/K/ Start-of Stream Delimiter pair. If an SSD pair is detected, carrier sense is asserted until either /T/R/ End-of-Stream Delimiter pair or a pair of IDLE symbols is detected. Carrier is negated after the /T/ symbol or the first IDLE. If /T/ is not followed by /R/, then carrier is maintained. Carrier is treated similarly for IDLE followed by some non-IDLE symbol.

(2) Collision Detect

A collision is the occurrence of simultaneous transmit and receive operations. The CO_COL output is asserted to indicate that a collision has been detected. CO_COL remains active for the duration of the collision. CO_COL is changed asynchronously to both CO_RX_CLK and TX_CLK. The CO_COL output becomes inactive during full duplex mode.

CO_COL may be tested by setting register 0, bit 7 high. This enables the collision test. CO_COL will be asserted within 512 bit times of CO_TX_EN rising and will be de-asserted within 4 bit times of CO_TX_EN falling.

In 10M mode, CO_COL pulses for approximately 10 bit times (1us), 2us after each transmitted packet (de-assertion of CO_TX_EN). This is the Signal Quality Error (SQE) signal and indicates that the transmission was successful.

(3) Isolate Mode

The ordinary external PHY LSI has a ability to make PHY data paths electrically isolated from the MII by setting register 0, bit 10 to a logic one.

But this PHY core is on-chip type so that this function is not supported.

(4) Link integrity Test

This PHY performs the link integrity test as outlined in the IEEE 802.3u (Clause 24-15) Link Monitor state diagram. The link status is multiplexed with the 10Mbps link status to form the reportable link status bit in Serial Management Register 1, and is driven to LINK LED.

The DSP indicates a valid MLT-3 waveform present on the RXP and RXM signals as defined by the ANSI X3.263 TP-PMD standard, to the Link Monitor state-machine, using internal signal called DATA_VALID. When DATA_VALID is asserted the control logic moves into a Link-Ready state, and waits for an enable from the Auto Negotiation block. When received, the Link-Up state is entered, and the Transmit and Receive logic blocks become active. Should Auto Negotiation be disabled, the link integrity logic moves immediately to the Link-Up state, when the DATA_VALID is asserted.

Note that to allow the line to stabilize, the link integrity logic will wait a minimum of 330 msec from the time DATA_VALID is asserted until the Link-Ready state is entered. Should the DATA_VALID input be negated at any time, this logic will immediately negate the Link signal and enter the Link-Down state.

When the 10/100 digital block is in 10Base-T mode, the link status is from the 10Base-T receiver logic.

(5) Power-Down modes

There is a power-down modes for the core:

- Power-Down
  This power-down is controlled by register 0, bit 11. In this mode the entire PHY, except the management interface, is powered-down and stays in that condition as long as bit 0.11 is HIGH. When bit 0.11 is cleared, the PHY powers up and is automatically reset.

(6) Reset

The core PHY has 4 reset sources:

- Module reset (co_resetb):

  It is connected to the co_resetb of PHYIFCR, and to the internal POR signal.

  If the co_resetb is asserted(write "0"), it should be held "0" for at least 100 us to ensure that the core is properly reset.
- The Power-On-Reset (POR) :

  POR(Power-On-Reset) signal, which is driven out of the core through the co_pwruprst of PHYIFSR, is asserted for approximately 16 ms after the first time that power is supplied to the chip.
- Software (SW) reset: (Do not use with this product.)

  Activated by writing register 0, bit 15 high. This signal is self- clearing. After the register-write, internal logic extends the reset by 256μs to allow PLL-stabilization before releasing the logic from reset.

  The IEEE 802.3u standard, clause 22 (22.2.4.1.1) states that the reset process should be completed within 0.5s from the setting of this bit.
- Power-Down reset:

  Automatically activated when the PHY comes out of power-down mode. The internal power-down reset is extended by 256μs after exiting the power-down mode to allow the PLLs to stabilize before the logic is released from reset.

These 4 reset sources are Module reset(Low active) and none Module reset(PHY power on reset, software reset, power down reset(High active) combined together in the digital block to create the internal "general reset", SYSRST, which is an asynchronous reset and is active HIGH. This SYSRST directly drives the PCS, DSP and MII blocks. It is also input to the Central Bias block in order to generate a short reset for the PLLs.

The SMI mechanism and registers are reset only by the Module reset, PHY power-on reset and Software reset. During Power-Down, the SMI registers are not reset. Note that some SMI register bits are not cleared by Software reset - these are marked "NASR" in the register tables.

For the first 16us after coming out of reset, the MII will run at 2.5 MHz. After that it will switch to 25 MHz if auto-negotiation is enabled.

(7) LED Description

The PHY provides four LED signals. These provide a convenient means to determine the mode of operation of the core. All LED signals are active low.

- The CRS LED :

  Its output is driven low when CRS is active (high). When CRS becomes inactive, the Activity LED output is extended by 128 ms.
- The Link LED:

  Its output is driven low whenever the PHY detects a valid link. The use of the 10Mbps or 100Mbps link test status is determined by the condition of the internally determined speed selection.
- The Speed LED:

  Its output is driven low when the operating speed is 100Mbit/s or during Auto-negotiation. This LED will go inactive when the operating speed is 10Mbit/s.
- The Full-Duplex LED

  Its output is driven low when the link is operating in Full-Duplex mode.

(8) Loopback Operation

The 10/100 digital has an independent loop-back mode: Internal loopback.

- Internal loopback

  The internal loopback mode is enabled by setting bit register 0 bit 14 to logic one. In this mode, the scrambled transmit data (output of the scrambler) is looped into the receive logic (input of the descrambler). The CO_COL signal will be inactive in this mode, unless collision test (bit 0.7) is active.

  In this mode, during transmission (CO_TX_EN is HIGH), nothing is transmitted to the line and the transmitters are powered down.

22.11 Internal I/O Signals

The I/O signals interface the logic of the core PHY to other modules on this LSI. The input signals can either be connected to other modules on the chip so that they can be connected to pins and driven externally, or they can be tied high or low inside the chip to set the behavior of the core PHY.

The following abbreviations are used:

- I: Input. Digital TTL levels.
- O: Output. Digital TTL levels.
- AI: Input. Analog levels.
- AO: Output. Analog levels.
- AI/O: Input or Output. Analog levels.

- MII signals

Signal Name	Type	Description
CO_MII_TXD0	I	Transmit Data 0: Bit 0 of the 4 data bits that are accepted by the PHY for transmission.
CO_MII_TXD1	I	Transmit Data 1: Bit 1 of the 4 data bits that are accepted by the PHY for transmission.
CO_MII_TXD2	I	Transmit Data 2: Bit 2 of the 4 data bits that are accepted by the PHY for transmission.
CO_MII_TXD3	I	Transmit Data 3: Bit 3 of the 4 data bits that are accepted by the PHY for transmission.
CO_TX_EN	I	Transmit Enable: Indicates that valid data is presented on the CO_MII_TXD[3:0] signals, for transmission.
CO_RX_ER (RXD4)	OO	Receive Error: Asserted to indicate that an error was detected somewhere in the frame presently being transferred from the PHY. In Symbol Interface (5B Decoding) mode, this signal is the MII Receive Data 4: the MSB of the received 5-bit symbol code-group.
CO_COL	O	MII Collision Detect: Asserted to indicate detection of collision condition.
CO_MII_RXD0	O	Receive Data 0: Bit 0 of the 4 data bits that are sent by the PHY in the receive path.
CO_MII_RXD1	O	Receive Data 1: Bit 1 of the 4 data bits that are sent by the PHY in the receive path.
CO_MII_RXD2	O	Receive Data 2: Bit 2 of the 4 data bits that sent by the PHY in the receive path.
CO_MII_RXD3	O	Receive Data 3: Bit 3 of the 4 data bits that sent by the PHY in the receive path.
CO_TX_ER (TXD4)	I	MII Transmit Error: When driven high, the 4B/5B encode process substitutes the Transmit Error code-group (/H/) for the encoded data word. This input is ignored in 10BaseT operation. In Symbol Interface (5B Decoding) mode, this signal becomes the MII Transmit Data 4: the MSB of the 5-bit symbol code-group.
CO_CRS	O	Carrier Sense: Indicate detection of carrier.
CO_RX_DV	O	Receive Data Valid: Indicates that recovered and decoded data nibbles are being presented on CO_MII_RXD[3:0].
CO_TX_CLK	O	Transmit Clock: 25MHz in 100Base-TX mode. 2.5MHz in 10Base-T mode.
CO_RX_CLK	O	Receive Clock: 25MHz in 100Base-TX mode. 2.5MHz in 10Base-T mode.

- Management signals

Signal Name	Type	Description
CO_MDI	I	Management Data Input: Serial management data input.
CO_MDO	O	Management Data Output: Serial management data output.
CO_MDC	I	Management Clock: Serial management clock.
CO_MDIO_DIR	O	Management Data Direction: May be used to control output enabled buffer for MDIO.

- General signals

Signal Name	Type	Description
CO_CLKIN	I	Clock Input - PHY clock. Can be 25MHz either from mck of CPG module or from CK_PHY pin.

22.12 Signals Relevant to PHY-IF

This PHY core has a part set up by the PHY-IF module.

(1) PHY address

The PHY address initialized by PHYIFADDR of PHY-IF, is same as the one that the ordinary external PHY LSI has. It gives each PHY a unique address. This address is latched into an internal register during Module reset and PHY power on reset. Originally, it enables a function to manage each PHY via the unique address in a multi-PHY application.

About this PHY module, you can not connect multiple PHYs to the MII interface within the LSI. But PHY address is also used to seed the scrambler, so that please accord the configuration of PHYIFADDRR and the PHY address on the management interface.

(2) Operation mode

The co_st_mode of the PHYIFCR of PHY-IF controls the configuration of 10/100 digital block.

co_st_mode[2:0] of PHYIFCR	Mode Definitions	Default Register Bit Values	
		Register 0 [13,12,10,8]	Register 4 [8,7,6,5]
000	10Base-T Half Duplex. Auto-negotiation disabled.	0000	N/A
001	10Base-T Full Duplex. Auto-negotiation disabled.	0001	N/A
010	100Base-TX Half Duplex. Auto-negotiation disabled.CRS is active during Transmit & Receive.	1000	N/A
011	100Base-TX Full Duplex. Auto-negotiation disabled.CRS is active during Receive.	1001	N/A
100	100Base-TX Half Duplex is advertised. Auto-negotiation enabled.CRS is active during Transmit & Receive.	1100	0100
101	Reserved.(Do not set this mode)	1100	0100
110	Power Down mode. In this mode the PHY wake-up in Power-Down mode.	N/A	N/A
111	All capable. Auto-negotiation enabled.	X10X	1111

22.13 Usage Notes

(1) Input clock to PHY module

The initial clock to PHY module is internal clock, mck (= ick/4), but it does work only when it is 25MHz, which is acceptable to PHY module.

It corresponds to power down mode. For example, even in the application which doesn't use the on-chip PHY module, you have to set up the clock to the on-chip PHY so that it could be low power consumption mode with power down mode.

(2) Treatment of Pins When PHY Power Supply is Not Used

Even when the on-chip PHY is not used, supply power to the analog power supply pins for the PHY (Vcc1A, Vcc2A, and Vcc3A) and connect the analog ground pins for the PHY (Vss1A and Vss2A) to the ground. Pull up the CK-PHY pin to VccQ through a resistor or pull down the CK-PHY pin to VssQ through a register. Connect pins TxP, TxM, RxP, and RxM to the PHY analog ground. Connect the EXERS1 pin to the PHY analog power supply without going through a resistor. Do not connect anything to the TSTBUSA pin.

(3) Software Reset of the PHY

The software reset of the on-chip PHY of this LSI has a defect in characteristics, which can prevent correct resetting of the PHY in some cases. Because of this, the PHY should be reset by a module reset, which is generated by setting the PHYIFCR register (in the PHY-IF module).

Note: 1. Software reset refers to the reset which is executed by bit 15 of register 0 (basic control) described in section 22.4.2, SMI Register Mapping.
2. Module reset refers to the reset which is executed by bit 14 of PHYIFCR (PHY-IF control register) described in section 23.2, Register Descriptions, in section 23, PHY Interface (PHY-IF).

(4) Waveform Adjustment

The Ethernet PHY module of this LSI has test registers for adjustment of differential output waveforms. Using these test registers in their initial values produces no problem, but their specifications are shown below to facilitate printed circuit board design by the customer.

(a) Adjustment of Tx100 Waveform Output

The on-chip PHY module of this LSI has the following adjustment registers as SIM registers, which allow waveform adjustment in the Tx100 operation. These registers have been designed so that they are not accidentally written. To change their values, follow the example procedure shown in "How to Use" that is described later.

- Register 20: Register for changing modes
- Register 23: Register for waveform adjustment
  (The register numbers are decimal)

- Meanings of the value written to register 23

Bit	Bit Name	Initial Value	R/W	Description
15	—	1	RO	Reserved The write value should always be 1.
14 to 9	—	0	RO	Reserved The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	D1CMP	1	R/W	These bits adjust the slope.
7	D0CMP	1	R/W	00: Three steps up
				01: Two steps up
				10: One step up
				11: Regular
6	D2A	1	R/W	These bits adjust the amplitude.
5	D1A	0	R/W	000: Amp 4 stp+
4	D0A	0	R/W	001: Amp 3 stp+
				010: Amp 2 stp+
				011: Amp 1 stp+
				100: Regular
				101: Amp 1 stp-
				110: Amp 2 stp-
				111: Amp 3 stp-
3	DASL	1	R/W	These bits adjust the transition time.
2	DBSL	0	R/W	00: One step up
				01: One step down
				10: Regular
				11: Two steps down
1, 0	—	0	RO	Reserved
				The write value should always be 0.

- How to Use (Example)

  Write to the SIM registers in the following sequence.

Step	Register to be Written	Value for Writing	Description
1	0	H'2100	Select Tx100 mode. (This operation can be omitted if Tx100 full-duplex or Tx100 half-duplex mode has been selected by auto-negotiation,)*
2	20	H'0000	Start register write mode setting.
3	20	H'0000	Register write mode setting (continued)
4	20	H'0400	Register write mode setting (continued)
5	20	H'0000	Register write mode setting (continued)
6	20	H'0400	Finish register write mode setting.
7	23	H'xxxx	Write the setting value. (The initial value of this register is H'81C8. Change the setting as necessary.)
8	20	H'4416	Validate the setting value (always write this value).
9	20	H'0000	Terminate the register write mode (return to normal mode).

Note: The setting of this register is initialized during the auto-negotiation process or when the PHY module is reset (including a system reset of the LSI). Accordingly, when waveform adjustment is to be performed by this register, the above steps must be carried out every time the register is initialized.

(b) Adjustment Register for Tx10 Waveform Output

Note: This register has little effect. The following descriptions of the register are only for your reference.

The adjustment register for Tx10 waveform output has DnTAMP (n = 1, 0) bits to adjust the amplitude of waveforms and DnTCMP (n = 1, 0) bits to adjust the slope of waveforms. The number of the register is 23 (decimal). When the value of writing of the step 8 in the adjustment register for Tx100 waveform output, described in (a), is H'4418 instead of H'4416, the value of writing of the step 7 is written as the setting value for the adjustment register for Tx10 waveform output. As described in the table below, the values written in bit15 and bit14 in the adjustment register for Tx10 waveform output are used as the setting values for DnTAMP (n = 1, 0) bits, while the values written in bit13 and bit12 are used as the setting values for DnTCMP (n = 1, 0) bits. However, based our testing, the adjustment of the amplitude by DnTAMP (n = 1, 0) bits effects only in several millivolts.

- Adjustment register for Tx10 waveform output

Bit	Bit Name	Initial Value	R/W	Description
15	D1TAMP	0	R/W	These bits adjust the amplitude.
14	D0TAMP	1	R/W	11: Amp 2 stp+ 10: Amp 1 stp+ 01: Regular 00: Amp 1 stp-
13	D1TCMP	0	R/W	These bits adjust the slope (transition time) (the more steps up, the gentler the slope is). 11: Three steps up 10: Two steps up 01: One step up 00: Regular
12	D0TCMP	0	R/W	
11 to 0	—	0	RO	Reserved The write value should always be 0.

- How to Use (Example)

  Write to the SIM registers in the following sequence.

Step	Register to be Written	Value for Writing	Description
1	0	H'2100	Select Tx100 mode (the setting by the reset of the PHY is also valid).*
2	20	H'0000	Start register write mode setting.
3	20	H'0000	Register write mode setting (continued)
4	20	H'0400	Register write mode setting (continued)
5	20	H'0000	Register write mode setting (continued)
6	20	H'0400	Finish register write mode setting.
7	23	H'xxxx	Write the setting value (in the “Regular” case, the initial value of this register is H'4000).
8	20	H'4418	Validate the setting value (in Tx10 case).
9	20	H'0000	Terminate the register write mode (return to normal mode).

Note: * To make the LSI enter the mode for setting the waveform adjustment, the Tx100 mode must be selected, instead of the Tx10 mode.

The setting of the waveform adjustment is initialized during the auto-negotiation process or when the PHY module is reset (including a system reset of the LSI).

(c) Detailed Descriptions

The detailed descriptions of the functions of the adjustment registers for Tx100 waveform output are given below.

1. External Specification for Waveform Generation

   Compliance tests include the items of the Rise Time (+/-ve) and Fall Time (+/-ve) in the "Tx100". The specified values are from 3 ns to 5 ns, respectively.

   Therefore, the on-chip PHY module of this LSI is designed to transfer from 0 V to 1 V in 4 ns.

2. Mechanism of Waveform Generation

Waveforms are generated in two ways; divided in time and voltage.

For an example of divisions in time and voltage, the case of a transfer from 0 V to 1 V in 4 ns is shown below.

time	Output voltage level
0 ～	250 [mv]
1 [ns] ～	500 [mv]
2 [ns] ～	750 [mv]
3 [ns] ～	1 [v]

- Time ranges

  In this case, four-divided time ranges are generated on internal clocks, at first. Rise times are controlled as the divided numbers are controlled.

  Total transition time is controlled as each timing in each time range is shifted on the DnSL bits in the adjustment registers.

  Each slope in each time range is set on the DnCMP bits.

- Voltage levels

  The voltage levels are also divided in four. The levels are modified at once as the maximum amplitude, the standard, is controlled on the DnA bits.

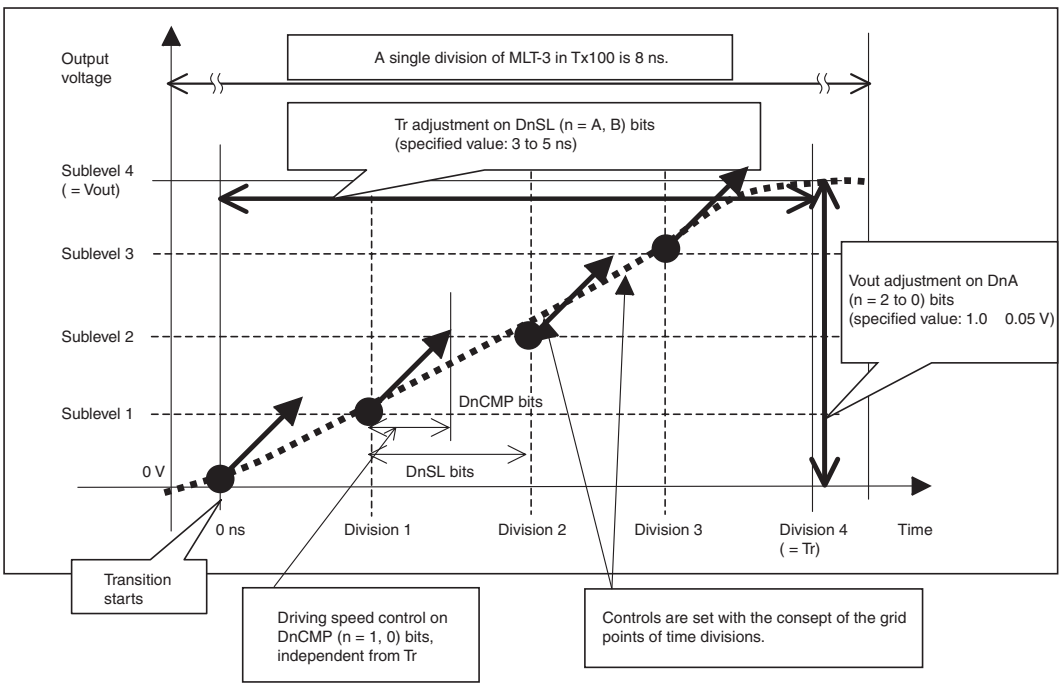
3. Control Method with Adjustment Registers for Tx100 Waveform Output

- Description of the bits of the adjustment registers for Tx100 waveform output

  The figure below shows the adjustment of the rising waveform from 0 V to 1 V, with the symbols;

  Vout: Maximum amplitude (target in manufacturing is 1 V)

  Tr: Transition time (target in manufacturing is 4 ns)



- Targets in manufacturing

  The table below shows the targets in manufacturing. Each target can be adjusted with the adjustment registers.

Time division		Voltage level corresponding to time division	
Division #	Target in manufacturing	Voltage level #	Target in manufacturing
Division 1	1 ns	Sublevel 1	250 mV
Division 2	2 ns	Sublevel 2	500 mV
Division 3	3 ns	Sublevel 3	750 mV
Division 4	4 ns	Sublevel 4	1 V

Time ranges evenly divided (Tr/4) are adjusted on DnSL bits, and consequently the total Tr value is also adjusted.
→"Transition time is controlled on DnSL"

Each sublevel is adjusted on the Vout value, controlled on DnA bits (evenly divided by Vout/4).
→"Amplitude is controlled on DnA"

- Adjustment effects

  The amplitude and the transition time (the slope) are controlled independently, as shown above.

  The slope is controlled on the DnSL bits and DnCMP bits together. However, since it is difficult to express the generated analog waveforms quantitatively, the waveforms must be ensured on the actual boards.

(d) Other Control Methods

The methods, shown below for your reference, may have some bad effects or disadvantages. Therefore, if the methods will be used, it is necessary to confirm the advantage and disadvantage sufficiently.

1. Common Amplitude Adjustment Method in Tx10 and Tx100
   - Advantage:

     As a common amplitude adjustment method in Tx10 and Tx100, there is a method of modifying the resistances of the resistors R1 and R2 in figure 22.10. The amplitudes in Tx10 and Tx100 have correlations with the resistances of both the resistors R1 and R2. Increasing the resistances increases the amplitudes in both Tx10 and Tx100, while decreasing the resistances decreases the amplitudes.
   - Disadvantage:

     Unfavorable outcomes may often appear in harmonic content testings.

2. Amplitude Adjustment Method in Tx10
   - Advantage:

     The amplitudes in Tx10 depend on VccnA (meaning PVCC in the example of connection above; n = 1 to 3). Increasing VccnA increases the amplitudes, while decreasing VccnA decreases the amplitudes.

     The amplitudes in Tx100 also depend on VccnA, though, less than in Tx100. Therefore, the amplitudes in Tx10 can be adjusted with modifying VccnA, with no influence on the results in Tx100.
   - Disadvantage:

     However, since VccQ and VccnA are connected with diode inside this LSI, the permanent potential difference in them may damage the LSI's reliability. Therefore, the method has the disadvantage that VccQ must be adjusted simultaneously.

22.14 Guidelines for Layout

22.14.1 General Guidelines

The guidelines for four-layer boards are shown below.

(1) Configuration of Board Layers

- Layer 1: Top layer (component side), which is a signal layer
- Layer 2: Ground layer
- Layer 3: Power layer
- Layer 4: Bottom layer (solder side), which is a signal layer

(2) Impedance Control

Ideally, impedance control should satisfy the following.

- Single ended traces: 51 ohm ±10%
- Differential pairs: 99 ohm ±10%
- No restrictions on the impedance of short power/grand traces

(3) Vias

Vias are a source of impedance mismatches and distorted waveforms on transmission lines, which can cause problems of signal integrity (noise) and EMI issues. For differential signals and fast signal traces, avoid using vias on the signal lines whenever possible. If vias are used on such signal traces, ensure that they do not create problems by simulation or other means.

(4) Notes on Routing

Stubs (branching) cause signal reflections, so they should be 12.7 mm (0.5 inch) or shorter for critical nets.

Stagger is a bad source of crosstalk, so all the signal traces around the PHY should be 25.4 mm (1 inch) or shorter.

(5) Terminations

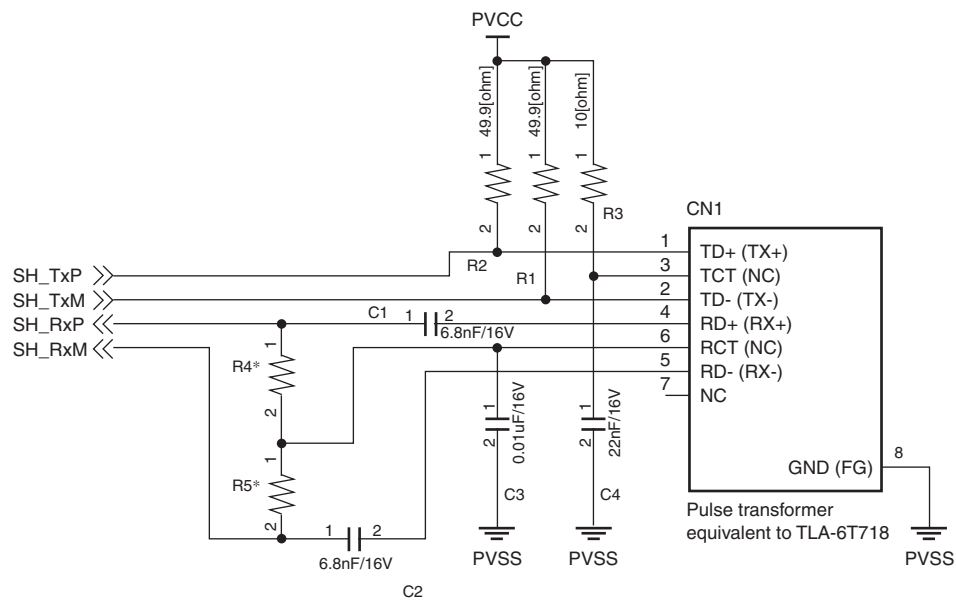
To reduce signal reflections caused by impedance mismatches, provide damper or terminating resistance at the end-points of signal nets. Damper resistance should be placed close to the signal source, while terminating resistance should be placed at the farthest end-points of their nets. The distance from the signal source or farthest end-point must be shorter than 12.7 mm (0.5 inch).

22.14.2 Guidelines for Layout

Since the signals of the PHY are analog signals with high frequencies and small amplitudes, they are susceptible to digital noise. So, routing and placement must be done with extra care.

An example of connection with a pulse transformer (RJ45) is shown in figure 22.10. The codes such as C1 and R2 in the following explanation are the part numbers indicated in figure 22.10.

(1) Example of Connection with a Pulse Transformer (RJ45)



Note: PVCC: Analog power supply
PVSS: Analog ground
* R4, R5: Please set the terminating resistance as 50 ohm, summing up with both side of the magnetic (even if the magnetic is in the RJ45 connector).

Figure 22.10 Example of Connection with a Pulse Transformer (RJ45)

(2) Sample Placement

1. The pulse transformer should be placed close to the PHY-related pins of this LSI.
2. Components should be placed so that the signal traces of differential pairs, TxP/TxM and RxP/RxM, do not cross each other.
3. R4 and R5, which are terminating resistors, should be placed close to this LSI.
4. R1 and R2 should be placed close to the pulse transformer (RJ45).
5. R3 and C4, which form a filter, should be placed close to the pulse transformer (RJ45).
6. C4 of the center tap should be placed close to the pulse transformer (RJ45).
7. Do not place any components on the bottom side.

(3) Ground Planes

Layer 2 is divided into logic ground plane and frame ground plane.

The logic ground is the combination of digital ground and analog ground. The frame ground is connected to the system ground and the shielding of the RJ45 socket so that it is grounded. Beware that this ground plane cuts impact the routing on adjacent signal layers.

Signal traces of L1 and L4 should not run across the cuts in the ground plane to avoid impedance mismatches and EMI problems. Minimize the frame ground area so as to make the logic ground as large and solid as possible. Connect the logic ground and frame ground by a ferrite bead or thick signal trace to provide a DC path. For safety, exclude the area near the leads of the RJ45 from the ground area.

(4) Common Power Plane

Layer 3 consists of multiple power planes of Vcc and Vcc for PLL1 and PLL2, which supply 1.8 V, and VccQ and VccnA (n = 1 to 3), which supply 3.3 V. VccnA is made up of an area of analog power for the RJ45 (connector-type pulse transformer) and an area of analog power for this LSI.

(5) Sample Routing

In the above example, the ground layer is simply divided into two planes while the power layer is divided into more planes. Therefore, the top layer (component side) is superior to the bottom layer (solder side) in terms of signal integrity. If possible, all the critical signals of the PHY, differential signal pairs for example, should be wired in the top layer without any vias.

Another important thing to be noted about differential signal pairs is that the pair of traces of a pair must be strictly equal in length to minimize duty cycle distortion and common mode radiation.

(6) Clock Layout

In addition to the clock input for the CPU, an external clock for the PHY (CKPHY) can also be input to this LSI.

Analog power supply and analog signals should be placed far away from the oscillator, resonator, and digital devices that produce much noise. Clock signal lines should be wired in a layer higher than the ground layer (the top layer (component side) in this example). In addition, clock traces should be kept as far away from other traces as possible. The minimum spacing is three times of the trace width.

Section 23 PHY Interface (PHY-IF)

This is an interface for operation of the on-chip PHY on this LSI.

23.1 Features

- Selectable operation to enable the on-chip PHY or to disable (= utilizing an external PHY LSI) by pin function controller of ports.
- Below settings for the on-chip PHY are available.

  The module reset

  Selectable operation clock of the PHY module, the internal clock or the exclusive external clock for PHY.

  But the clock of the on-chip PHY module has 25 MHz, fixed frequency.

Figure 23.1 shows the block diagram of PHY-IF.

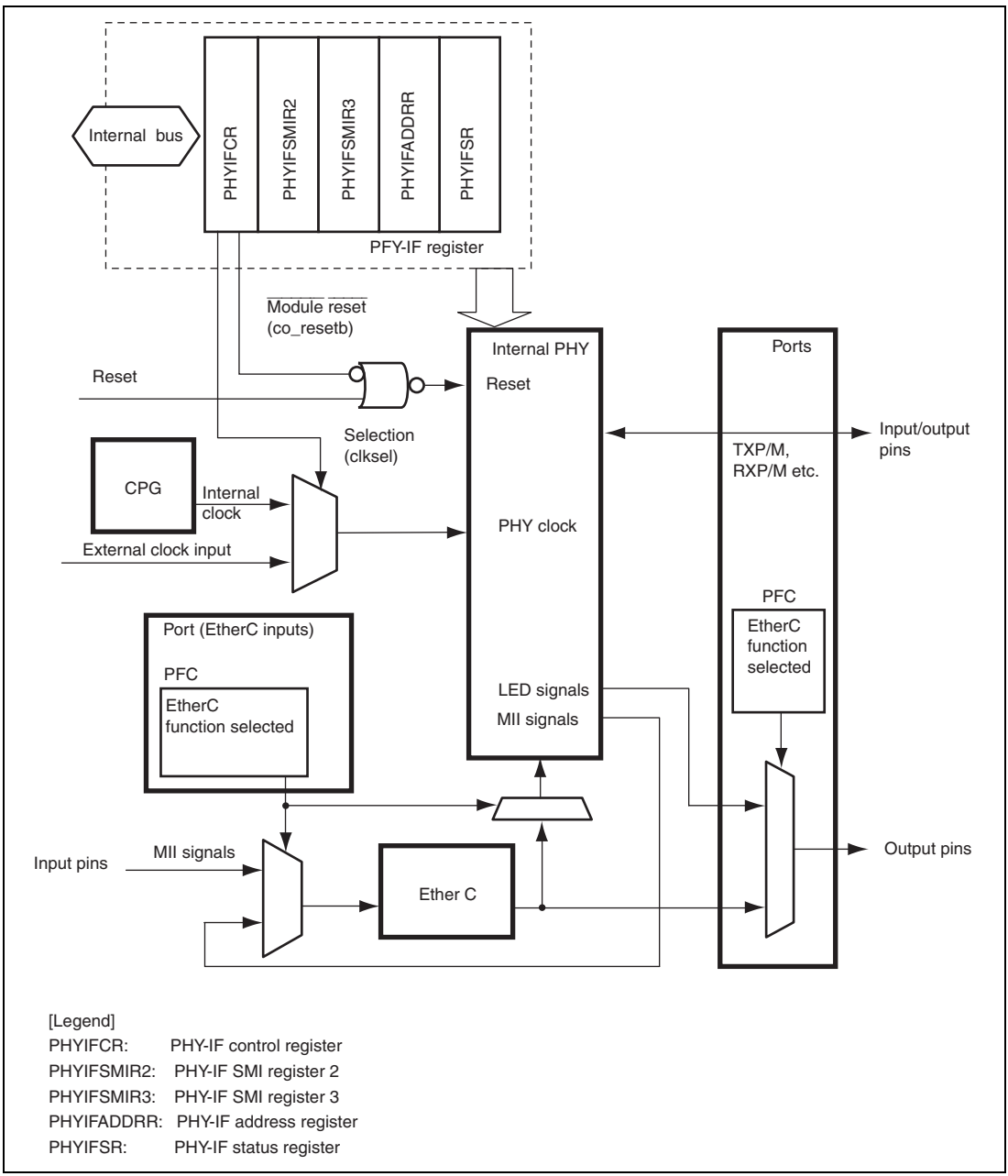


Figure 23.1 Block Diagram of PHY-IF

23.2 Register Descriptions

PHY-IF has below registers. Refer to section 24, List of Registers, about the addresses and the status under each operating condition.

- PHY-IF control register (PHYIFCR)
- PHY-IF SMI register 2 (PHYIFSMIR2)
- PHY-IF SMI register 3 (PHYIFSMIR3)
- PHY-IF address register (PHYIFADDRR)
- PHY-IF status register (PHYIFSR)

23.2.1 PHY-IF Control Register (PHYIFCR)

PHYIFCR is a 16-bit readable/writeable register, which sets the operation mode of the on-chip PHY module. The changed bit values except co_resetb are taken by the module reset of the on-chip PHY with co_resetb.

PHYIFCR is initialized by power-on-reset. It is also initialized as H'C000 in the standby mode.

Bit	Bit name	Initial value	R/W	Description
15	—	1	R	Reserved. This bit is always read as 1. The write value should always be 1.
14	co_resetb	1	R/W	Module reset Resets the on-chip PHY with software. 0: reset state 1: reset state is released (an initial value)
13	clksel	0	R/W	Clock selection Selects which to provide to on-chip PHY, the internal clock or the external clock. 0: Uses the internal clock(mck) (an initial value) 1: Uses the external clock (CK_PHY)
12 to 3	—	0	R/W	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit name	Initial value	R/W	Description
2	co_st_mode[2]	1	R/W	PHY mode
1	co_st_mode[1]	1		Decides the initial value of the PHY mode.
0	co_st_mode[0]	0		000: 10Base-T Half Duplex. Auto-negotiation disabled.
				001: 10Base-T Full Duplex. Auto-negotiation disabled.
				010: 100Base-TX Half Duplex. Auto-negotiation disabled.
				CRS is active during Transmit & Receive.
				011: 100Base-TX Full Duplex. Auto-negotiation disabled.
				CRS is active during Receive.
				100: 100Base-TX Half Duplex is advertised. Auto-negotiation enabled.
				CRS is active during Transmit & Receive.
				101: Reserved. (Do not set this mode.)
				110: Power Down mode. In this mode the PHY wake-up in Power-Down mode (an initial value)
				111: All capable. Auto-negotiation enabled.

23.2.2 PHY-IF SMI Register 2 (PHYIFSMIR2)

PHYIFSMIR2 is a 16-bit readable/writeable register, which sets the initial value of SMI register 2 in the case of the module reset the on-chip PHY module.

The changes of this register are taken by the on-chip PHY module reset with co_resetb.

