REJ09B0256-0100

Everywhere you imagine.

32

SH7763

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC Engine Family
SH-4A Series

R5S77630

Hardware Manual

Rev.1.00
Revision Date: Oct. 01, 2007

Renesas Technology
www.renesas.com

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 they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

5. Reading from/Writing to Reserved Bit of Each Register

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

Configuration of This Manual

This manual comprises the following items:

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

Preface

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

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

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the above users.

Notes on reading this manual:

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

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

Abbreviations

ALU	Arithmetic Logic Unit
ASID	Address Space Identifier
BGA	Ball Grid Array
CMT	Timer/Counter (Compare Match Timer)
CPG	Clock Pulse Generator
CPU	Central Processing Unit
DDR	Double Data Rate
DDRIF	DDR-SDRAM Interface
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
FIFO	First-In First-Out
FPU	Floating-point Unit
HAC	Audio Codec
H-UDI	User Debugging Interface
INTC	Interrupt Controller
JTAG	Joint Test Action Group
LBSC	Local Bus State Controller
LRAM	L Memory
LRU	Least Recently Used
LSB	Least Significant Bit
MMCIF	Multimedia Card Interface
MMU	Memory Management Unit

MSB	Most Significant Bit
PC	Program Counter
PCI	Peripheral Component Interconnect
PCIC	PCI (local bus) Controller
PFC	Pin Function Controller
RISC	Reduced Instruction Set Computer
RTC	Realtime Clock
SCIF	Serial Communication Interface with FIFO
SIOF	Serial Interface with FIFO
SSI	Serial Sound Interface
TAP	Test Access Port
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watchdog Timer

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

Contents

Section 1 Overview 1
1.1 Features of the SH7763 1
1.2 Block Diagram 13
1.3 Pin Arrangement 14

Section 2 Programming Model 37
2.1 Data Formats 37
2.2 Register Descriptions 38
2.2.1 Privileged Mode and Banks 38
2.2.2 General Registers 42
2.2.3 Floating-Point Registers 43
2.2.4 Control Registers 45
2.2.5 System Registers 47
2.3 Memory-Mapped Registers 51
2.4 Data Formats in Registers 52
2.5 Data Formats in Memory 52
2.6 Processing States 53
2.7 Usage Note 55
2.7.1 Notes on Self-Modified Codes 55

Section 3 Instruction Set 57
3.1 Execution Environment 57
3.2 Addressing Modes 59
3.3 Instruction Set 64

Section 4 Pipelining 79
4.1 Pipelines 79
4.2 Parallel-Executability 90
4.3 Issue Rates and Execution Cycles 94

Section 5 Exception Handling 105
5.1 Summary of Exception Handling 105
5.2 Register Descriptions 105
5.2.1 TRAPA Exception Register (TRA) 106
5.2.2 Exception Event Register (EXPEVT) 107
5.2.3 Interrupt Event Register (INTEVT) 108
5.3 Exception Handling Functions 109

5.3.1 Exception Handling Flow 109
5.3.2 Exception Handling Vector Addresses 109
5.4 Exception Types and Priorities 110
5.5 Exception Flow 112
5.5.1 Exception Flow 112
5.5.2 Exception Source Acceptance 114
5.5.3 Exception Requests and BL Bit 115
5.5.4 Return from Exception Handling 115
5.6 Description of Exceptions 116
5.6.1 Resets 116
5.6.2 General Exceptions 118
5.6.3 Interrupts 132
5.6.4 Priority Order with Multiple Exceptions 133
5.7 Usage Notes 135

Section 6 Memory Management Unit (MMU) 137
6.1 Overview of MMU 137
6.1.1 Address Spaces 140
6.2 Register Descriptions 146
6.2.1 Page Table Entry High Register (PTEH) 147
6.2.2 Page Table Entry Low Register (PTEL) 148
6.2.3 Translation Table Base Register (TTB) 149
6.2.4 TLB Exception Address Register (TEA) 149
6.2.5 MMU Control Register (MMUCR) 149
6.2.6 Physical Address Space Control Register (PASCR) 152
6.2.7 Instruction Re-Fetch Inhibit Control Register (IRMCR) 153
6.3 TLB Functions 156
6.3.1 Unified TLB (UTLB) Configuration 156
6.3.2 Instruction TLB (ITLB) Configuration 159
6.3.3 Address Translation Method 160
6.4 MMU Functions 162
6.4.1 MMU Hardware Management 162
6.4.2 MMU Software Management 162
6.4.3 MMU Instruction (LDTLB) 163
6.4.4 Hardware ITLB Miss Handling 164
6.4.5 Avoiding Synonym Problems 165
6.5 MMU Exceptions 166
6.5.1 Instruction TLB Multiple Hit Exception 166
6.5.2 Instruction TLB Miss Exception 167
6.5.3 Instruction TLB Protection Violation Exception 168

6.5.4 Data TLB Multiple Hit Exception 169
6.5.5 Data TLB Miss Exception 169
6.5.6 Data TLB Protection Violation Exception 170
6.5.7 Initial Page Write Exception 171
6.6 Memory-Mapped TLB Configuration 172
6.6.1 ITLB Address Array 173
6.6.2 ITLB Data Array 174
6.6.3 UTLB Address Array 175
6.6.4 UTLB Data Array 176
6.7 32-Bit Address Extended Mode 177
6.7.1 Overview of 32-Bit Address Extended Mode 178
6.7.2 Transition to 32-Bit Address Extended Mode 178
6.7.3 Privileged Space Mapping Buffer (PMB) Configuration 179
6.7.4 PMB Function 181
6.7.5 Memory-Mapped PMB Configuration 182
6.7.6 Notes on Using 32-Bit Address Extended Mode 184
6.8 Usage Notes 186

Section 7 Caches 187
7.1 Features 187
7.2 Register Descriptions 190
7.2.1 Cache Control Register (CCR) 191
7.2.2 Queue Address Control Register 0 (QACR0) 193
7.2.3 Queue Address Control Register 1 (QACR1) 194
7.2.4 On-Chip Memory Control Register (RAMCR) 195
7.3 Operand Cache Operation 197
7.3.1 Read Operation 197
7.3.2 Prefetch Operation 198
7.3.3 Write Operation 199
7.3.4 Write-Back Buffer 201
7.3.5 Write-Through Buffer 201
7.3.6 OC Two-Way Mode 201
7.4 Instruction Cache Operation 202
7.4.1 Read Operation 202
7.4.2 Prefetch Operation 203
7.4.3 IC Two-Way Mode 203
7.5 Cache Operation Instruction 204
7.5.1 Coherency between Cache and External Memory 204
7.5.2 Prefetch Operation 205
7.6 Memory-Mapped Cache Configuration 206

7.6.1 IC Address Array 206
7.6.2 IC Data Array 208
7.6.3 OC Address Array 209
7.6.4 OC Data Array 210
7.7 Store Queues 212
7.7.1 SQ Configuration 212
7.7.2 Writing to SQ 212
7.7.3 Transfer to External Memory 213
7.7.4 Determination of SQ Access Exception 214
7.7.5 Reading from SQ 214
7.8 Notes on Using 32-Bit Address Extended Mode 215

Section 8 L Memory 217
8.1 Features 217
8.2 Register Descriptions 218
8.2.1 On-Chip Memory Control Register (RAMCR) 220
8.2.2 L Memory Transfer Source Address Register 0 (LSA0) 221
8.2.3 L Memory Transfer Source Address Register 1 (LSA1) 223
8.2.4 L Memory Transfer Destination Address Register 0 (LDA0) 225
8.2.5 L Memory Transfer Destination Address Register 1 (LDA1) 227
8.3 Operation 229
8.3.1 Access from the CPU and FPU 229
8.3.2 Access from the SuperHyway Bus Master Module 229
8.3.3 Block Transfer 229
8.4 L Memory Protective Functions 231
8.5 Usage Notes 232
8.5.1 Page Conflict 232
8.5.2 L Memory Coherency 232
8.5.3 Sleep Mode 232
8.6 Note on Using 32-Bit Address Extended Mode 232

Section 9 Interrupt Controller (INTC) 233
9.1 Features 233
9.1.1 Interrupt Method 235
9.1.2 Interrupt Types in INTC 236
9.2 Input/Output Pins 240
9.3 Register Descriptions 241
9.3.1 Interrupt Control Register 0 (ICR0) 246
9.3.2 Interrupt Control Register 1 (ICR1) 248
9.3.3 Interrupt Priority Register (INTPRI) 249

9.3.4 Interrupt Source Register (INTREQ) 250
9.3.5 Interrupt Mask Register 0 (INTMSK0) 251
9.3.6 Interrupt mask register 1 (INTMSK1) 253
9.3.7 Interrupt mask register 2 (INTMSK2) 254
9.3.8 Interrupt Mask Clear Register 0 (INTMSKCLR0) 257
9.3.9 Interrupt mask clear register 1 (INTMSKCLR1) 259
9.3.10 Interrupt mask clear register 2 (INTMSKCLR2) 260
9.3.11 NMI Flag Control Register (NMIFCR) 263
9.3.12 User Interrupt Mask Level Register (USERIMASK) 264
9.3.13 On-chip module Interrupt Priority Registers (INT2PRI0 to INT2PRI13) 266
9.3.14 Interrupt Source Register 0 (Mask State is not affected) (INT2A0) 268
9.3.15 Interrupt Source Register 01 (Mask State is not affected) (INT2A01) 269
9.3.16 Interrupt Source Register (Mask State is affected) (INT2A1) 272
9.3.17 Interrupt Source Register 11 (Mask State is affected) (INT2A11) 274
9.3.18 Interrupt Mask Register (INT2MSKR) 276
9.3.19 Interrupt Mask Register 1 (INT2MSKR1) 277
9.3.20 Interrupt Mask Clear Register (INT2MSKCR) 279
9.3.21 Interrupt Mask Clear Register 1 (INT2MSKCR1) 281
9.3.22 On-chip Module Interrupt Source Registers (INT2B0 to INT2B7 and INT2B9 to INT2B11) 283
9.3.23 GPIO Interrupt Set Register (INT2GPIC) 289
9.4 Interrupt Sources 292
9.4.1 NMI Interrupt 292
9.4.2 IRQ Interrupts 292
9.4.3 IRL Interrupts 293
9.4.4 On-chip Module Interrupts 295
9.4.5 Interrupt Priority Level of On-chip Module Interrupts 295
9.4.6 Interrupt Exception Handling and Priority 296
9.5 Operation 305
9.5.1 Interrupt Sequence 305
9.5.2 Multiple Interrupts 307
9.5.3 Interrupt Masking by MAI Bit 307
9.6 Interrupt Response Time 308
9.7 Usage Notes 309
9.7.1 Example of Interrupt Handling Routine for Level-Encoded IRL and Level-Sensed IRQ 309
9.7.2 Notes on Setting IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ Pin Function 310
9.7.3 To Clear IRQ and IRL Interrupt Requests 310