PHYIFSMIR2 is initialized by power-on-reset. It is also initialized as H'0000 in the standby mode.

Bit	Bit name	Initial value	R/W	Description
15 to 0	co_reg2_oui_in[15-0]	All 0	R/W	The initial value of SMI register 2 (= PHY identifier 1)[15-0]

23.2.3 PHY-IF SMI Register 3 (PHYIFSMIR3)

PHYIFSMIR3 is a 16-bit readable/writeable register, which sets the initial value of SMI register 3 in the case of the module reset the on-chip PHY module.

The changes of this register are taken by the on-chip PHY module reset with co_resetb.

PHYIFSMIR2 is initialized by power-on-reset. It is also initialized as H'0000 in the standby mode.

Bit	Bit name	Initial value	R/W	Description
15 to 0	co_reg3_oui_in[15-0]	All 0	R/W	The initial value of SMI register 3 (= PHY identifier 2)[15-0]

23.2.4 PHY-IF Address Register (PHYIFADDRR)

PHYIFADDRR is a 16-bit readable/writeable register, which sets the PHY address of the on-chip PHY module.

The changes of this register are taken by the on-chip PHY module reset with co_resetb.

PHYIFADDRR is initialized by power-on-reset. It is also initialized as H'0000 in the standby mode.

Bit	Bit name	Initial value	R/W	Description
15 to 5	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
4 to 0	co_st_phyadd[4-0]	All 0	R/W	The initial value of PHY address

23.2.5 PHY-IF status Register (PHYIFSR)

PHYIFSR is a 16-bit read-only register that shows the status of the on-chip PHY module.

PHYIFSR is initialized by power-on-reset.

Bit	Bit name	Initial value	R/W	Description
15	co_pwruprst	1 (Ref. Description)	R	Power Up Reset This bit goes to "1" only on detection of Power up of the on-chip PHY power (Vcc1A to Vcc3A) and stays at "1" for 21 ms, automatically.
14 to 0	—	0	R	Reserved. These bits are always read as 0. The write value should always be 0.

23.3 PHY-IF Operation

PHY-IF is basically for initializing the on-chop PHY module.

Following the procedures described in the following sections, please set up the on-chip PHY module with the MII interface like as for the external PHY LSI. The PHY module itself goes to power down mode with the initial values of co_st_mode of PHYIFCR after power-on-reset of the whole LSI at power-up.

23.3.1 The Procedures of Setting Up the On-Chip PHY

Please set up with below procedures.

1. Release of module stop

   First of all, release the module stop (MSTP20 of STBCR4), if PHY-IF is in module stop mode.
2. Power Up Reset

   Check the release of power up reset mode, shown in the co_pwruprst-bit of PHYIFSR with value "0".
3. Activation of the on-chip PHY module

   To activate the on-chip PHY module, set the pin function registers of Port C as something but EtherC function, that is, I/O ports and LED outputs of the on-chip PHY.
   - PCCRH2 = H'0000
   - PCCRL1 = H'0000
   - PCCRL2 = H'FF00

   In this case, the LNKSTA input pin of the EtherC is deselected. As the link output of the on-chip PHY and link input of the EtherC are connected in this LSI, the link signal change interrupt can be generated in the same way as the external PHY LSI is used.

4. Set up of the clock

   In the case of utilizing the internal clock from CPG, you have to set up the MCLKCR during the reset period of the on-chip PHY. Set the input clock of the PHY module as 25 MHz by adjusting the FRQCR and MCLKCR.

   Do this set up before module reset of the on-chip PHY.

5. The reset of the on-chip PHY

   Before you reset the on-chip PHY module, please set the register sets of PHY-IF parts as you need, except PHYIFCR. After that, set the co_resetb of PHYIFCR as zero, to make the on-chip PHY reset state.

   At this moment, you should set the other bits of PHYIFCR, which corresponds to the operating mode of the on-chip PHY. Please adjust the waiting time with software-loop, etc., so that you can keep the reset period is over 100 μs.
6. Release of the reset of the on-chip PHY.

   Set only the co_resetb of PHYIFCR as "1", for releasing the reset state of the on-chip PHY.

   After releasing the reset, adjust the waiting time with software loops, etc. as over 20 ms for propagation of reset signal within the PHY.
7. Set up the on-chip PHY module with the MII management frame.

   The procedures after this step are set up by the MII management frame like an external PHY LSI on the market.

   Please refer the section of PHY module about the each settings of it.

23.3.2 The Procedures of Set Up the External PHY LSI

In the case of utilizing the external PHY LSI, select the EtherC function of the pin function controllers and then set up the internal registers of the PHY LSI with the MII management frame.

1. Activation of the external PHY LSI.

   Select the EtherC functions with pin function controller.
   - PCCRH2 = H'0155
   - PCCRL1 = H'5555
   - PCCRL2 = H'5555
2. Set up the external PHY LSI with the MII management frame.

   Following procedures are set up by the MII management frame.

   About the each settings of the PHY LSI that you utilize, please refer the documents of it.

Section 24 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- Reserved addresses are indicated by — in the register name column.
  Do not access the reserved addresses.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given.
- Registers are classified according to functional modules.
- The numbers of Access Cycles are given.

2. Register bits
- Bit configurations of the registers are listed in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- Space in the bit name field indicates that the entire register is allocated to either the counter or data.
- For the registers of 16 or 32 bits, the MSB is listed first.

3. Register states in each operating mode
- Register states are listed in the same order as the register addresses.
- The register states shown here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

24.1 Register Addresses (Address Order)

Entries under Access size indicates numbers of bits.

The number of access cycles indicate the number of cycles of the given reference clock. B, W, and L indicate values for 8-, 16-, and 32-bit accesses, respectively.

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.

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
DMA source address register_0	SAR_0	32	H'F8010020	DMAC	16/32
DMA destination address register_0	DAR_0	32	H'F8010024	DMAC	16/32
DMA transfer count register_0	DMATCR_0	32	H'F8010028	DMAC	16/32
DMA channel control register_0	CHCR_0	32	H'F801002C	DMAC	8/16/32
DMA source address register_1	SAR_1	32	H'F8010030	DMAC	16/32
DMA destination address register_1	DAR_1	32	H'F8010034	DMAC	16/32
DMA transfer count register_1	DMATCR_1	32	H'F8010038	DMAC	16/32
DMA channel control register_1	CHCR_1	32	H'F801003C	DMAC	8/16/32
DMA source address register_2	SAR_2	32	H'F8010040	DMAC	16/32
DMA destination address register_2	DAR_2	32	H'F8010044	DMAC	16/32
DMA transfer count register_2	DMATCR_2	32	H'F8010048	DMAC	16/32
DMA channel control register_2	CHCR_2	32	H'F801004C	DMAC	8/16/32
DMA source address register_3	SAR_3	32	H'F8010050	DMAC	16/32
DMA destination address register_3	DAR_3	32	H'F8010054	DMAC	16/32
DMA transfer count register_3	DMATCR_3	32	H'F8010058	DMAC	16/32
DMA channel control register_3	CHCR_3	32	H'F801005C	DMAC	8/16/32
DMA operation register	DMAOR	16	H'F8010060	DMAC	16
Port A data register H	PADRH	16	H'F8050000	I/O	8/16
Port A IO register H	PAIORH	16	H'F8050004	I/O	8/16
Port A control register H1	PACRH1	16	H'F8050008	I/O	8/16
Port A control register H2	PACRH2	16	H'F805000A	I/O	8/16
Port B data register L	PBDRL	16	H'F8050012	I/O	8/16
Port B IO register L	PBIORL	16	H'F8050016	I/O	8/16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Port B control register L1	PBCRL1	16	H'F805001C	I/O	8/16
Port B control register L2	PBCRL2	16	H'F805001E	I/O	8/16
Port C data register H	PCDRH	16	H'F8050020	I/O	8/16
Port C data register L	PCDRL	16	H'F8050022	I/O	8/16
Port C IO register H	PCIORH	16	H'F8050024	I/O	8/16
Port C IO register L	PCIORL	16	H'F8050026	I/O	8/16
Port C control register H2	PCCRH2	16	H'F805002A	I/O	8/16
Port C control register L1	PCCRL1	16	H'F805002C	I/O	8/16
Port C control register L2	PCCRL2	16	H'F805002E	I/O	8/16
Port D data register L	PDDRL	16	H'F8050032	I/O	8/16
Port D IO register L	PDIORL	16	H'F8050036	I/O	8/16
Port D control register L2	PDCRL2	16	H'F805003E	I/O	8/16
Port E data register H	PEDRH	16	H'F8050040	I/O	8/16
Port E data register L	PEDRL	16	H'F8050042	I/O	8/16
Port E IO register H	PEIORH	16	H'F8050044	I/O	8/16
Port E IO register L	PEIORL	16	H'F8050046	I/O	8/16
Port E control register H1	PECRH1	16	H'F8050048	I/O	8/16
Port E control register H2	PECRH2	16	H'F805004A	I/O	8/16
Port E control register L1	PECRL1	16	H'F805004C	I/O	8/16
Port E control register L2	PECRL2	16	H'F805004E	I/O	8/16
Interrupt priority register C	IPRC	16	H'F8080000	INTC	16
Interrupt priority register D	IPRD	16	H'F8080002	INTC	16
Interrupt priority register E	IPRE	16	H'F8080004	INTC	16
Interrupt priority register F	IPRF	16	H'F8080006	INTC	16
Interrupt priority register G	IPRG	16	H'F8080008	INTC	16
DMA extended resource selector 0	DMARS0	16	H'F8090000	DMAC	16
DMA extended resource selector 1	DMARS1	16	H'F8090004	DMAC	16
Standby control register 3	STBCR3	8	H'F80A0000	Power-down mode	8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Standby control register 4	STBCR4	8	H'F80A0004	Power-down mode	8
PHY-LSI clock frequency control register	MCLKCR	8	H'F80A000C	CPG	8/16*
Instruction register	SDIR	16	H'F8100200	H-UDI	16
ID register	SDID	32	H'F8100214	H-UDI	32/16
Interrupt control register 0	ICR0	16	H'F8140000	INTC	8/16
IRQ control register	IRQCR	16	H'F8140002	INTC	8/16
IRQ status register	IRQSR	16	H'F8140004	INTC	8/16
Interrupt priority register A	IPRA	16	H'F8140006	INTC	8/16
Interrupt priority register B	IPRB	16	H'F8140008	INTC	8/16
Frequency control register	FRQCR	16	H'F815FF80	CPG	16
Standby control register	STBCR	8	H'F815FF82	Power-down mode	8
Watch dog timer counter	WTCNT	8	H'F815FF84	WDT	8/16*
Watch dog timer control/status register	WTCSR	8	H'F815FF86	WDT	8/16*
Standby control register 2	STBCR2	8	H'F815FF88	Power-down mode	8
Serial mode register_0	SCSMR_0	16	H'F8400000	SCIF_0	16
Bit rate register_0	SCBRR_0	8	H'F8400004	SCIF_0	8
Serial control register_0	SCSCR_0	16	H'F8400008	SCIF_0	16
Transmit FIFO data register_0	SCFTDR_0	8	H'F840000C	SCIF_0	8
Serial status register_0	SCFSR_0	16	H'F8400010	SCIF_0	16
Receive FIFO data register_0	SCFRDR_0	8	H'F8400014	SCIF_0	8
FIFO control register_0	SCFCR_0	16	H'F8400018	SCIF_0	16
FIFO data count register_0	SCFDR_0	16	H'F840001C	SCIF_0	16
Serial port register_0	SCSPTR_0	16	H'F8400020	SCIF_0	16
Line status register_0	SCLSR_0	16	H'F8400024	SCIF_0	16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Serial mode register_1	SCSMR_1	16	H'F8410000	SCIF_1	16
Bit rate register_1	SCBRR_1	8	H'F8410004	SCIF_1	8
Serial control register_1	SCSCR_1	16	H'F8410008	SCIF_1	16
Transmit FIFO data register_1	SCFTDR_1	8	H'F841000C	SCIF_1	8
Serial status register_1	SCFSR_1	16	H'F8410010	SCIF_1	16
Receive FIFO data register_1	SCFRDR_1	8	H'F8410014	SCIF_1	8
FIFO control register_1	SCFCR_1	16	H'F8410018	SCIF_1	16
FIFO data count register_1	SCFDR_1	16	H'F841001C	SCIF_1	16
Serial Port register_1	SCSPTR_1	16	H'F8410020	SCIF_1	16
Line status register_1	SCLSR_1	16	H'F8410024	SCIF_1	16
Serial mode register_2	SCSMR_2	16	H'F8420000	SCIF_2	16
Bit rate register_2	SCBRR_2	8	H'F8420004	SCIF_2	8
Serial control register_2	SCSCR_2	16	H'F8420008	SCIF_2	16
Transmit FIFO data register_2	SCFTDR_2	8	H'F842000C	SCIF_2	8
Serial status register_2	SCFSR_2	16	H'F8420010	SCIF_2	16
Receive FIFO data register_2	SCFRDR_2	8	H'F8420014	SCIF_2	8
FIFO control register_2	SCFCR_2	16	H'F8420018	SCIF_2	16
FIFO data count register_2	SCFDR_2	16	H'F842001C	SCIF_2	16
Serial port register_2	SCSPTR_2	16	H'F8420020	SCIF_2	16
Line status register_2	SCLSR_2	16	H'F8420024	SCIF_2	16
Mode register	SIMDR	16	H'F8480000	SIOF	16
Clock select register	SISCR	16	H'F8480002	SIOF	16
Transmit data assign register	SITDAR	16	H'F8480004	SIOF	16
Receive data assign register	SIRDAR	16	H'F8480006	SIOF	16
Control data assign register	SICDAR	16	H'F8480008	SIOF	16
Control register	SICTR	16	H'F848000C	SIOF	16
FIFO control register	SIFCTR	16	H'F8480010	SIOF	16
Status register	SISTR	16	H'F8480014	SIOF	16
Interrupt enable register	SIIER	16	H'F8480016	SIOF	16
Transmit data register	SITDR	32	H'F8480020	SIOF	32
Receive data register	SIRDR	32	H'F8480024	SIOF	32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Transmit control data register	SITCR	32	H'F8480028	SIOF	32
Receive control data register	SIRCR	32	H'F848002C	SIOF	32
SPI control register	SPICR	16	H'F8480030	SIOF	16
PHY-IF control register	PHYIFCR	16	H'F8490000	PHY-IF	8/16
PHY-IF SMI register 2	PHYIFSMIR2	16	H'F8490004	PHY-IF	8/16
PHY-IF SMI register 3	PHYIFSMIR3	16	H'F8490008	PHY-IF	8/16
PHY-IF address register	PHYIFADDRR	16	H'F849000C	PHY-IF	8/16
PHY-IF status register	PHYIFSR	16	H'F8490010	PHY-IF	8/16
Compare match timer start register	CMSTR	16	H'F84A0070	CMT	8/16
Compare match timer control/status register_0	CMCSR_0	16	H'F84A0072	CMT	8/16
Compare match counter_0	CMCNT_0	16	H'F84A0074	CMT	8/16
Compare match timer constant register_0	CMCOR_0	16	H'F84A0076	CMT	8/16
Compare match timer control/status register_1	CMCSR_1	16	H'F84A0078	CMT	8/16
Compare match counter_1	CMCNT_1	16	H'F84A007A	CMT	8/16
Compare match timer constant register_1	CMCOR_1	16	H'F84A007C	CMT	8/16
HIF index register	HIFIDX	32	H'F84D0000	HIF	32
HIF general status register	HIFGSR	32	H'F84D0004	HIF	32
HIF status/control register	HIFSCR	32	H'F84D0008	HIF	32
HIF memory control register	HIFMCR	32	H'F84D000C	HIF	32
HIF internal Interrupt control register	HIFIICR	32	H'F84D0010	HIF	32
HIF external Interrupt control register	HIFEICR	32	H'F84D0014	HIF	32
HIF address register	HIFADR	32	H'F84D0018	HIF	32
HIF data register	HIFDATA	32	H'F84D001C	HIF	32
HIFDREQ trigger register	HIFDTR	32	H'F84D0020	HIF	32
HIF bank Interrupt control register	HIFBICR	32	H'F84D0024	HIF	32
HIF boot control register	HIFBCR	32	H'F84D0040	HIF	32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Common control register	CMNCR	32	H'F8FD0000	BSC	32
Bus control register for area 0	CS0BCR	32	H'F8FD0004	BSC	32
Bus control register for area 3	CS3BCR	32	H'F8FD000C	BSC	32
Bus control register for area 4	CS4BCR	32	H'F8FD0010	BSC	32
Bus control register for area 5B	CS5BBCR	32	H'F8FD0018	BSC	32
Bus control register for area 6B	CS6BBCR	32	H'F8FD0020	BSC	32
Wait control register for area 0	CS0WCR	32	H'F8FD0024	BSC	32
Wait control register for area 3	CS3WCR	32	H'F8FD002C	BSC	32
Wait control register for area 4	CS4WCR	32	H'F8FD0030	BSC	32
Wait control register for area 5B	CS5BWCR	32	H'F8FD0038	BSC	32
Wait control register for area 6B	CS6BWCR	32	H'F8FD0040	BSC	32
SDRAM control register	SDCR	32	H'F8FD0044	BSC	32
Refresh timer control/status register	RTCSR	32	H'F8FD0048	BSC	32
Refresh timer counter	RTCNT	32	H'F8FD004C	BSC	32
Refresh time constant register	RTCOR	32	H'F8FD0050	BSC	32
E-DMAC mode register	EDMR	32	H'FB000000	E-DMAC	32
E-DMAC transmit request register	EDTRR	32	H'FB000004	E-DMAC	32
E-DMAC receive request register	EDRRR	32	H'FB000008	E-DMAC	32
Transmit descriptor list start address register	TDLAR	32	H'FB00000C	E-DMAC	32
Receive descriptor list start address register	RDLAR	32	H'FB000010	E-DMAC	32
EtherC/E-DMAC status register	EESR	32	H'FB000014	E-DMAC	32
EtherC/E-DMAC status interrupt permission register	EESIPR	32	H'FB000018	E-DMAC	32
Transmit/receive status copy enable register	TRSCER	32	H'FB00001C	E-DMAC	32
Receive missed-frame counter register	RMFCR	32	H'FB000020	E-DMAC	32
Transmit FIFO threshold register	TFTR	32	H'FB000024	E-DMAC	32
FIFO depth register	FDR	32	H'FB000028	E-DMAC	32
Receiving method control register	RMCR	32	H'FB00002C	E-DMAC	32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
E-DMAC operation control register	EDOCR	32	H'FB000030	E-DMAC	32
Flow control FIFO threshold register	FCFTR	32	H'FB000034	E-DMAC	32
Transmit Interrupt setting register	TRIMD	32	H'FB00003C	E-DMAC	32
Receive buffer write address register	RBWAR	32	H'FB000040	E-DMAC	32
Receive descriptor fetch address register	RDFAR	32	H'FB000044	E-DMAC	32
Transmit buffer read address register	TBRAR	32	H'FB00004C	E-DMAC	32
Transmit descriptor fetch address register	TDFAR	32	H'FB000050	E-DMAC	32
EtherC mode register	ECMR	32	H'FB000160	EtherC	32
EtherC status register	ECSR	32	H'FB000164	EtherC	32
EtherC interrupt permission register	ECSIPR	32	H'FB000168	EtherC	32
PHY interface register	PIR	32	H'FB00016C	EtherC	32
MAC address high register	MAHR	32	H'FB000170	EtherC	32
MAC address low register	MALR	32	H'FB000174	EtherC	32
Receive frame length register	RFLR	32	H'FB000178	EtherC	32
PHY status register	PSR	32	H'FB00017C	EtherC	32
Transmit retry over counter register	TROCR	32	H'FB000180	EtherC	32
Delayed collision detect counter register	CDCR	32	H'FB000184	EtherC	32
Lost carrier counter register	LCCR	32	H'FB000188	EtherC	32
Carrier not detect counter register	CNDCR	32	H'FB00018C	EtherC	32
CRC error frame receive counter register	CEFCR	32	H'FB000194	EtherC	32
Frame receive error counter register	FRECR	32	H'FB000198	EtherC	32
Too-short frame receive counter register	TSFRCR	32	H'FB00019C	EtherC	32
Too-long frame receive counter register	TLFRCR	32	H'FB0001A0	EtherC	32
Residual-bit frame counter register	RFCR	32	H'FB0001A4	EtherC	32
Multicast address frame receive counter register	MAFCR	32	H'FB0001A8	EtherC	32
IPG setting register	IPGR	32	H'FB0001B4	EtherC	32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Automatic PAUSE frame set register	APR	32	H'FB0001B8	EtherC	32
Manual PAUSE frame set register	MPR	32	H'FB0001BC	EtherC	32
Automatic PAUSE frame retransfer count set register	TPAUSER	32	H'FB0001C4	EtherC	32
Break data register B	BDRB	32	H'FFFFFF90	UBC	32
Break data mask register B	BDMRB	32	H'FFFFFF94	UBC	32
Break control register	BRCR	32	H'FFFFFF98	UBC	32
Execution times break register	BETR	16	H'FFFFFF9C	UBC	16
Break address register B	BARB	32	H'FFFFFFA0	UBC	32
Break address mask register B	BAMRB	32	H'FFFFFFA4	UBC	32
Break bus cycle register B	BBRB	16	H'FFFFFFA8	UBC	16
Branch source register	BRSR	32	H'FFFFFFAC	UBC	32
Break address register A	BARA	32	H'FFFFFFB0	UBC	32
Break address mask register A	BAMRA	32	H'FFFFFFB4	UBC	32
Break bus cycle register A	BBRA	16	H'FFFFFFB8	UBC	16
Branch destination register	BRDR	32	H'FFFFFFBC	UBC	32
Cache control register 1	CCR1	32	H'FFFFFFEC	Cache	32

Note: * The numbers of access cycles are eight bits when reading and 16 bits when writing.

24.2 Register Bits

Register addresses and 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
SAR_0									DMAC
DAR_0									
DMATCR_0									
CHCR_0	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_1									
DAR_1									

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
DMATCR_1									DMAC
CHCR_1	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_2									
DAR_2									
DMATCR_2									
CHCR_2	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
SAR_3									

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
DAR_3									DMAC
DMATCR_3									
CHCR_3	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	DL	DS	TB	TS1	TS0	IE	TE	DE	
DMAOR	—	—	CMS1	CMS0	—	—	PR1	PR0	
	—	—	—	—	—	AE	NMIF	DME	
PADRH	—	—	—	—	—	—	PA25DR	PA24DR	I/O
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR	
PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR	
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR	
PACRH1	—	—	—	—	—	—	—	—	
	—	—	—	—	PA25MD1	PA25MD0	PA24MD1	PA24MD0	
PACRH2	PA23MD1	PA23MD0	PA22MD1	PA22MD0	PA21MD1	PA21MD0	—	PA20MD0	
	—	PA19MD0	—	PA18MD0	—	PA17MD0	—	PA16MD0	
PBDRL	—	—	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PBIORL	—	—	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	
	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
PBCRL1	—	—	—	—	—	PB13MD0	—	PB12MD0	
	—	PB11MD0	—	PB10MD0	—	PB9MD0	—	PB8MD0	
PBCRL2	—	PB7MD0	—	PB6MD0	—	PB5MD0	—	PB4MD0	
	—	PB3MD0	—	PB2MD0	—	PB1MD0	—	PB0MD0	

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
PCDRH	—	—	—	—	—	—	—	—	I/O
	—	—	—	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR	
PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PCIORH	—	—	—	—	—	—	—	—	
	—	—	—	PC20IOR	PC19IOR	PC18IOR	PC17IOR	PC16IOR	
PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	
	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
PCCRH2	—	—	—	—	—	—	—	PC20MD0	
	—	PC19MD0	—	PC18MD0	—	PC17MD0	—	PC16MD0	
PCCRL1	—	PC15MD0	—	PC14MD0	—	PC13MD0	—	PC12MD0	
	—	PC11MD0	—	PC10MD0	—	PC9MD0	—	PC8MD0	
PCCRL2	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	
	—	PC3MD0	—	PC2MD0	—	PC1MD0	—	PC0MD0	
PDDRL	—	—	—	—	—	—	—	—	
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PDIORL	—	—	—	—	—	—	—	—	
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
PDCRL2	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
PEDRH	—	—	—	—	—	—	—	PE24DR	
	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PEIORH	—	—	—	—	—	—	—	PE24IOR	
	PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PECRH1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PE24MD1	PE24MD0	

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
PECRH2	PE23MD1	PE23MD0	PE22MD1	PE22MD0	PE21MD1	PE21MD0	PE20MD1	PE20MD0	I/O
	PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16MD0	
PECRL1	PE15MD1	PE15MD0	PE14MD1	PE14MD0	PE13MD1	PE13MD0	PE12MD1	PE12MD0	
	PE11MD1	PE11MD0	PE10MD1	PE10MD0	PE9MD1	PE9MD0	—	PE8MD0	
PECRL2	—	PE7MD0	PE6MD1	PE6MD0	—	PE5MD0	PE4MD1	PE4MD0	
	—	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
IPRC	IPRC15	IPRC14	IPRC13	IPRC12	IPRC11	IPRC10	IPRC9	IPRC8	INTC
	IPRC7	IPRC6	IPRC5	IPRC4	IPRC3	IPRC2	IPRC1	IPRC0	
IPRD	IPRD15	IPRD14	IPRD13	IPRD12	IPRD11	IPRD10	IPRD9	IPRD8	
	IPRD7	IPRD6	IPRD5	IPRD4	—	—	—	—	
IPRE	IPRE15	IPRE14	IPRE13	IPRE12	IPRE11	IPRE10	IPRE9	IPRE8	
	—	—	—	—	—	—	—	—	
IPRF	IPRF15	IPRF14	IPRF13	IPRF12	IPRF11	IPRF10	IPRF9	IPRF8	
	IPRF7	IPRF6	IPRF5	IPRF4	IPRF3	IPRF2	IPRF1	IPRF0	
IPRG	IPRG15	IPRG14	IPRG13	IPRG12	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0	DMAC
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0	
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0	
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0	
STBCR3	—	—	—	MSTP15	—	MSTP13	MSTP12	MSTP11	Power-down mode
STBCR4	—	—	—	MSTP23	—	MSTP21	MSTP20	MSTP19	
MCLKCR	FLSCS1	FLSCS0	—	—	—	FLDIVS2	FLDIVS1	FLDIVS0	CPG
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	H-UDI
	—	—	—	—	—	—	—	—	
SDID	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16	
	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	

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
ICR0	NMIL	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IRQCR	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IRQSR	IRQ7L	IRQ6L	IRQ5L	IRQ4L	IRQ3L	IRQ2L	IRQ1L	IRQ0L	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
IPRA	IPRA15	IPRA14	IPRA13	IPRA12	IPRA11	IPRA10	IPRA9	IPRA8	
	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0	
IPRB	IPRB15	IPRB14	IPRB13	IPRB12	IPRB11	IPRB10	IPRB9	IPRB8	
	IPRB7	IPRB6	IPRB5	IPRB4	IPRB3	IPRB2	IPRB1	IPRB0	
FRQCR	—	—	—	CKOEN	—	STC2	STC1	STC0	CPG
	—	—	—	—	—	PFC2	PFC1	PFC0	
STBCR	STBY	—	—	—	MDCHG	—	—	—	Power-down mode
WTCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	WDT
WTCSR	TME	WT/IT	—	WOVF	IOVF	CKS2	CKS1	CKS0	
STBCR2	MSTP10	MSTP9	MSTP8	—	—	MSTP5	MSTP4	—	Power-down mode
SCSMR_0	—	—	—	—	—	—	—	—	SCIF_0
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_0	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	

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
SCFDR_0	—	—	—	T4	T3	T2	T1	T0	SCIF_0
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_0	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_1	—	—	—	—	—	—	—	—	SCIF_1
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_1	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_1	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_1	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_2	—	—	—	—	—	—	—	—	SCIF_2
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_2	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	

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
SCFRDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SCIF_2
SCFCR_2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_2	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_2	—	—	—	—	—	—	—	—	
	— (Reserved)	— (Reserved)	— (Reserved)	— (Reserved)	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SIMDR	TRMD1	TRMD0	SYNCAT	REDG	FL3	FL2	FL1	FL0	SIOF
	TXDIZ	RCIM	—	SYNCDL	—	—	—	—	
SISCR	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0	
	—	—	—	—	—	BRDV2	BRDV1	BRDV0	
SITDAR	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0	
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0	
SIRDAR	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0	
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0	
SICDAR	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0	
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0	
SICTR	SCKE	FSE	—	—	—	—	TXE	RXE	
	—	—	—	—	—	—	TXRST	RXRST	
SIFCTR	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0	
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0	
SISTR	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	
	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF	
SIIER	TDMAE	TCRDYE	TFEMPE	TDREQE	RDMAE	RCRDYE	RFFULE	RDREQE	
	—	—	SAERRE	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVFE	

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	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	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	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08	
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00	
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18	
	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10	
SIRCR	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08	
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00	
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18	
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10	
SPICR	SPIM	—	CPHA	CPOL	—	—	—	SS0E	
	—	—	SSAST1	SSAST0	—	—	FLD1	FLD0	
PHYIFCR	—	co_resetb	cksel	—	—	—	—	—	PHY-IF
	—	—	—	—	—	co_st_ mode[2]	co_st_ mode[1]	co_st_ mode[0]	
PHYIFSMIR2	co_reg2_o ui_in[15]	co_reg2_o ui_in[14]	co_reg2_o ui_in[13]	co_reg2_o ui_in[12]	co_reg2_o ui_in[11]	co_reg2_o ui_in[10]	co_reg2_o ui_in[9]	co_reg2_o ui_in[8]	
	co_reg2_o ui_in[7]	co_reg2_o ui_in[6]	co_reg2_o ui_in[5]	co_reg2_o ui_in[4]	co_reg2_o ui_in[3]	co_reg2_o ui_in[2]	co_reg2_o ui_in[1]	co_reg2_o ui_in [0]	
PHYIFSMIR3	co_reg3_o ui_in[15]	co_reg3_o ui_in[14]	co_reg3_o ui_in[13]	co_reg3_o ui_in[12]	co_reg3_o ui_in[11]	co_reg3_o ui_in[10]	co_reg3_o ui_in[9]	co_reg3_o ui_in[8]	
	co_reg3_o ui_in[7]	co_reg3_o ui_in[6]	co_reg3_o ui_in[5]	co_reg3_o ui_in[4]	co_reg3_o ui_in[3]	co_reg3_o ui_in[2]	co_reg3_o ui_in[1]	co_reg3_o ui_in[0]	
PHYIFADDRR	—	—	—	—	—	—	—	—	
	—	—	—	co_st_phy add[4]	co_st_phy add[3]	co_st_phy add[2]	co_st_phy add[1]	co_st_phy add[0]	

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
PHYIFSR	co_pwrupr st	—	—	—	—	—	—	—	PHY-IF
	—	—	—	—	—	—	—	—	
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCOR_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCOR_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
HIFIDX	—	—	—	—	—	—	—	—	HIF
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0	
HIFGSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	STATUS15	STATUS14	STATUS13	STATUS12	STATUS11	STATUS10	STATUS9	STATUS8	
	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATUS0	
HIFSCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	DMD	DPOL	BMD	BSEL	
	—	—	MD1	—	—	WBSWP	EDN	BO	

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
HIFMCR	—	—	—	—	—	—	—	—	HIF
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	LOCK	—	WT	—	RD	—	—	AI/AD	
HIFIICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR	
HIFEICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	EIC6	EIC5	EIC4	EIC3	EIC2	EIC1	EIC0	EIR	
HIFADR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	A9	A8	
	A7	A6	A5	A4	A3	A2	—	—	
HIFDATA	D31	D30	D29	D28	D27	D26	D25	D24	
	D23	D22	D21	D20	D19	D18	D17	D16	
	D15	D14	D13	D12	D11	D10	D9	D8	
	D7	D6	D5	D4	D3	D2	D1	D0	
HIFDTR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DTRG	
HIFBICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	BIE	BIF	

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
HIFBCR	—	—	—	—	—	—	—	—	HIF
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	AC	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	MAP	—	—	—	—	
	—	—	—	—	ENDIAN	—	HIZMEM	HIZCNT	
CS0BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS3BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS4BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS6BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	

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	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS3WCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	
CS3WCR (when SDRAM is in use)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	WTRP1	WTRP0	—	WTRCD1	WTRCD0	—	A3CL1	
	A3CL0	—	—	TRWL1	TRWL0	—	WTRC1	WTRC0	
CS4WCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR (when PCMCIA is in use)	—	—	—	—	—	—	—	—	
	—	—	SA1	SA0	—	—	—	—	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	
CS6BWCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	

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
CS6BWCR (when PCMCIA is in use)	—	—	—	—	—	—	—	—	BSC
	—	—	SA1	SA0	—	—	—	—	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0	
SDCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	RFSH	RMODE	—	BACTV	
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0	
RTCSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CMF	—	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0	
RTCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RTCOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDMR	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	DE	DL1	DL0	—	—	—	SWR	
EDTRR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	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
EDRRR	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RR	
TDLAR	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	
RDLAR	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24	
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16	
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8	
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0	
EESR	—	TWB	—	—	—	TABT	RABT	RFCOF	
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF	
	—	—	—	—	CND	DLC	CD	TRO	
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF	
EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP	
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP	
TRSCER	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CNDCE	DLCCE	CDCE	TROCE	
	RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE	
RMFCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8	
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0	

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
TFTR	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFT10	TFT9	TFT8	
	TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1	TFT0	
FDR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFD2	TFD1	TFD0	
	—	—	—	—	—	RFD2	RFD1	RFD0	
RMCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RNC	
EDOCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FEC	AEC	EDH	—	
FCFTR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFF2	RFF1	RFF0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFD2	RFD1	RFD0	
TRIMD	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	TIS	
RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24	
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16	
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8	
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0	

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
RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24	EtherC
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16	
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8	
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0	
TBRAR	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	
TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA24	
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA16	
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8	
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0	
ECMR	—	—	—	—	—	—	—	—	
	—	—	—	—	ZPF	PFR	RXF	TXF	
	—	—	—	PRCEF	—	—	MPDE	—	
	—	PE	TE	—	ILB	ELB	DM	PRM	
ECSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	PSRTO	—	LCHNG	MPD	ICD	
ECSIPR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP	
PIR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	MDI	MDO	MMD	MDC	

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
MAHR	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40	EtherC
	MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32	
	MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24	
	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16	
MALR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	
	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	
RFLR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	RFL11	RFL10	RFL9	RFL8	
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0	
PSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LMON	
TROCR	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24	
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16	
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8	
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0	
CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC24	
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC16	
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC8	
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC0	
LCCR	LCC31	LCC30	LCC29	LCC28	LCC27	LCC26	LCC25	LCC24	
	LCC23	LCC22	LCC21	LCC20	LCC19	LCC18	LCC17	LCC16	
	LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8	
	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0	

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
CNDCR	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC24	EtherC
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC16	
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8	
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0	
CEFCR	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	
FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24	
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16	
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8	
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0	
TSFRCR	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24	
	TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16	
	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8	
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0	
TLFRCR	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24	
	TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16	
	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8	
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0	
RFCR	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24	
	RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16	
	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8	
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0	
MAFCR	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24	
	MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16	
	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8	
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0	

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
IPGR	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0	
APR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	AP15	AP14	AP13	AP12	AP11	AP10	AP9	AP8	
	AP7	AP6	AP5	AP4	AP3	AP2	AP1	AP0	
MPR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MP15	MP14	MP13	MP12	MP11	MP10	MP9	MP8	
	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	
TPAUSER	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	TPAUSE 15	TPAUSE 14	TPAUSE 13	TPAUSE 12	TPAUSE 11	TPAUSE 10	TPAUSE 9	TPAUSE 8	
	TPAUSE7	TPAUSE6	TPAUSE5	TPAUSE4	TPAUSE3	TPAUSE2	TPAUSE1	TPAUSE0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	

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
BETR	—	—	—	—	—	—	—	—	UBC
	—	—	—	—	—	—	—	—	
	—	—	—	—	BET11	BET10	BET9	BET8	
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0	
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	
BBRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	