Section 10 SuperHyway Bus Bridge (SBR) 313
10.1 Features 313
10.2 Register Descriptions 314
10.2.1 Bus Arbitration Priority Level Setting Register (SBRIVCLV) 315
10.2.2 SuperHyway Bus Priority Control Resister (PRPRICR) 316
10.3 Operation 317
10.3.1 SuperHyway Bus Interface 317
10.3.2 Bus Arbitration 317

Section 11 Local Bus State Controller (LBSC) 319
11.1 Features 319
11.2 Input/Output Pins 322
11.3 Area Overview 324
11.3.1 Space Divisions 324
11.3.2 Memory Bus Width 328
11.3.3 Data Alignment 329
11.3.4 PCMCIA Support 329
11.4 Register Descriptions 333
11.4.1 Memory Address Map Select Register (MMSELR) 334
11.4.2 Bus Control Register (BCR) 336
11.4.3 CSn Bus Control Register (CSnBCR) 340
11.4.4 CSn Wait Control Register (CSnWCR) 346
11.4.5 CSn PCMCIA Control Register (CSnPCR) 351
11.5 Operation 356
11.5.1 Endian/Access Size and Data Alignment 356
11.5.2 Areas 361
11.5.3 SRAM interface 365
11.5.4 Burst ROM Interface 373
11.5.5 PCMCIA Interface 375
11.5.6 MPX Interface 386
11.5.7 Byte Control SRAM Interface 399
11.5.8 Wait Cycles between Accesses 403
11.5.9 Bus Arbitration 405
11.5.10 Master Mode 407
11.5.11 Cooperation between Master and Slave 408

Section 12 DDR-SDRAM Interface (DDRIF) 409
12.1 Features 409
12.2 Input/Output Pins 411

12.3 Data Conversion 412
12.3.1 Data Alignment 412
12.3.2 Data Alignment in Peripheral Modules 414
12.4 Register Descriptions 415
12.4.1 Memory Interface Mode Register (MIM) 417
12.4.2 DDR-SDRAM Control Register (SCR) 421
12.4.3 DDR-SDRAM Timing Register (STR) 423
12.4.4 DDR-SDRAM Row Attribute Register (SDR) 426
12.4.5 DDR-SDRAM Mode Register (SDMR) 427
12.4.6 DDR-SDRAM Back-up Register (DBK) 429
12.5 Operation 430
12.5.1 DDR-SDRAM Access 430
12.5.2 DDR-SDRAM Initialization Sequence 430
12.5.3 Supported DDR-SDRAM Commands 432
12.5.4 DDR-SDRAM Access Mode 433
12.5.5 Power-Down Modes 433
12.5.6 Registers that Set DDR-SDRAM Timing Restrictions 434
12.5.7 Operating Frequency 435
12.5.8 Note on Clock Stop 435
12.5.9 Using SCR to Issue REFA Commands (Outside the Initialization Sequence) 435
12.5.10 Note on Timing of Connected DDR-SDRAM 435
12.5.11 Note on Setting Auto-Refresh Interval 436
12.5.12 Address Multiplexing 436
12.5.13 DDR-SDRAM Access Arbitration 436
12.5.14 Coherency When Accessing DDR-SDRAM 437
12.6 DDRIF Basic Timing 438

Section 13 PCI Controller (PCIC) 449
13.1 Features 449
13.2 Input/Output Pins 452
13.3 Register Descriptions 455
13.3.1 PCIC Enable Control Register (PCIECR) 460
13.3.2 Configuration Registers 461
13.3.3 Local Register 486
13.4 Operation 527
13.4.1 Supported PCI Commands 527
13.4.2 PCIC Initialization 528
13.4.3 Master Access 529
13.4.4 Target Access 537
13.4.5 Host Bus Bridge Mode 546

13.4.6 Normal mode 549
13.4.7 Power Management 549
13.4.8 PCI Local Bus Basic Interface 550
13.5 Usage Notes 562
13.5.1 Notes on PCIC Target Reading 562
13.5.2 Notes on Host Mode 562

Section 14 Direct Memory Access Controller (DMAC) 565
14.1 Features 565
14.2 Input/Output Pins 567
14.3 Register Descriptions 569
14.3.1 DMA Source Address Registers (SAR0 to SAR5) 572
14.3.2 DMA Source Address Registers (SARB0 to SARB3) 573
14.3.3 DMA Destination Address Registers (DAR0 to DAR5) 573
14.3.4 DMA Destination Address Registers (DARB0 to DARB3) 574
14.3.5 DMA Transfer Count Registers (TCR0 to TCR5) 574
14.3.6 DMA Transfer Count Registers (TCRB0 to TCRB3) 575
14.3.7 DMA Channel Control Registers (CHCR0 to CHCR5) 576
14.3.8 DMA Operation Register (DMAOR) 584
14.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2) 587
14.4 Operation 591
14.4.1 DMA Transfer Requests 591
14.4.2 Channel Priority 595
14.4.3 DMA Transfer Types 598
14.4.4 DMA Transfer Flow 606
14.4.5 Repeat Mode Transfer 608
14.4.6 Reload Mode Transfer 609
14.4.7 DREQ Pin Sampling Timing 610
14.5 Usage Notes 614
14.5.1 Module Stop 614
14.5.2 Address Error 614
14.5.3 Notes on Burst Mode Transfer 614
14.5.4 DACK and TEND Output Divisions 615
14.5.5 $\overline{CS}$ Output Settings and Transfer Size Larger than External Bus Width 615
14.5.6 DACK and TEND Assertion and DREQ Sampling 615
14.5.7 DMA Transfer to DMAC Prohibited 619
14.5.8 NMI Interrupt 619

Section 15 External CPU Interface (EXCPU) 621
15.1 Features 621
15.2 Input/Output Pins 622
15.3 Register Descriptions 623
15.3.1 External CPU Control Register (EXCCTRL) 624
15.3.2 External CPU Memory Space Select Register (EXCMSETR) 625
15.3.3 External CPU Interrupt Output Control Register (EXCINOR) 626
15.4 Operation 627

Section 16 Clock Pulse Generator (CPG) 633
16.1 Features 633
16.2 Input/Output Pins 636
16.3 Clock Operating Mode 637
16.4 Register Descriptions 638
16.4.1 Frequency Control Register (FRQCR) 639
16.4.2 PLL Control Register (PLLCR) 641
16.5 Notes on Board Design 642

Section 17 Watchdog Timer and Reset (WDT) 645
17.1 Features 645
17.2 Input/Output Pins 647
17.3 Register Descriptions 648
17.3.1 Watchdog Timer Stop Time Register (WDTST) 649
17.3.2 Watchdog Timer Control/Status Register (WDTCSR) 650
17.3.3 Watchdog timer Base Stop Time Register (WDTBST) 652
17.3.4 Watchdog Timer Counter (WDTCNT) 653
17.3.5 Watchdog Timer Base Counter (WDTBCNT) 653
17.4 Operation 654
17.4.1 Reset request 654
17.4.2 Using watchdog timer mode 655
17.4.3 Using Interval timer mode 656
17.4.4 Time for WDT Overflow 656
17.4.5 Clearing WDT Counter 658
17.5 Status Pin Change Timing during Reset 659
17.5.1 Power-On Reset by PRESET 659
17.5.2 Power-On Reset by Watchdog Timer Overflow 662
17.5.3 Manual Reset by Watchdog Timer Overflow 664

Section 18 Power-Down Mode 667
18.1 Features 667
18.1.1 Types of Power-Down Modes 667
18.2 Input/Output Pins 669
18.3 Register Descriptions 670
18.3.1 Standby Control Register (STBCR) 671
18.3.2 Module Stop Register 0 (MSTPCR0) 672
18.3.3 Module Stop Register 1 (MSTPCR1) 673
18.4 Sleep Mode 678
18.4.1 Transition to Sleep Mode 678
18.4.2 Canceling Sleep Mode 678
18.5 Software Standby Mode 679
18.5.1 Transition to Software Standby Mode 679
18.5.2 Canceling Software Standby Mode 679
18.6 Module Standby Mode 680
18.6.1 Transition to Module Standby Mode 680
18.6.2 Canceling Module Standby Mode 680
18.7 DDR-SDRAM Power Supply Backup 681
18.7.1 Control of Self-Refresh and Initialization 681
18.7.2 DDR-SDRAM Backup Sequence when Turning Off System Power Supply 682
18.8 RTC Power Supply Backup 684
18.8.1 Transition to RTC Power Supply Backup Mode 684
18.8.2 Canceling RTC Power Supply Backup Mode 684
18.9 STATUS Pin Signal Change Timing 685
18.9.1 Timing at Reset 685
18.9.2 Timing at Sleep Mode Cancellation 685

Section 19 Timer Unit (TMU) 687
19.1 Features 687
19.2 Input/Output Pins 689
19.3 Register Descriptions 690
19.3.1 Timer Output Control Register (TOCR) 692
19.3.2 Timer Start Register (TSTR) 693
19.3.3 Timer Constant Register (TCORn) (n = 0 to 5) 695
19.3.4 Timer Counter (TCNTn) (n = 0 to 5) 695
19.3.5 Timer Control Registers (TCRn) (n = 0 to 5) 696
19.3.6 Input Capture Register 2 (TCPR2) 698
19.4 Operation 699
19.4.1 Counter Operation 699

19.4.2 Input Capture Function 703
19.5 Interrupts 704
19.6 Usage Notes 705
19.6.1 Register Writes 705
19.6.2 Reading from TCNT 705
19.6.3 External Clock Frequency 705