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
BBRA	—	—	—	—	—	—	—	—	UBC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
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	
CCR1	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	CB	WT	CE	

24.3 Register States in Each Processing State

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
DMAC	SAR_0	H'F8010020	Initialized	Retained	Retained	Retained
	DAR_0	H'F8010024	Initialized	Retained	Retained	Retained
	DMATCR_0	H'F8010028	Initialized	Retained	Retained	Retained
	CHCR_0	H'F801002C	Initialized	Retained	Retained	Retained
	SAR_1	H'F8010030	Initialized	Retained	Retained	Retained
	DAR_1	H'F8010034	Initialized	Retained	Retained	Retained
	DMATCR_1	H'F8010038	Initialized	Retained	Retained	Retained
	CHCR_1	H'F801003C	Initialized	Retained	Retained	Retained
	SAR_2	H'F8010040	Initialized	Retained	Retained	Retained
	DAR_2	H'F8010044	Initialized	Retained	Retained	Retained
	DMATCR_2	H'F8010048	Initialized	Retained	Retained	Retained
	CHCR_2	H'F801004C	Initialized	Retained	Retained	Retained
	SAR_3	H'F8010050	Initialized	Retained	Retained	Retained
	DAR_3	H'F8010054	Initialized	Retained	Retained	Retained
	DMATCR_3	H'F8010058	Initialized	Retained	Retained	Retained
	CHCR_3	H'F801005C	Initialized	Retained	Retained	Retained
	DMAOR	H'F8010060	Initialized	Retained	Retained	Retained
I/O	PADRH	H'F8050000	Initialized	Retained	—*3	Retained
	PAIORH	H'F8050004	Initialized	Retained	—*3	Retained
	PACRH1	H'F8050008	Initialized	Retained	—*3	Retained
	PACRH2	H'F805000A	Initialized	Retained	—*3	Retained
	PBDRL	H'F8050012	Initialized	Retained	—*3	Retained
	PBIORL	H'F8050016	Initialized	Retained	—*3	Retained
	PBCRL1	H'F805001C	Initialized	Retained	—*3	Retained
	PBCRL2	H'F805001E	Initialized	Retained	—*3	Retained
	PCDRH	H'F8050020	Initialized	Retained	—*3	Retained
	PCDRL	H'F8050022	Initialized	Retained	—*3	Retained
	PCIORH	H'F8050024	Initialized	Retained	—*3	Retained
	PCIORL	H'F8050026	Initialized	Retained	—*3	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
I/O	PCCRH2	H'F805002A	Initialized	Retained	—*3	Retained
	PCCRL1	H'F805002C	Initialized	Retained	—*3	Retained
	PCCRL2	H'F805002E	Initialized	Retained	—*3	Retained
	PDDRL	H'F8050032	Initialized	Retained	—*3	Retained
	PDIORL	H'F8050036	Initialized	Retained	—*3	Retained
	PDCRL2	H'F805003E	Initialized	Retained	—*3	Retained
	PEDRH	H'F8050040	Initialized	Retained	—*3	Retained
	PEDRL	H'F8050042	Initialized	Retained	—*3	Retained
	PEIORH	H'F8050044	Initialized	Retained	—*3	Retained
	PEIORL	H'F8050046	Initialized	Retained	—*3	Retained
	PECRH1	H'F8050048	Initialized	Retained	—*3	Retained
	PECRH2	H'F805004A	Initialized	Retained	—*3	Retained
	PECRL1	H'F805004C	Initialized	Retained	—*3	Retained
	PECRL2	H'F805004E	Initialized	Retained	—*3	Retained
INTC	IPRC	H'F8080000	Initialized	Retained	—*3	Retained
	IPRD	H'F8080002	Initialized	Retained	—*3	Retained
	IPRE	H'F8080004	Initialized	Retained	—*3	Retained
	IPRF	H'F8080006	Initialized	Retained	—*3	Retained
	IPRG	H'F8080008	Initialized	Retained	—*3	Retained
DMAC	DMARS0	H'F8090000	Initialized	Retained	Retained	Retained
	DMARS1	H'F8090004	Initialized	Retained	Retained	Retained
Power-down mode	STBCR3	H'F80A0000	Initialized	Retained	—*3	Retained
	STBCR4	H'F80A0004	Initialized	Retained	—*3	Retained
CPG	MCLKCR	H'F80A000C	Initialized	Retained	—*3	Retained
H-UDI	SDIR	H'F8100200	Initialized	Retained	Retained	Retained
	SDID	H'F8100214	Initialized	Retained	Retained	Retained
INTC	ICR0	H'F8140000	Initialized*1	Retained	—*3	Retained
	IRQCR	H'F8140002	Initialized	Retained	—*3	Retained
	IRQSR	H'F8140004	Initialized*1	Retained	—*3	Retained
	IPRA	H'F8140006	Initialized	Retained	—*3	Retained
	IPRB	H'F8140008	Initialized	Retained	—*3	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
CPG	FRQCR	H'F815FF80	Initialized*[2]	Retained	—*[3]	Retained
Power-down mode	STBCR	H'F815FF82	Initialized	Retained	—*[3]	Retained
WDT	WTCNT	H'F815FF84	Initialized*[2]	Retained	—*[3]	Retained
	WTCSR	H'F815FF86	Initialized*[2]	Retained	—*[3]	Retained
Power-down mode	STBCR2	H'F815FF88	Initialized	Retained	—*[3]	Retained
SCIF_0	SCSMR_0	H'F8400000	Initialized	Retained	Retained	Retained
	SCBRR_0	H'F8400004	Initialized	Retained	Retained	Retained
	SCSCR_0	H'F8400008	Initialized	Retained	Retained	Retained
	SCFTDR_0	H'F840000C	Undefined	Retained	Retained	Retained
	SCFSR_0	H'F8400010	Initialized	Retained	Retained	Retained
	SCFRDR_0	H'F8400014	Undefined	Retained	Retained	Retained
	SCFCR_0	H'F8400018	Initialized	Retained	Retained	Retained
	SCFDR_0	H'F840001C	Initialized	Retained	Retained	Retained
	SCSPTR_0	H'F8400020	Initialized*[1]	Retained	Retained	Retained
	SCLSR_0	H'F8400024	Initialized	Retained	Retained	Retained
SCIF_1	SCSMR_1	H'F8410000	Initialized	Retained	Retained	Retained
	SCBRR_1	H'F8410004	Initialized	Retained	Retained	Retained
	SCSCR_1	H'F8410008	Initialized	Retained	Retained	Retained
	SCFTDR_1	H'F841000C	Undefined	Retained	Retained	Retained
	SCFSR_1	H'F8410010	Initialized	Retained	Retained	Retained
	SCFRDR_1	H'F8410014	Undefined	Retained	Retained	Retained
	SCFCR_1	H'F8410018	Initialized	Retained	Retained	Retained
	SCFDR_1	H'F841001C	Initialized	Retained	Retained	Retained
	SCSPTR_1	H'F8410020	Initialized*[1]	Retained	Retained	Retained
	SCLSR_1	H'F8410024	Initialized	Retained	Retained	Retained
SCIF_2	SCSMR_2	H'F8420000	Initialized	Retained	Retained	Retained
	SCBRR_2	H'F8420004	Initialized	Retained	Retained	Retained
	SCSCR_2	H'F8420008	Initialized	Retained	Retained	Retained
	SCFTDR_2	H'F842000C	Undefined	Retained	Retained	Retained
	SCFSR_2	H'F8420010	Initialized	Retained	Retained	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
SCIF_2	SCFRDR_2	H'F8420014	Undefined	Retained	Retained	Retained
	SCFCR_2	H'F8420018	Initialized	Retained	Retained	Retained
	SCFDR_2	H'F842001C	Initialized	Retained	Retained	Retained
	SCSPTR_2	H'F8420020	Initialized*[1]	Retained	Retained	Retained
	SCLSR_2	H'F8420024	Initialized	Retained	Retained	Retained
SIOF	SIMDR	H'F8480000	Initialized	Retained	Retained	Retained
	SISCR	H'F8480002	Initialized	Retained	Retained	Retained
	SITDAR	H'F8480004	Initialized	Retained	Retained	Retained
	SIRDAR	H'F8480006	Initialized	Retained	Retained	Retained
	SICDAR	H'F8480008	Initialized	Retained	Retained	Retained
	SICTR	H'F848000C	Initialized	Retained	Retained	Retained
	SIFCTR	H'F8480010	Initialized	Retained	Retained	Retained
	SISTR	H'F8480014	Initialized	Retained	Retained	Retained
	SIIER	H'F8480016	Initialized	Retained	Retained	Retained
	SITDR	H'F8480020	Initialized	Retained	Retained	Retained
	SIRDR	H'F8480024	Initialized	Retained	Retained	Retained
	SITCR	H'F8480028	Initialized	Retained	Retained	Retained
	SIRCR	H'F848002C	Initialized	Retained	Retained	Retained
	SPICR	H'F8480030	Initialized	Retained	Retained	Retained
PHY-IF	PHYIFCR	H'F8490000	Initialized	Initialized	Retained	Retained
	PHYIFSMIR2	H'F8490004	Initialized	Initialized	Retained	Retained
	PHYIFSMIR3	H'F8490008	Initialized	Initialized	Retained	Retained
	PHYIFADDRR	H'F849000C	Initialized	Initialized	Retained	Retained
	PHYIFSR	H'F8490010	Initialized*[4]	Initialized	Retained	Retained
CMT	CMSTR	H'F84A0070	Initialized	Initialized	Retained	Retained
	CMCSR_0	H'F84A0072	Initialized	Initialized	Retained	Retained
	CMCNT_0	H'F84A0074	Initialized	Initialized	Retained	Retained
	CMCOR_0	H'F84A0076	Initialized	Initialized	Retained	Retained
	CMCSR_1	H'F84A0078	Initialized	Initialized	Retained	Retained
	CMCNT_1	H'F84A007A	Initialized	Initialized	Retained	Retained
	CMCOR_1	H'F84A007C	Initialized	Initialized	Retained	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
HIF	HIFIDX	H'F84D0000	Initialized	Retained	Retained	Retained
	HIFGSR	H'F84D0004	Initialized	Retained	Retained	Retained
	HIFSCR	H'F84D0008	Initialized*1	Retained	Retained	Retained
	HIFMCR	H'F84D000C	Initialized	Retained	Retained	Retained
	HIFIICR	H'F84D0010	Initialized	Retained	Retained	Retained
	HIFEICR	H'F84D0014	Initialized	Retained	Retained	Retained
	HIFADR	H'F84D0018	Initialized	Retained	Retained	Retained
	HIFDATA	H'F84D001C	Initialized	Retained	Retained	Retained
	HIFDTR	H'F84D0020	Initialized	Retained	Retained	Retained
	HIFBICR	H'F84D0024	Initialized	Retained	Retained	Retained
	HIFBCR	H'F84D0040	Initialized*1	Retained	Retained	Retained
BSC	CMNCR	H'F8FD0000	Initialized*1	Retained	—*3	Retained
	CS0BCR	H'F8FD0004	Initialized	Retained	—*3	Retained
	CS3BCR	H'F8FD000C	Initialized	Retained	—*3	Retained
	CS4BCR	H'F8FD0010	Initialized	Retained	—*3	Retained
	CS5BBCR	H'F8FD0018	Initialized	Retained	—*3	Retained
	CS6BBCR	H'F8FD0020	Initialized	Retained	—*3	Retained
	CS0WCR	H'F8FD0024	Initialized	Retained	—*3	Retained
	CS3WCR	H'F8FD002C	Initialized	Retained	—*3	Retained
	CS3WCR (SDRAM in use)	H'F8FD002C	Initialized	Retained	—*3	Retained
	CS4WCR	H'F8FD0030	Initialized	Retained	—*3	Retained
	CS5BWCR	H'F8FD0038	Initialized	Retained	—*3	Retained
	CS5BWCR (PCMCIA in use)	H'F8FD0038	Initialized	Retained	—*3	Retained
	CS6BWCR	H'F8FD0040	Initialized	Retained	—*3	Retained
	CS6BWCR (PCMCIA in use)	H'F8FD0040	Initialized	Retained	—*3	Retained
	SDCR	H'F8FD0044	Initialized	Retained	—*3	Retained
	RTCSR	H'F8FD0048	Initialized	Retained	—*3	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
E-DMAC	RTCNT	H'F8FD004C	Initialized	Retained	—*3	Retained
	RTCOR	H'F8FD0050	Initialized	Retained	—*3	Retained
	EDMR	H'FB000000	Initialized	Retained	Retained	Retained
	EDTRR	H'FB000004	Initialized	Retained	Retained	Retained
	EDRRR	H'FB000008	Initialized	Retained	Retained	Retained
	TDLAR	H'FB00000C	Initialized	Retained	Retained	Retained
	RDLAR	H'FB000010	Initialized	Retained	Retained	Retained
	EESR	H'FB000014	Initialized	Retained	Retained	Retained
	EESIPR	H'FB000018	Initialized	Retained	Retained	Retained
	TRSCER	H'FB00001C	Initialized	Retained	Retained	Retained
	RMFCR	H'FB000020	Initialized	Retained	Retained	Retained
	TFTR	H'FB000024	Initialized	Retained	Retained	Retained
	FDR	H'FB000028	Initialized	Retained	Retained	Retained
	RMCR	H'FB00002C	Initialized	Retained	Retained	Retained
	EDOCR	H'FB000030	Initialized	Retained	Retained	Retained
	FCFTR	H'FB000034	Initialized	Retained	Retained	Retained
	TRIMD	H'FB00003C	Initialized	Retained	Retained	Retained
	RBWAR	H'FB000040	Initialized	Retained	Retained	Retained
	RDFAR	H'FB000044	Initialized	Retained	Retained	Retained
	TBRAR	H'FB00004C	Initialized	Retained	Retained	Retained
	TDFAR	H'FB000050	Initialized	Retained	Retained	Retained
EtherC	ECMR	H'FB000160	Initialized	Retained	Retained	Retained
	ECSR	H'FB000164	Initialized	Retained	Retained	Retained
	ECSIPR	H'FB000168	Initialized	Retained	Retained	Retained
	PIR	H'FB00016C	Initialized*1	Retained	Retained	Retained
	MAHR	H'FB000170	Initialized	Retained	Retained	Retained
	MALR	H'FB000174	Initialized	Retained	Retained	Retained
	RFLR	H'FB000178	Initialized	Retained	Retained	Retained
	PSR	H'FB00017C	Initialized*1	Retained	Retained	Retained
	TROCR	H'FB000180	Initialized	Retained	Retained	Retained
	CDCR	H'FB000184	Initialized	Retained	Retained	Retained

Module	Abbr.	Address	Power-On Reset	Software Standby	Module Standby	Sleep
EtherC	LCCR	H'FB000188	Initialized	Retained	Retained	Retained
	CNDCR	H'FB00018C	Initialized	Retained	Retained	Retained
	CEFCR	H'FB000194	Initialized	Retained	Retained	Retained
	FRECR	H'FB000198	Initialized	Retained	Retained	Retained
	TSFRCR	H'FB00019C	Initialized	Retained	Retained	Retained
	TLFRCR	H'FB0001A0	Initialized	Retained	Retained	Retained
	RFCR	H'FB0001A4	Initialized	Retained	Retained	Retained
	MAFCR	H'FB0001A8	Initialized	Retained	Retained	Retained
	IPGR	H'FB0001B4	Initialized	Retained	Retained	Retained
	APR	H'FB0001B8	Initialized	Retained	Retained	Retained
	MPR	H'FB0001BC	Initialized	Retained	Retained	Retained
	TPAUSER	H'FB0001C4	Initialized	Retained	Retained	Retained
UBC	BDRB	H'FFFFFF90	Initialized	Retained	Retained	Retained
	BDMRB	H'FFFFFF94	Initialized	Retained	Retained	Retained
	BRCR	H'FFFFFF98	Initialized	Retained	Retained	Retained
	BETR	H'FFFFFF9C	Initialized	Retained	Retained	Retained
	BARB	H'FFFFFFA0	Initialized	Retained	Retained	Retained
	BAMRB	H'FFFFFFA4	Initialized	Retained	Retained	Retained
	BBRB	H'FFFFFFA8	Initialized	Retained	Retained	Retained
	BRSR	H'FFFFFFAC	Initialized	Retained	Retained	Retained
	BARA	H'FFFFFFB0	Initialized	Retained	Retained	Retained
	BAMRA	H'FFFFFFB4	Initialized	Retained	Retained	Retained
	BBRA	H'FFFFFFB8	Initialized	Retained	Retained	Retained
	BRDR	H'FFFFFFBC	Initialized*[1]	Retained	Retained	Retained
Cache	CCR1	H'FFFFFFEC	Initialized	Retained	Retained	Retained

Notes: 1. Some bits are not initialized.
2. Not initialized by a power-on reset caused by the WDT.
3. This module does not enter the module standby mode.
4. Initialization by applying the PHY power supply, not by a reset through power-on reset pin.

Section 25 Electrical Characteristics

25.1 Absolute Maximum Ratings

Table 25.1 shows the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

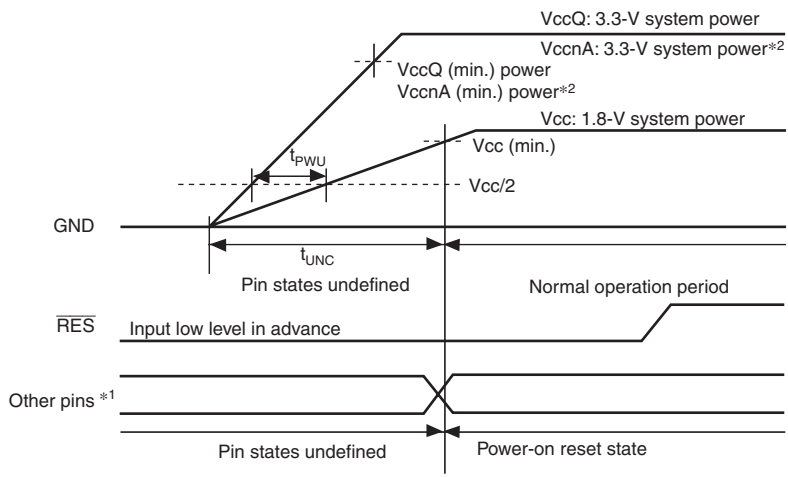
Item	Symbol	Value	Unit
Power supply voltage (I/O)	$V_{CC}Q$	–0.3 to +3.8	V
Power supply voltage (internal)	V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	–0.3 to +2.1	V
Input voltage	V_{in}	–0.3 to $V_{CC}Q$ + 0.3	V
Analog power supply (PHY)	$V_{CC}1A$ $V_{CC}2A$ $V_{CC}3A$	–0.3 to +3.8	V
Operating temperature	T_{opr}	See the operating temperatures given in appendix B, Product Code Lineup.	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

25.2 Power-On and Power-Off Order

- Order of turning on 1.8-V system power (Vcc, Vcc (PLL1), and Vcc (PLL2)) and 3.3-V system power (VccQ, Vcc1A, Vcc2A, and Vcc3A)
  - — First turn on the 3.3-V system power, then turn on the 1.8-V system power within 1 ms. This time should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation.
  - — Until voltage is applied to all power supplies and a low level is input to the $\overline{RES}$ 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.

  Waveforms at power-on are shown in the following figure.



Notes: 1. Except power/GND and clock related pins
2. n = 1 to 3

Table 25.2 Recommended Timing at Power-On

Item	Symbol	Maximum Value	Unit
Time difference between turning on VccQ, VccnA (n = 1 to 3), and Vcc	t_{PWU}	1	ms
Time over which the internal state is undefined	t_{UNC}	100	ms

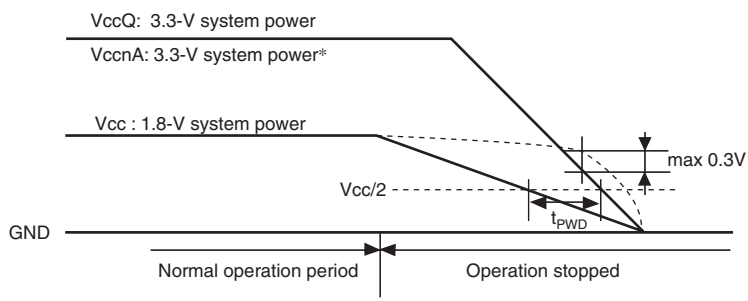
Note: * The values shown in table 25.2 are recommended values, so they represent guidelines rather than strict requirements.

The time over which the internal state is undefined means the time taken to reach Vcc (min.).

The pin states become settled when VccQ and VccnA (n = 1 to 3) reached the VccQ (min.). The timing when a power-on reset ($\overline{RES}$) is normally accepted is after Vcc reaches Vcc (min.) and oscillation becomes stable (when using the on-chip oscillator).

Ensure that the time over which the internal state is undefined is less than or equal to 100 ms.

- Power-off order
  - — In the reverse order of power-on, first turn off the 1.8-V system power, then turn off the 3.3-V system power within 10 ms. This time should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation. In some systems, Vcc may exceed 3.3-V system power (Vcc > 3.3-V system power) temporarily, on the falling edge. Even in this case, the inverted potential difference must be 0.3 V or less.
  - — Pin states are undefined while only the 1.8-V system power is turned off. The system design must ensure that these undefined states do not cause erroneous system operation.



Note: * n = 1 to 3

Table 25.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value	Unit
Time difference between turning off VccQ, VccnA (n = 1 to 3), and Vcc	t_{PWD}	10	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

25.3 DC Characteristics

Tables 25.4 and 25.5 show the DC characteristics.

Table 25.4 DC Characteristics (1)

Conditions: For Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current consumption	Normal operation	I_{CC}	—	250	300	mA	V_{CC} = 1.8 V $V_{CC}Q$ = 3.3 V Iϕ = 125 MHz Bϕ = 62.5 MHz
		$I_{CC}Q$	—	60	100	mA	
	Standby mode	I_{stby} (V_{CC})	—	700*	—	μA	T_a = 25°C V_{CC} = 1.8 V $V_{CC}Q$ = 3.3 V *: Reference value
		I_{stby} ($V_{CC}Q$, $V_{CC}nA$ (n = 1 to 3))	—	20*	—		
	Sleep mode	I_{sleep}	—	70	150	mA	V_{CC} = 1.8 V $V_{CC}Q$ = 3.3 V B ϕ = 62.5 MHz
Input leakage current	All pins	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Tri-state leakage current	I/O pins, all output pins (off state)	$\lvert I_{STI} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Input capacitance	RxP/M	C	—	—	30	pF	
	Other than above		—	—	10		

Table 25.4 DC Characteristics (2)

Conditions: For Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply		$V_{CC}Q$	3.0	3.3	3.6	V	
		V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	1.71	1.8	1.89		
		$V_{CC}1A$ $V_{CC}2A$ $V_{CC}3A$	3.0	3.3	3.6		Note: The voltage should be the same as $V_{CC}Q$.
Input high voltage	$\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, HIFMD, $\overline{TRST}$	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	EXTAL, CK_PHY		$V_{CC}Q - 0.3$	—	$V_{CC}Q + 0.3$		
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, HIFMD, $\overline{TRST}$	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$		
	EXTAL, CK_PHY		–0.3	—	$V_{CC}Q \times 0.2$		
	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} = –200 μA
			2.0	—	—		$V_{CC}Q$ = 3.0 V I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	$V_{CC}Q$ = 3.6 V I_{OL} = 2.0 mA

Notes: 1. The Vcc and Vss pins must be connected to the V_{CC} and V_{SS}.

2. Current consumption values are for V_{IH} min. = $V_{CC}Q$ – 0.5 V and V_{IL} max. = 0.5 V with all output pins unloaded.

Table 25.5 Permissible Output Currents

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	ΣI_{OL}	—	—	120	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 25.5.

25.4 AC Characteristics

Signals input to this LSI are basically handled as signals synchronized with the clock. Unless otherwise noted, setup and hold times for individual signals must be followed.

Table 25.6 Maximum Operating Frequency

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Operating frequency	CPU, cache (Iϕ)	f	20	—	125	MHz	
	External bus (Bϕ)		20	—	62.5		
	On-chip peripheral module (Pϕ)		5	—	31.25		

25.4.1 Clock Timing

Table 25.7 Clock Timing

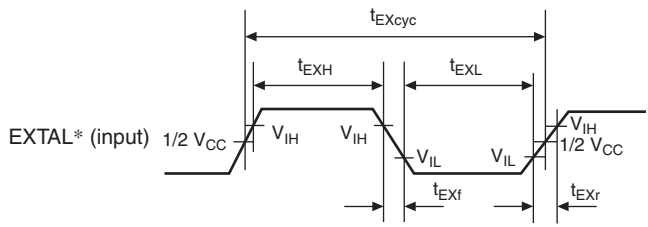
Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup; External bus operating frequency (Max.) = 62.5 MHz

Item	Symbol	Min.	Max.	Unit.	Reference Figures
EXTAL clock input frequency	f_{EX}	10	25	MHz	Figure 25.1
EXTAL clock input cycle time	t_{Excyc}	40	100	ns	
EXTAL clock input low pulse width	t_{EXL}	10	—	ns	
EXTAL clock input high pulse width	t_{EXH}	10	—	ns	
EXTAL clock rising time	t_{Exr}	—	4	ns	
EXTAL clock falling time	t_{Exf}	—	4	ns	
CKIO clock output frequency	f_{OP}	20	62.5	MHz	Figure 25.2
CKIO clock output cycle time	t_{cyc}	16	50	ns	
CKIO clock low pulse width	t_{CKOL}	3.5	—	ns	
CKIO clock high pulse width	t_{CKOH}	3.5	—	ns	
CKIO clock rising time	t_{CKOr}	—	4.5	ns	
CKIO clock falling time	t_{CKOf}	—	4.5	ns	
CK_PHY clock input frequency	f_{CKPHY}	25 –100 ppm*[1]	25 +100 ppm*[1]	MHz	
CK_PHY clock input low pulse width	t_{CKPHYL}	12	—	ns	
CK_PHY clock input high pulse width	t_{CKPHYH}	12	—	ns	
CK_PHY clock input rising time	t_{CKPHYr}	—	6	ns	
CK_PHY clock input falling time	t_{CKPHYf}	—	6	ns	
Oscillation settling time (power-on)	t_{OSC1}	10	—	ms	Figure 25.3
$\overline{RES}$ setup time	t_{RESS}	25	—	ns	Figures 25.3 and 25.4
$\overline{RES}$ assert time	t_{RESW}	20	—	t_{bcyc}*[2]	

Item	Symbol	Min.	Max.	Unit.	Reference Figures
Oscillation settling time 1 (leaving standby mode)	t_{OSC2}	10	—	ms	Figure 25.4
Oscillation settling time 2 (leaving standby mode)	t_{OSC3}	—	10	ms	Figure 25.5
PLL synchronize settling time	t_{PLL}	—	100	μs	Figure 25.6

Notes: 1. Error margin means frequency tolerance (reference value). Recommending under 100 ps of peak to peak jitter.

2. t_{bcyc} indicates the period of the external bus clock (Bϕ).



Note: * When the clock is input to the EXTAL pin

Figure 25.1 External Clock Input Timing

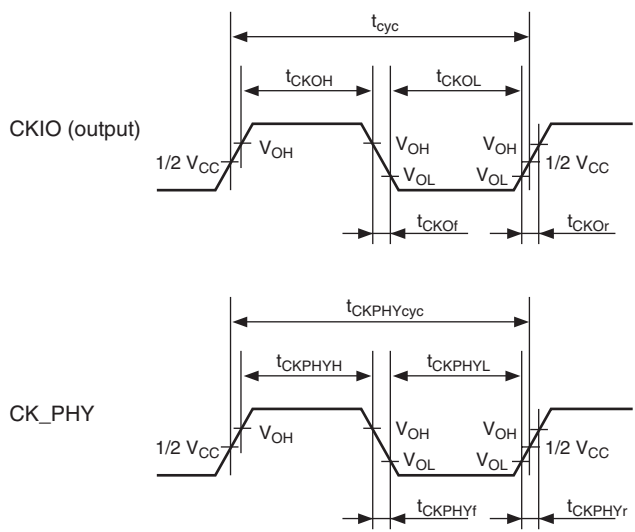
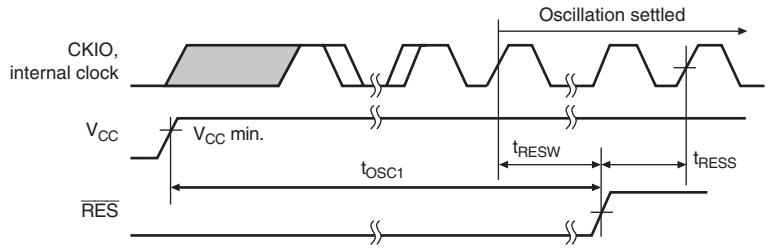
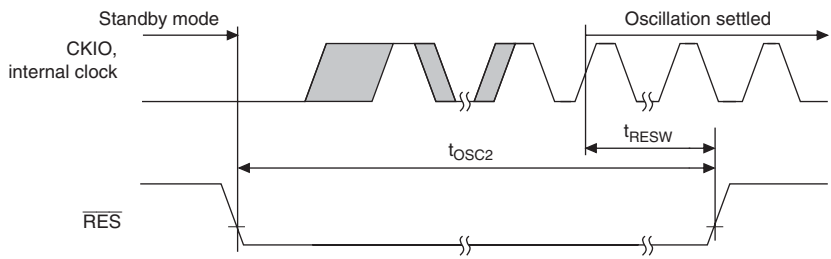


Figure 25.2 CKIO Clock Output Timing and CK_PHY Clock Input Timing



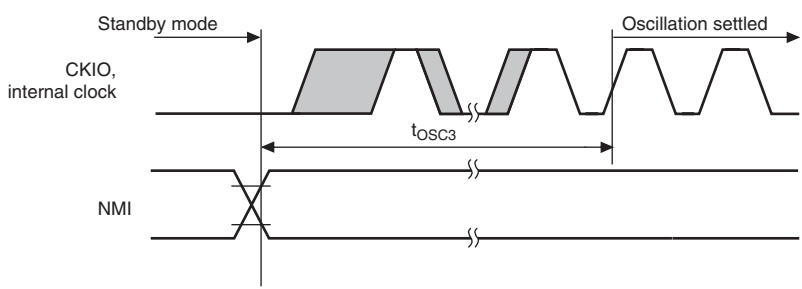
Note: Oscillation settling time when the internal oscillator is in use

Figure 25.3 Oscillation Settling Timing after Power-On



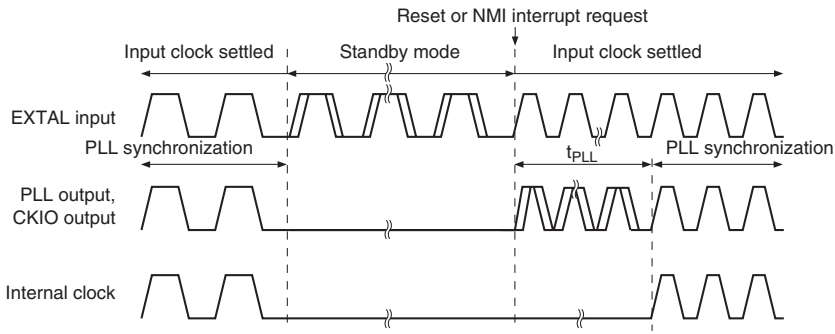
Note: Oscillation settling time when the internal oscillator is in use

Figure 25.4 Oscillation Settling Timing after Standby Mode (By Reset)



Note: Oscillation settling time when the internal oscillator is in use

Figure 25.5 Oscillation Settling Timing after Standby Mode (By NMI or IRQ)



Note: PLL oscillation settling time when the clock is input to the EXTAL pin

Figure 25.6 PLL Synchronize Settling Timing By Reset or NMI

25.4.2 Control Signal Timing

Table 25.8 Control Signal Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
$\overline{RES}$ pulse width	t_{RESW}	20*2	—	t_{bcyc}*3	Figures 25.7 and 25.8
$\overline{RES}$ setup time*1	t_{RESS}	25	—	ns	
$\overline{RES}$ hold time	t_{RESH}	15	—	ns	
NMI setup time*1	t_{NMIS}	12	—	ns	Figure 25.8
NMI hold time	t_{NMIH}	10	—	ns	
IRQ7 to IRQ0 setup time*1	t_{IRQS}	12	—	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	10	—	ns	
Bus tri-state delay time 1	t_{BOFF1}	—	20	ns	Figure 25.9
Bus tri-state delay time 2	t_{BOFF2}	—	20	ns	
Bus buffer on time 1	t_{BON1}	—	20	ns	
Bus buffer on time 2	t_{BON2}	—	20	ns	

Notes:
1. The $\overline{RES}$, NMI, and IRQ7 to IRQ0 signals are asynchronous signals. When the setup time is satisfied, a signal change is detected at the rising edge of the clock signal. When the setup time is not satisfied, a signal change may be delayed to the next rising edge.
2. In standby mode, $t_{RESW} = t_{OSC2}$ (10 ms). When changing the clock multiplication, $t_{RESW} = t_{PLL1}$ (100 μs).
3. t_{bcyc} indicates the period of the external bus clock (Bϕ).

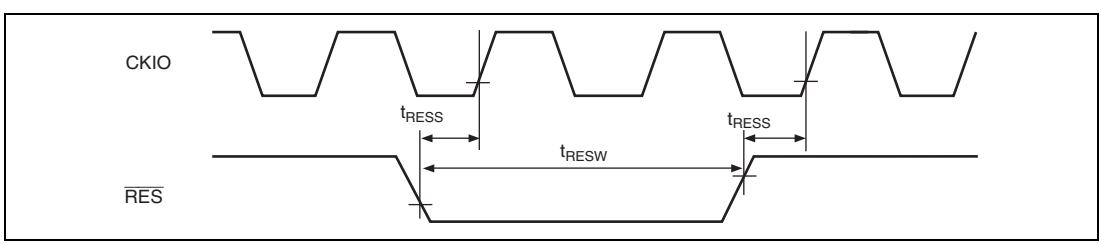


Figure 25.7 Reset Input Timing

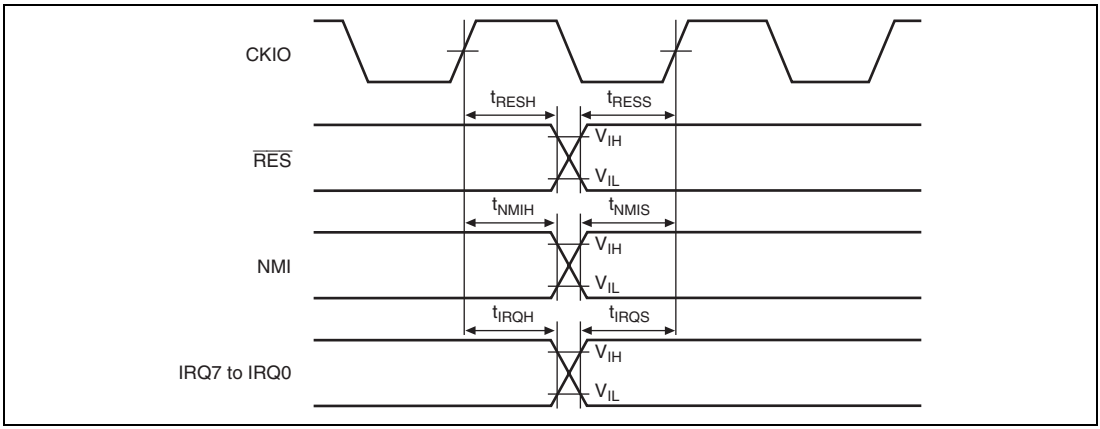


Figure 25.8 Interrupt Input Timing

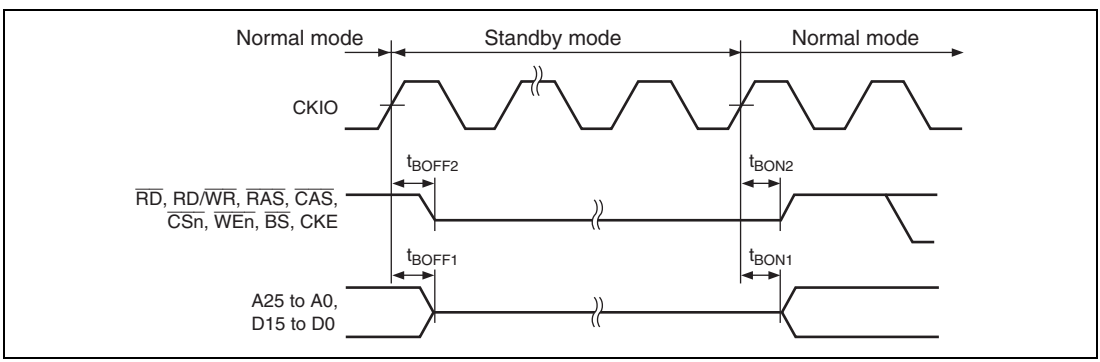


Figure 25.9 Pin Drive Timing in Standby Mode

25.4.3 AC Bus Timing

Table 25.9 Bus Timing

Conditions: Clock mode = 1/2/5/6, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V;
for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
Address delay time 1	t_{AD1}	1	14	ns	Figures 25.10 to 25.36
Address setup time	t_{AS}	3	—	ns	Figures 25.10 to 25.13
Address hold time	t_{AH}	3	—	ns	Figures 25.10 to 25.13
$\overline{BS}$ delay time	t_{BSD}	—	14	ns	Figures 25.10 to 25.29 and 25.33 to 25.36
$\overline{CS}$ delay time 1	t_{CSD1}	1	14	ns	Figures 25.10 to 25.36
Read write delay time	t_{RWD1}	1	14	ns	Figures 25.10 to 25.36
Read strobe time	t_{RSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 13$	ns	Figures 25.10 to 25.15, 25.33, and 25.34
Read data setup time 1	t_{RDS1}	$1/2 \times t_{bcyc} + 10$	—	ns	Figures 25.10 to 25.15 and 25.33 to 25.36
Read data setup time 2	t_{RDS2}	10	—	ns	Figures 25.16 to 25.19, Figures 25.24 to 25.26
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 25.10 to 25.15 and 25.33 to 25.36
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 25.16 to 25.19 and 25.24 to 25.26
Write enable delay time 1	t_{WED1}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 10$	ns	Figures 25.10 to 25.14, 25.33, and 25.34
Write enable delay time 2	t_{WED2}	—	13	ns	Figure 25.15
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 25.10 to 25.15 and 25.33 to 25.36
Write data delay time 2	t_{WDD2}	—	14	ns	Figures 25.20 to 25.23 and 25.27 to 25.29
Write data hold time 1	t_{WDH1}	2	—	ns	Figures 25.10 to 25.15 and 25.33 to 25.36

Item	Symbol	Min.	Max.	Unit	Reference Figures
Write data hold time 2	t_{WDH2}	2	—	ns	Figures 25.20 to 25.23 and 25.27 to 25.29
Write data hold time 3	t_{WDH3}	0	—	ns	Figures 25.10 to 25.13
$\overline{WAIT}$ setup time	t_{WTS1}*	$1/2 \times t_{bcyc} + 11$	—	ns	Figures 25.12 to 25.15, 25.34, and 25.36
$\overline{WAIT}$ hold time	t_{WTH1}*	$1/2 \times t_{bcyc} + 10$	—	ns	Figures 25.12 to 25.15, 25.34, and 25.36
$\overline{RAS}$ delay time	t_{RASD1}	1	14	ns	Figures 25.16 to 25.27 and 25.29 to 25.32
$\overline{CAS}$ delay time	t_{CASD1}	1	14	ns	Figures 25.16 to 25.32
DQM delay time	t_{DQMD1}	1	14	ns	Figures 25.16 to 25.29
CKE delay time	t_{CKED1}	—	14	ns	Figure 25.31
$\overline{ICIORD}$ delay time	t_{ICRSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 15$	ns	Figures 25.35 and 25.36
$\overline{ICIOWR}$ delay time	t_{ICWSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 15$	ns	Figures 25.35 and 25.36
$\overline{IOIS16}$ setup time	t_{IO16S}	$1/2 \times t_{bcyc} + 11$	—	ns	Figure 25.36
$\overline{IOIS16}$ hold time	t_{IO16H}	$1/2 \times t_{bcyc} + 10$	—	ns	Figure 25.36

Note: * The AC timing specification of $\overline{WAIT}$ is as follows.

Input setup time + hold time of $\overline{WAIT}$

= 11 [ns] + 10 [ns] = 21 [ns]

As the frequency, 47.62 [MHz]

Therefore, when the bus clock is 47.62 MHz or more, at least either setup time or hold time cannot be satisfied during 1-bus clock. The following notes should be confirmed.

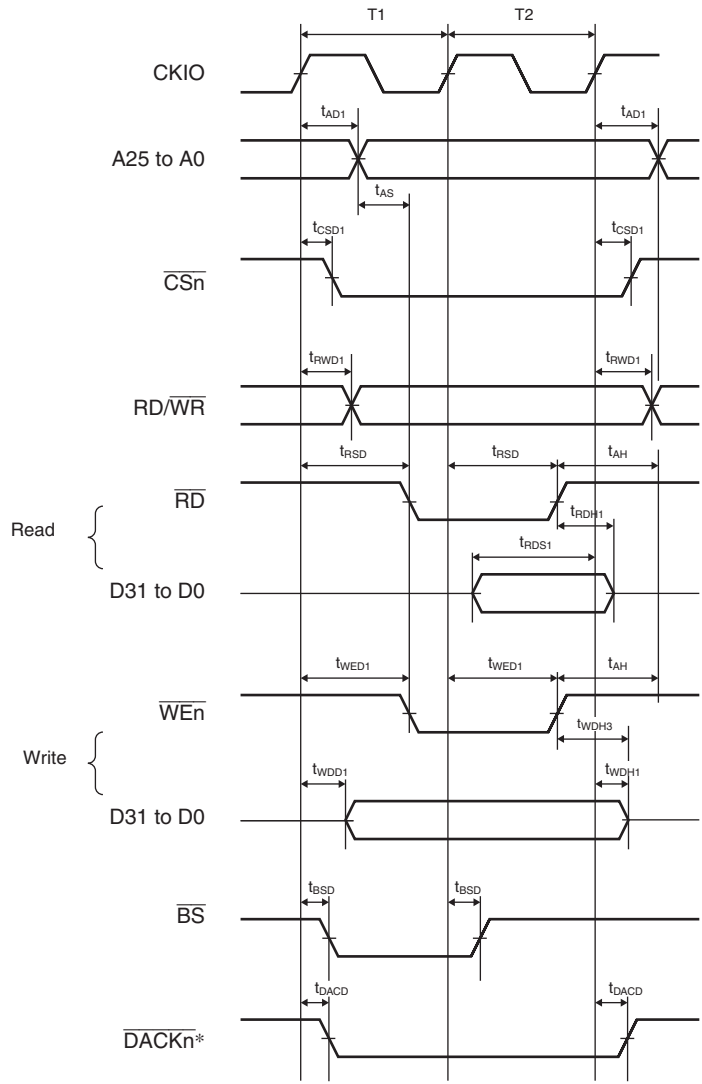
- When the hardware-wait function is used synchronously

The bus clock frequency must be low enough to satisfy the AC specification above.

- When the hardware-wait function is used asynchronously

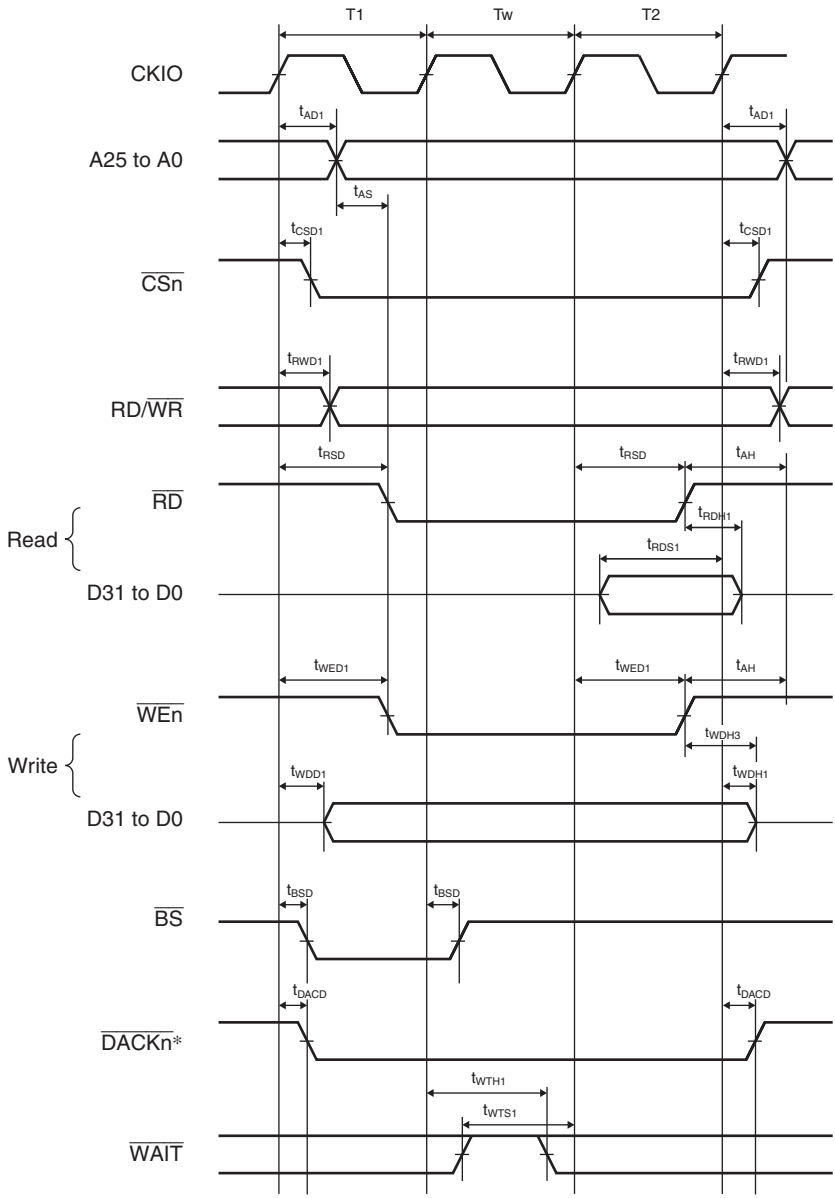
To ensure the setup time until the start of the input assertion of $\overline{WAIT}$, insert appropriate number of the software wait after the T1 state. Then, even if the AC specification above cannot be satisfied, the accesses can be executed correctly.

25.4.4 Basic Timing



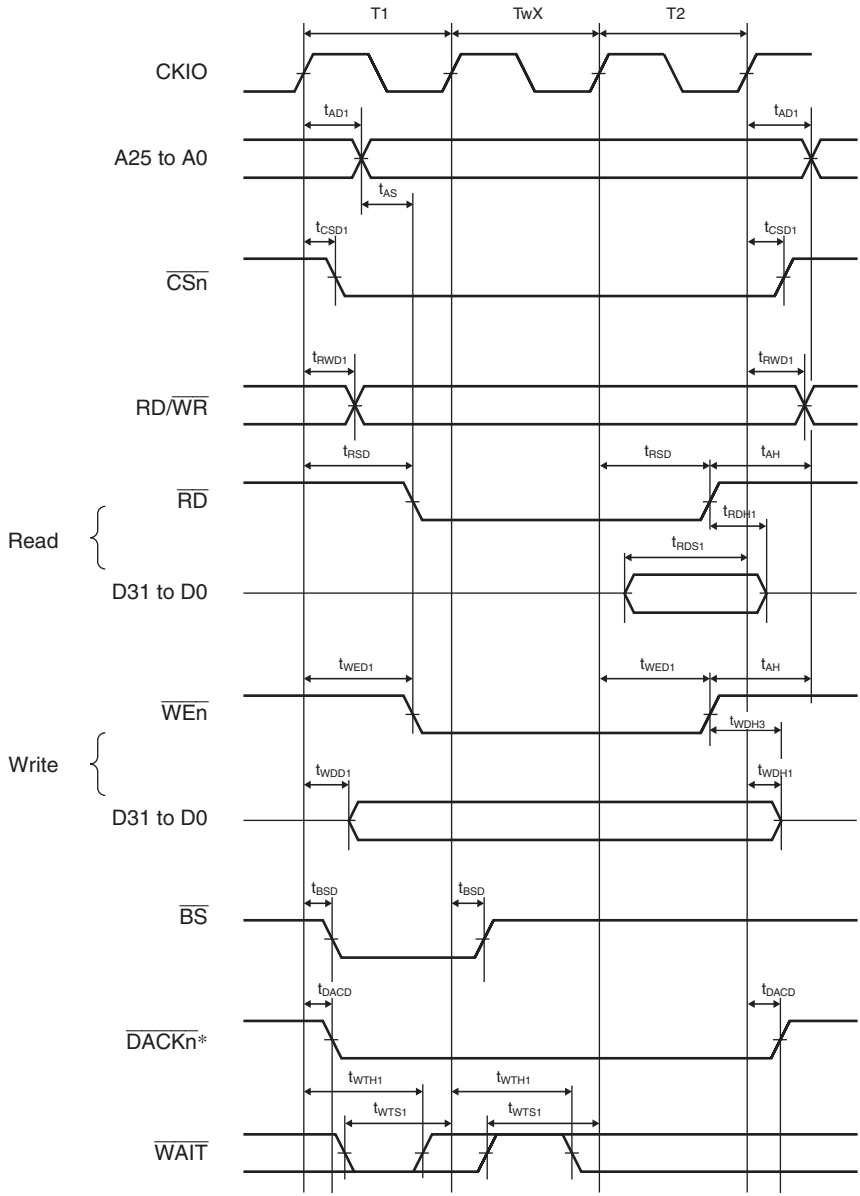
Note : * $\overline{\text{DACKn}}$ is the waveform when active low is selected.

Figure 25.10 Basic Bus Timing: No Wait Cycle



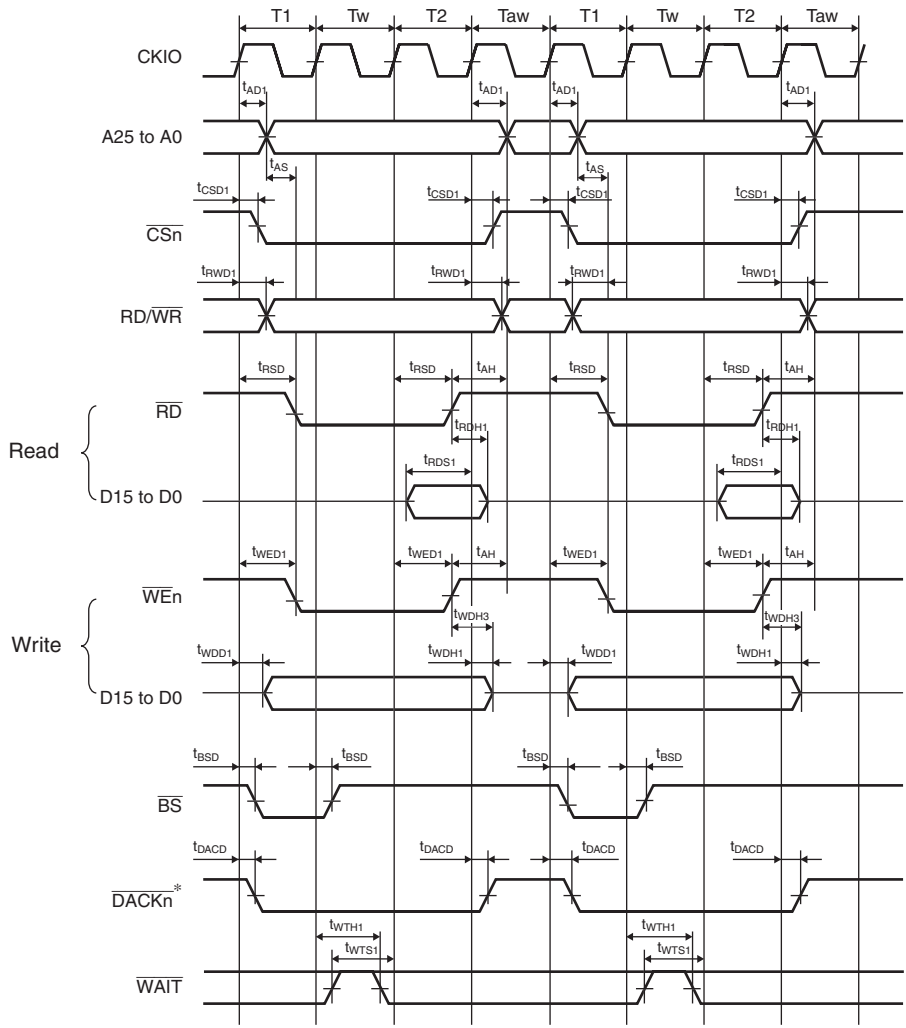
Note : * DACKn is the waveform when active low is selected.

Figure 25.11 Basic Bus Timing: One Software Wait Cycle



Note : * $\overline{\text{DACKn}}$ is the waveform when active low is selected.

Figure 25.12 Basic Bus Timing: One External Wait Cycle



Note : * $\overline{\text{DACKn}}$ is the waveform when active low is selected.

Figure 25.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle

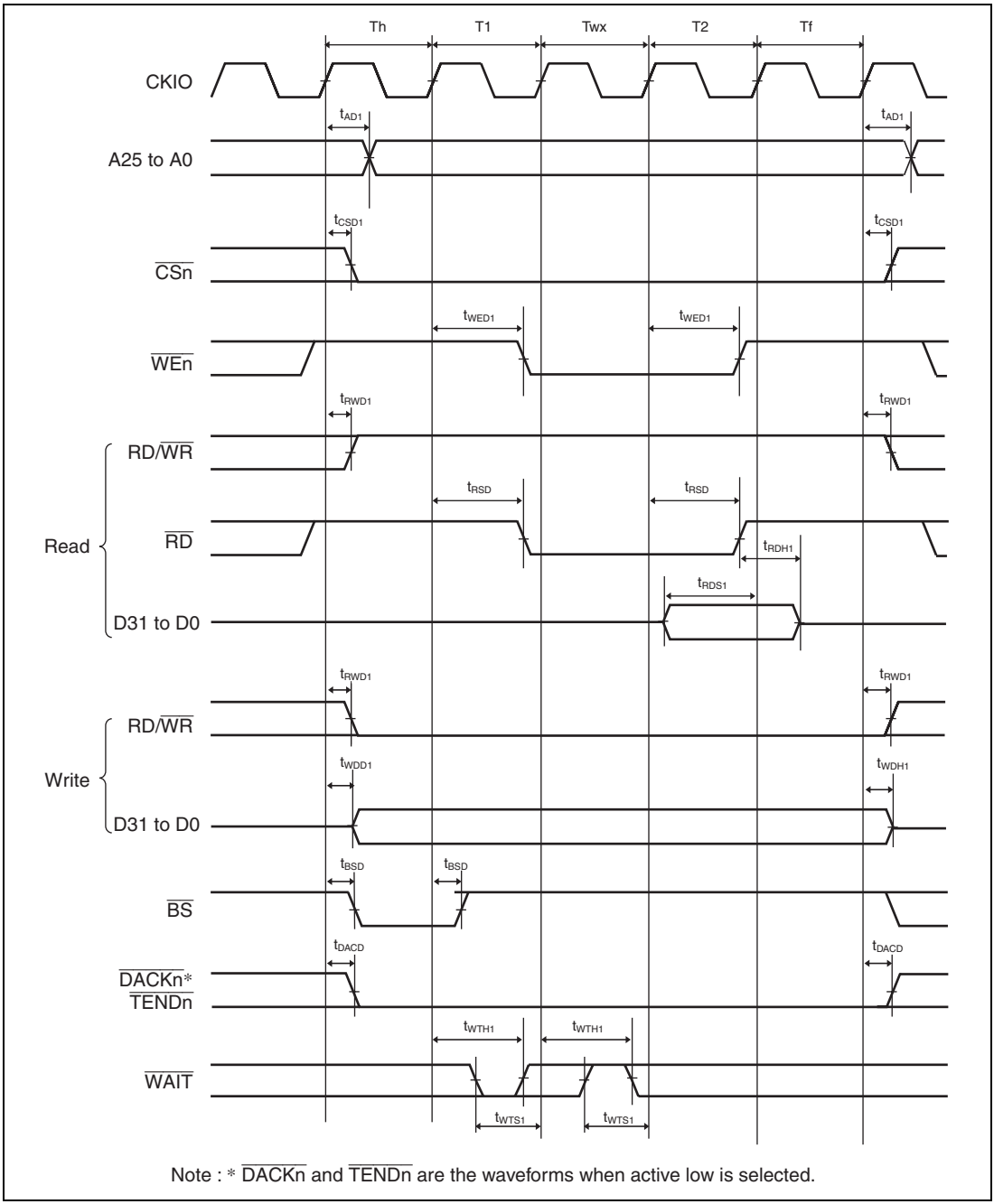


Figure 25.14 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 0 (UB-/LB-Controlled Write Cycle)

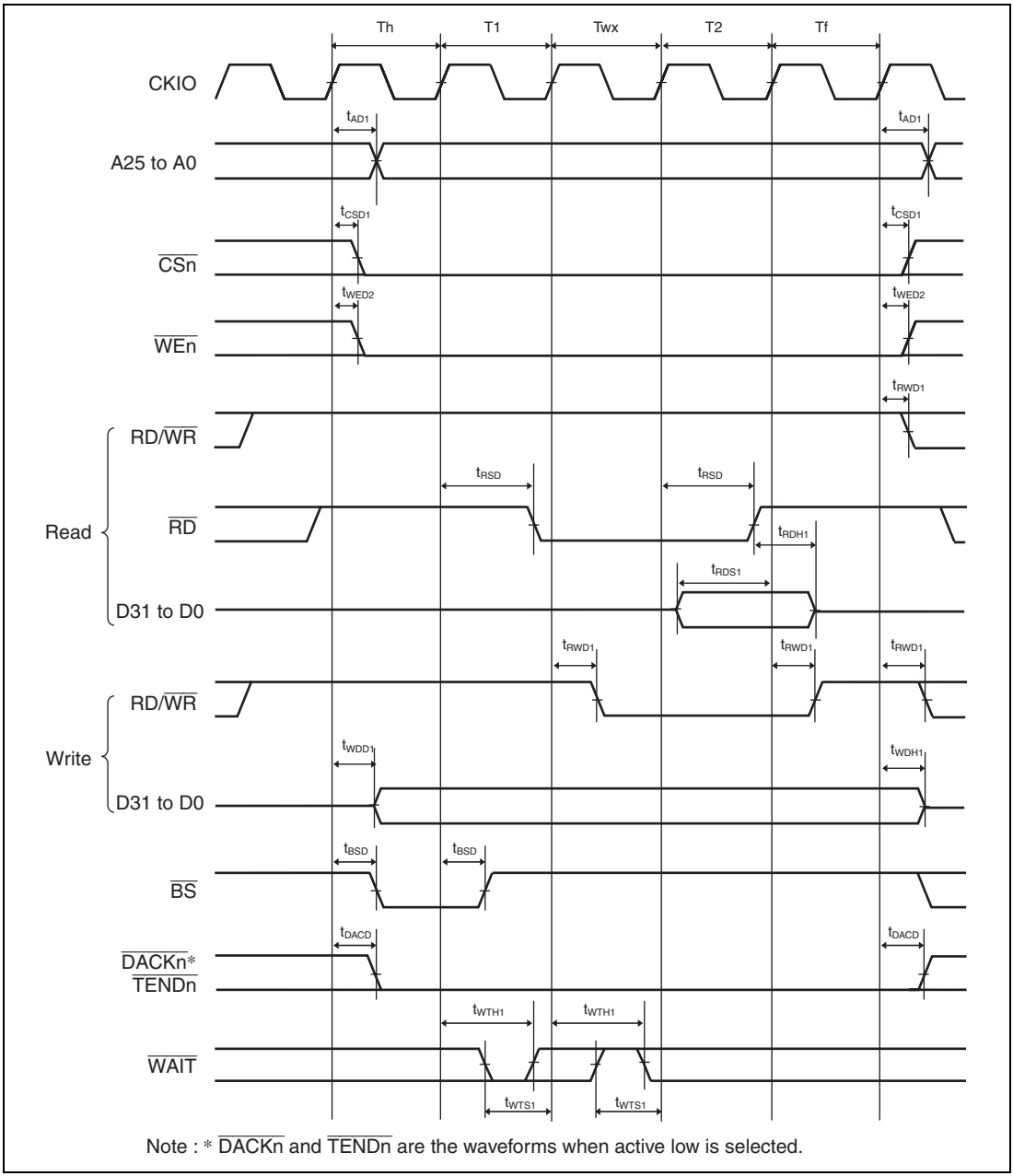
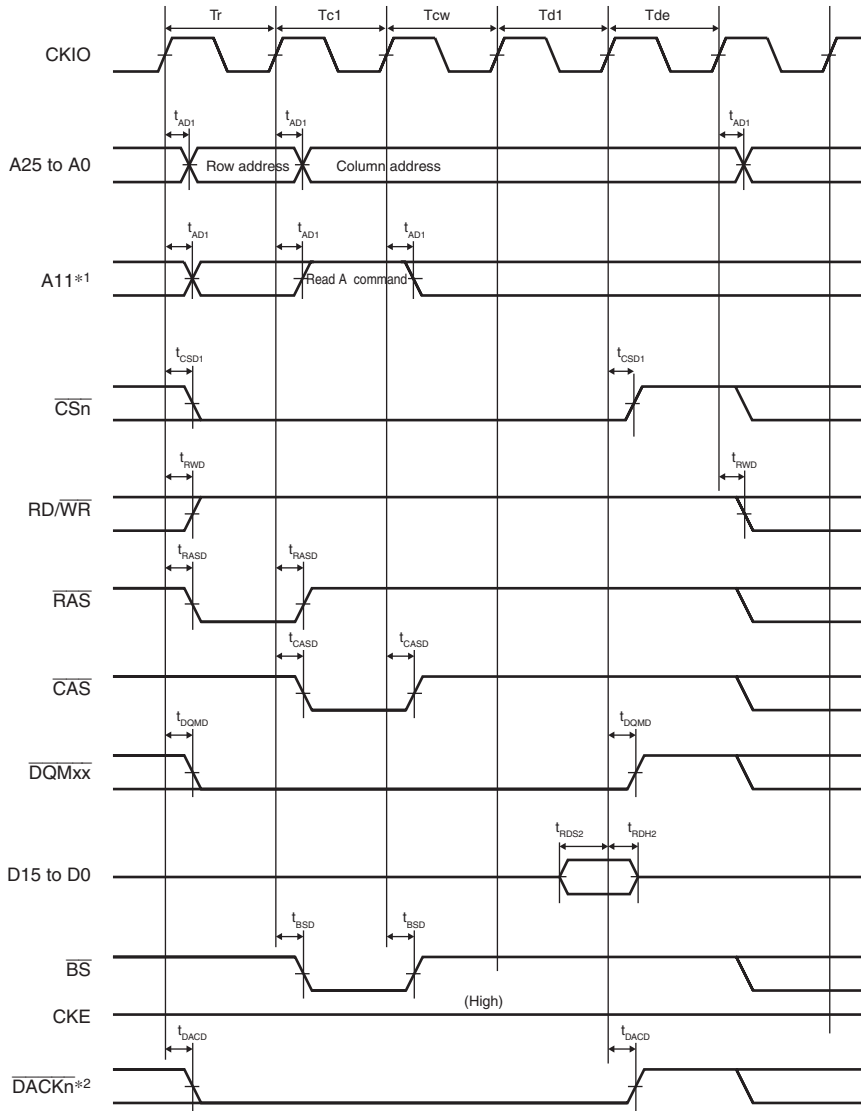


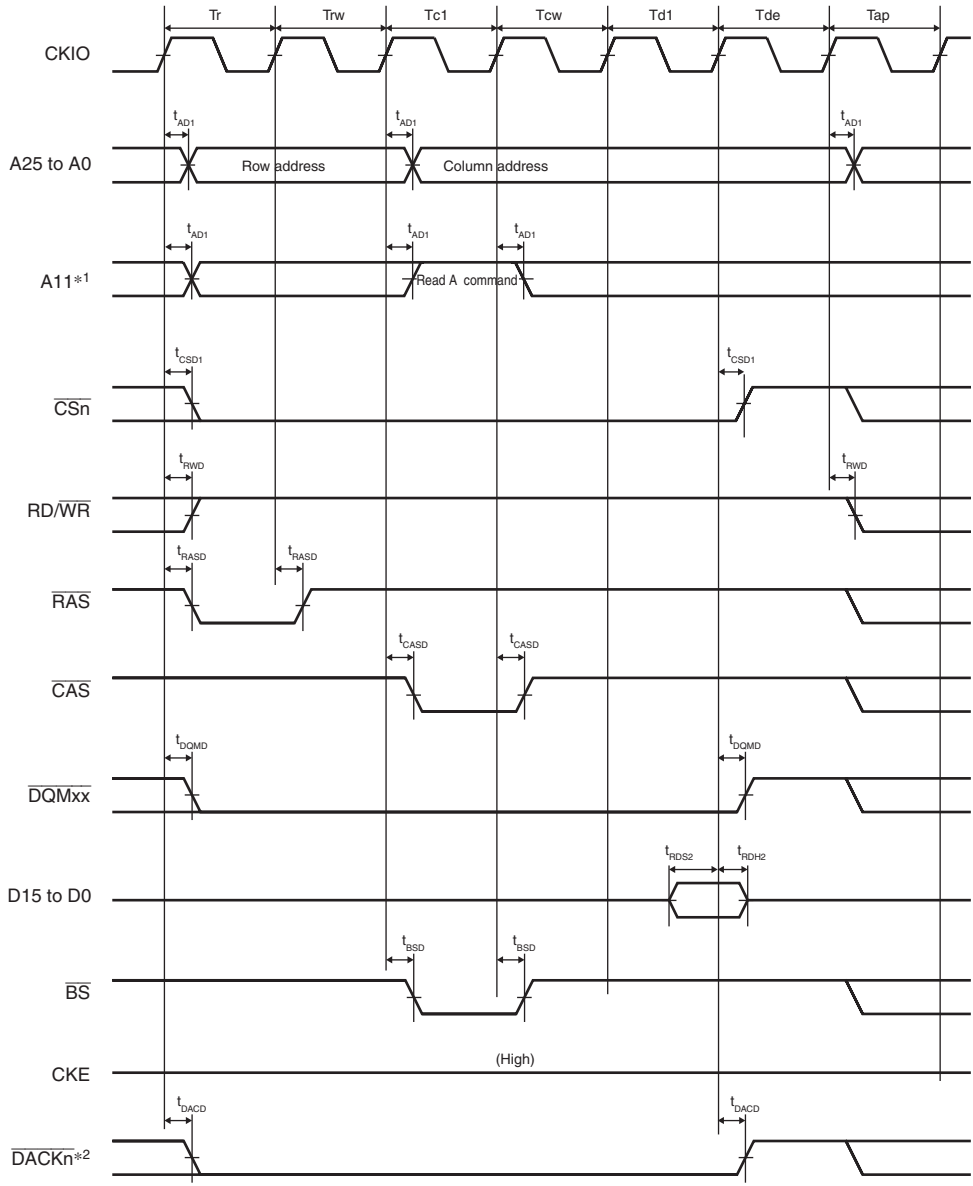
Figure 25.15 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 1 (WE-Controlled Write Cycle)

25.4.5 Synchronous DRAM Timing



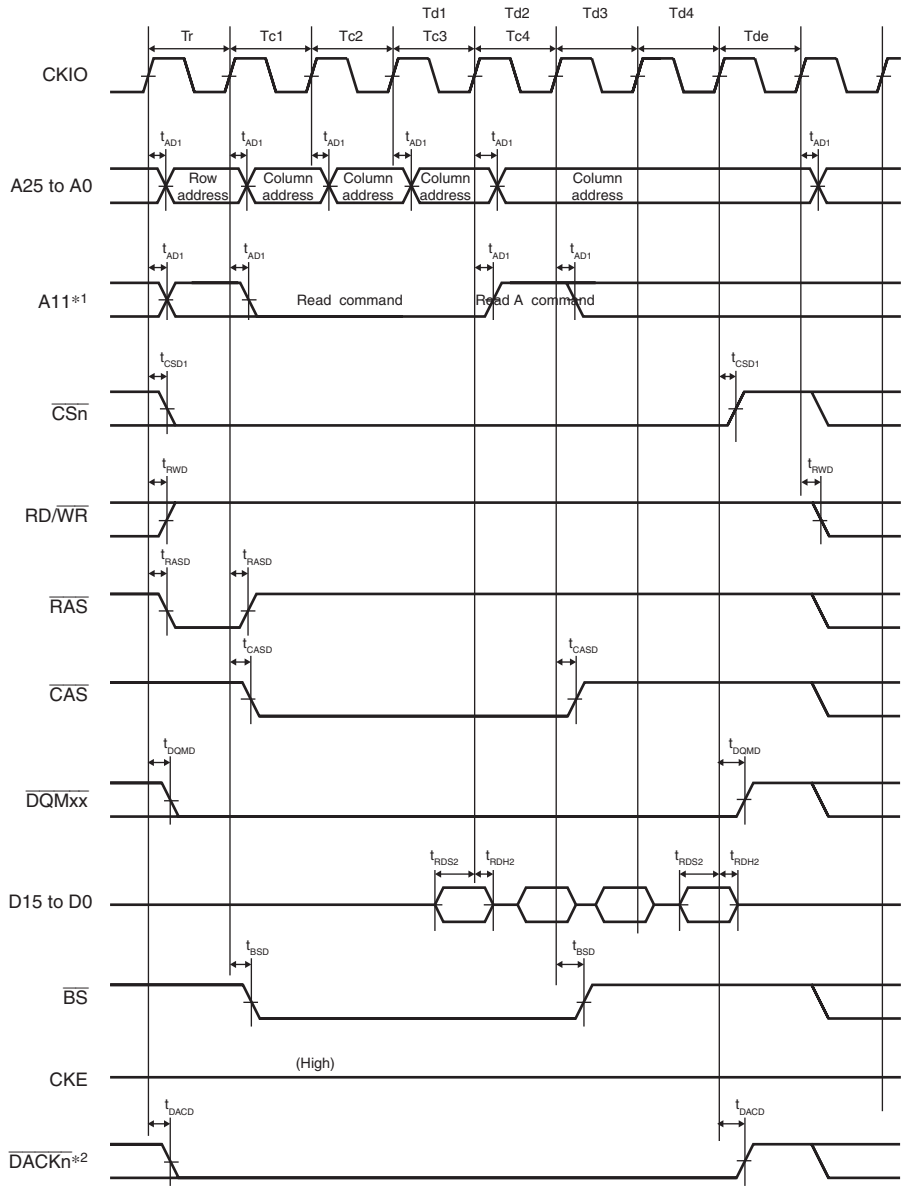
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