Section 20 16-Bit Timer Pulse Unit (TPU) 707
20.1 Features 707
20.2 Input/Output Pins 710
20.3 Register Descriptions 711
20.3.1 Timer Control Registers (TCR) 715
20.3.2 Timer Mode Registers (TMDR) 719
20.3.3 Timer I/O Control Registers (TIOR) 721
20.3.4 Timer Interrupt Enable Registers (TIER) 723
20.3.5 Timer Status Registers (TSR) 725
20.3.6 Timer Counters (TCNT) 727
20.3.7 Timer General Registers (TGR) 727
20.3.8 Timer Start Register (TSTR) 728
20.4 Operation 729
20.4.1 Overview 729
20.4.2 Basic Functions 730
20.4.3 Buffer Operation 734
20.4.4 PWM Modes 737
20.4.5 Phase Counting Mode 740
20.5 Usage Notes 746

Section 21 Compare Match Timer (CMT) 747
21.1 Features 747
21.2 Register Descriptions 749
21.2.1 Compare Match Timer Start Register (CMSTR) 751
21.2.2 Compare Match Timer Control/Status Register (CMCSR) 752
21.2.3 Compare Match Timer Counter (CMCNT) 754
21.2.4 Compare Match Timer Constant Register (CMCOR) 754
21.3 Operation 755
21.3.1 Counter Operation 755
21.3.2 Counter Size 756
21.3.3 Timing for Counting by CMCNT 756
21.3.4 DMA Transfer Requests and Internal Interrupt Requests to CPU 757
21.3.5 Compare Match Flag Set Timing (All Channels) 757

Section 22 Realtime Clock (RTC) 759
22.1 Features 759
22.1.1 Block Diagram 760
22.2 Input/Output Pins 761
22.3 Register Descriptions 762
22.4 Register Descriptions 764
22.4.1 64 Hz Counter (R64CNT) 764
22.4.2 Second Counter (RSECCNT) 764
22.4.3 Minute Counter (RMINCNT) 765
22.4.4 Hour Counter (RHRCNT) 765
22.4.5 Day-of-Week Counter (RWKCNT) 766
22.4.6 Day Counter (RDAYCNT) 767
22.4.7 Month Counter (RMONCNT) 768
22.4.8 Year Counter (RYRCNT) 768
22.4.9 Second Alarm Register (RSECAR) 769
22.4.10 Minute Alarm Register (RMINAR) 769
22.4.11 Hour Alarm Register (RHRAR) 770
22.4.12 Day-of-Week Alarm Register (RWKAR) 770
22.4.13 Day Alarm Register (RDAYAR) 771
22.4.14 Month Alarm Register (RMONAR) 772
22.4.15 RTC Control Register 1 (RCR1) 772
22.4.16 RTC Control Register 2 (RCR2) 774
22.4.17 RTC Control Register (RCR3) and Year-Alarm Register (RYRAR) 777
22.5 Operation 778
22.5.1 Time Setting Procedures 778
22.5.2 Time Reading Procedures 779
22.5.3 Alarm Function 780
22.6 Interrupts 781
22.7 Usage Notes 781
22.7.1 Register Initialization 781
22.7.2 Crystal Oscillator Circuit 781
22.7.3 Interrupt source and request generating order 782

Section 23 Gigabit Ethernet Controller (GETHER) 783
23.1 Features 783
23.2 Input/Output Pins 785
23.3 Register Descriptions 790
23.3.1 Software Reset Register (ARSTR) 807
23.3.2 E-MAC Mode Register (ECMR) 808
23.3.3 E-MAC Status Register (ECSR) 814

23.3.4 E-MAC Interrupt Permission Register (ECSIPR) 816
23.3.5 PHY Interface Register (PIR) 817
23.3.6 MAC Address High Register (MAHR) 818
23.3.7 MAC Address Low Register (MALR) 819
23.3.8 Receive Frame Length Register (RFLR) 820
23.3.9 PHY Status Register (PSR) 821
23.3.10 PHY_INT Polarity Register (PIPR) 822
23.3.11 Transmit Retry Over Counter Register (TROCR) 823
23.3.12 Delayed Collision Detect Counter Register (CDCR) 824
23.3.13 Lost Carrier Counter Register (LCCR) 825
23.3.14 CRC Error Frame Receive Counter Register (CEFCR) 826
23.3.15 Frame Receive Error Counter Register (FRECR) 827
23.3.16 Too-Short Frame Receive Counter Register (TSFRCR) 828
23.3.17 Too-Long Frame Receive Counter Register (TLFRCR) 829
23.3.18 Residual-Bit Frame Receive Counter Register (RFCR) 830
23.3.19 Carrier Extension Loss Counter Register (CERCR) 831
23.3.20 Carrier Extension Error Counter Register (CEECR) 832
23.3.21 Multicast Address Frame Receive Counter Register (MAFCR) 833
23.3.22 Automatic PAUSE Frame Register (APR) 834
23.3.23 Manual PAUSE Frame Register (MPR) 835
23.3.24 Automatic PAUSE Frame Retransmit Count Register (TPAUSER) 836
23.3.25 PAUSE Frame Transmit Counter Register (PFTCR) 837
23.3.26 PAUSE Frame Receive Counter Register (PFRCR) 838
23.3.27 GETHER Mode Register (GECMR) 839
23.3.28 Burst Cycle Count Upper-Limit Register (BCULR) 840
23.3.29 TSU Counter Reset Register (TSU_CTRST) 841
23.3.30 Relay Enable Register (Port 0 to 1) (TSU_FWEN0) 842
23.3.31 Relay Enable Register (Port 1 to 0) (TSU_FWEN1) 843
23.3.32 Relay FIFO Size Select Register (TSU_FCM) 844
23.3.33 Relay FIFO Overflow Alert Set Register (Port 0) (TSU_BSYSL0) 845
23.3.34 Relay FIFO Overflow Alert Set Register (Port 1) (TSU_BSYSL1) 847
23.3.35 Transmit/Relay Priority Control Mode Register (Port 0) (TSU_PRISL0) 849
23.3.36 Transmit/Relay Priority Control Mode Register (Port 1) (TSU_PRISL1) 851
23.3.37 Receive/Relay Function Set Register (Port 0 to 1) (TSU_FWSL0) 853
23.3.38 Receive/Relay Function Set Register (Port 1 to 0) (TSU_FWSL1) 855
23.3.39 Relay Function Set Register (Common) (TSU_FWSLC) 857
23.3.40 Qtag Addition/Deletion Set Register (Port 0 to 1) (TSU_QTAG0) 859
23.3.41 Qtag Addition/Deletion Set Register (Port 1 to 0) (TSU_QTAG1) 860
23.3.42 Relay Status Register (TSU_FWSR) 861
23.3.43 Relay Status Interrupt Mask Register (TSU_FWINMK) 864

23.3.44 Added Qtag Value Set Register (Port 0 to 1) (TSU_ADQT0) 867
23.3.45 Added Qtag Value Set Register (Port 1 to 0) (TSU_ADQT1) 868
23.3.46 VLANtag Set Register (Port 0) (TSU_VTAG0) 869
23.3.47 VLANtag Set Register (Port 1) (TSU_VTAG1) 870
23.3.48 CAM Entry Table Busy Register (TSU_ADSBSY) 871
23.3.49 CAM Entry Table Enable Register (TSU_TEN) 872
23.3.50 CAM Entry Table POST1 Register (TSU_POST1) 877
23.3.51 CAM Entry Table POST2 Register (TSU_POST2) 880
23.3.52 CAM Entry Table POST3 Register (TSU_POST3) 883
23.3.53 CAM Entry Table POST4 Register (TSU_POST4) 886
23.3.54 CAM Entry Table 0H to 31H Registers (TSU_ADRH0 to TSU_ADRH31) 889
23.3.55 CAM Entry Table 0L to 31L Registers (TSU_ADRL0 to TSU_ADRL31) 890
23.3.56 Transmit Frame Counter Register (Port 0) (Normal Transmission Only) (TXNLCR0) 891
23.3.57 Transmit Frame Counter Register (Port 0) (Normal and Erroneous Transmission) (TXALCR0) 892
23.3.58 Receive Frame Counter Register (Port 0) (Normal Reception Only) (RXNLCR0) 893
23.3.59 Receive Frame Counter Register (Port 0) (Normal and Erroneous Reception) (RXALCR0) 894
23.3.60 Relay Frame Counter Register (Port 1 to 0) (Normal Relay Only) (FWNLCR0) 895
23.3.61 Relay Frame Counter Register (Port 1 to 0) (Normal and Erroneous Transmission) (FWALCR0) 896
23.3.62 Transmit Frame Counter Register (Port 1) (Normal Transmission Only) (TXNLCR1) 897
23.3.63 Transmit Frame Counter Register (Port 1) (Normal and Erroneous Transmission) (TXALCR1) 898
23.3.64 Receive Frame Counter Register (Port 1) (Normal Reception Only) (RXNLCR1) 899
23.3.65 Receive Frame Counter Register (Port 1) (Normal and Erroneous Reception) (RXALCR1) 900
23.3.66 Relay Frame Counter Register (Port 0 to 1) (Normal Relay Only) (FWNLCR1) 901
23.3.67 Relay Frame Counter Register (Port 0 to 1) (Normal and Erroneous Transmission) (FWALCR1) 902
23.3.68 E-DMAC Start Register (EDSR) 903
23.3.69 E-DMAC Mode Register (EDMR) 904
23.3.70 E-DMAC Transmit Request Register (EDTRR) 906
23.3.71 E-DMAC Receive Request Register (EDRRR) 907

23.3.72 Transmit Descriptor List Start Address Register (TDLAR) 908
23.3.73 Receive Descriptor List Start Address Register (RDLAR) 909
23.3.74 E-MAC/E-DMAC Status Register (EESR) 910
23.3.75 E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR) 916
23.3.76 Transmit/Receive Status Copy Enable Register (TRSCER) 920
23.3.77 Receive Missed-Frame Counter Register (RMFCR) 923
23.3.78 Transmit FIFO Threshold Register (TFTR) 924
23.3.79 FIFO Depth Register (FDR) 925
23.3.80 Receiving Method Control Register (RMCR) 926
23.3.81 Receive Descriptor Fetch Address Register (RDFAR) 927
23.3.82 Receive Descriptor Finished Address Register (RDFXR) 928
23.3.83 Receive Descriptor Final Flag Register (RDFFR) 929
23.3.84 Transmit Descriptor Fetch Address Register (TDFAR) 930
23.3.85 Transmit Descriptor Finished Address Register (TDFXR) 931
23.3.86 Transmit Descriptor Final Flag Register (TDFFR) 932
23.3.87 Overflow Alert FIFO Threshold Register (FCFTR) 933
23.3.88 Receive Data Padding Insert Register (RPADIR) 935
23.4 Operation 936
23.4.1 Descriptors and Descriptor List 939
23.4.2 Transmission 956
23.4.3 Reception 962
23.4.4 Relay 968
23.4.5 CAM Function 969
23.4.6 Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/Multi-Descriptor) 971
23.4.7 Padding Insertion in Receive Data 973
23.4.8 Interrupt Processing 974
23.4.9 Activation Procedure 978
23.4.10 Flow Control 980
23.4.11 Magic Packet Detection 981
23.4.12 Direction for IEEE802.1Q Qtag 982
23.5 Connection to PHY-LSI 984
23.5.1 MII Frame Transmission/Reception Timing 984
23.5.2 GMII/MII Frame Reception Timing 986
23.5.3 RMII Frame Transmission/Reception Timing 988
23.5.4 Accessing MII Registers 989
23.5.5 Mll-RMII Interface Conversion 991
23.6 Usage Notes 993
23.6.1 Checksum Calculation of Ethernet Frames 993
23.6.2 Notes on TSU Use 993