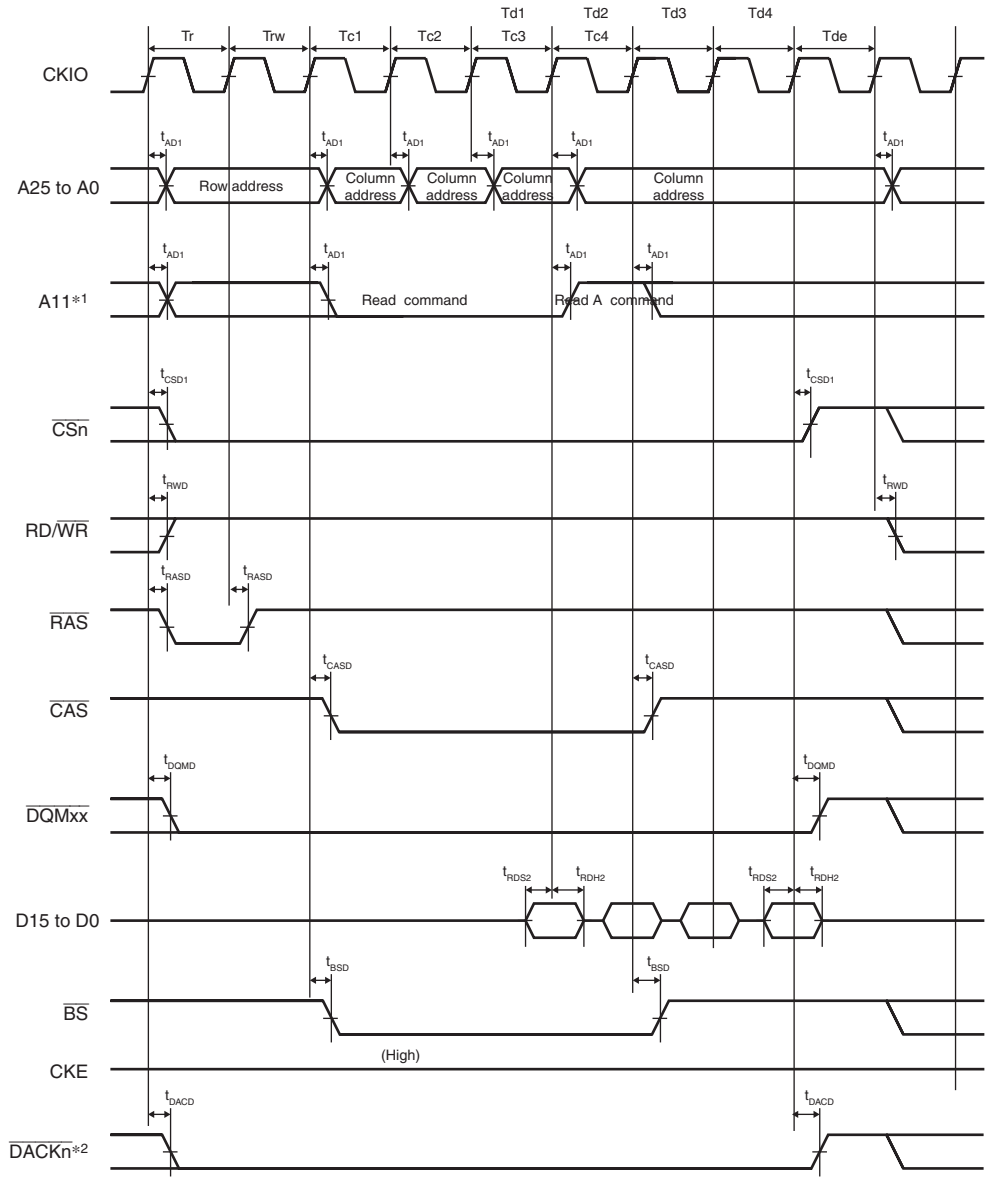
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



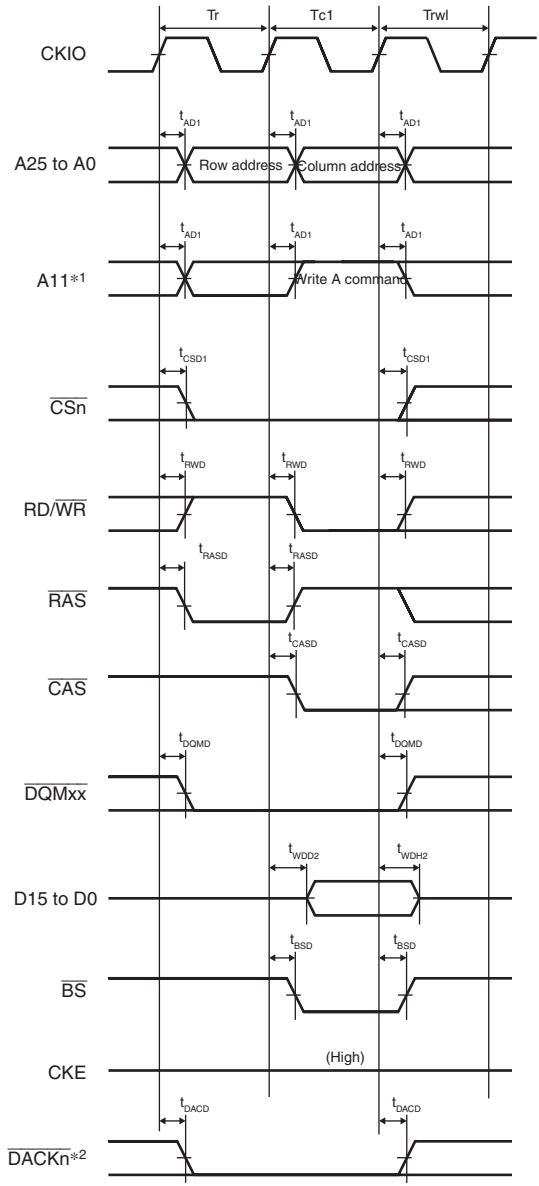
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



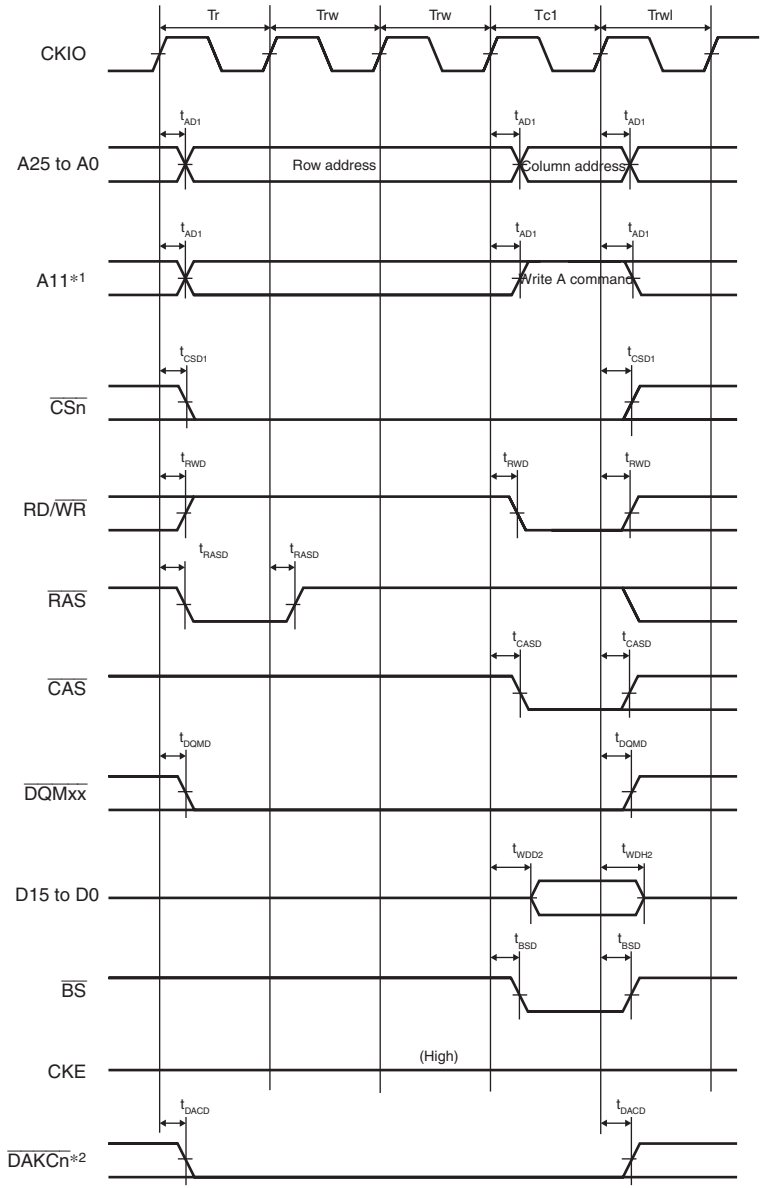
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



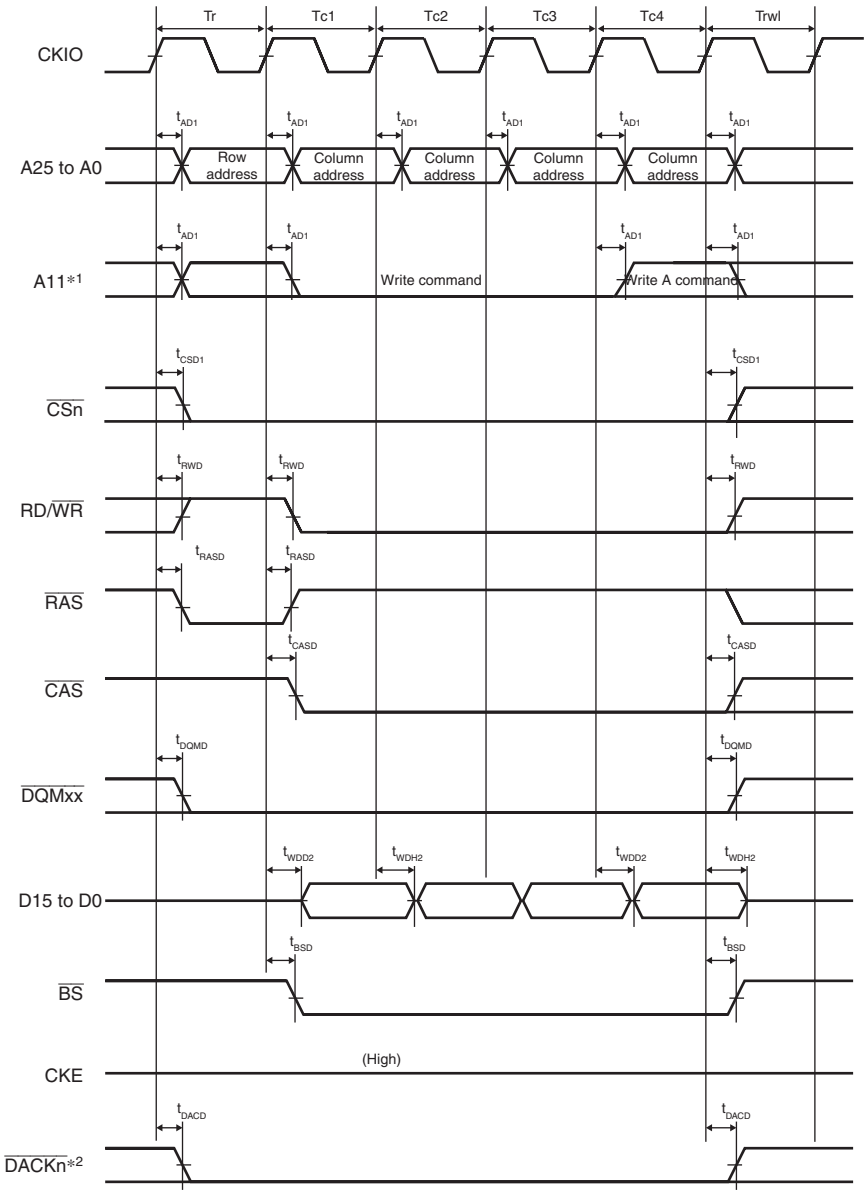
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle)



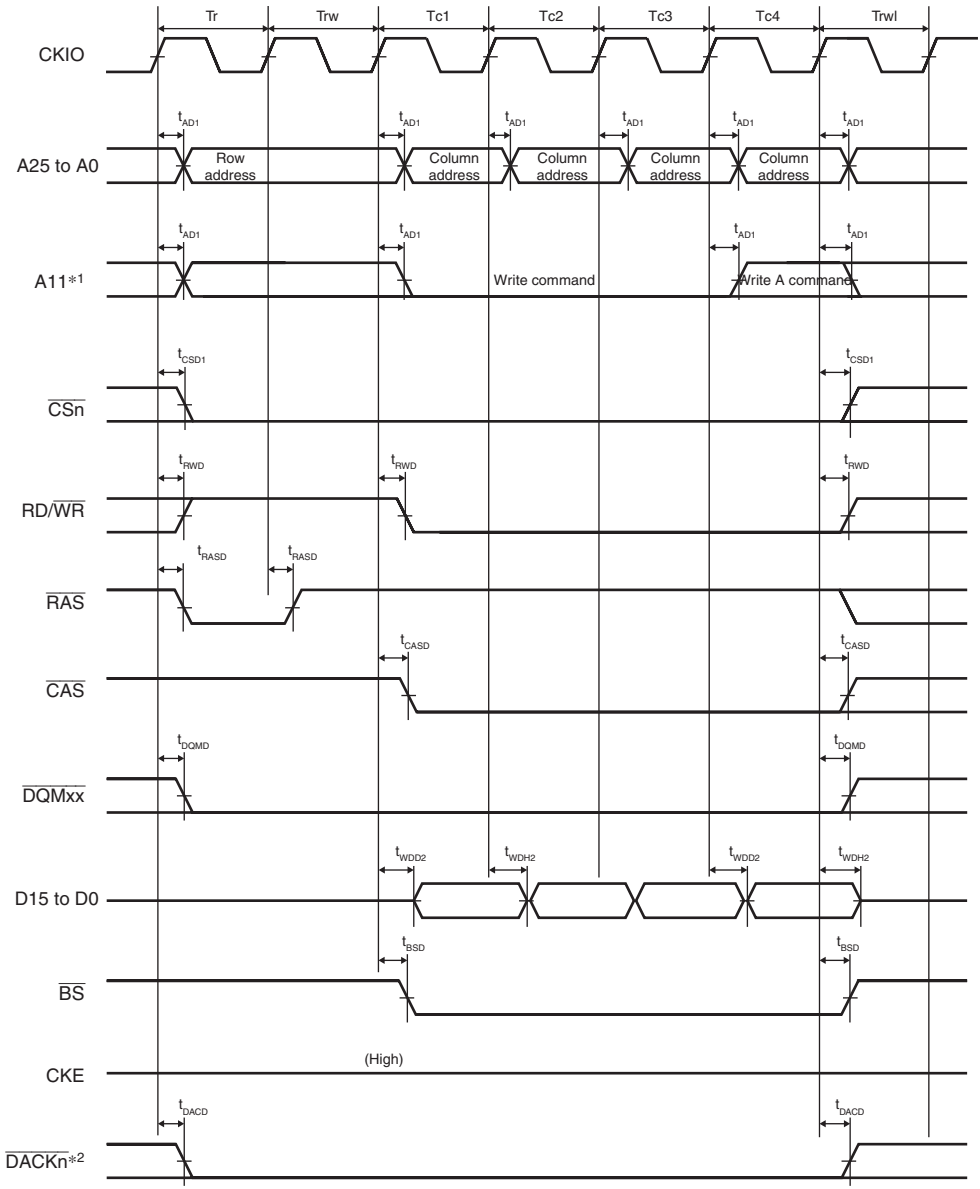
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.21 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



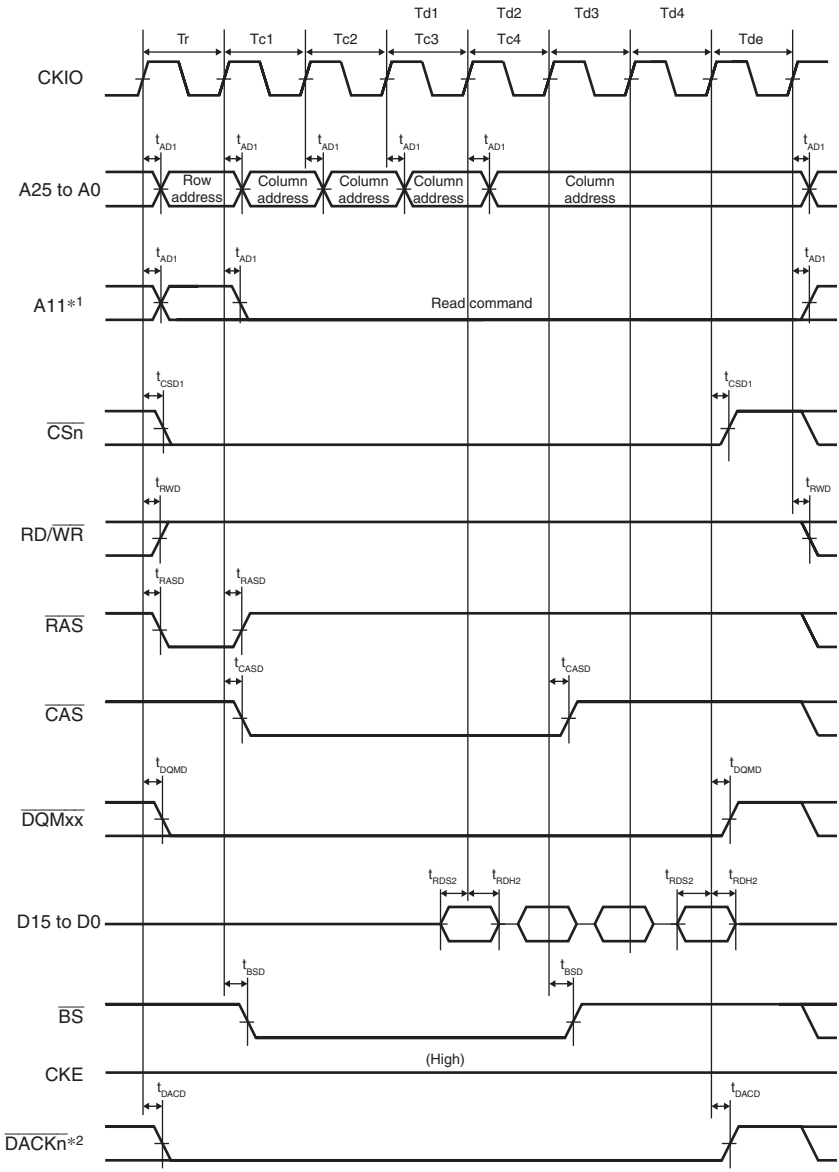
Notes: * 1. Address pins connected to A10 in SDRAM
2. $\overline{\text{DACKn}}$ is the waveform when active low is selected.

Figure 25.22 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



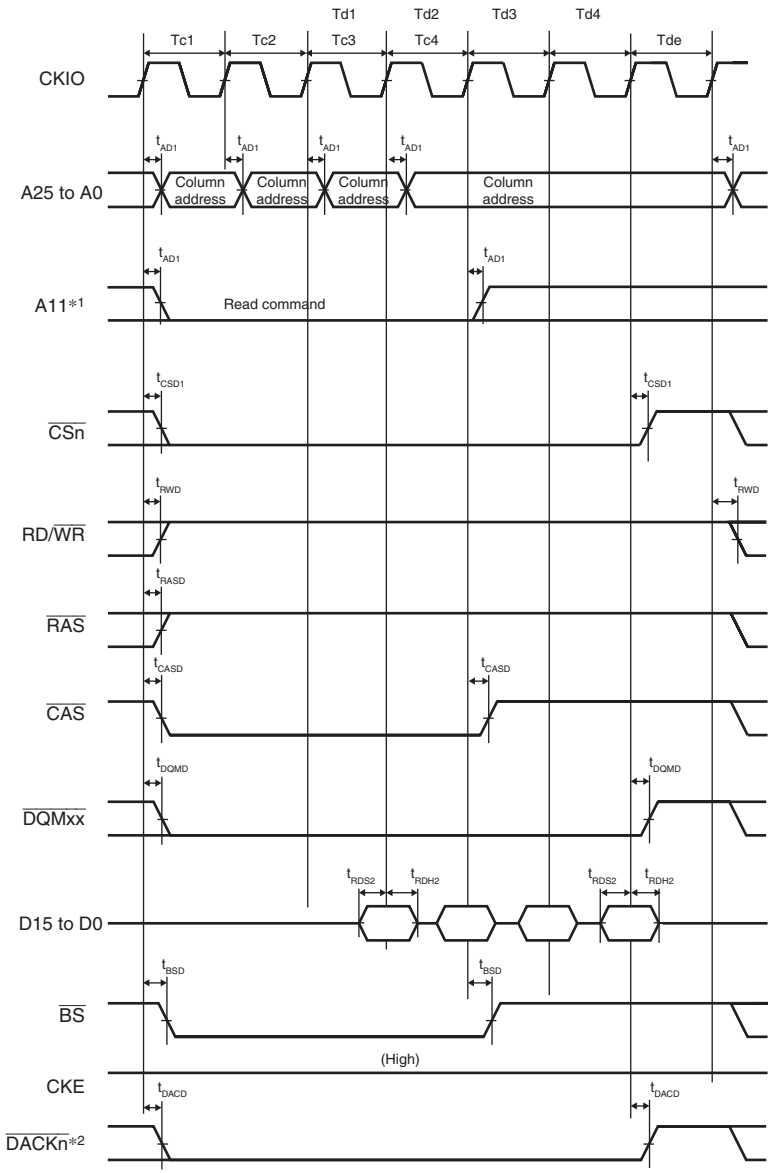
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



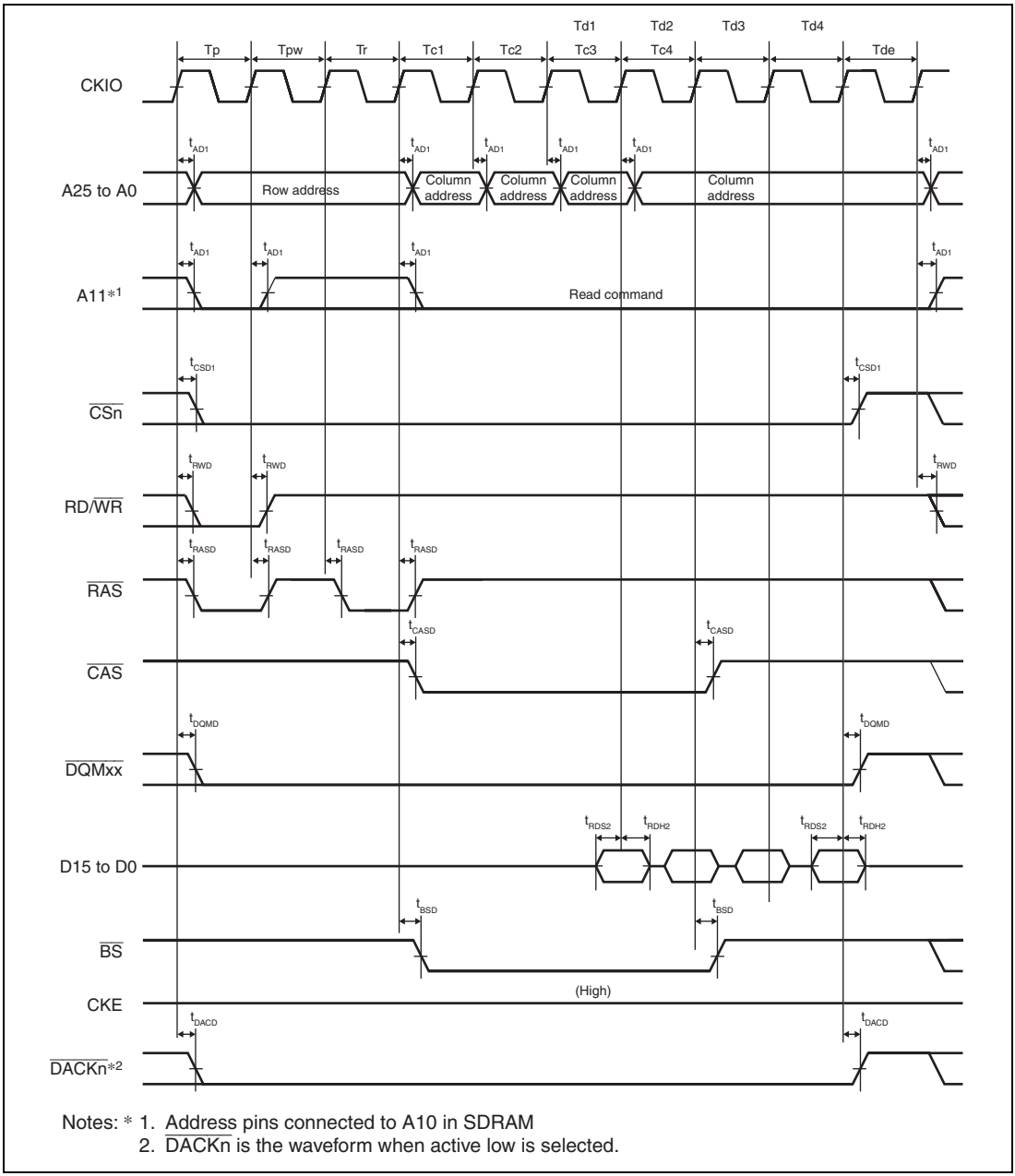
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0 Cycle)



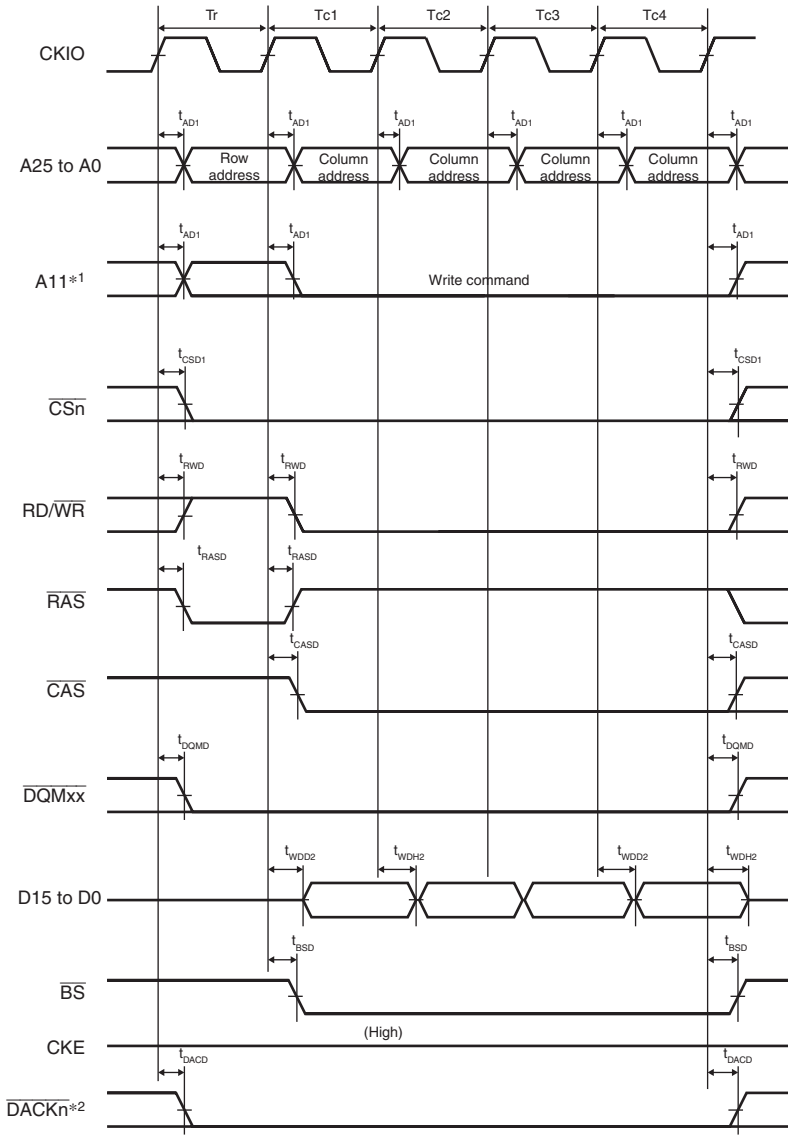
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: READ Command, Same Row Address, CAS Latency = 2, WTRCD = 0 Cycle)



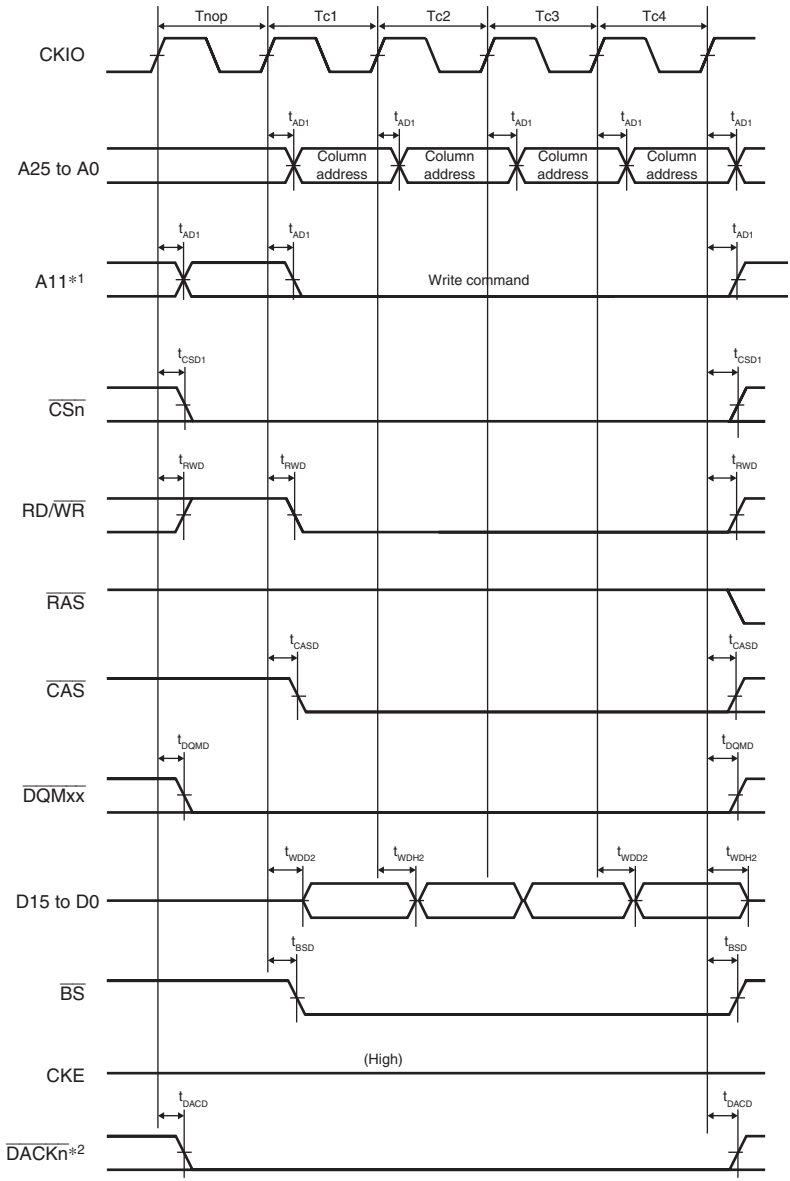
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.26 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency = 2, WTRCD = 0 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.27 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)

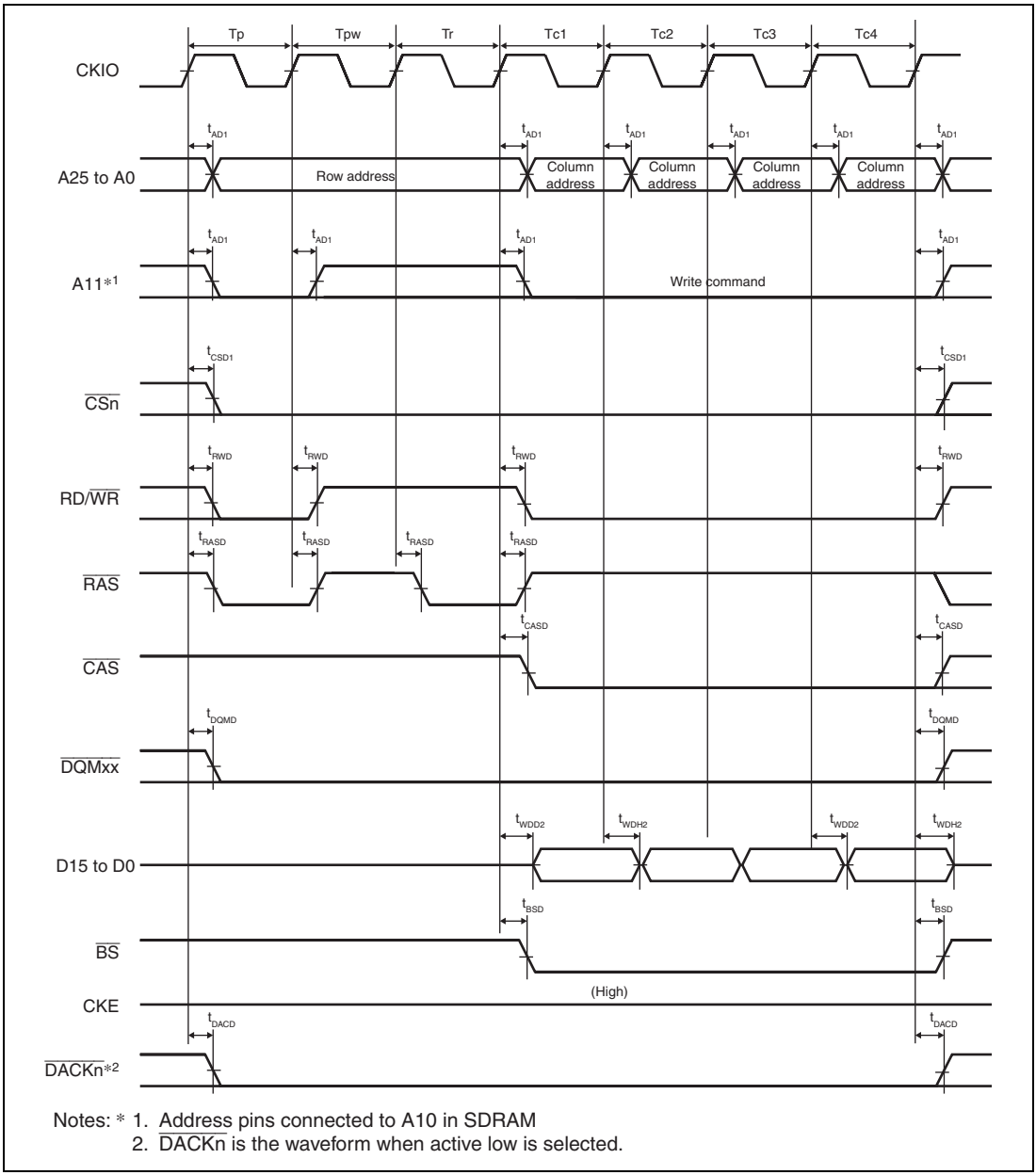
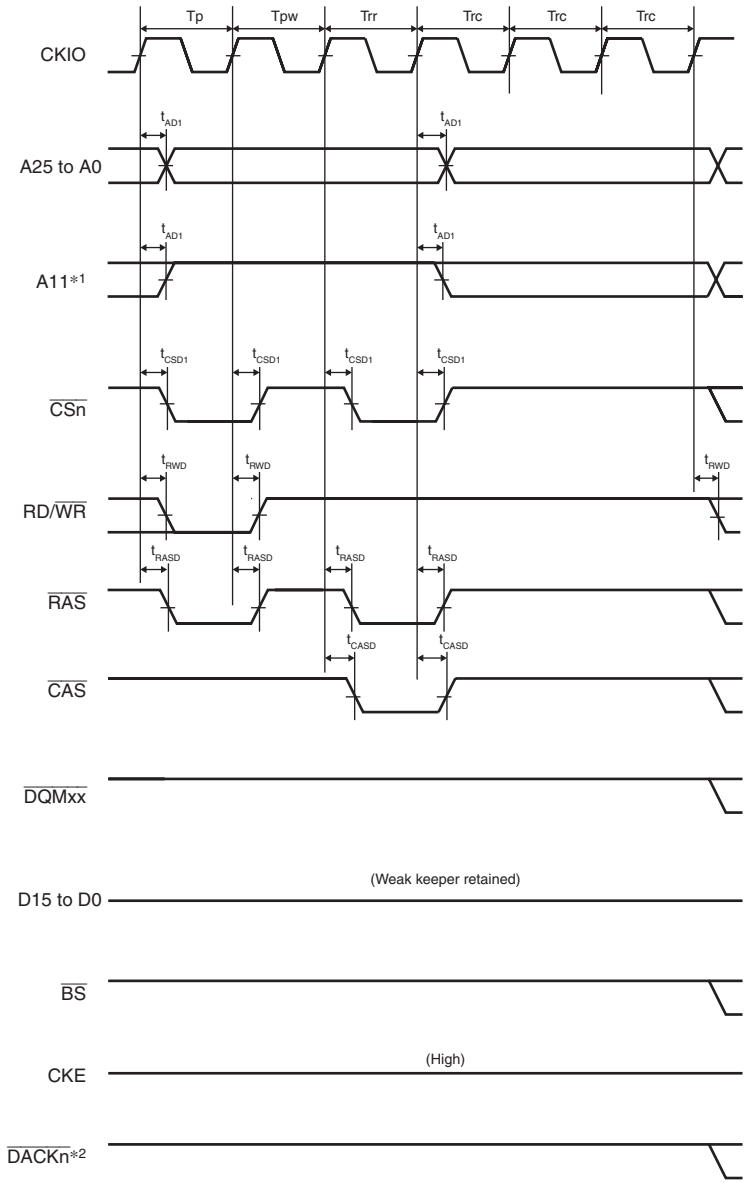
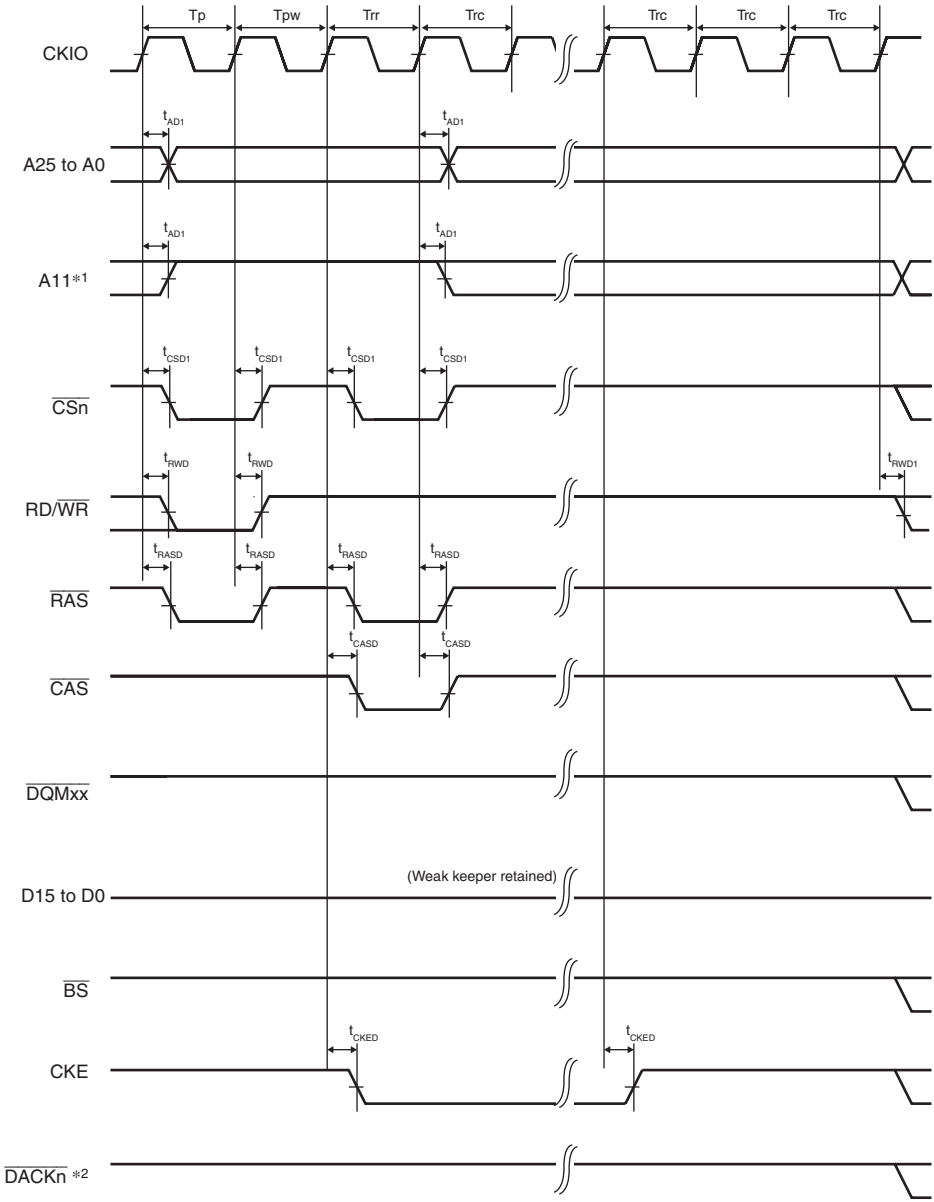


Figure 25.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



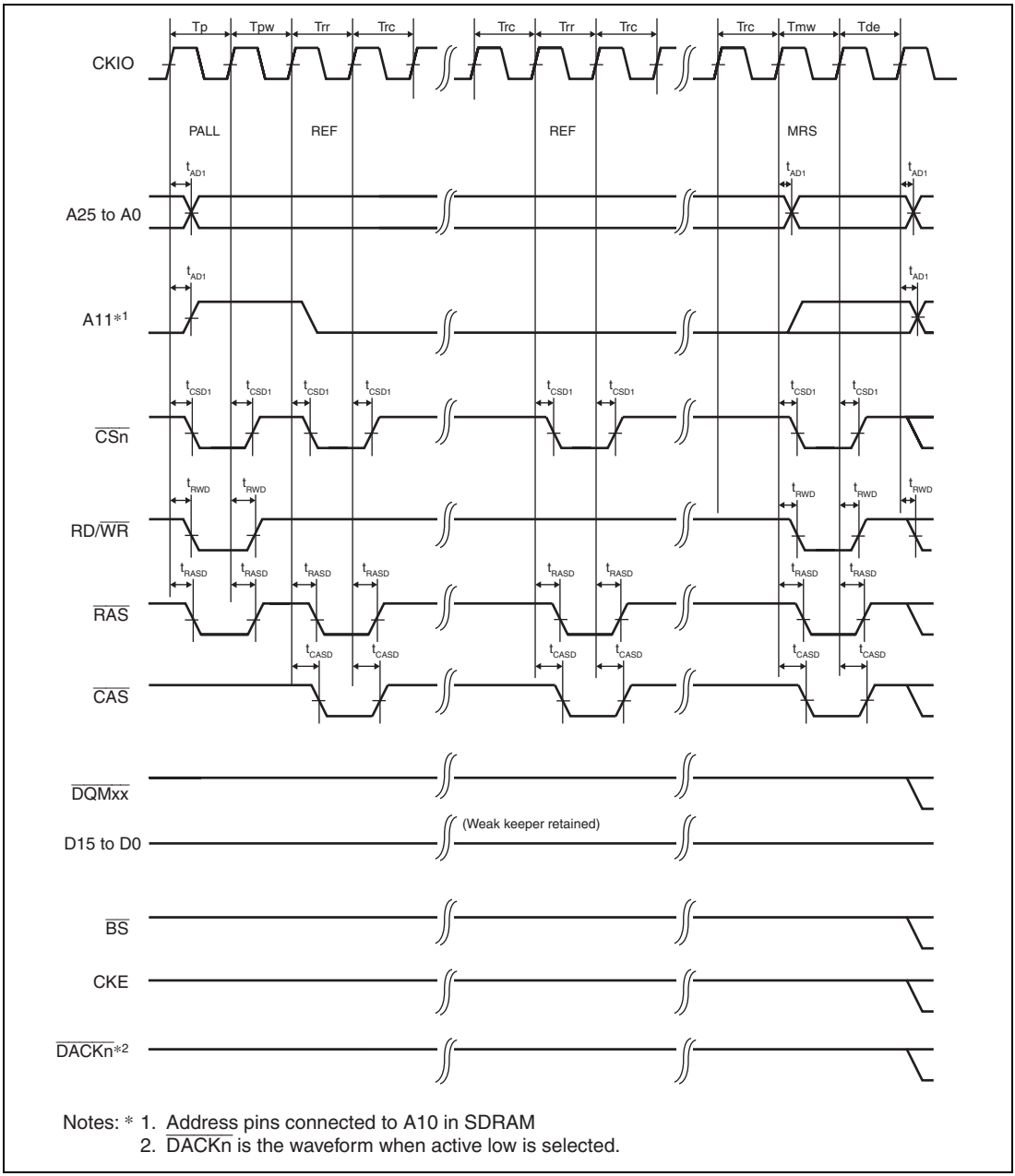
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)

25.4.6 PCMCIA Timing

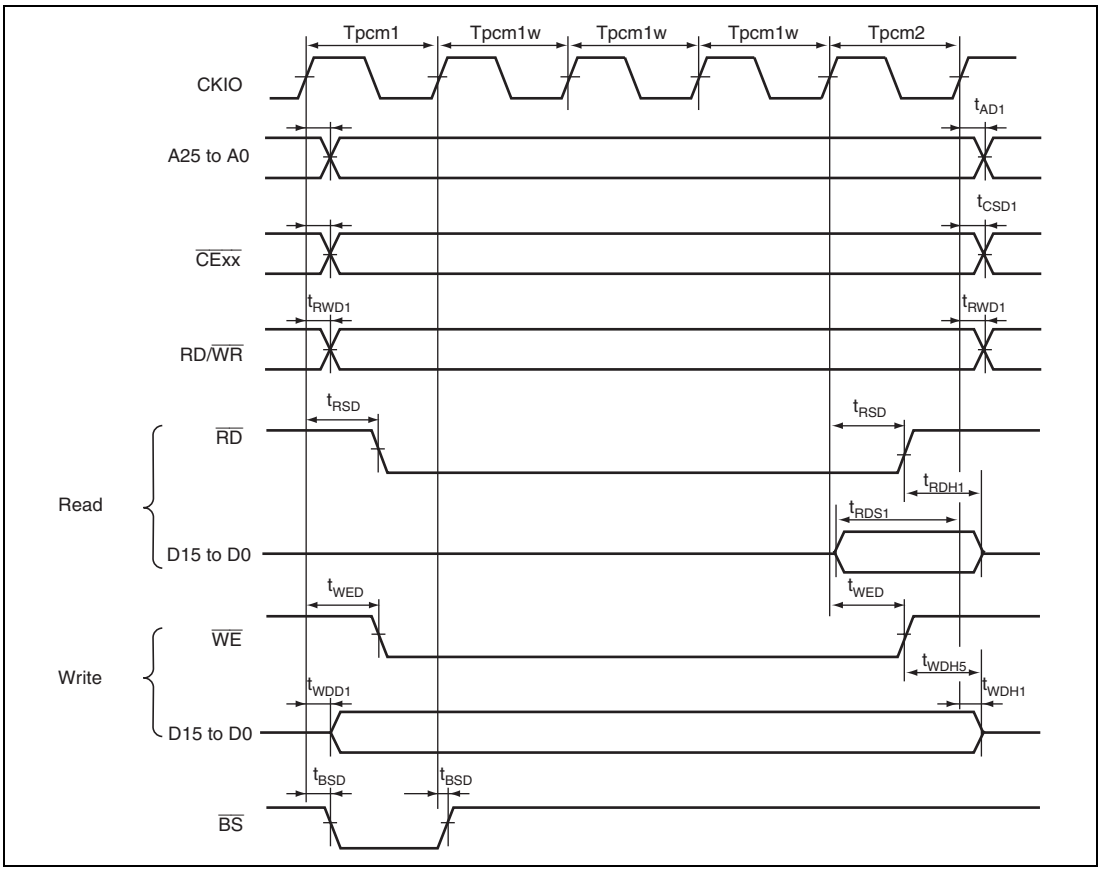


Figure 25.33 PCMCIA Memory Card Interface Bus Timing

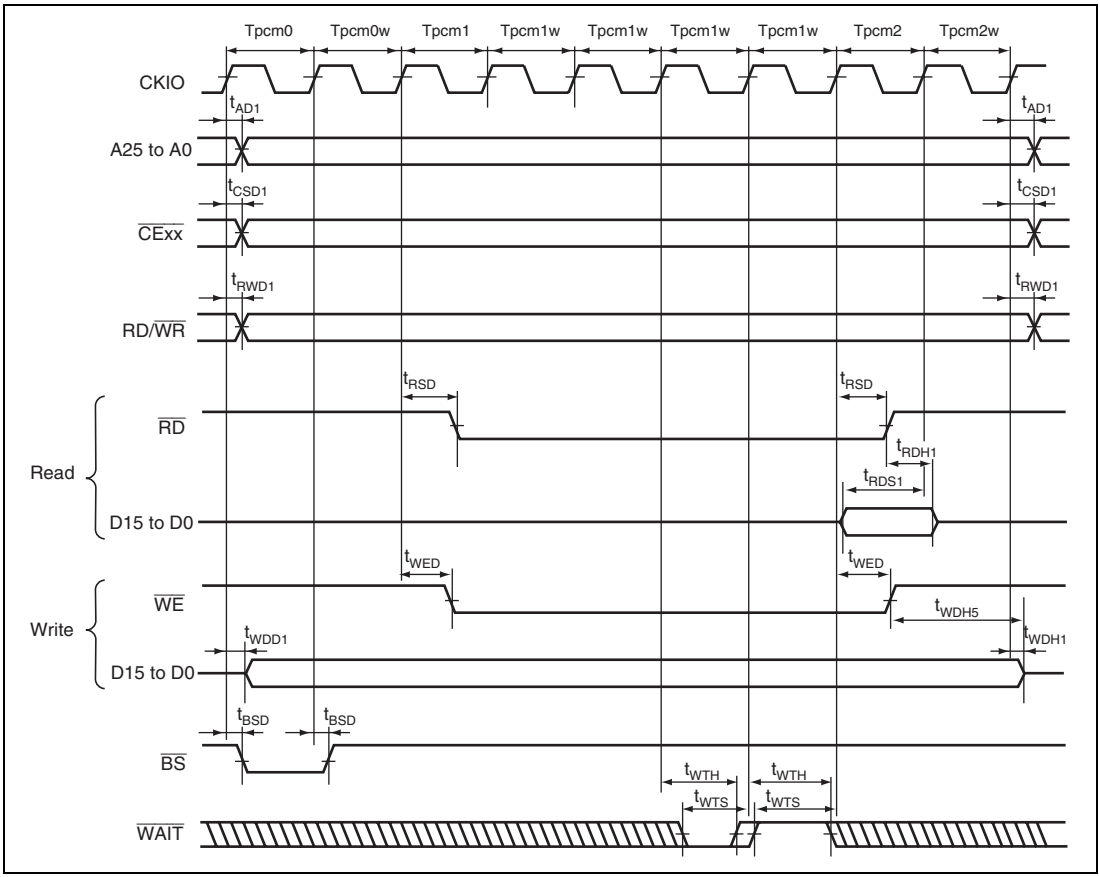


Figure 25.34 PCMCIA Memory Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle)

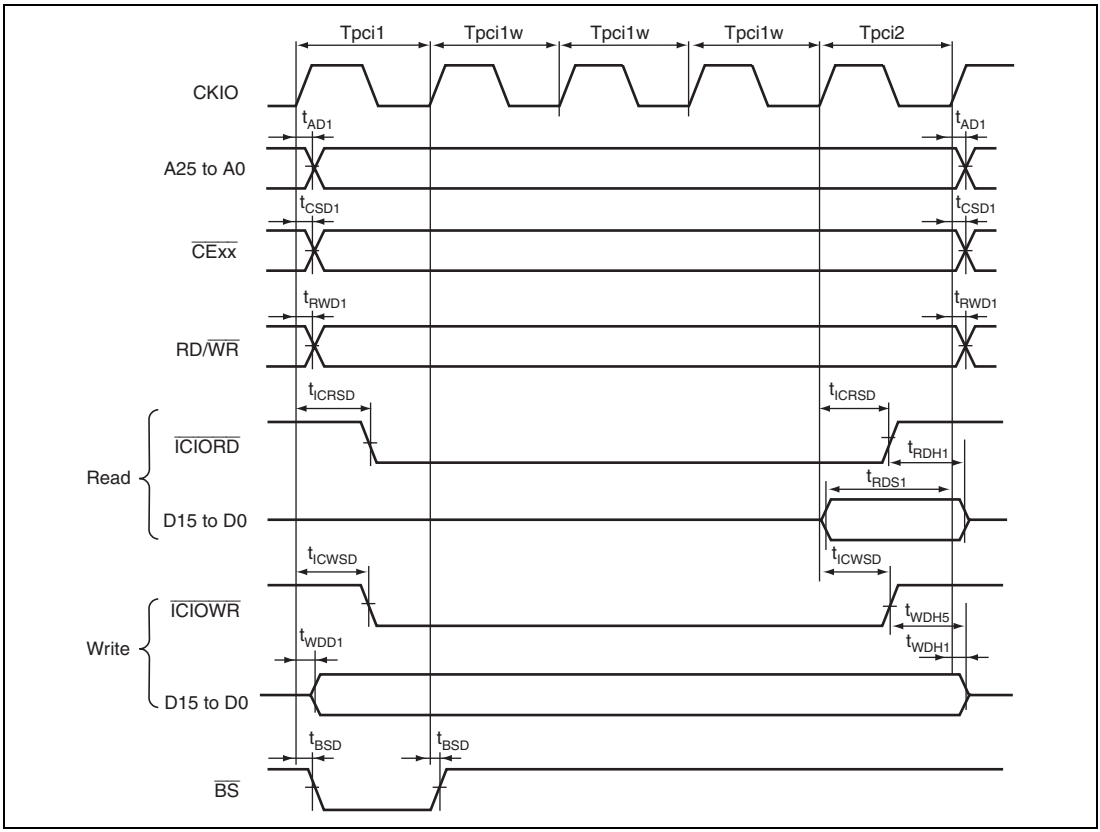


Figure 25.35 PCMCIA I/O Card Interface Bus Timing

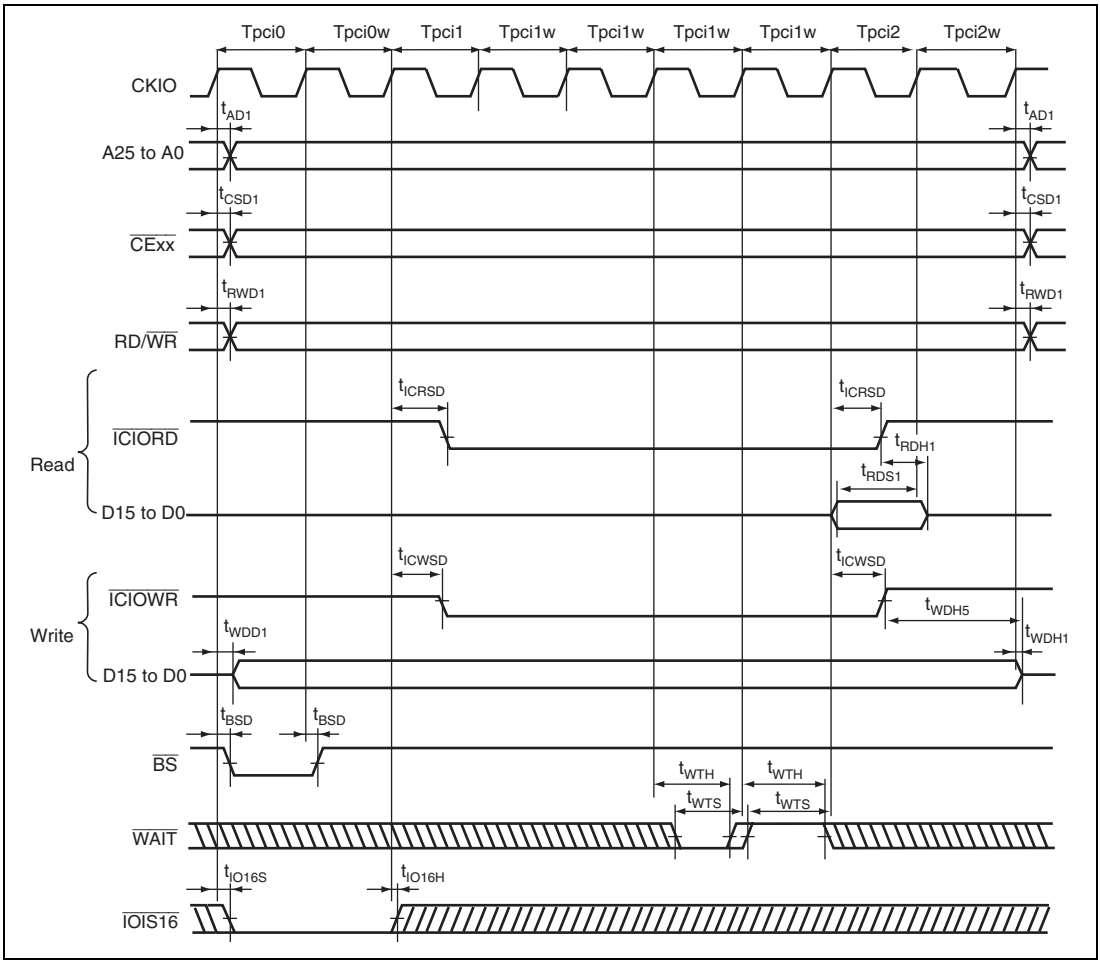


Figure 25.36 PCMCIA I/O Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle)

25.4.7 DMAC Signal Timing

Table 25.10 DMAC Signal Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
DREQn set-up time	t_{DRQS}	10	—	ns	Figure 25.37
DREQn hold time	t_{DRQH}	10	—	ns	Figure 25.37
TENDn, DACKn delay time	t_{DACD}	—	10	ns	Figure 25.38

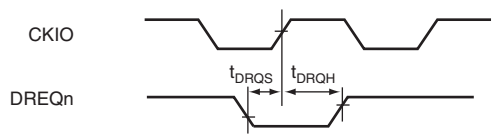


Figure 25.37 DREQ Input Timing

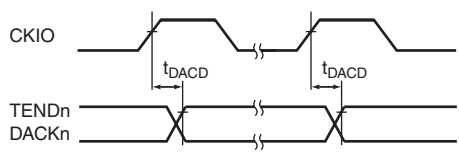


Figure 25.38 TENDn, DACKn Output Timing

25.4.8 SCIF Timing

Table 25.11 SCIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Max.	Unit	Reference Figures
Input clock cycle	Clock synchronous	t_{Scyc}	12	—	t_{pcyc}	Figures 25.39 and 25.40
	Asynchronous		4	—		
Input clock rising time		t_{SCKR}	—	0.8	t_{pcyc}	Figure 25.39
Input clock falling time		t_{SCKF}	—	0.8	t_{pcyc}	
Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time		t_{TXD}	—	$3 \times t_{pcyc}$*+ 50	ns	Figure 25.40
Receive data setup time (clocked synchronous)		t_{RXS}	3	—	t_{pcyc}	
Receive data hold time (clocked synchronous)		t_{RXH}	3	—	t_{pcyc}	
$\overline{RTS}$ delay time		t_{RTSD}	—	100	ns	
$\overline{CTS}$ setup time (clocked synchronous)		t_{CTSS}	100	—	ns	
$\overline{CTS}$ hold time (clocked synchronous)		t_{CTSH}	100	—	ns	

Note: * t_{pcyc} indicates the period of the peripheral module clock (Pϕ).

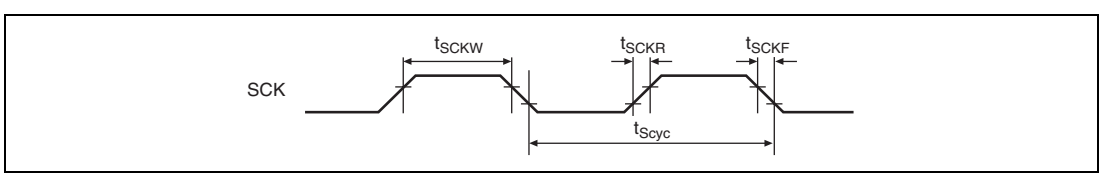


Figure 25.39 SCK Input Clock Timing

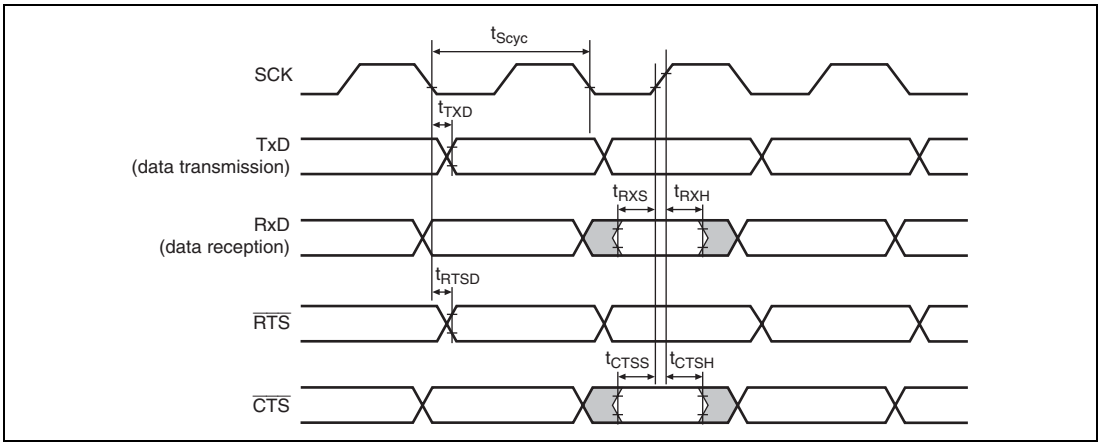


Figure 25.40 SCI Input/Output Timing in Clocked Synchronous Mode

25.4.9 SIOF Module Signal Timing

Table 25.12 SCIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
SIOMCLK clock input cycle time	t_{Mcyc}	32	—	ns	Figure 25.41
SIOMCLK input high pulse width	t_{MWH}	0.4 × t_{Mcyc}	—		
SIOMCLK input low pulse width	t_{MWL}	0.4 × t_{Mcyc}	—		
SCK_SIO clock cycle time	t_{SIcyc}	2 × t_{pcyc}*	—		Figures 25.42 to 25.46
SCK_SIO output high pulse width	t_{SWHO}	0.4 × t_{SIcyc}	—		Figures 25.42 to 25.45
SCK_SIO output low pulse width	t_{SWLO}	0.4 × t_{SIcyc}	—		
SIOFSYNC output delay time	t_{FSD}	—	20		
SCK_SIO input high pulse width	t_{SWHI}	0.4 × t_{SIcyc}	—		Figure 25.46
SCK_SIO input low pulse width	t_{SWLI}	0.4 × t_{SIcyc}	—		
SIOFSYNC input set-up time	t_{FSS}	20	—		
SIOFSYNC input hold time	t_{FSH}	20	—		

Item	Symbol	Min.	Max.	Unit	Reference Figures
TXD_SIO output delay time	t_{STDD}	—	20	ns	Figures 25.42 to 25.46
RXD_SIO input set-up time	t_{SRDS}	20	—		
RXD_SIO input hold time	t_{SRDH}	20	—		

Note: * t_{pcyc} indicates the period of the peripheral module clock (Pϕ).

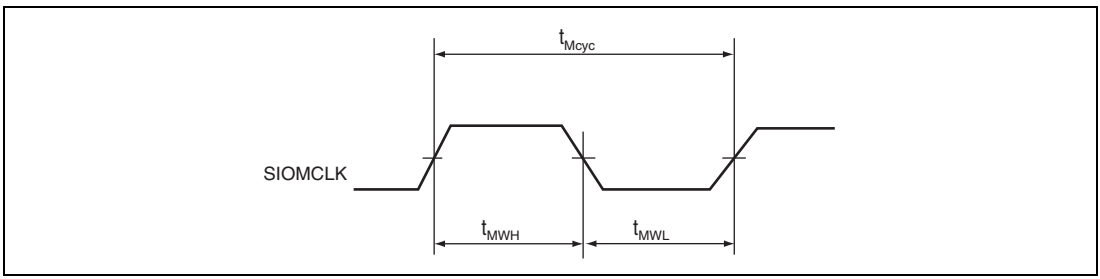


Figure 25.41 SIOMCLK Input Timing

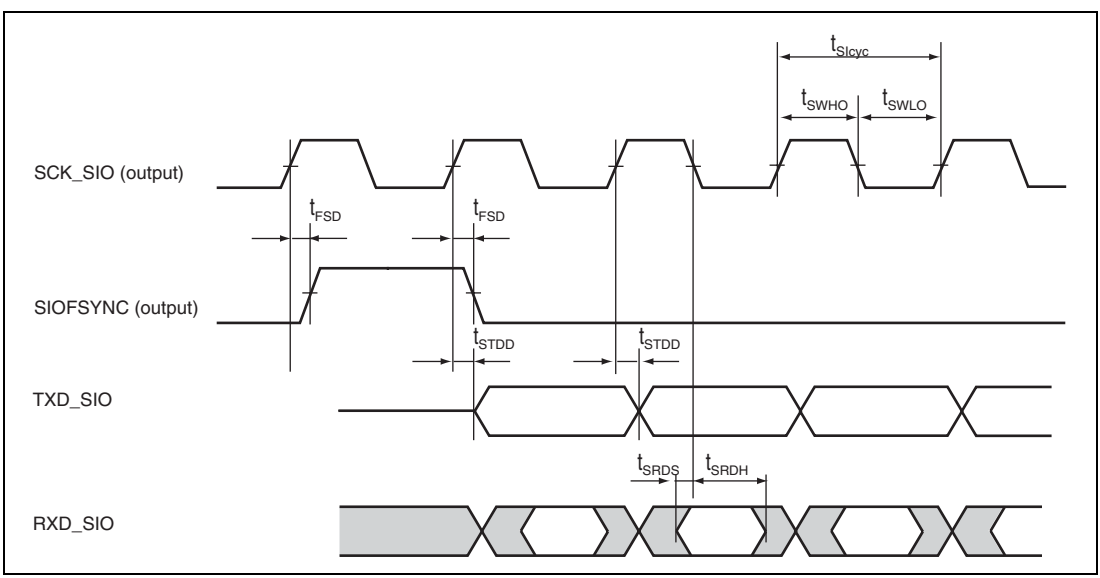


Figure 25.42 SIOF Transmit/Receive Timing (Master Mode 1/Falling Edge Sampling)

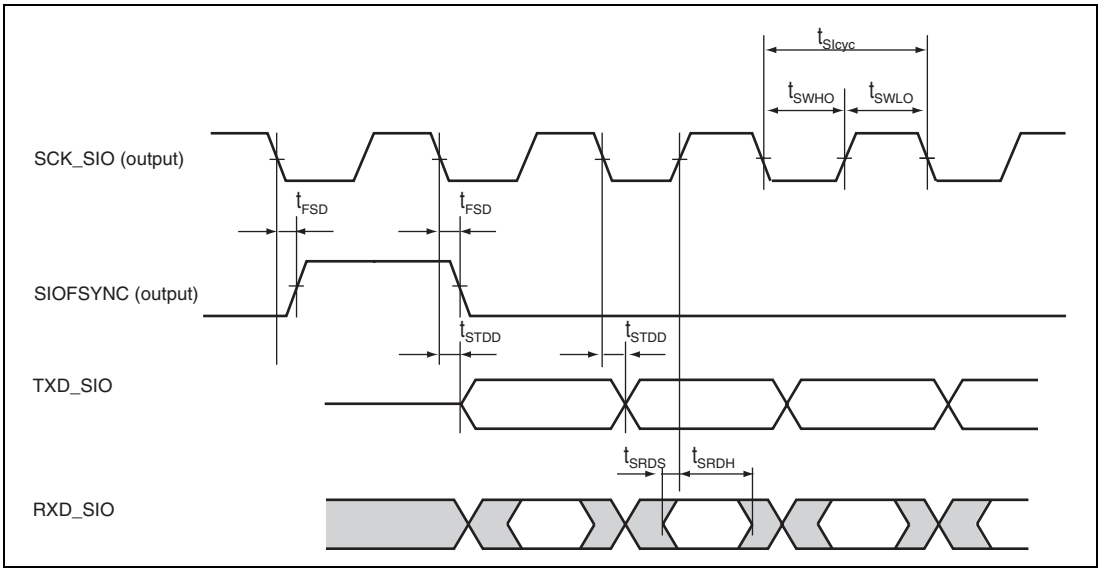


Figure 25.43 SIOF Transmit/Receive Timing (Master Mode 1/Rising Edge Sampling)

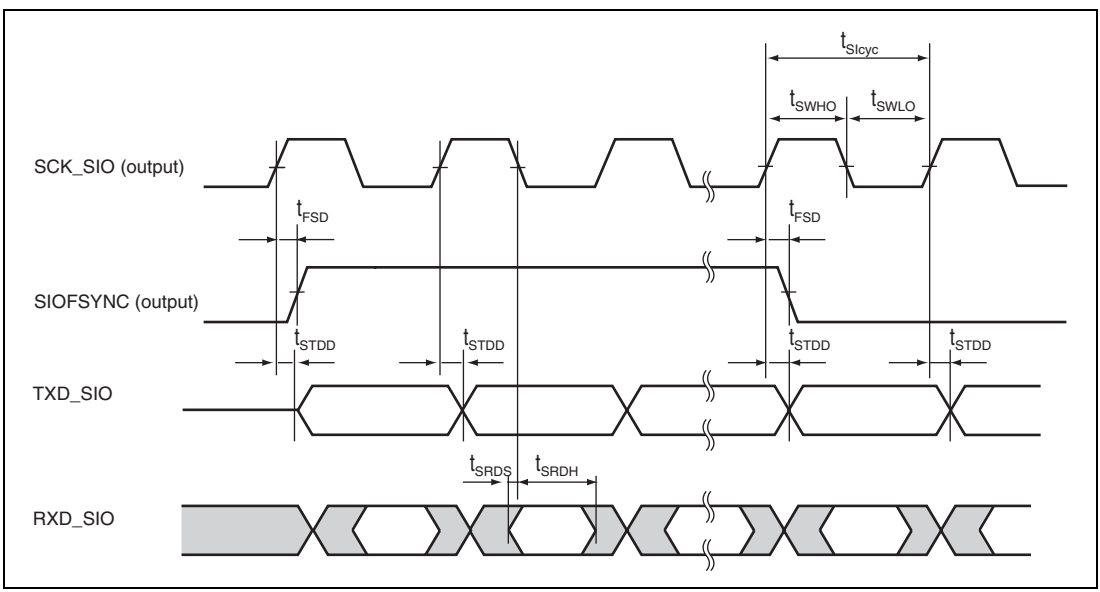


Figure 25.44 SIOF Transmit/Receive Timing (Master Mode 2/Falling Edge Sampling)

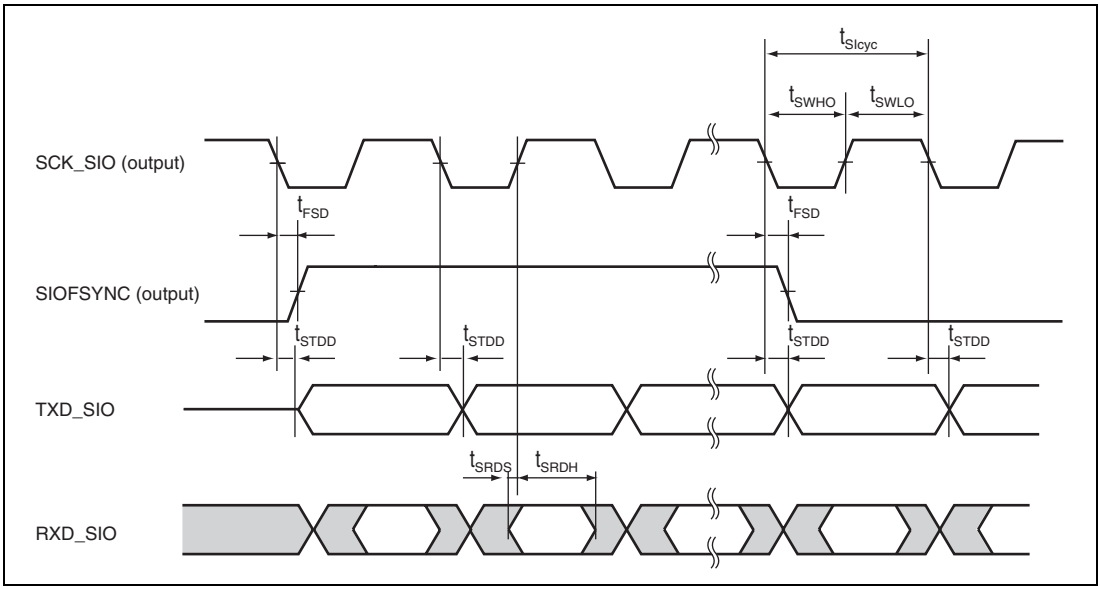


Figure 25.45 SIOF Transmit/Receive Timing (Master Mode 2/Rising Edge Sampling)