Section 24 IP Security Accelerator (SECURITY) 995

Section 25 Stream Interface (STIF) 997
25.1 Features 997
25.2 Input/Output Pins 999
25.3 Register Descriptions 1000
25.3.1 Mode Registers 0, 1 (STIMDR0, STIMDR1) 1002
25.3.2 Control Registers 0, 1 (STICR0, STICR1) 1006
25.3.3 Interrupt Status Registers 0, 1 (STIISR0, STIISR1) 1007
25.3.4 Interrupt Enable Registers 0, 1 (STIIER0, STIIER1) 1009
25.3.5 Time Stamp Counter Registers 0, 1 (STITSC0, STITSC1) 1011
25.3.6 Transmit/Receive Packet Count Registers 0, 1 (STIPNR0, STIPNR1) 1012
25.3.7 Transmit/Receive Packet Counter Registers 0, 1 (STIPCR0, STIPCR1) 1013
25.3.8 Transmit/Receive FIFO Data Registers 0, 1 (STIFIFO0, STIFIFO1) 1014
25.3.9 Operation 1015
25.3.10 External Memory Configuration for Stream Data Transmission/Reception 1015
25.3.11 Stream Data Receive Operation 1016
25.3.12 Stream Data Transmit Operation 1020

Section 26 I^2C Bus Interface (IIC) 1025
26.1 Features 1025
26.2 Input/Output Pins 1026
26.3 Register Descriptions 1026
26.3.1 Slave Control Register (ICSCR) 1029
26.3.2 Slave Status Register (ICSSR) 1031
26.3.3 Slave Interrupt Enable Register (ICSIER) 1034
26.3.4 Slave Address Register (ICSAR) 1035
26.3.5 Master Control Register (ICMCR) 1036
26.3.6 Master Status Register (ICMSR) 1038
26.3.7 Master Interrupt Enable Register (ICMIER) 1040
26.3.8 Master Address Register (ICMAR) 1041
26.3.9 Clock Control Register (ICCCR) 1041
26.3.10 Receive and Transmit Data Registers (ICRXD and ICTXD) 1043
26.4 Operations 1044
26.4.1 Data and Clock Filters 1044
26.4.2 Clock Generator 1044
26.4.3 Master/Slave Interfaces 1044
26.4.4 Software Status Interlocking 1044
26.4.5 I^2C Bus Data Format 1046
26.4.6 7-Bit Address Format 1047

26.4.7 10-Bit Address Format 1048
26.4.8 Master Transmit Operation 1050
26.4.9 Master Receive Operation 1052
26.5 Programming Examples 1054
26.5.1 Master Transmitter 1054
26.5.2 Master Receiver 1055
26.5.3 Master Transmitter—Restart—Master Receiver 1056

Section 27 Serial Communication Interface with FIFO (SCIF) 1059
27.1 Features 1059
27.2 Input/Output Pins 1065
27.3 Register Descriptions 1066
27.3.1 Receive Shift Register (SCRSR) 1068
27.3.2 Receive FIFO Data Register (SCFRDR) 1068
27.3.3 Transmit Shift Register (SCTSR) 1069
27.3.4 Transmit FIFO Data Register (SCFTDR) 1069
27.3.5 Serial Mode Register (SCSMR) 1070
27.3.6 Serial Control Register (SCSCR) 1073
27.3.7 Serial Status Register (SCFSR) 1077
27.3.8 Bit Rate Register (SCBRR) 1083
27.3.9 FIFO Control Register (SCFCR) 1084
27.3.10 Transmit FIFO Data Count Register (SCTFDR) 1086
27.3.11 Receive FIFO Data Count Register (SCRFDR) 1086
27.3.12 Serial Port Register (SCSPTR) 1087
27.3.13 Line Status Register (SCLSR) 1090
27.3.14 Serial Error Register (SCRER) 1091
27.4 Operation 1092
27.4.1 Overview 1092
27.4.2 Operation in Asynchronous Mode 1094
27.4.3 Operation in Clocked Synchronous Mode 1104
27.5 SCIF Interrupt Sources and the DMAC 1113
27.6 Usage Notes 1115

Section 28 Serial Communication Interface with FIFO/IrDA Interface (SCIF/IrDA) 1119
28.1 Features 1119
28.2 Input/Output Pins 1123
28.3 Register Descriptions 1124
28.3.1 Receive Shift Register (SCRSR) 1126
28.3.2 Receive FIFO Data Register (SCFRDR) 1126

28.3.3 Transmit Shift Register (SCTSR) 1127
28.3.4 Transmit FIFO Data Register (SCFTDR) 1127
28.3.5 Serial Mode Register (SCSMR) 1128
28.3.6 Serial Control Register (SCSCR) 1131
28.3.7 Serial Status Register (SCFSR) 1135
28.3.8 Bit Rate Register (SCBRR) 1141
28.3.9 FIFO Control Register (SCFCR) 1142
28.3.10 FIFO Data Count Register (SCFDR) 1144
28.3.11 Serial Port Register (SCSPTR) 1145
28.3.12 Line Status Register (SCLSR) 1147
28.3.13 BRG Frequency Division Register (BRGDL2) 1148
28.3.14 BRG Clock Select Register (BRGCKS2) 1149
28.3.15 IrDA Serial Mode Register (SCSMRIR) 1150
28.4 Operation 1151
28.4.1 Overview 1151
28.4.2 Operation in Asynchronous Mode 1155
28.4.3 Operation in Clocked Synchronous Mode 1165
28.4.4 SCIF Interrupt Sources and the DMAC 1174
28.4.5 Usage Notes 1176
28.5 Infrared Data Communication Interface 1179
28.5.1 Infrared Data Communication Format 1179
28.5.2 Operation of Infrared Data Communication Interface 1180
28.6 Baud Rate Generator for External Clock (BRG) 1181
28.6.1 BRG Block Diagram 1181
28.6.2 Restrictions on the BRG 1182

Section 29 Serial I/O with FIFO (SIOF) 1185
29.1 Features 1185
29.2 Input/Output Pins 1187
29.3 Register Descriptions 1188
29.3.1 Mode Register (SIMDR) 1192
29.3.2 Clock Select Register (SISCR) 1194
29.3.3 Control Register (SICTR) 1196
29.3.4 Transmit Data Register (SITDR) 1199
29.3.5 Receive Data Register (SIRDR) 1200
29.3.6 Transmit Control Data Register (SITCR) 1201
29.3.7 Receive Control Data Register (SIRCR) 1202
29.3.8 Status Register (SISTR) 1203
29.3.9 Interrupt Enable Register (SIIER) 1209
29.3.10 FIFO Control Register (SIFCTR) 1211

29.3.11 Transmit Data Assign Register (SITDAR) 1213
29.3.12 Receive Data Assign Register (SIRDAR) 1214
29.3.13 Control Data Assign Register (SICDAR) 1215
29.4 Operation 1217
29.4.1 Serial Clocks 1217
29.4.2 Serial Timing 1218
29.4.3 Transfer Data Format 1220
29.4.4 Register Allocation of Transfer Data 1222
29.4.5 Control Data Interface 1225
29.4.6 FIFO 1227
29.4.7 Transmit and Receive Procedures 1229
29.4.8 Interrupts 1234
29.4.9 Transmit and Receive Timing 1236

Section 30 SIM Card Module (SIM) 1241
30.1 Features 1241
30.2 Input/Output Pins 1243
30.3 Register Descriptions 1244
30.3.1 Serial Mode Register (SCSMR) 1246
30.3.2 Bit Rate Register (SCBRR) 1247
30.3.3 Serial Control Register (SCSCR) 1248
30.3.4 Transmit Shift Register (SCTSR) 1250
30.3.5 Transmit Data Register (SCTDR) 1251
30.3.6 Serial Status Register (SCSSR) 1252
30.3.7 Receive Shift Register (SCRSR) 1258
30.3.8 Receive Data Register (SCRDR) 1258
30.3.9 Smart Card Mode Register (SCSCMR) 1259
30.3.10 Serial Control 2 Register (SCSC2R) 1261
30.3.11 Guard Extension Register (SCGRD) 1262
30.3.12 Wait Time Register (SCWAIT) 1263
30.3.13 Sampling Register (SCSMPL) 1264
30.4 Operation 1265
30.4.1 Overview 1265
30.4.2 Data Format 1266
30.4.3 Register Settings 1267
30.4.4 Clocks 1269
30.4.5 Data Transmit/Receive Operation 1270
30.5 Usage Notes 1278