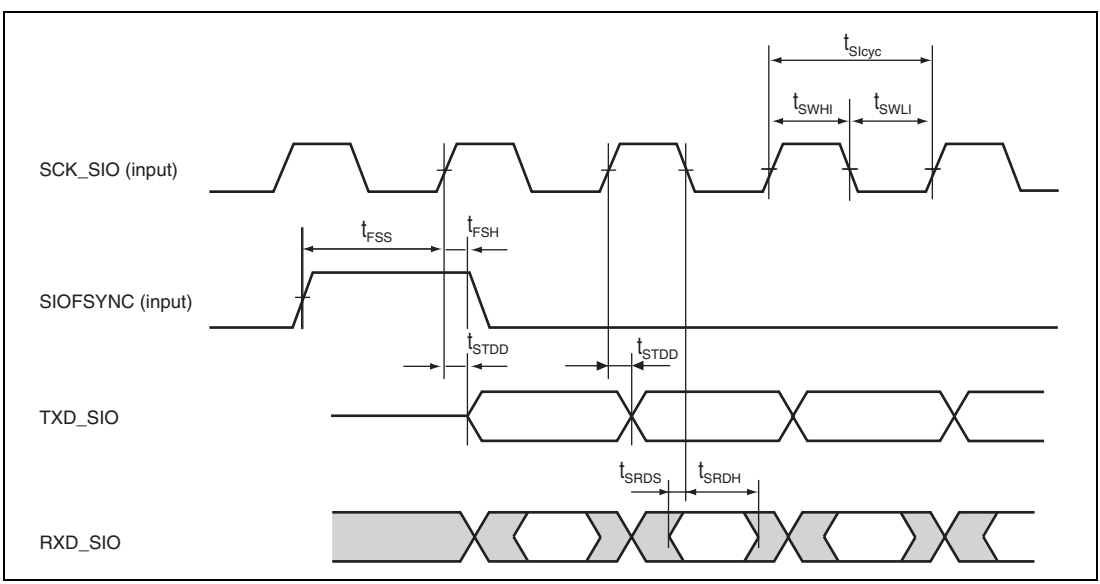


Figure 25.46 SIOF Transmit/Receive Timing (Slave Mode 1/ Slave Mode 2)

25.4.10 Port Timing

Table 25.13 Port Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
Output data delay time	t_{PORTD}	—	20	ns	Figure 25.47
Input data setup time	t_{PORTS}	16	—	ns	
Input data hold time	t_{PORTH}	10	—	ns	

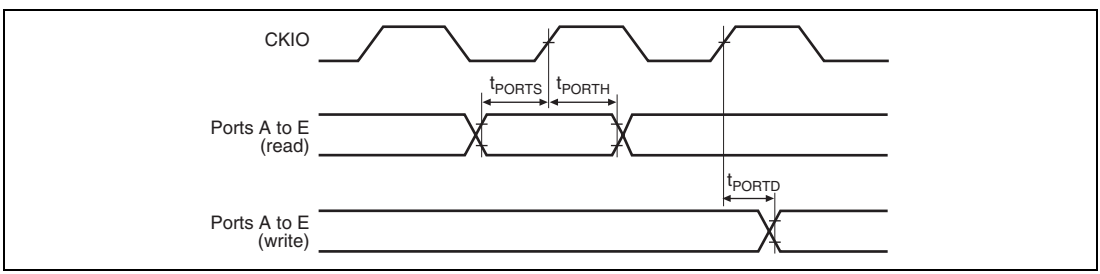


Figure 25.47 I/O Port Timing

25.4.11 HIF Timing

Table 25.14 HIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
Read bus cycle time	$t_{HIFCYCR}$	4	—	t_{pcyc}	Figure 25.48
Write bus cycle time	$t_{HIFCYCW}$	4	—	t_{pcyc}	
Address setup time (HIFSCR.DMD = 0)	t_{HIFAS}	16	—	ns	
Address setup time (HIFSCR.DMD = 1)	t_{HIFAS}	0	—	ns	
Address hold time (HIFSCR.DMD = 0)	t_{HIFAH}	16	—	ns	
Address hold time (HIFSCR.DMD = 1)	t_{HIFAH}	0	—	ns	
Read low width (read)	t_{HIFWRL}	2.5	—	t_{pcyc}	
Write low width (write)	t_{HIFWWL}	2.5	—	t_{pcyc}	
Read/write high width	$t_{HIFWRWH}$	2.0	—	t_{pcyc}	
Read data delay time	t_{HIFRDD}	—	$2 \times t_{pcyc} + 16$	ns	
Read data hold time	t_{HIFRDH}	0	—	ns	
Write data setup time	t_{HIFWDS}	$t_{pcyc} + 10$	—	ns	
Write data hold time	t_{HIFWDH}	10	—	ns	
$\overline{\text{HIFINT}}$ output delay time	t_{HIFITD}	—	20	ns	Figure 25.49
HIFRDY output delay time	t_{HIFRYD}	—	10	t_{pcyc}	Figure 25.50
HIFDREQ output delay time	t_{HIFDQD}	—	20	ns	Figure 25.49
HIF pin enable delay time	t_{HIFEBD}	—	20	ns	Figure 25.50
HIF pin disable delay time	t_{HIFDBD}	—	20	ns	

Notes:
1. t_{pcyc} indicates the period of the peripheral module clock (Pϕ).
2. t_{HIFAS} is given from the start of the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ (or $\overline{\text{HIFWR}}$) signals are low levels.
3. t_{HIFAH} is given from the end of the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ (or $\overline{\text{HIFWR}}$) signals are low levels.
4. t_{HIFWRL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ signals are low levels.
5. t_{HIFWWL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFWR}}$ signals are low levels.
6. When reading the register specified by bits REG5 to REG0 after writing to the HIF index register (HIFIDX), $t_{HIFWRWH}$ (min.) = $2 \times t_{pcyc} + 5$ ns.

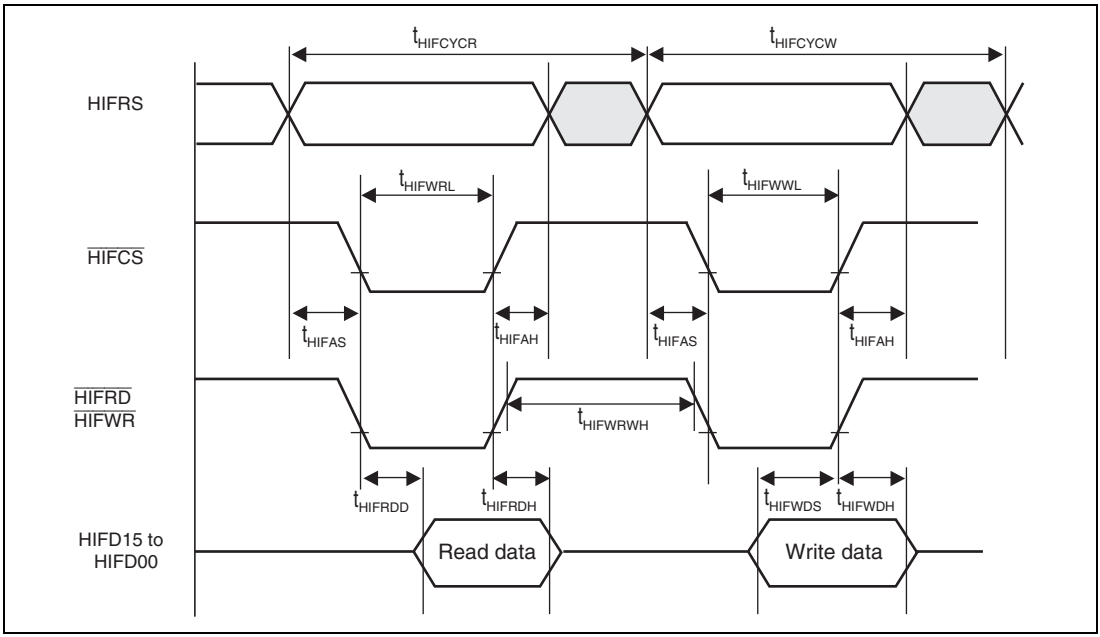


Figure 25.48 HIF Access Timing

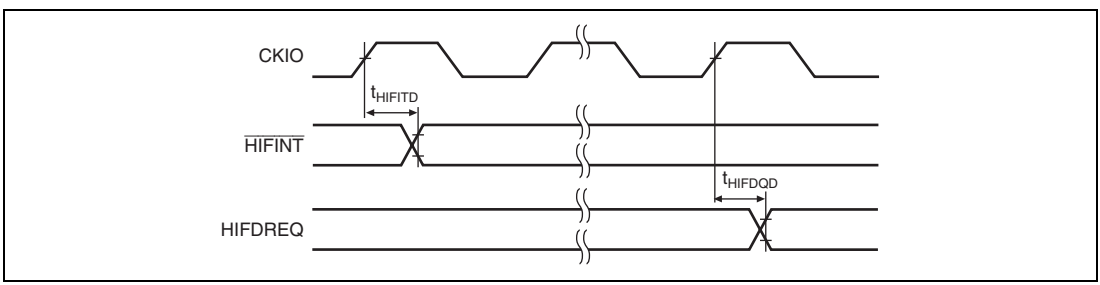


Figure 25.49 $\overline{\text{HIFINT}}$ and HIFDREQ Timing

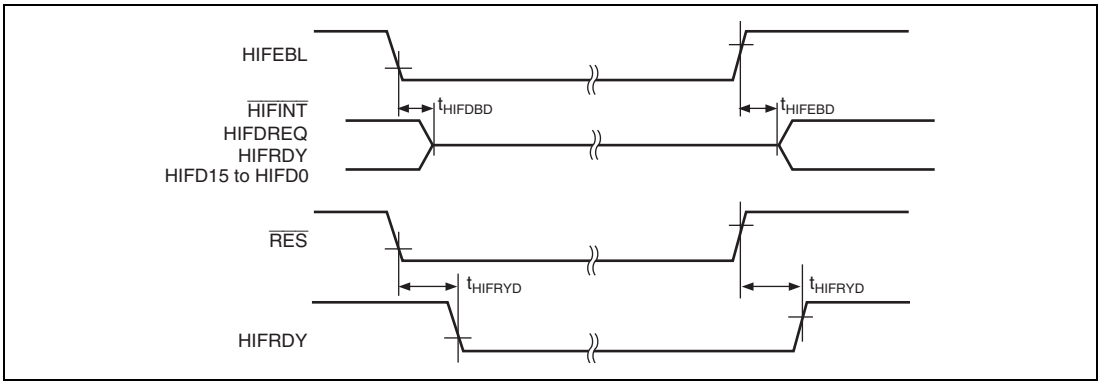


Figure 25.50 HIFRDY and HIF Pin Enable/Disable Timing

25.4.12 EtherC Timing

Note: These characteristics are valid in the case of utilizing the external PHY LSI. For example, "MDIO output" means that the MDIO input/output function is selected as the function of the pin PC16/MDIO, and the pin is used as the MDIO output pin of the on-chip EtherC.

Table 25.15 EtherC Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
TX-CLK cycle time	t_{Tcyc}	40	—	ns	—
TX-EN output delay time	t_{TENd}	1	20	ns	Figure 25.51
MII_TXD[3:0] output delay time	t_{MTDd}	1	20	ns	
CRS setup time	t_{CRSs}	10	—	ns	
CRS hold time	t_{CRSh}	10	—	ns	
COL setup time	t_{COLs}	10	—	ns	Figure 25.52
COL hold time	t_{COLh}	10	—	ns	
RX-CLK cycle time	t_{Rcyc}	40	—	ns	—
RX-DV setup time	t_{RDVs}	10	—	ns	Figure 25.53
RX-DV hold time	t_{RDVh}	10	—	ns	
MII_RXD[3:0] setup time	t_{MRDs}	10	—	ns	
MII_RXD[3:0] hold time	t_{MRDh}	10	—	ns	
RX-ER setup time	t_{RERs}	10	—	ns	Figure 25.54
RX-ER hold time	t_{RERh}	10	—	ns	
MDIO setup time	t_{MDIOs}	10	—	ns	Figure 25.55
MDIO hold time	t_{MDIOh}	10	—	ns	
MDIO output data hold time	t_{MDIOdh}	5	18	ns	Figure 25.56
WOL output delay time	t_{WOLd}	1	20	ns	Figure 25.57
EXOUT output delay time	t_{EXOUTd}	1	20	ns	Figure 25.58

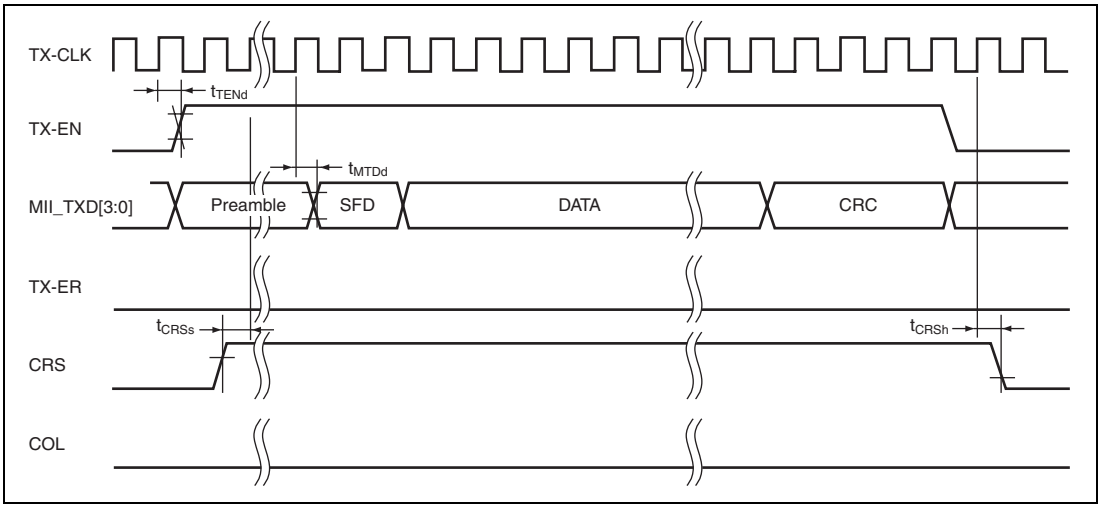


Figure 25.51 MII Transmission Timing (Normal Operation)

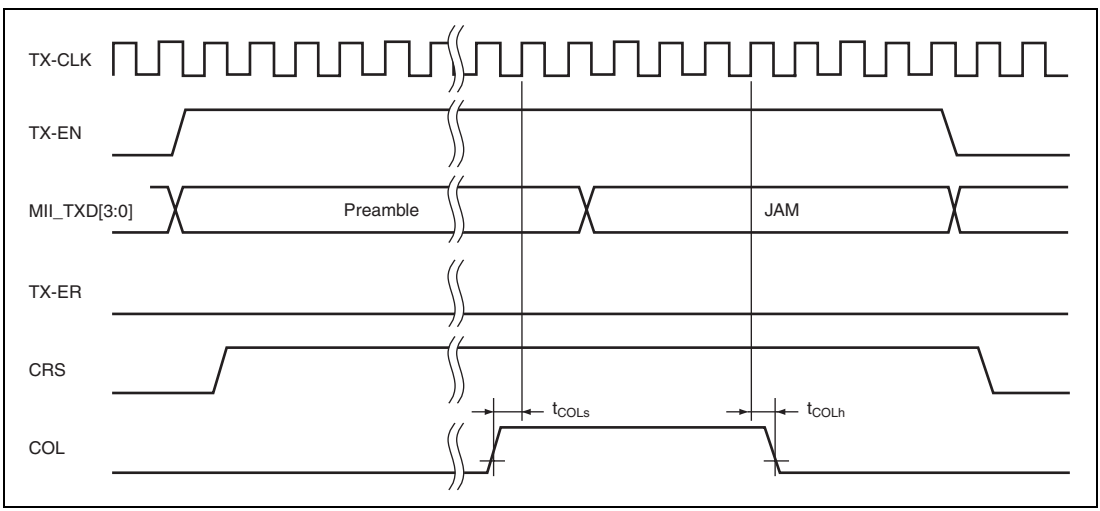


Figure 25.52 MII Transmission Timing (Collision Occurred)

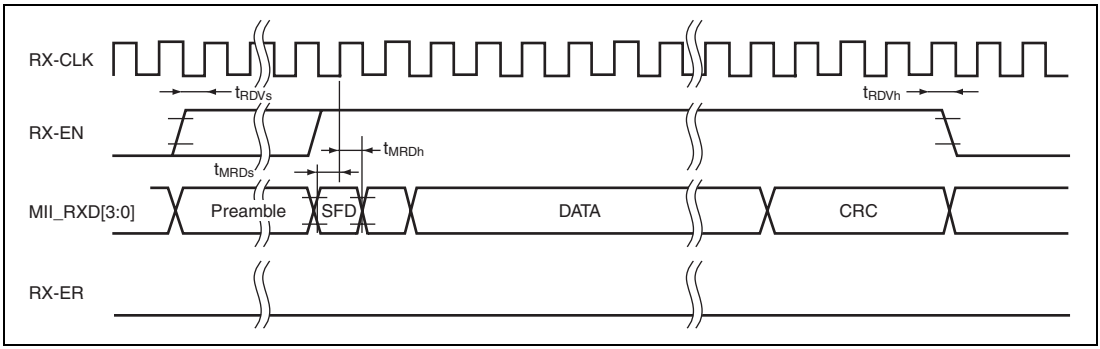


Figure 25.53 MII Reception Timing (Normal Operation)

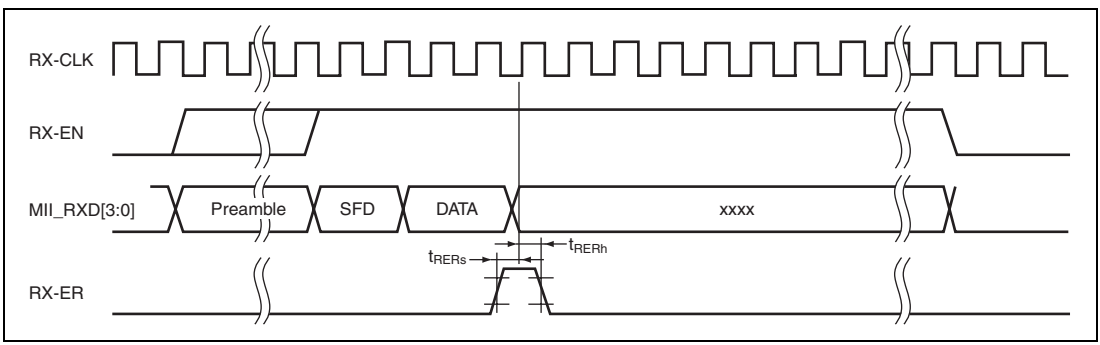


Figure 25.54 MII Reception Timing (Error Occurred)

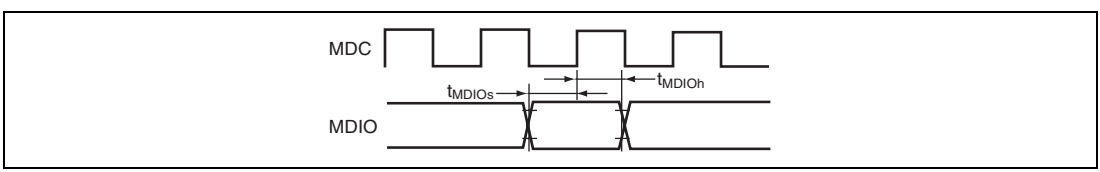


Figure 25.55 MDIO Input Timing

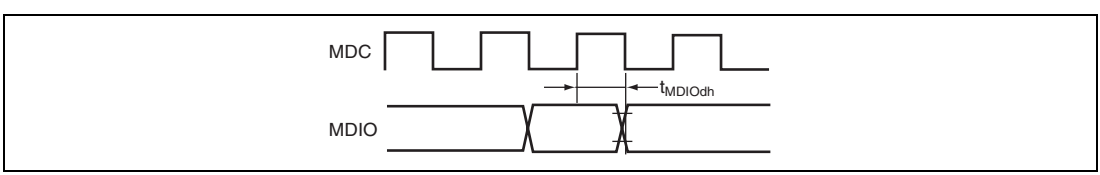


Figure 25.56 MDIO Output Timing

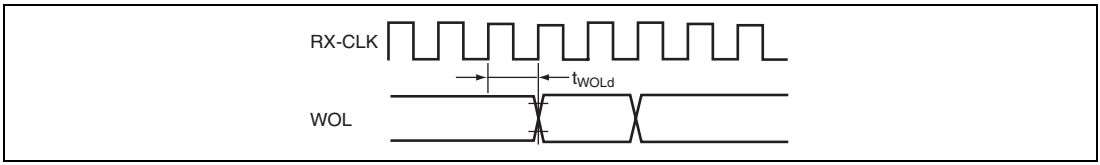


Figure 25.57 WOL Output Timing

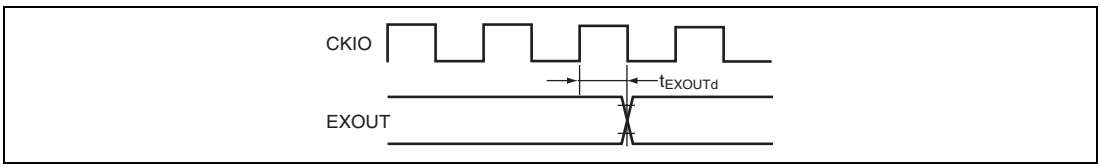


Figure 25.58 EXOUT Output Timing

25.4.13 H-UDI Related Pin Timing

Table 25.16 H-UDI Related Pin Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference Figures
TCK cycle time	t_{TCKcyc}	50	—	ns	Figure 25.59
TCK high pulse width	t_{TCKH}	19	—	ns	
TCK low pulse width	t_{TCKL}	19	—	ns	
TCK rising/falling time	t_{TCKrf}	—	4	ns	
TRST setup time	t_{TRSTS}	10	—	t_{bcyc}*	Figure 25.60
TRST hold time	t_{TRSTH}	50	—	t_{bcyc}*	
TDI setup time	t_{TDIS}	10	—	ns	Figure 25.61
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	19	ns	

Note: * t_{bcyc} indicates the period of the external bus clock (Bϕ).

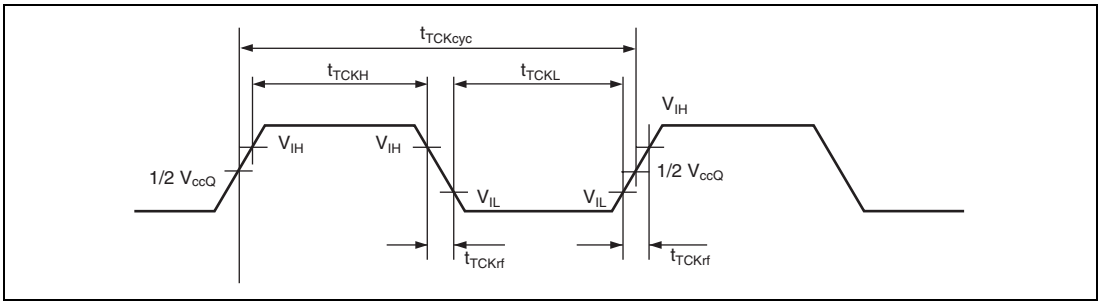


Figure 25.59 TCK Input Timing

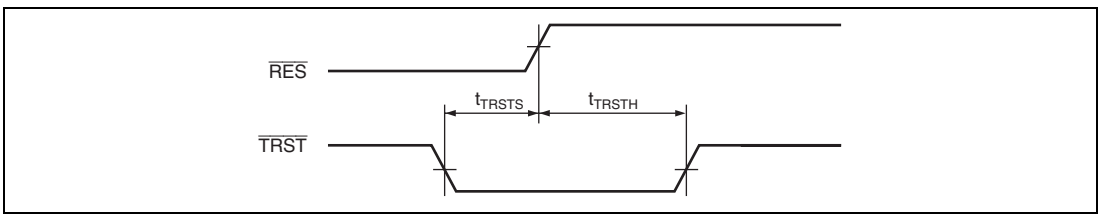


Figure 25.60 TCK Input Timing in Reset Hold State

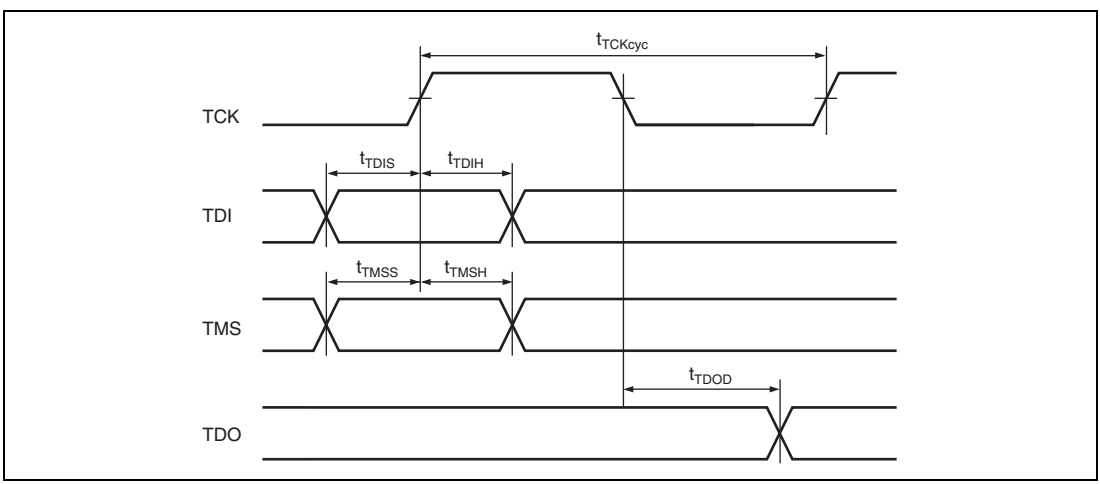
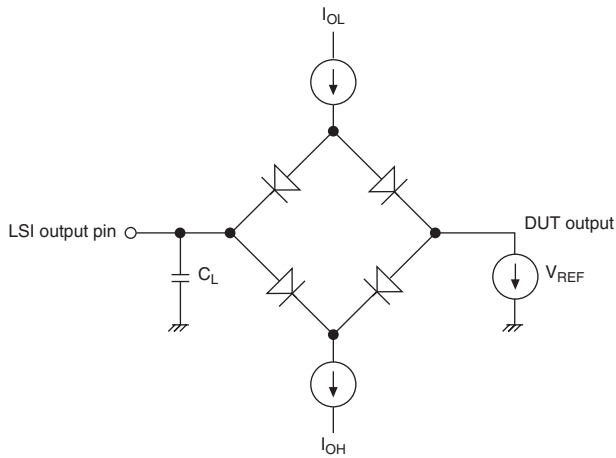


Figure 25.61 H-UDI Data Transmission Timing

25.4.14 AC Characteristic Test Conditions

- I/O signal reference level: $V_{CC}Q/2$ ($V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V)
- Input pulse level: V_{SS} to $V_{CC}Q$ ($\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, $\overline{HIFMD}$, $\overline{TRST}$, and EXTAL), V_{SS} to 3.0 V (other pins)
- Input rising and falling times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set for all pins as 30 pF.
2. I_{OL} and I_{OH} are shown in table 25.5.

Figure 25.62 Output Load Circuit

25.5 Physical Layer Ttransceiver (PHY) Characteristics (Reference Values)

Table 25.17 shows the characteristics of the physical layer transceiver (PHY).

Table 25.17 PHY Characteristics

Conditions: $V_{CC}1A = V_{CC}2A = V_{CC}3A = 3.3$ V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Transformer secondary-side differential output voltage	100BASE-TX output high level	V_{OH100}	+0.95	—	+1.05	V	—
	100BASE-TX output middle level	V_{OM100}	-50	—	+50	mV	—
	100BASE-TX output low level	V_{OL100}	-1.05	—	-0.95	V	—
	10BASE-TX output high level	V_{OH10}	2.2	—	2.8	V	—

Appendix

A. Port States in Each Pin State

Table A.1 Port States in Each Pin State

		Reset State		Power-Down Mode		
Classification	**Abbr.**	**Power-On (HIFMD = Low)**	**Power-On (HIFMD = High)**	**Software Standby**	**Sleep**	**H-UDI Module Standby**
Clock	EXTAL	I	I	I	I	I
	XTAL	O*[1]	O*[1]	O*[1]	O*[1]	O*[1]
	CKIO	O*[1]	O*[1]	ZO*[5]	O*[1]	O*[1]
	CK_PHY	I	I	I	I	I
System control	$\overline{\text{RES}}$	I	I	I	I	I
Operating mode control	MD5, MD3 to MD0	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I
	IRQ7 to IRQ0	—	—	I	I	I
Address bus	A25 to A16	—	—	ZHL*[4]	O	O
	A15 to A0	O	O	ZHL*[4]	O	O
Data bus	D31 to D16	—	—	Z	IO	IO
	D15 to D0	Z	Z	Z	IO	IO
Bus control	$\overline{\text{WAIT}}$	—	—	Z	I	I
	$\overline{\text{IOIS16}}$	—	—	Z	I	I
	CKE	—	—	ZO*[2]	O	O
	$\overline{\text{CAS}}$, $\overline{\text{RAS}}$	—	—	ZO*[2]	O	O
	$\overline{\text{WE0}}$/DQMLL	H	H	ZH*[4]	O	O
	$\overline{\text{WE1}}$/DQMLU/ $\overline{\text{WE}}$	H	H	ZH*[4]	O	O
	$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$	—	—	ZH*[4]	O	O

Classification	Abbr.	Reset State		Power-Down Mode		
		Power-On (HIFMD = Low)	Power-On (HIFMD = High)	Software Standby	Sleep	H-UDI Module Standby
Bus control	$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$	—	—	ZH*4	O	O
	$\overline{RD}$	H	H	ZH*4	O	O
	RDWR	H	H	ZH*4	O	O
	$\overline{CE2B}$, $\overline{CE2A}$	—	—	ZH*4	O	O
	$\overline{CS6B}$/$\overline{CE1B}$, $\overline{CS5B}$/$\overline{CE1A}$	—	—	ZH*4	O	O
	$\overline{CS4}$, $\overline{CS3}$	—	—	ZH*4	O	O
	$\overline{CS0}$	H	H	ZH*4	O	O
	$\overline{BS}$	—	—	ZH*4	O	O
Ethernet controller	ERXD3 to ERXD0	—	—	I	I	I
	ETXD3 to ETXD0	—	—	O	O	O
	RX_DV	—	—	I	I	I
	RX_ER	—	—	I	I	I
	RX_CLK	—	—	I	I	I
	TX_ER	—	—	O	O	O
	TX_EN	—	—	O	O	O
	TX_CLK	—	—	I	I	I
	COL	—	—	I	I	I
	CRS	—	—	I	I	I
	MDIO	—	—	IO	IO	IO
	MDC	—	—	O	O	O
	LNKSTA	—	—	Z	I	I
	EXOUT	—	—	Z	O	O
	WOL	—	—	Z	O	O

Classification	Abbr.	Reset State		Power-Down Mode		
		Power-On (HIFMD = Low)	Power-On (HIFMD = High)	Software Standby	Sleep	H-UDI Module Standby
DMAC	DREQ1, DREQ0	—	—	Z	I	I
	DACK1, DACK0	—	—	Z	O	O
	TEND1, TEND0	—	—	Z	O	O
SCIF	TxD2 to TxD0	—	—	Z	O	O
	RxD2 to RxD0	—	—	Z	I	I
	SCK2, SCK1	—	—	Z	O	O
	SCK0	—	—	Z	I	I
	RTS1, RTS0	—	—	Z	O	O
	CTS1, CTS0	—	—	Z	I	I
SIOF	SIOMCLK0	—	—	Z	I	I
	SCK_SIO0	—	—	Z	O	O
	SIOFSYNC0	—	—	Z	O	O
	TXD_SIO0	—	—	Z	O	O
	RXD_SIO0	—	—	Z	I	I
Host interface	HIFEBL	—	Z	Z	I	I
	HIFRDY	—	O	O	O*3	O*3
	HIFDREQ	—	Z	Z	O*3	O*3
	HIFMD	I	I	I	I*3	I*3
	$\overline{\text{HIFINT}}$	—	Z	Z	O*3	O*3
	$\overline{\text{HIFRD}}$	—	Z	Z	I*3	I*3
	$\overline{\text{HIFWR}}$	—	Z	Z	I*3	I*3
	HIFRS	—	Z	Z	I*3	I*3
	$\overline{\text{HIFCS}}$	—	Z	Z	I*3	I*3
	HIFD15 to HIFD0	—	Z	Z	IO*3	IO*3

Classification	Abbr.	Reset State		Power-Down Mode		
		Power-On (HIFMD = Low)	Power-On (HIFMD = High)	Software Standby	Sleep	H-UDI Module Standby
User debugging interface (H-UDI)	$\overline{\text{TRST}}$	I	I	I	I	I
	TCK	I	I	I	I	I
	TMS	I	I	I	I	I
	TDI	I	I	I	I	I
	TDO	Z	Z	ZO*6	ZO*6	Z
	$\overline{\text{ASEMD}}$	I	I	I	I	I
I/O port	PA25 to PA16	Z	Z	Z	P	I/O
	PB13 to PB0	Z	Z	Z	P	I/O
	PC20 to PC0	Z	Z	Z	P	I/O
	PD7 to PD0	Z	Z	Z	P	I/O
	PE24 to PE4, PE2 to PE0	Z	—	Z	P	I/O
	PE3	—	—	Z	P	I/O
Test mode	$\overline{\text{TESTMD}}$	I	I	I	I	I
	$\overline{\text{TESTOUT}}$	O	O	O	O	O
PHY	TxP	O	O	O	O	O
	TxM	O	O	O	O	O
	RxP	I	I	I	I	I
	RxM	I	I	I	I	I
	$\overline{\text{SPEED100}}$	—	—	O	O	O
	$\overline{\text{LINK}}$	—	—	O	O	O
	$\overline{\text{CRS}}$	—	—	O	O	O
	$\overline{\text{DUPLEX}}$	—	—	O	O	O
	EXRES1	I	I	I	I	I
	TSTBUSA	Z	Z	Z	Z	Z

[Legend]

—: This pin function is not selected as an initial state.

I: Input

O: Output

IO: Input/output

H: High level output

L: Low level output

Z: High-impedance

P: Input or output depending on the register setting

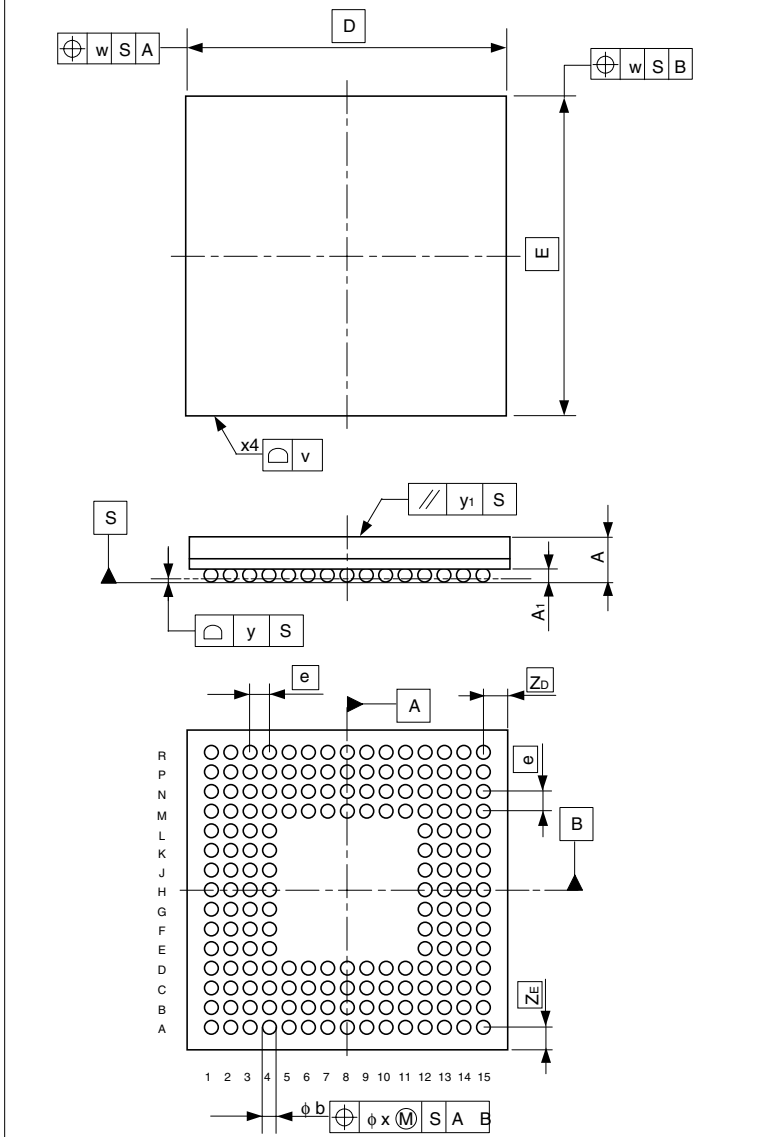
Notes:
1. Depends on the clock mode (setting of pins MD2 to MD0).
2. Depends on the HIZCNT bit in CMNCR.
3. High-impedance when HIFEBL = low
4. Depends on the HIZMEM bit in CMNCR.
5. Depends on the HIZCNT bit in CMNCR or the CKOEN bit in FRQCR.
6. This pin becomes output state only when reading data from the H-UDI and retains high-impedance state when the pin is not output state.
7. In all pins having the weak keeper circuit, even with the "Z" (meaning high-impedance) description, each weak keeper circuit is always operating. For details on the weak keeper circuit, see section 19.6, Usage Notes.

B. Product Code Lineup

Product Code	Catalogue Code	Operating Temperature	Solder ball Composition	Package Code
DS76190B125BGV	R4S76190B125BGV	–20 to 70°C	Pb-free solder	PLBG0176GA-A
DS76190N125BGV	R4S76190N125BGV	–20 to 85°C	Pb-free solder	
DS76190W125BGV	R4S76190W125BGV	–20 to 85°C	Pb-free solder	
DS76190D125BGV	R4S76190D125BGV	–40 to 85°C	Pb-free solder	
DS76190B125BG	R4S76190B125BG	–20 to 70°C	Non-Pb-free solder	
DS76190N125BG	R4S76190N125BG	–20 to 85°C	Non-Pb-free solder	
DS76190W125BG	R4S76190W125BG	–20 to 85°C	Non-Pb-free solder	
DS76190D125BG	R4S76190D125BG	–40 to 85°C	Non-Pb-free solder	
DS76191B125BGV	R4S76191B125BGV	–20 to 70°C	Pb-free solder	
DS76191N125BGV	R4S76191N125BGV	–20 to 85°C	Pb-free solder	
DS76191W125BGV	R4S76191W125BGV	–20 to 85°C	Pb-free solder	
DS76191D125BGV	R4S76191D125BGV	–40 to 85°C	Pb-free solder	

C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Figure C.1 Package Dimensions (BP-176)

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
Table 1.1 Features of SH7619	4	Amended **Items**: Ethernet controller (EtherC) **Specification**: • MAC (Media Access Control) function — Data frame assembly/disassembly (frame format conforming to IEEE802.3) — CSMA/CD link management (collision prevention and collision processing) — CRC processing — 512 bytes each for transmit/receive FIFO — Full-duplex transmit/receive support — Short frame/long frame detectable
Table 1.2 Pin Functions	16	Amended **Classification**: Physical layer transceiver (PHY) **Abbr.**: TSTBUSA **Description**: Input/output pin for testing the on-chip PHY. This pin should be open.
6.6.1 Interrupt Sequence	102	Deleted 4. The CPU ... Instead of executing the decoded instruction, the CPU starts interrupt exception handling ~~(see figure 6.5)~~.
6.7 Interrupt Response Time	104	Deleted Table 6.3 lists … handling routine begins. ~~Figure 6.5 shows an example of the pipeline operation when an IRQ interrupt is accepted.~~
Table 7.12 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (1)	160, 161	Deleted and amended **Setting**: ~~A2/3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0] **Setting**: ~~A2/3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]

Item	Page	Revision (See Manual for Details)
Table 7.13 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (2)	161, 162	Deleted and amended **Setting**: ~~A2/3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0] **Setting**: ~~A2/3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]
Table 7.14 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (3)	163	Deleted and amended **Setting**: ~~A2/3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]
Table 7.15 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (4)	164, 165	Deleted **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0] **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]
Table 7.16 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (5)	165, 166	Deleted **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0] **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]
Table 7.17 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (6)	167, 168	Deleted **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0] **Setting**: ~~A3~~ BSZ [1:0], A3 ROW [1:0], A3 COL [1:0]
11.4 Operation	253	Amended The overview of the Ethernet controller (EtherC) are shown below. The EtherC transmits and receives PAUSE frames conforming to the Ethernet/IEEE802.3x frames.

Item	Page	Revision (See Manual for Details)
Figure 11.5 MII Management Frame Format	258	Amended [Legend] IDLE: Wait time until next MII management format input (a) Write: Independent bus release (notation: X) performed (b) Read: Bus already released in TA; data control unnecessary
11.6 Usage Notes • Flow Control Defect 1 • Flow Control Defect 2	264, 265	Added Note: This defect may be generated only in the R4S76190. In the R4S76191, the defect has been corrected.
12.2.6 EtherC/E-DMAC Status Register (EESR)	275	Amended **Bit** / **Description** 21 / Frame Transmit Complete Indicates that all the data specified by the transmit descriptor has been transmitted to the EtherC. The transfer status is written back to the relevant descriptor. For 1-frame/1-buffer processing, when 1-frame transmission is completed and the transmission descriptor valid bit (TACT) in the next descriptor is not set, transmission is completed and this bit is set to 1. Likewise, for multiple-frame buffer processing, when the last data in the frame is transmitted and the transmission descriptor valid bit (TACT) in the next descriptor is not set, transmission is completed and this bit is set to 1. After frame transmission, the E-DMAC writes the transmission status back to the descriptor. 0: Transfer not complete, or no transfer directive 1: Transfer complete
13.3.4 DMA Channel Control Registers 0 to 3 (CHCR_0 to CHCR_3) [Notice] [Workaround]	335	Added

Item	Page	Revision (See Manual for Details)
13.3.5 DMA Operation Register (DMAOR) [Notice] [Workaround]	338	Added
15.3.9 FIFO Control Register (SCFCR)	407	Amended

Bit	Description
3	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. In synchronous mode, clear this bit to 0. This bit is available only in SCFCR_0 and SCFCR_1. In SCFCR_2, this bit is reserved. The initial value is 0 and the write value should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * The $\overline{\text{CTS}}$ pin state has no effect on transmit operations regardless of the input value. The $\overline{\text{RTS}}$ pin state has no effect on receive operations, either.

16.3.1 Mode Register (SIMDR)	450	Amended

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Item	Page	Revision (See Manual for Details)
16.3.4 Receive Data Register (SIRDR)	456	Added SIRDR is a 32-bit read-only register that reads receive data of the SIOF. SIRDR stores data in the receive FIFO and is initialized to undefined value by the conditions specified in section 24, List of Registers, or by a receive reset caused by the RXRST bit in SICTR. Amended

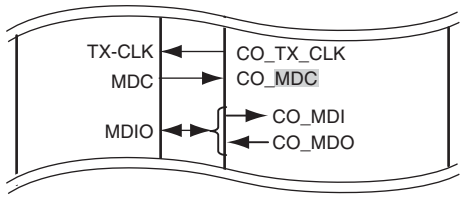
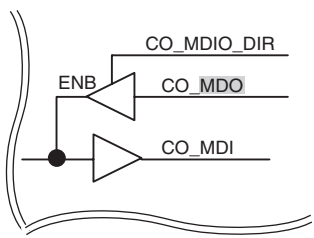
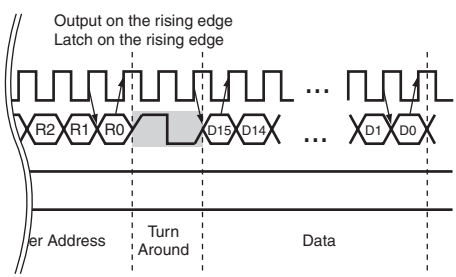
Bit	Bit Name	Initial Value	R/W
31 to 16	SIRDL 15 to 0	Undefined	R
15 to 0	SIRDR 15 to 0	Undefined	R

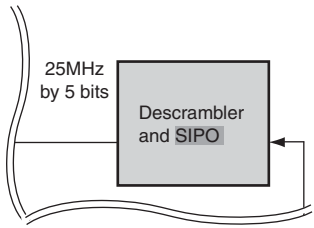
16.3.7 Status Register (SISTR)	464	Deleted

Bit	Description
0	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF indicates overflow, and receive data is lost. • ~~This bit is valid when the RXE bit in SICTR is 1.~~ • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Table 16.7 Audio Mode Specification for Receive Data	482	Deleted Note: Left and right same audio mode is not supported in receive data. To execute ~~8-bit~~ monaural transmission or reception, use the left channel.

Item	Page	Revision (See Manual for Details)
16.4.7 Transmit and Receive Procedures Transmission/Reception and Transmission in Master Mode:	487	Amended Figure 16.9 (1) shows an example of settings and operation for master mode transmission/reception. Figure 16.9 (2) shows an example of settings and operation for master mode transmission.
Figure 16.9 (1) Transmission/Reception Operation in Master Mode (Example of Reception and Full-Duplex Transmission by the CPU with TDMAE=0)	488	Replaced from Figure 16.9, Example of Transmit Operation in Master Mode
Figure 16.9 (2) Transmission Operation in Master Mode (Example of Half-Duplex Transmission by the CPU with TDMAE=0)	489	Added
16.4.9 Transmit and Receive Timing [Notes on Usage]	500, 501	Added
18.2 Notes on Usage	577	Added
19.6 Usage Notes	591	Amended 2. The weak keeper circuit is included in all pins except MD5, MD3, MD2, MD1, MD0, $\overline{\text{ASEMD}}$, $\overline{\text{TESTMD}}$, EXTAL, XTAL, TxP, TxM, RxP, RxM, EXRES1, and TSTBUSA. The weak keeper is a circuit, always operating while the power is on, that fixes the input in I/O pins to low or high when the pins are not driven from outside. Notes on processing the input pins are as follows: — When using pins having the weak keeper circuit as input pins and driving these pins to a certain level from outside, adjust the resistance of pull-up/pull-down resistors to let the weak keeper circuit keep the intended levels. (2 kΩ and 8 kΩ are recommended respectively.) The larger the resistance is, the longer the transition time is. In addition, a large resistance may fail to let the weak keeper circuit to keep the intended levels. Therefore, when the resistors adjusted comparatively large are used, ensure that any transition does not delay in the system.

Item	Page	Revision (See Manual for Details)
Figure 22.1 The Block Diagram around PHY Module	632	Amended 
22.4.1 Serial Management Interface (SMI)	635	Amended The Serial Management Interface is used to control this PHY core and obtain its status. This interface supports registers 0 through 6 as required by Clause 22 of the IEEE802.3 standard. Non-supported registers (7 to 15) will be read as hexadecimal "FFFF".
Figure 22.3 How to Derive MDIO Signal from Core Signals	635	Amended 
22.4.1 Serial Management Interface (SMI)	635	Amended The CO_MDC signal is an a-periodic clock provided by the station management controller (SMC), part of the EtherC. The CO_MDI signal receives serial data (commands) from the controller SMC. The CO_MDO sends serial data (status) to the SMC.
Figure 22.5 MDIO Timing and Frame Structure (WRITE Cycle)	636	Amended 

Item	Page	Revision (See Manual for Details)
Figure 22.7 Receive Data Path	651	Amended 
22.6 100Base-TX Receive (4) Descrambling	652	Deleted Special logic in the descrambler ensures synchronization with the remote PHY by searching for IDLE symbols within a window of 4000 bytes ~~(40us)~~. This window ensures that a maximum packet size of 1514 bytes, allowed by the IEEE 802.3 standard, can be received with no interference.
22.11 Internal I/O Signals • Management signals	667	Amended

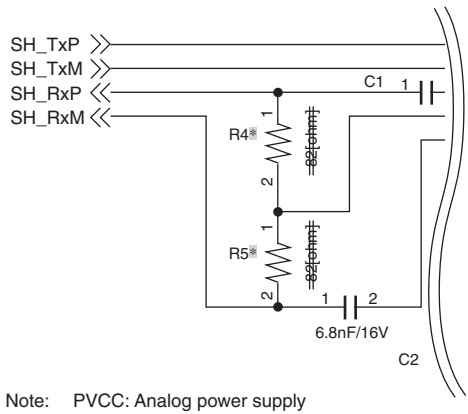
Signal Name	Type	Description
CO_MDI	I	Management Data Input: Serial management data input.
CO_MDO	O	Management Data Output: Serial management data output.
CO_MDC	I	Management Clock: Serial management clock.
CO_MDIO_DIR	O	Management Data Direction: May be used to control output enabled buffer for MDIO.

22.13 Usage Notes	669	Deleted (4) Waveform Adjustment ~~for Tx100 Output~~

Item	Page	Revision (See Manual for Details)
• Meanings of the value written to register 23	670	Amended

Bit	Description
8	These bits adjust the slope.
7	00: Three steps up
	01: Two steps up
	10: One step up
	11: Regular
3	These bits adjust the transition time.
2	00: One step up
	01: One step down
	10: Regular
	11: Two steps down

~~Figure 22.10 Role of Each Bit Field (Example of Rising Waveform) Slope is Controlled in Four Segments~~	—	Deleted
(b) Adjustment Register for Tx10 Waveform Output (c) Detailed Descriptions (d) Other Control Methods	672 to 677	Added

Item	Page	Revision (See Manual for Details)
Figure 22.10 Example of Connection with a Pulse Transformer (RJ45)	679	Amended  Note: PVCC: Analog power supply PVSS: Analog ground * R4, R5: Please set the terminating resistance as 50 ohm, summing up with both side of the magnetic (even if the magnetic is in the RJ45 connector).
23.3.1 The Procedures of Setting Up the On-Chip PHY	689	Amended 3. Activation of the on-chip PHY module To activate the on-chip PHY module, set the pin function registers of Port C as something but EtherC function, that is, I/O ports and LED outputs of the on-chip PHY. • PCCRH2 = H'0000 • PCCRL1 = H'0000 • PCCRL2 = H'FF00 ……..
23.3.2 The Procedures of Set Up the External PHY LSI	690	Amended In the case of utilizing the external PHY LSI, select the EtherC function of the pin function controllers and then set up the internal registers of the PHY LSI with the MII management frame. 1. Activation of the external PHY LSI. Select the EtherC functions with pin function controller. • PCCRH2 = H'0155 • PCCRL1 = H'5555 • PCCRL2 = H'5555

Item	Page	Revision (See Manual for Details)
24.2 Register Bits	707	Amended

Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4
SIMDR	TRMD1	TRMD0	SYNCAT	REDG
	TXDIZ	RCIM	—	SYNCDL
SISCR	MSSEL	MSIMM	—	BRPS4
	—	—	—	—

25.2 Power-On and Power-Off Order	731	Added

- Power-off order
  - In the reverse order of power-on, first turn off the 1.8-V system power, then turn off the 3.3-V system power within 10 ms. This time should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation. In some systems, Vcc may exceed 3.3-V system power (Vcc > 3.3-V system power) temporarily, on the falling edge. Even in this case, the inverted potential difference must be 0.3 V or less.

Item	Page	Revision (See Manual for Details)
25.2 Power-On and Power-Off Order	731	Amended

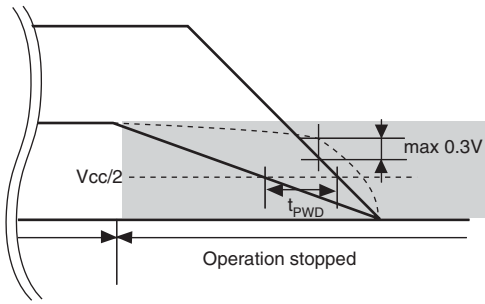


Table 25.7 Clock Timing — 735, 736 — Amended

Item	Symbol	Min.	Max.	Unit.
CK_PHY clock input frequency	f_{CKPHY}	25 −100 ppm*1	25 +100 ppm*1	MHz
~~CK_PHY clock input cycle time~~	~~$T_{CKPHYcyc}$~~	~~39.996~~	~~40.004~~	~~ns~~
CK_PHY clock input low pulse width	t_{CKPHYL}	12	—	ns
$\overline{RES}$ assert time	t_{RESW}	20	—	t_{bcyc}*2

Notes: 1. Error margin means frequency tolerance (reference value). Recommending under 100 ps of peak to peak jitter.

2. t_{bcyc} indicates the period of the external bus clock (Bϕ).

Item	Page	Revision (See Manual for Details)
Table 25.9 Bus Timing	742	Added

Item	Symbol	Min.
Write data hold time 2	t_{WDH2}	2
Write data hold time 3	t_{WDH3}	0
$\overline{WAIT}$ setup time	t_{WTS1} *	$1/2 \times t_{bcyc} + 11$
$\overline{WAIT}$ hold time	t_{WTH1} *	$1/2 \times t_{bcyc} + 10$

Note: * The AC timing specification of $\overline{WAIT}$ is as follows.

Input setup time + hold time of $\overline{WAIT}$

= 11 [ns] + 10 [ns] = 21 [ns]

As the frequency, 47.62 [MHz]

Therefore, when the bus clock is 47.62 MHz or more, at least either setup time or hold time cannot be satisfied during 1-bus clock. The following notes should be confirmed.

- When the hardware-wait function is used synchronously

The bus clock frequency must be low enough to satisfy the AC specification above.

- When the hardware-wait function is used asynchronously

To ensure the setup time until the start of the input assertion of $\overline{WAIT}$, insert appropriate number of the software wait after the T1 state. Then, even if the AC specification above cannot be satisfied, the accesses can be executed correctly.

Item	Page	Revision (See Manual for Details)
25.4.12 EtherC Timing	780	Added Note: These characteristics are valid in the case of utilizing the external PHY LSI. For example, "MDIO output" means that the MDIO input/output function is selected as the function of the pin PC16/MDIO, and the pin is used as the MDIO output pin of the on-chip EtherC.
Appendix Table A.1 Port States in Each Pin State	791	Added Notes: 7. In all pins having the weak keeper circuit, even with the "Z" (meaning high-impedance) description, each weak keeper circuit is always operating. For details on the weak keeper circuit, see section 19.6, Usage Notes.
B. Product Code Lineup	792	Added

Product Code	Catalogue Code	Operating Temperature	Solder ball Composition
DS76190B125BG	R4S76190B125BG	–20 to 70°C	Non-Pb-free solder
DS76190N125BG	R4S76190N125BG	–20 to 85°C	Non-Pb-free solder
DS76190W125BG	R4S76190W125BG	–20 to 85°C	Non-Pb-free solder
DS76190D125BG	R4S76190D125BG	–40 to 85°C	Non-Pb-free solder
DS76191B125BGV	R4S76191B125BGV	–20 to 70°C	Pb-free solder
DS76191N125BGV	R4S76191N125BGV	–20 to 85°C	Pb-free solder
DS76191W125BGV	R4S76191W125BGV	–20 to 85°C	Pb-free solder
DS76191D125BGV	R4S76191D125BGV	–40 to 85°C	Pb-free solder

Index

Numerics

100Base-TX receive 651
100Base-TX transmit..... 648
100M phase lock loop (PLL)..... 651
100M receive data across the MII..... 654
100M receive input..... 651
100M transmit data across the MII..... 648
100M transmit driver..... 650
10Base-T receive..... 656
10Base-T transmit..... 654
10M receive data across the MII..... 656
10M receive input and squelch..... 656
10M transmit data across the MII..... 655
10M transmit drivers..... 655
4B/5B encoding..... 648
5B/4B decoding..... 653

A

Access wait control..... 154
Accessing MII registers..... 258
Address array..... 60
Address error exception handling..... 73
Address error sources 73
Address multiplexing..... 159
Addressing modes..... 32
Alignment..... 653
Arithmetic operation instructions 45
Asynchronous mode 414
Auto-negotiation..... 658
Auto-negotiation disabling 660
Auto-refreshing..... 181
Auto-request mode 343

B

Bank active 174
Basic timing 149
Basic timing for I/O card interface..... 195
Basic timing for
memory card interface 193
Baud rate generator..... 477
Bit rate..... 398
Boundary scan..... 629
Branch instructions 48
Burst mode..... 354
Burst read..... 168
Burst write..... 172
Bus state controller (BSC) 107
Byte-selection SRAM interface 186

C

Cache 53
Cache structure..... 53
Carrier sense 661
Cases when exceptions are accepted..... 78
Changing clock operating mode..... 211
Changing division ratio..... 211
Changing frequency..... 210
Changing multiplication ratio 210
Clock operating modes 204
Clock pulse generator (CPG)..... 201
Coherency of cache and
external memory 59
Collision detect 661
Compare match timer (CMT) 367
Connection to PHY-LSI..... 263
Control registers..... 25
CPU..... 23
Cycle-steal mode..... 352

D

Data array 61
Data register..... 579
Data transfer instructions..... 43
Descrambling..... 652
Direct memory access controller
(DMAC) 325
Divided areas and cache 55
Dual address mode..... 349

E

Endian/access size and data alignment ... 143
Equalizer, baseline wander correction
and clock and data recovery 652
EtherC receiver..... 255
EtherC transmitter 253
Ethernet controller (EtherC) 233
Ethernet controller direct memory
access controller (E-DMAC) 267
Exception handling..... 67
Exception handling operations 68
Exception handling vector table 69
Extension of chip select ($\overline{\text{CSn}}$)
assertion period..... 156
External request mode 343

F

Features of instructions..... 29
Fixed mode..... 345
Flow control..... 262

G

General illegal instructions..... 77
General registers (Rn)..... 25
General signals 667

H

Half-duplex and full-duplex..... 661
Host interface (HIF)..... 511
H-UDI interrupt 628
H-UDI reset 628

I

I/O ports..... 579
Illegal slot instructions..... 77
Immediate data formats..... 29
Initial values of registers..... 27
Input clock to PHY module 668
Instruction formats..... 35
Instruction set..... 39
Intermittent mode..... 353
Interrupt controller (INTC)..... 83
Interrupt exception handling..... 75
Interrupt exception handling
vector table..... 100
Interrupt priority 75
Interrupt response time 104
Interrupt sequence..... 102
Interrupt sources 74
IRQ interrupts 98
Isolate mode..... 662

J

Jabber detection 657

L

LED description..... 664
Link integrity test..... 662
Logic operation instructions 47
Loopback operation 664

M

MAC interface 657
Magic packet detection 261
Management signals 667
Manchester decoding 656
Manchester encoding 655
Memory data formats 28
Memory-mapped cache 60
MII 657
MII frame timing 256
MII signals 666
Miscellaneous functions 661
Module standby mode 232
Multi-buffer frame
transmit/receive processing 304
multiplexed pin 543

N

NMI interrupt 97
Normal space interface 149
NRZI and MLT-3 decoding 652
NRZI and MLT3 encoding 650

O

On-chip peripheral module interrupts 99
On-chip peripheral module
request mode 344
Operation by IPG setting 262
Operation mode 667

P

Parallel detection 660
PCMCIA interface 191
PHY (on-chip PHY module) 631
PHY address 667
PHY interface (PHY-IF) 683
PHY management control 635
Pin assignments 8
Pin function controller (PFC) 543
Pin functions 9
Power-down modes 223, 662
Power-on reset 71
Power-on sequence 184

R

Read access 58
Receive data valid signal 653
Receive descriptor 0 (RD0) 297
Receive descriptor 1 (RD1) 300
Receive descriptor 2 (RD2) 300
Receiver errors 654
Receiving serial data
(asynchronous mode) 423
Receiving serial data
(synchronous mode) 432
Refreshing 181
Register data format 28
Registers
APR 251, 699, 719, 728
BAMRA 596, 699, 720, 728
BAMRB 598, 699, 720, 728
BARA 595, 699, 720, 728
BARB 597, 699, 720, 728
BBRA 596, 699, 721, 728
BBRB 599, 699, 720, 728
BDMRB 599, 699, 719, 728
BDRB 598, 699, 719, 728
BETR 604, 699, 720, 728
BRCR 601, 699, 719, 728
BRDR 605, 699, 721, 728
BRSR 604, 699, 720, 728
CCR1 ..... 56, 699, 721, 728
CDCR 248, 698, 717, 727
CEFCR 249, 698, 718, 728
CHCR 330
CHCR_0 ..... 692, 700, 722

CHCR_1 ..... 692, 701, 722
CHCR_2 ..... 692, 701, 722
CHCR_3 ..... 692, 702, 722
CMCNT 370
CMCNT_0 ..... 696, 709, 725
CMCNT_1 ..... 696, 709, 725
CMCOR 370
CMCOR_0 ..... 696, 709, 725
CMCOR_1 ..... 696, 709, 725
CMCSR 369
CMCSR_0 ..... 696, 709, 725
CMCSR_1 ..... 696, 709, 725
CMNCR 116, 697, 711, 726
CMSTR 368, 696, 709, 725
CNDCR 248, 698, 718, 728
CS0BCR 697, 711, 726
CS0WCR 122, 697, 712, 726
CS3BCR 697, 711, 726
CS3WCR 124, 132, 697, 712, 726
CS4BCR 697, 711, 726
CS4WCR 125, 697, 712, 726
CS5BBCR 697, 711, 726
CS5BWCR 128, 134, 697, 712, 726
CS6BBCR 697, 711, 726
CS6BWCR 130, 134, 697, 712, 726
CSnBCR 117
CSnWCR 122
DAR 329
DAR_0 ..... 692, 700, 722
DAR_1 ..... 692, 700, 722
DAR_2 ..... 692, 701, 722
DAR_3 ..... 692, 702, 722
DMAOR 336, 692, 702, 722
DMARS 339
DMARS0 ..... 693, 704, 723
DMARS1 ..... 693, 704, 723
DMATCR 329
DMATCR_0 ..... 692, 700, 722
DMATCR_1 ..... 692, 701, 722
DMATCR_2 ..... 692, 701, 722
DMATCR_3 ..... 692, 702, 722
ECMR 238, 698, 716, 727
ECSIPR 243, 698, 716, 727
ECSR 241, 698, 716, 727
EDMR 269, 697, 713, 727
EDOCR 287, 698, 715, 727
EDRRR 271, 697, 714, 727
EDTRR 270, 697, 713, 727
EESIPR 278, 697, 714, 727
EESR 273, 697, 714, 727
FCFTR 289, 698, 715, 727
FDR 285, 697, 715, 727
FRECR 249, 698, 718, 728
FRQCR 206, 694, 705, 724
HIFADR 524, 696, 710, 726
HIFBCR 525, 696, 711, 726
HIFBICR 527, 696, 710, 726
HIFDATA 525, 696, 710, 726
HIFDTR 526, 696, 710, 726
HIFEICR 523, 696, 710, 726
HIFGSR 518, 696, 709, 726
HIFIDX 515, 696, 709, 726
HIFIICR 523, 696, 710, 726
HIFMCR 521, 696, 710, 726
HIFSCR 518, 696, 709, 726
ICR0 ..... 86, 694, 705, 723
IPGR 251, 698, 719, 728
IPR 95
IPRA 694, 705, 723
IPRB 694, 705, 723
IPRC 693, 704, 723
IPRD 693, 704, 723
IPRE 693, 704, 723
IPRF 693, 704, 723
IPRG 693, 704, 723
IRQCR 87, 694, 705, 723
IRQSR 90, 694, 705, 723
LCCR 248, 698, 717, 728
MAFCR 250, 698, 718, 728
MAHR 245, 698, 717, 727

MALR..... 245, 698, 717, 727
MCLKCR 208, 694, 704, 723
MPR..... 252, 699, 719, 728
PACRH1..... 554, 692, 702, 722
PACRH2..... 554, 692, 702, 722
PADRH..... 579, 692, 702, 722
PAIORH 554, 692, 702, 722
PBCRL1 ..... 557, 693, 702, 722
PBCRL2 ..... 557, 693, 702, 722
PBDRL 581, 692, 702, 722
PBIORL..... 557, 692, 702, 722
PCCRH2..... 561, 693, 703, 723
PCCRL1 ..... 561, 693, 703, 723
PCCRL2 ..... 561, 693, 703, 723
PCDRH..... 584, 693, 703, 722
PCDRL 584, 693, 703, 722
PCIORH 561, 693, 703, 722
PCIORL..... 561, 693, 703, 722
PDCRL2 ..... 567, 693, 703, 723
PDDRL 586, 693, 703, 723
PDIORL..... 566, 693, 703, 723
PECRH1 ..... 569, 693, 703, 723
PECRH2 ..... 569, 693, 704, 723
PECRL1..... 569, 693, 704, 723
PECRL2..... 569, 693, 704, 723
PEDRH 589, 693, 703, 723
PEDRL 589, 693, 703, 723
PEIORH..... 569, 693, 703, 723
PEIORL 569, 693, 703, 723
PHYIFADDRR..... 687, 696, 708, 725
PHYIFCR 685, 696, 708, 725
PHYIFSMIR2..... 686, 696, 708, 725
PHYIFSMIR3..... 687, 696, 708, 725
PHYIFSR..... 688, 696, 709, 725
PIR..... 244, 698, 716, 727
PSR..... 247, 698, 717, 727
RBWAR 288, 698, 715, 727
RDFAR..... 288, 698, 716, 727
RDLAR..... 272, 697, 714, 727
Register 0 (basic control)..... 643
Register 1 (basic status) 644
Register 2 (PHY identifier 1)..... 644
Register 3 (PHY identifier 2)..... 645
Register 4
(auto negotiation advertisement)..... 645
Register 5 (auto negotiation
link partner ability) 646
Register 6
(auto negotiation expansion)..... 647
RFCR 250, 698, 718, 728
RFLR 246, 698, 717, 727
RMCR..... 286, 697, 715, 727
RMFCR..... 283, 697, 714, 727
RTCNT 141, 697, 713, 727
RTCOR 142, 697, 713, 727
RTCSR..... 139, 697, 713, 726
SAR..... 329
SAR_0..... 692, 700, 722
SAR_1..... 692, 700, 722
SAR_2..... 692, 701, 722
SAR_3..... 692, 701, 722
SCBRR..... 398
SCBRR_0..... 694, 705, 724
SCBRR_1..... 695, 706, 724
SCBRR_2..... 695, 706, 724
SCFCR 405
SCFCR_0 ..... 694, 705, 724
SCFCR_1 ..... 695, 706, 724
SCFCR_2 ..... 695, 707, 725
SCFDR..... 408
SCFDR_0..... 694, 706, 724
SCFDR_1..... 695, 706, 724
SCFDR_2..... 695, 707, 725
SCFRDR 382
SCFRDR_0 ..... 694, 705, 724
SCFRDR_1 ..... 695, 706, 724
SCFRDR_2 ..... 695, 707, 725
SCFSR 390
SCFSR_0 ..... 694, 705, 724
SCFSR_1 ..... 695, 706, 724

SCFSR_2 ..... 695, 706, 724
SCFTDR 383
SCFTDR_0 ..... 694, 705, 724
SCFTDR_1 ..... 695, 706, 724
SCFTDR_2 ..... 695, 706, 724
SCLSR 413
SCLSR_0 ..... 694, 706, 724
SCLSR_1 ..... 695, 706, 724
SCLSR_2 ..... 695, 707, 725
SCRSR 382
SCSCR 386
SCSCR_0 ..... 694, 705, 724
SCSCR_1 ..... 695, 706, 724
SCSCR_2 ..... 695, 706, 724
SCSMR 383
SCSMR_0 ..... 694, 705, 724
SCSMR_1 ..... 695, 706, 724
SCSMR_2 ..... 695, 706, 724
SCSPTR 409
SCSPTR_0 ..... 694, 706, 724
SCSPTR_1 ..... 695, 706, 724
SCSPTR_2 ..... 695, 707, 725
SCTSR 382
SDBPR 617
SDBSR 618
SDCR 138, 697, 713, 726
SDID 625, 694, 704, 723
SDIR 617, 694, 704, 723
SICDAR 473, 695, 707, 725
SICTR 695
SICTR 452, 707, 725
SIFCTR 695
SIFCTR 467, 707, 725
SIIER 465, 695, 707, 725
SIMDR 449, 695, 707, 725
SIRCR 458, 696, 708, 725
SIRDAR 472, 695, 707, 725
SIRDR 456, 695, 708, 725
SISCR 469, 695, 707, 725
SISTR 459, 695, 707, 725
SITCR 457, 696, 708, 725
SITDAR 470, 695, 707, 725
SITDR 455, 695, 708, 725
SPICR 474, 696, 708, 725
STBCR 225, 694, 705, 724
STBCR2 ..... 226, 694, 705, 724
STBCR3 ..... 227, 693, 704, 723
STBCR4 ..... 228, 694, 704, 723
TBRAR 288, 698, 716, 727
TDFAR 289, 698, 716, 727
TDLAR 272, 697, 714, 727
TFTR 284, 697, 715, 727
TLFRCR 250, 698, 718, 728
TPAUSER 252, 699, 719, 728
TRIMD 290, 698, 715, 727
TROCR 247, 698, 717, 727
TRSCER 281, 697, 714, 727
TSFRCR 249, 698, 718, 728
WTCNT 217, 694, 705, 724
WTCSR 217, 694, 705, 724
Relationship between
refresh requests and bus cycles 184
Reset 71, 663
Re-starting auto-negotiation 660
RISC-type 29
Round-robin mode 346

S

SCIF initialization
(asynchronous mode) 418
SCIF initialization
(synchronous mode) 428
Scrambling 650
SDRAM direct connection 157
SDRAM interface 157
Searching cache 57
Self-refreshing 183
Serial communication interface with
FIFO (SCIF) 377

Serial I/O with FIFO (SIOF) 445
Serial management interface (SMI) 635
Shift instructions 48
Signals relevant to PHY-IF 667
Single address mode 351
Single read 172
Single write 174
Sleep mode 229
SMI register 642
Software standby mode 230
SPI mode 502
Stack states after exception handling ends 79
State transition 51
Synchronous mode 427
System control instructions 49
System registers 26

T

TAP controller 626
The procedures of set up the external PHY LSI 690
The procedures of setting up the on-chip PHY 689
Transmit descriptor 0 (TD0) 293
Transmit descriptor 1 (TD1) 295
Transmit descriptor 2 (TD2) 295
Transmitting and receiving serial data simultaneously (synchronous mode) 434
Transmitting serial data (asynchronous mode) 420
Transmitting serial data (synchronous mode) 430
Trap instructions 76
Treatment of pins when PHY power supply is not used 668
Types of exception handling and priority 67
Types of exceptions triggered by instructions 76
Types of power-down modes 223

U

U memory 65
User break controller (UBC) 593
User break interrupt 99
User debugging interface (H-UDI) 615

W

Wait between access cycles 198
Watchdog timer (WDT) 215
Write access 59
Write-back buffer 59

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7619 Group

Publication Date:	Rev.1.00, Mar. 18, 2005
	Rev.6.00, Jul. 15, 2009
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

© 2009. 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 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 6.2

SH7619 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0237-0600