Section 31 Multimedia Card Interface (MMCIF) 1285
31.1 Features 1285
31.2 Input/Output Pins 1287
31.3 Register Descriptions 1288
31.3.1 Command Type Register (CMDTYR) 1292
31.3.2 Response Type Register (RSPTYR) 1293
31.3.3 Transfer Byte Number Count Register (TBCR) 1297
31.3.4 Transfer Block Number Counter (TBNCR) 1298
31.3.5 Command Registers 0 to 5 (CMDR0 to CMDR5) 1298
31.3.6 Response Registers 0 to 16, D (RSPR0 to RSPR16, RSPRD) 1300
31.3.7 Command Start Register (CMDSTRT) 1302
31.3.8 Operation Control Register (OPCR) 1304
31.3.9 Command Timeout Control Register (CTOCR) 1306
31.3.10 Data Timeout Register (DTOUTR) 1307
31.3.11 Card Status Register (CSTR) 1308
31.3.12 Interrupt Control Registers 0 and 1 (INTCR0, INTCR1) 1310
31.3.13 Interrupt Status Registers 0 and 1 (INTSTR0, INTSTR1) 1312
31.3.14 Transfer Clock Control Register (CLKON) 1316
31.3.15 VDD/Open-Drain Control Register (VDCNT) 1317
31.3.16 Data Register (DR) 1318
31.3.17 FIFO Pointer Clear Register (FIFOCLR) 1318
31.3.18 DMA Control Register (DMACR) 1319
31.3.19 Interrupt Control Register 2 (INTCR2) 1320
31.3.20 Interrupt Status Register 2 (INTSTR2) 1321
31.3.21 Card Switch Register (CSWR) 1322
31.3.22 Switch Status Register (SWSR) 1323
31.3.23 Chattering Elimination Pulse Setting Register (CHATR) 1324
31.4 Operation 1326
31.4.1 Operations in MMC Mode 1326
31.5 Operations when Using DMAC 1351
31.5.1 Operation in Read Sequence 1351
31.5.2 Operation in Write Sequence 1354
31.6 MMCIF Interrupt Sources 1357
31.7 Procedure to Apply the Card Detection Function 1358

Section 32 PC Card Controller (PCC) 1359
32.1 Features 1359
32.1.1 PCMCIA Support 1360
32.2 Input/Output Pins 1364

32.3 Register Descriptions 1365
32.3.1 Area 6 Interface Status Register (PCC0ISR) 1366
32.3.2 Area 6 General Control Register (PCC0GCR) 1369
32.3.3 Area 6 Card Status Change Register (PCC0CSCR) 1372
32.3.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER) 1376
32.4 Operation 1380
32.4.1 PC card Connection Specification (Interface Diagram, Pin Correspondence) 1380
32.4.2 PC Card Interface Timing 1384
32.5 Usage Notes 1389

Section 33 Audio Codec Interface (HAC) 1391
33.1 Features 1391
33.2 Input/Output Pins 1392
33.3 Register Descriptions 1393
33.3.1 Control and Status Register (HACCR) 1394
33.3.2 Command/Status Address Register (HACCSAR) 1396
33.3.3 Command/Status Data Register (HACCSDR) 1398
33.3.4 PCM Left Channel Register (HACPCML) 1399
33.3.5 PCM Right Channel Register (HACPCMR) 1401
33.3.6 TX Interrupt Enable Register (HACTIER) 1402
33.3.7 TX Status Register (HACTSR) 1403
33.3.8 RX Interrupt Enable Register (HACRIER) 1405
33.3.9 RX Status Register (HACRSR) 1406
33.3.10 HAC Control Register (HACACR) 1407
33.4 AC 97 Frame Slot Structure 1409
33.5 Operation 1410
33.5.1 Receiver 1410
33.5.2 Transmitter 1411
33.5.3 DMA 1411
33.5.4 Interrupts 1411
33.5.5 Initialization Sequence 1412
33.5.6 Notes 1417
33.5.7 Reference 1417

Section 34 Serial Sound Interface (SSI) 1419
34.1 Features 1419
34.2 Input/Output Pins 1421
34.3 Register Descriptions 1422
34.3.1 Control Register (SSICR) 1424

34.3.2 Status Register (SSISR) 1431
34.3.3 Transmit Data Register (SSITDR) 1436
34.3.4 Receive Data Register (SSIRDR) 1436
34.4 Operation 1437
34.4.1 Bus Format 1437
34.4.2 Non-Compressed Modes 1438
34.4.3 Operation Modes 1448
34.4.4 Transmit Operation 1449
34.4.5 Receive Operation 1452
34.4.6 Serial Clock Control 1455
34.5 Usage Note 1456
34.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation 1456
34.5.2 Restrictions for Operation in Slave Mode 1456

Section 35 USB Host Controller (USBH) 1457
35.1 Features 1457
35.2 Pin Description 1459
35.3 Register Description 1460
35.3.1 HcRevision Register (USBHR) 1462
35.3.2 HcControl Register (USBHC) 1463
35.3.3 HcCommandStatus Register (USBHCS) 1465
35.3.4 HcInterruptStatus Register (USBHIS) 1466
35.3.5 HcInterruptEnable Register (USBHIE) 1468
35.3.6 HcInterruptDisable Register (USBHID) 1469
35.3.7 HcHCCA Register (USBHHCCA) 1471
35.3.8 HcPeriodCurrentED Register (USBHPCED) 1471
35.3.9 HcControlHeadED Register (USBHCHED) 1472
35.3.10 HcControlCurrentED Register (USBHCCED) 1472
35.3.11 HcBulkHeadED Register (USBHBHED) 1473
35.3.12 HcBulkCurrentED Register (USBHBCED) 1473
35.3.13 HcDoneHead Register (USBHDHED) 1474
35.3.14 HcFmInterval Register (USBHFI) 1475
35.3.15 HcFrameRemaining Register (USBHFR) 1476
35.3.16 HcFmNumber Register (USBHFN) 1477
35.3.17 HcPeriodicStart Register (USBHPS) 1478
35.3.18 HcLSThreshold Register (USBHLST) (Not supporting LowSpeed mode) 1479
35.3.19 HcRhDescriptorA Register (USBHRDA)
(Only one port is supported by this LSI.) 1480
35.3.20 HcRhDescriptorB Register (USBHRDB)
(Only one port is supported by this LSI.) 1482

35.3.21 HcRhStatus Register (USBHRS) 1484
35.3.22 HcRhPortStatus[2] Register (USBHRPS2) 1486
35.3.23 ConfigurationControl Register (USBHSC) 1489
35.4 Functional Description 1491
35.4.1 General Functionality 1491
35.5 Connection Example of an External Circuit 1493
35.6 Usage Notes 1493
35.6.1 External memory that USBH accesses 1493
35.6.2 Issuing USB Bus Reset 1493

Section 36 USB Function Controller (USBF) 1495
36.1 Features 1495
36.2 Input/Output Pins 1497
36.3 Register Descriptions 1498
36.3.1 Interrupt Flag Register 0 (IFR0) 1504
36.3.2 Interrupt Flag Register 1 (IFR1) 1507
36.3.3 Interrupt Flag Register 2 (IFR2) 1509
36.3.4 Interrupt Flag Register 3 (IFR3) 1511
36.3.5 Interrupt Flag Register 4 (IFR4) 1514
36.3.6 Interrupt Select Register 0 (ISR0) 1515
36.3.7 Interrupt Select Register 1 (ISR1) 1516
36.3.8 Interrupt Select Register 2 (ISR2) 1517
36.3.9 Interrupt Select Register 3 (ISR3) 1518
36.3.10 Interrupt Select Register 4 (ISR4) 1519
36.3.11 Interrupt Enable Register 0 (IER0) 1520
36.3.12 Interrupt Enable Register 1 (IER1) 1521
36.3.13 Interrupt Enable Register 2 (IER2) 1522
36.3.14 Interrupt Enable Register 3 (IER3) 1523
36.3.15 Interrupt Enable Register 4 (IER4) 1524
36.3.16 EP0i Data Register (EPDR0i) 1525
36.3.17 EP0o Data Register (EPDR0o) 1526
36.3.18 EP0s Data Register (EPDR0s) 1527
36.3.19 EP1 Data Register (EPDR1) 1528
36.3.20 EP2 Data Register (EPDR2) 1529
36.3.21 EP3 Data Register (EPDR3) 1530
36.3.22 EP4 Data Register (EPDR4) 1531
36.3.23 EP5 Data Register (EPDR5) 1532
36.3.24 EP0o Receive Data Size Register (EPSZ0o) 1533
36.3.25 EP1 Receive Data Size Register (EPSZ1) 1534
36.3.26 EP4 Receive Data Size Register (EPSZ4) 1535

36.3.27 Trigger Register (TRG) 1536
36.3.28 Data Status Register (DASTS) 1537
36.3.29 FIFO Clear Register 0 (FCLR0) 1538
36.3.30 FIFO Clear Register 1 (FCLR1) 1539
36.3.31 DMA Transfer Setting Register (DMA) 1540
36.3.32 Endpoint Stall Register 0 (EPSTL0) 1541
36.3.33 Endpoint Stall Register 1 (EPSTL1) 1542
36.3.34 Configuration Value Register (CVR) 1543
36.3.35 Time Stamp Register (TSRH/TSRL) 1544
36.3.36 Control Register 0 (CTLR0) 1546
36.3.37 Control Register 1 (CTLR1) 1548
36.3.38 Endpoint Information Register (EPIR) 1549
36.3.39 Timer Register (TMRH/TMRL) 1555
36.3.40 Set Time Out Register (STOH/STOL) 1557
36.4 Operation 1559
36.4.1 Cable Connection 1559
36.4.2 Cable Disconnection 1560
36.4.3 EP1 Bulk-Out Transfer (Dual FIFOs) 1566
36.4.4 EP2 Bulk-In Transfer (Dual FIFOs) 1567
36.4.5 EP3 Interrupt-In Transfer 1569
36.5 EP4 Isochronous-Out Transfer 1570
36.6 EP5 Isochronous-In Transfer 1573
36.7 Processing of USB Standard Commands and Class/Vendor Commands 1576
36.7.1 Processing of Commands Transmitted by Control Transfer 1576
36.8 Stall Operations 1577
36.8.1 Overview 1577
36.8.2 Forcible Stall by Application 1577
36.8.3 Automatic Stall by USB Function Controller 1579
36.9 Examples of External Circuit 1581
36.9.1 Example of the Connection between USB Function Controller 1581
36.10 Usage Notes 1582
36.10.1 Setup Data Reception 1582
36.10.2 FIFO Clear 1582
36.10.3 Overreading/Overwriting of Data Register 1582
36.10.4 Assigning EP0 Interrupt Sources 1583
36.10.5 FIFO Clear when DMA Transfer is Set 1583
36.10.6 Note on Using TR Interrupt 1583

Section 37 LCD Controller (LCDC) 1585
37.1 Features 1585
37.2 Input/Output Pins 1587
37.3 Register Configuration 1588
37.3.1 LCDC Input Clock Register (LDICKR) 1591
37.3.2 LCDC Module Type Register (LDMTR) 1593
37.3.3 LCDC Data Format Register (LDDFR) 1596
37.3.4 LCDC Scan Mode Register (LDSMR) 1598
37.3.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU) 1599
37.3.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL) 1601
37.3.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR) 1602
37.3.8 LCDC Palette Control Register (LDPALCR) 1603
37.3.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF) 1604
37.3.10 LCDC Horizontal Character Number Register (LDHCNR) 1605
37.3.11 LCDC Horizontal Sync Signal Register (LDHSYNR) 1606
37.3.12 LCDC Vertical Display Line Number Register (LDVDLNR) 1607
37.3.13 LCDC Vertical Total Line Number Register (LDVTLNR) 1608
37.3.14 LCDC Vertical Sync Signal Register (LDVSYNR) 1609
37.3.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR) 1610
37.3.16 LCDC Interrupt Control Register (LDINTR) 1611
37.3.17 LCDC Power Management Mode Register (LDPMMR) 1614
37.3.18 LCDC Power-Supply Sequence Period Register (LDPSPR) 1616
37.3.19 LCDC Control Register (LDCNTR) 1618
37.3.20 LCDC User Specified Interrupt Control Register (LDUINTR) 1619
37.3.21 LCDC User Specified Interrupt Line Number Register (LDUINTLNR) 1621
37.3.22 LCDC Memory Access Interval Number Register (LDLIRNR) 1622
37.4 Operation 1623
37.4.1 LCD Module Sizes which can be Displayed in this LCDC 1623
37.4.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM) 1625
37.4.3 Color Palette Specification 1628
37.4.4 Data Format 1629
37.4.5 Setting the Display Resolution 1633
37.4.6 Power-Supply Control Sequence 1633
37.4.7 Operation for Hardware Rotation 1638
37.5 Clock and LCD Data Signal Examples 1641
37.6 Usage Notes 1653
37.6.1 Procedure for Halting Access to Display Data Storage VRAM (DDR-SDRAM in Area 3) 1653

37.6.2 Notes on Using NMI Interrupt 1653

Section 38 A/D Converter 1655
38.1 Features 1655
38.2 Input Pins 1657
38.3 Register Descriptions 1658
38.3.1 A/D Data Registers A to D (ADDRA to ADDRD) 1658
38.3.2 A/D Control/Status Registers (ADCSR) 1660
38.4 Operation 1663
38.4.1 Single Mode (MDS1 = 0, MDS0 = 0) 1663
38.4.2 Multi Mode (MDS[1:0] = 10) 1664
38.4.3 Scan Mode (MDS1 = 1, MDS0 = 1) 1666
38.4.4 A/D Conversion Time 1668
38.5 Interrupts 1668
38.6 Definitions of A/D Conversion Accuracy 1669
38.7 Usage Notes 1671
38.7.1 Setting Analog Input Voltage 1671
38.7.2 Processing of Analog Input Pins 1671
38.7.3 Pck0 Clock and Clock Division Ratio Settings 1672
38.7.4 A/D Conversion Stop 1672

Section 39 D/A Converter (DAC) 1673
39.1 Features 1673
39.2 Input/Output Pins 1674
39.3 Register Descriptions 1674
39.3.1 D/A Data Registers 0 and 1 (DADR0, DADR1) 1675
39.3.2 D/A Control Register (DACR) 1676
39.4 Operation 1677
39.5 Usage Notes 1678

Section 40 General Purpose I/O (GPIO) 1679
40.1 Features 1679
40.2 Register Descriptions 1690
40.2.1 Port A Control Register (PACR) 1693
40.2.2 Port B Control Register (PBCR) 1695
40.2.3 Port C Control Register (PCCR) 1696
40.2.4 Port D Control Register (PDCR) 1698
40.2.5 Port E Control Register (PECR) 1700
40.2.6 Port F Control Register (PFCR) 1701
40.2.7 Port G Control Register (PGCR) 1703

40.2.8 Port H Control Register (PHCR) 1705
40.2.9 Port I Control Register (PICR) 1707
40.2.10 Port J Control Register (PJCR) 1709
40.2.11 Port K Control Register (PKCR) 1711
40.2.12 Port L Control Register (PLCR) 1713
40.2.13 Port M Control Register (PMCR) 1715
40.2.14 Port N Control Register (PNCR) 1717
40.2.15 Port O Control Register (POCR) 1719
40.2.16 Port A Data Register (PADR) 1721
40.2.17 Port B Data Register (PBDR) 1722
40.2.18 Port C Data Register (PCDR) 1723
40.2.19 Port D Data Register (PDDR) 1724
40.2.20 Port E Data Register (PEDR) 1725
40.2.21 Port F Data Register (PFDR) 1726
40.2.22 Port G Data Register (PGDR) 1727
40.2.23 Port H Data Register (PHDR) 1728
40.2.24 Port I Data Register (PIDR) 1729
40.2.25 Port J Data Register (PJDR) 1730
40.2.26 Port K Data Register (PKDR) 1731
40.2.27 Port L Data Register (PLDR) 1732
40.2.28 Port M Data Register (PMDR) 1733
40.2.29 Port N Data Register (PNDR) 1734
40.2.30 Port O Data Register (PODR) 1735
40.2.31 Port I Pull-Up Control Register (PIPUPR) 1736
40.2.32 Port J Pull-Up Control Register (PJPUPR) 1737
40.2.33 Port K Pull-Up Control Register (PKPUPR) 1738
40.2.34 Port L Pull-Up Control Register (PLPUPR) 1739
40.2.35 Port M Pull-Up Control Register (PMPUPR) 1740
40.2.36 Port N Pull-Up Control Register (PNPUPR) 1742
40.2.37 Port O Pull-Up Control Register (POPUPR) 1743
40.2.38 Input-Pin Pull-Up Control Register (PPUPR) 1744
40.2.39 Pin Select Register 0 (PSEL0) 1745
40.2.40 Pin Select Register 1 (PSEL1) 1746
40.2.41 Pin Select Register 2 (PSEL2) 1749
40.2.42 Pin Select Register 3 (PSEL3) 1752
40.2.43 Pin Select Register 4 (PSEL4) 1755
40.3 Usage Examples 1758
40.3.1 Port Output Function 1758
40.3.2 Port Input Function 1758
40.3.3 Peripheral Module Function 1758

Section 41 User Break Controller (UBC) 1759
41.1 Features 1759
41.2 Register Descriptions 1761
41.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1) 1763
41.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1) 1770
41.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1) 1772
41.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1) 1774
41.2.5 Match Data Setting Register 1 (CDR1) 1776
41.2.6 Match Data Mask Setting Register 1 (CDMR1) 1777
41.2.7 Execution Count Break Register 1 (CETR1) 1778
41.2.8 Channel Match Flag Register (CCMFR) 1779
41.2.9 Break Control Register (CBCR) 1780
41.3 Operation Description 1781
41.3.1 Definition of Words Related to Accesses 1781
41.3.2 User Break Operation Sequence 1782
41.3.3 Instruction Fetch Cycle Break 1784
41.3.4 Operand Access Cycle Break 1785
41.3.5 Sequential Break 1787
41.3.6 Program Counter Value to be Saved 1788
41.4 User Break Debugging Support Function 1789
41.5 User Break Examples 1791
41.6 Usage Notes 1795

Section 42 User Debugging Interface (H-UDI) 1797
42.1 Features 1797
42.2 Input/Output Pins 1799
42.3 Boundary Scan TAP Controllers (IDCODE, EXTEST, SAMPLE/PRELOAD, and BYPASS) 1800
42.4 Register Descriptions 1802
42.4.1 Instruction Register (SDIR) 1803
42.4.2 Interrupt Source Register (SDINT) 1804
42.4.3 Bypass Register (SDBPR) 1805
42.4.4 Boundary Scan Register (SDBSR) 1805
42.5 Operation 1823
42.5.1 TAP Control 1823
42.5.2 H-UDI Reset 1824
42.5.3 H-UDI Interrupt 1824
42.6 Usage Notes 1824

Section 43 Electrical Characteristics 1825
43.1 Absolute Maximum Ratings 1825
43.2 Power-On and Power-Off Order 1826
43.2.1 Power-On Order 1826
43.2.2 Power-Off Order 1826
43.2.3 Power-Off and Power-On Order in RTC Power-Supply Backup Mode (Hardware Standby) 1828
43.2.4 Power-Off and Power-On Order in DDR-SDRAM Power-Supply Backup Mode 1828
43.3 DC Characteristics 1829
43.4 AC Characteristics 1833
43.4.1 Clock and Control Signal Timing 1834
43.4.2 Control Signal Timing 1838
43.4.3 Bus Timing 1840
43.4.4 DDRIF Signal Timing 1858
43.4.5 INTC Module Signal Timing 1861
43.4.6 External CPU Interface Read/Write Access Timing 1863
43.4.7 PCIC Module Signal Timing 1865
43.4.8 DMAC Module Signal Timing 1867
43.4.9 TMU Module Signal Timing 1868
43.4.10 16-bit Timer Pulse Unit (TPU) Timing 1869
43.4.11 GETHER Module Signal Timing 1870
43.4.12 Stream Interface Module Timing 1876
43.4.13 I^2C Bus Interface Timing 1880
43.4.14 SCIF Module Signal Timing 1882
43.4.15 SIOF Module Signal Timing 1884
43.4.16 SIM Module Signal Timing 1888
43.4.17 MMCIF Module Signal Timing 1889
43.4.18 HAC Interface Module Signal Timing 1891
43.4.19 SSI Interface Module Signal Timing 1893
43.4.20 USB Module Signal Timing 1895
43.4.21 LCDC Module Signal Timing 1896
43.4.22 GPIO Signal Timing 1897
43.4.23 H-UDI Module Signal Timing 1898
43.5 A/D, D/A Converter Characteristics 1900
43.5.1 A/D Converter Characteristics 1900
43.5.2 D/A Converter Characteristics 1900
43.6 AC Characteristic Test Conditions 1901
43.7 Change in Delay Time Based on Load Capacitance 1902

Appendix 1903
A. CPU Operation Mode Register (CPUOPM) 1903
B. Instruction Prefetching and Its Side Effects 1905
C. Speculative Execution for Subroutine Return 1906
D. List of Mode Control Pins and Schematic Diagram of External Cicuits 1907
E. Notes on Board Design 1909
F. Package Dimensions 1912
G. Pin States 1913
H. Handling of Unused Pins 1931
I. Version Registers 1943
J. Heat Radiation 1944
J.1 Heat Resistance Simulation Conditions 1944
J.2 Analysis Results of Heat Resistance Simulation 1945

Index 1947

Figures

Section 1 Overview
Figure 1.1 SH7763 Block Diagram 13
Figure 1.2 Pin Arrangement 15

Section 2 Programming Model
Figure 2.1 Data Formats 37
Figure 2.2 CPU Register Configuration in Each Processing Mode 41
Figure 2.3 General Registers 42
Figure 2.4 Floating-Point Registers 44
Figure 2.5 Relationship between SZ bit and Endian 50
Figure 2.6 Formats of Byte Data and Word Data in Register 52
Figure 2.7 Data Formats in Memory 53
Figure 2.8 Processing State Transitions 54

Section 4 Pipelining
Figure 4.1 Basic Pipelines 79
Figure 4.2 Instruction Execution Patterns (1) 81
Figure 4.2 Instruction Execution Patterns (2) 82
Figure 4.2 Instruction Execution Patterns (3) 83
Figure 4.2 Instruction Execution Patterns (4) 84
Figure 4.2 Instruction Execution Patterns (5) 85
Figure 4.2 Instruction Execution Patterns (6) 86
Figure 4.2 Instruction Execution Patterns (7) 87
Figure 4.2 Instruction Execution Patterns (8) 88
Figure 4.2 Instruction Execution Patterns (9) 89

Section 5 Exception Handling
Figure 5.1 Instruction Execution and Exception Handling 113
Figure 5.2 Example of General Exception Acceptance Order 114

Section 6 Memory Management Unit (MMU)
Figure 6.1 Role of MMU 139
Figure 6.2 Virtual Address Space (AT in MMUCR = 0) 140
Figure 6.3 Virtual Address Space (AT in MMUCR = 1) 141
Figure 6.4 P4 Area 143
Figure 6.5 Physical Address Space 144
Figure 6.6 UTLB Configuration 156
Figure 6.7 Relationship between Page Size and Address Format 158
Figure 6.8 ITLB Configuration 159

Figure 6.9 Flowchart of Memory Access Using UTLB 160
Figure 6.10 Flowchart of Memory Access Using ITLB 161
Figure 6.11 Operation of LDTLB Instruction 164
Figure 6.12 Memory-Mapped ITLB Address Array 173
Figure 6.13 Memory-Mapped ITLB Data Array 174
Figure 6.14 Memory-Mapped UTLB Address Array 176
Figure 6.15 Memory-Mapped UTLB Data Array 177
Figure 6.16 Physical Address Space (32-Bit Address Extended Mode) 177
Figure 6.17 PMB Configuration 179
Figure 6.18 Memory-Mapped PMB Address Array 183
Figure 6.19 Memory-Mapped PMB Data Array 183

Section 7 Caches

Figure 7.1 Configuration of Operand Cache (OC) 188
Figure 7.2 Configuration of Instruction Cache (IC) 189
Figure 7.3 Configuration of Write-Back Buffer 201
Figure 7.4 Configuration of Write-Through Buffer 201
Figure 7.5 Memory-Mapped IC Address Array 207
Figure 7.6 Memory-Mapped IC Data Array 208
Figure 7.7 Memory-Mapped OC Address Array 210
Figure 7.8 Memory-Mapped OC Data Array 211
Figure 7.9 Store Queue Configuration 212

Section 9 Interrupt Controller (INTC)

Figure 9.1 Block Diagram of INTC 234
Figure 9.2 Example of IRL Interrupt Connection 293
Figure 9.3 On-chip Module Interrupt Priority 296
Figure 9.4 Interrupt Operation Flowchart 306
Figure 9.5 Example of Interrupt Handling Routine 309

Section 10 SuperHyway Bus Bridge (SBR)

Figure 10.1 SBR Block Diagram 313
Figure 10.2 Bus Arbitration by the SBR 317

Section 11 Local Bus State Controller (LBSC)

Figure 11.1 LBSC Block Diagram 321
Figure 11.2 Correspondence between Virtual Address Space and External Memory Space 325
Figure 11.3 External Memory Space Allocation 327
Figure 11.4 Basic Timing of SRAM Interface 366
Figure 11.5 Example of 32-Bit Data-Width SRAM Connection 367
Figure 11.6 Example of 16-Bit Data-Width SRAM Connection 368
Figure 11.7 Example of 8-Bit Data-Width SRAM Connection 368

Figure 11.8 SRAM Interface Wait Timing (Software Wait Only) 369
Figure 11.9 SRAM Interface Wait Cycle Timing (Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal)..... 370
Figure 11.10 SRAM Interface Wait State Timing (Read-Strobe Negate Timing Setting)..... 372
Figure 11.11 Burst ROM Basic Access Timing 374
Figure 11.12 Burst ROM Wait Access Timing..... 374
Figure 11.13 Burst ROM Wait Access Timing..... 375
Figure 11.14 $\overline{\text{CExx}}$ and $\overline{\text{DACK}}$ Output of ATA Complete Mode in DMA Transfer..... 377
Figure 11.15 Example of PCMCIA Interface 380
Figure 11.16 Basic Timing for PCMCIA Memory Card Interface 381
Figure 11.17 Wait Timing for PCMCIA Memory Card Interface 382
Figure 11.18 Basic Timing for PCMCIA I/O Card Interface 383
Figure 11.19 Wait Timing for PCMCIA I/O Card Interface 384
Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface 385
Figure 11.21 Example of 32-Bit Data Width MPX Connection 387
Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External Wait) 387
Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait Inserted)..... 388
Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No External Wait) 389
Figure 11.25 MPX Interface Timing 4
(Single Write Cycle, IW = 1, One External Wait Inserted)..... 390
Figure 11.26 MPX Interface Timing 5 (Burst Read Cycle, IW = 0, No External Wait)..... 391
Figure 11.27 MPX Interface Timing 6 (Burst Read Cycle, IW = 0, External Wait Control) 392
Figure 11.28 MPX Interface Timing 7 (Burst Write Cycle, IW = 0, No External Wait)..... 393
Figure 11.29 MPX Interface Timing 8 (Burst Write Cycle, IW = 1, External Wait Control) 394
Figure 11.30 MPX Interface Timing 9 (Burst Read Cycle, IW = 0,
No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer) 395
Figure 11.31 MPX Interface Timing 10 (Burst Read Cycle, IW = 0,
External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)..... 396
Figure 11.32 MPX Interface Timing 11 (Burst Write Cycle, IW = 0,
No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer) 397
Figure 11.33 MPX Interface Timing 12 (Burst Write Cycle, IW = 1,
External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)..... 398
Figure 11.34 Example of 32-Bit Data-Width Byte-Control SRAM 399
Figure 11.35 Byte-Control SRAM Basic Read Cycle (No Wait) 400
Figure 11.36 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cycle)..... 401
Figure 11.37 Byte-Control SRAM Basic Read Cycle
(One Internal Wait + One External Wait) 402
Figure 11.38 Wait Cycles between Access Cycles 404
Figure 11.39 Arbitration Sequence 406

Section 12 DDR-SDRAM Interface (DDRIF)

Figure 12.1 DDRIF Block Diagram 410
Figure 12.2 Data Alignment in DDR-SDRAM and DDRIF 414
Figure 12.3 Relationship between Write Values in SDMR and Output Signals to Memory Pins 428
Figure 12.4 DDR-SDRAM Access 430
Figure 12.5 Basic DDRIF Timing (1 Burst Read: 1, 2, 4, or 8 Bytes; Without Auto-Precharge) 438
Figure 12.6 Basic DDRIF Timing (1 Burst Write: 1, 2, 4, or 8 Bytes; Without Auto-Precharge) 439
Figure 12.7 Basic DDRIF Timing (1 Burst Read: 1, 2, 4, or 8 Bytes; With Auto-Precharge) 440
Figure 12.8 Basic DDRIF Timing (1 Burst Write: 1, 2, 4, or 8 Bytes; With Auto-Precharge) 441
Figure 12.9 Basic DDRIF Timing (4 Burst Read: 32 Bytes; Without Auto-Precharge) 442
Figure 12.10 Basic DDRIF Timing (4 Burst Write: 32 Bytes; Without Auto-Precharge) 443
Figure 12.11 Basic DDRIF Timing (Precharge all Banks (PREALL) to Bank Activate (ACT)) 444
Figure 12.12 Basic DDRIF Timing (Mode Register Set (MRS)) 445
Figure 12.13 Basic DDRIF Timing (Auto-Refresh (REFA) Enter/Exit to Bank Activate (ACT)) 446
Figure 12.14 Basic DDRIF Timing (Self-Refresh Entry from IDLE (REFS)/ Self-Refresh Exit (REFSX) to Any Command Input) 447

Section 13 PCI Controller (PCIC)

Figure 13.1 PCIC Block Diagram 451
Figure 13.2 SuperHyway Bus to PCI Local Bus Access 530
Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 0) 531
Figure 13.4 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 1) 532
Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 2) 532
Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/O) 533
Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus (Non-Byte Swapping: TBS = 0) 535
Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus (Byte Swapping: TBS = 1) 536
Figure 13.9 PCI local bus to SuperHyway bus Memory Map 537

Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation (Local Address Space 0/1) 539
Figure 13.11 PCI Local Bus to SuperHyway Bus Address Translation (PCIC I/O Space) 540
Figure 13.12 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 0) 542
Figure 13.13 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 1) 543
Figure 13.14 Cache Flush/Purge Execution Flow for PCI local Bus to SuperHyway Bus 545
Figure 13.15 Address Generation for Type 0 Configuration Access 547
Figure 13.16 PCI Local Bus Power Down State Transition 550
Figure 13.17 Master Write Cycle in Host Bus Bridge Mode (Single) 551
Figure 13.18 Master Read Cycle in Host Bus Bridge Mode (Single) 552
Figure 13.19 Master Write Cycle in Normal Mode (Burst) 553
Figure 13.20 Master Read Cycle in Normal Mode (Burst) 554
Figure 13.21 Target Read Cycle in Normal Mode (Single) 556
Figure 13.22 Target Write Cycle in Normal Mode (Single) 557
Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burst) 558
Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst) 559
Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with stepping) 560
Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with stepping) 561
Figure 13.27 Timing Example of Device ($\overline{\text{REQm}}$) Not Executing $\overline{\text{REQ}}$ Negation and $\overline{\text{FRAME}}$ Assertion Simultaneously 562

Section 14 Direct Memory Access Controller (DMAC)

Figure 14.1 Block Diagram of DMAC 566
Figure 14.2 Round-Robin Mode 596
Figure 14.3 Changes in Channel Priority in Round-Robin Mode 597
Figure 14.4 Data Flow of Dual Address Mode 598
Figure 14.5 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory) 599
Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection) 600
Figure 14.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection) 601
Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection) 601
Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection) 602
Figure 14.10 Bus State when Multiple Channels are Operating 606
Figure 14.11 DMA Transfer Flowchart 607
Figure 14.12 Reload Mode Transfer 609
Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection 610

Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection 610
Figure 14.15 Example of DREQ Input Detection in Burst Mode Edge Detection 611
Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detection 611
Figure 14.17 DMA Transfer End Signal (Cycle Steal Mode Level Detection) 612
Figure 14.18 Example of BSC Ordinary Memory Access
(No Wait, Idle Cycle 1, Longword Access to 16-Bit Device) 613

Section 15 External CPU Interface (EXCPU)
Figure 15.1 EXCPU Block Diagram 621
Figure 15.2 External CPU Access (Single Access) 630
Figure 15.3 External CPU Access (Burst Access) 631
Figure 15.4 Configuration of Connection with External CPU 632

Section 16 Clock Pulse Generator (CPG)
Figure 16.1 Block Diagram of CPG 634
Figure 16.2 Notes on Using Crystal Resonator 642
Figure 16.3 Notes on Using PLL or DLL Oscillator Circuit 643

Section 17 Watchdog Timer and Reset (WDT)
Figure 17.1 System Block Diagram 646
Figure 17.2 WDT Counting Up Operation 657
Figure 17.3 STATUS Output during Power-on 660
Figure 17.4 STATUS Output by Reset input during Normal Operation 661
Figure 17.5 STATUS Output by Reset input during Sleep Mode 661
Figure 17.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation 662
Figure 17.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode 663
Figure 17.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation 664
Figure 17.9 STATUS Output by Watchdog timer overflow Manual Reset during Sleep Mode 665

Section 18 Power-Down Mode
Figure 18.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off 682
Figure 18.2 Sequence for Turning Off System Power Supply after Entering Self-Refresh Mode 683
Figure 18.3 Sequence for Turning VDD Power Supply (1.2 V) On/Off 685
Figure 18.4 STATUS Output when an Interrupt Occurs in Sleep Mode 685

Section 19 Timer Unit (TMU)
Figure 19.1 Block Diagram of TMU 688
Figure 19.2 Example of Count Operation Setting Procedure 700
Figure 19.3 TCNT Auto-Reload Operation 701
Figure 19.4 Count Timing when Operating on Internal Clock 701
Figure 19.5 Count Timing when Operating on External Clock 702
Figure 19.6 Count Timing when Operating on on-chip RTC output Clock 702
Figure 19.7 Operation Timing when Using Input Capture Function 703

Section 20 16-Bit Timer Pulse Unit (TPU)
Figure 20.1 Block Diagram of TPU 709
Figure 20.2 Example of Counter Operation Setting Procedure 730
Figure 20.3 Free-Running Counter Operation 731
Figure 20.4 Periodic Counter Operation 732
Figure 20.5 Example of Setting Procedure for Waveform Output by Compare Match 732
Figure 20.6 Example of 0 Output/1 Output Operation 733
Figure 20.7 Example of Toggle Output Operation 733
Figure 20.8 Compare Match Buffer Operation 734
Figure 20.9 Example of Buffer Operation Setting Procedure 735
Figure 20.10 Example of Buffer Operation 736
Figure 20.11 Example of PWM Mode Setting Procedure 738
Figure 20.12 Example of PWM Mode Operation (1) 739
Figure 20.13 Examples of PWM Mode Operation (2) 739
Figure 20.14 Example of Phase Counting Mode Setting Procedure 741
Figure 20.15 Example of Phase Counting Mode 1 Operation 742
Figure 20.16 Example of Phase Counting Mode 2 Operation 743
Figure 20.17 Example of Phase Counting Mode 3 Operation 744
Figure 20.18 Example of Phase Counting Mode 4 Operation 745
Figure 20.19 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode 746

Section 21 Compare Match Timer (CMT)
Figure 21.1 Block Diagram of CMT 748
Figure 21.2 Counter Operation (One-Shot Operation) 755
Figure 21.3 Counter Operation (Free-Running Operation) 756
Figure 21.4 CMF Set Timing 757

Section 22 Realtime Clock (RTC)
Figure 22.1 Block Diagram of RTC 760
Figure 22.2 Examples of Time Setting Procedures 778
Figure 22.3 Examples of Time Reading Procedures 779
Figure 22.4 Example of Use of Alarm Function 780
Figure 22.5 Example of Crystal Oscillator Circuit Connection 782

Section 23 Gigabit Ethernet Controller (GETHER)

Figure 23.1 Configuration of GETHER 784
Figure 23.2 GETHER Data Path and Various Settings 938
Figure 23.3 Relationship between Transmit Descriptor and Transmit Buffer 940
Figure 23.4 Relationship between Receive Descriptor and Receive Buffer 946
Figure 23.5 Relationship between Transmit Descriptor and Transmit Buffer 952
Figure 23.6 Relationship between Receive Descriptor and Receive Buffer 953
Figure 23.7 Relationship between Transmit/Receive Descriptor and Descriptor Pointing Registers 955
Figure 23.8 Sample Transmission Flowchart (Single-Frame/Two-Description) 958
Figure 23.9 E-MAC Transmitter State Transitions 960
Figure 23.10 E-MAC Receiver State Transitions 963
Figure 23.11 Sample Reception Flowchart (Single-Frame/Two-Descriptor) 965
Figure 23.12 E-DMAC Operation after Transmit Error 971
Figure 23.13 E-DMAC Operation after Receive Error 972
Figure 23.14 Padding Insertion in Receive Data 973
Figure 23.15 Outlines of Qtag Additional Functions 982
Figure 23.16 Comparison of Normal Ethernet Frame and IEEE802.1Q Frame (with Qtag) 983
Figure 23.17 MII Frame Transmit Timing (Normal Transmission) 984
Figure 23.18 MII Frame Transmit Timing (Collision) 984
Figure 23.19 MII Frame Transmit Timing (Transmit Error) 985
Figure 23.20 MII Frame Receive Timing (Normal Reception) 985
Figure 23.21 MII Frame Receive Timing (Reception Error (1)) 985
Figure 23.22 MII Fame Receive Timing (Reception Error (2)) 985
Figure 23.23 GMII/MII Fame Receive Timing (Normal Reception) 986
Figure 23.24 GMII/MII Fame Receive Timing (with Carrier Extension) 986
Figure 23.25 GMII/MII Fame Receive Timing (Burst Reception) 986
Figure 23.26 GMII/MII Fame Receive Timing (Reception Error) 987
Figure 23.27 GMII/MII Fame Receive Timing (Error with Carrier Extension) 987
Figure 23.28 GMII/MII Fame Receive Timing (False Carrier Indication) 987
Figure 23.29 RMII Fame Receive Timing (Normal 100-Mbps Reception) 988
Figure 23.30 RMII Fame Receive Timing (100-Mbps Reception with Illegal Carrier Detected) 988
Figure 23.31 RMII Fame Transmit Timing (Normal 100-Mbps Transmission) 988
Figure 23.32 MII Management Frame Format 989
Figure 23.33 1-Bit Data Write Flowchart 990
Figure 23.34 Bus Release Flowchart (TA in Read in Figure 23.33) 990
Figure 23.35 1-Bit Data Read Flowchart 991
Figure 23.36 Independent Bus Release Flowchart (IDLE in Write in Figure 23.33) 991
Figure 23.37 MII-RMll Conversion Circuit 992

Figure 23.38 Data Subject to Checksum Calculation 993

Section 25 Stream Interface (STIF)

Figure 25.1 Block Diagram of STIF 998
Figure 25.2 Transmit/Receive Data Structure in External Memory (with 16-Byte Work Area) 1015
Figure 25.3 Clock Valid Reception Timing 1017
Figure 25.4 Strobe Reception Timing 1019
Figure 25.5 Clock Valid Transmission Timing 1021
Figure 25.6 Strobe Transmission Timing 1023

Section 26 I^2C Bus Interface (IIC)

Figure 26.1 Block Diagram for I^2C Bus Interface 1025
Figure 26.2 I^2C Bus Timing 1046
Figure 26.3 Master Data Transmit Format 1047
Figure 26.4 Master Data Receive Format 1047
Figure 26.5 Combination Transfer Format of Master Transfer 1048
Figure 26.6 10-Bit Address Data Transmit Format 1048
Figure 26.7 10-Bit Address Data Receive Format 1049
Figure 26.8 10-Bit Address Transmit/Receive Combined Format 1049
Figure 26.9 Data Transmit Mode Operation Timing 1051
Figure 26.10 Data Receive Mode Operation Timing 1053

Section 27 Serial Communication Interface with FIFO (SCIF)

Figure 27.1 Block Diagram of SCIF 1061
Figure 27.2 $\overline{\text{SCIFn_RTS}}$ Pin (n = 0, 1) 1062
Figure 27.3 $\overline{\text{SCIFn_CTS}}$ Pin (n = 0, 1) 1063
Figure 27.4 SCIFn_SCK Pin (n = 0, 1) 1064
Figure 27.5 SCIFn_TXD Pin (n = 0, 1) 1064
Figure 27.6 SCIFn_RXD Pin (n = 0, 1) 1065
Figure 27.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits) 1094
Figure 27.8 Sample SCIF Initialization Flowchart 1097
Figure 27.9 Sample Serial Transmission Flowchart 1098
Figure 27.10 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit) 1100
Figure 27.11 Sample Operation Using Modem Control ($\overline{\text{SCIF_CTS}}$) 1100
Figure 27.12 Sample Serial Reception Flowchart (1) 1101
Figure 27.12 Sample Serial Reception Flowchart (2) 1102
Figure 27.13 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit) 1103

Figure 27.14 Sample Operation Using Modem Control ($\overline{\text{SCIF0_RTS}}$) (Only in Channel 0) 1104
Figure 27.15 Data Format in Clocked Synchronous Communication 1104
Figure 27.16 Sample SCIF Initialization Flowchart 1106
Figure 27.17 Sample Serial Transmission Flowchart 1107
Figure 27.18 Sample SCIF Transmission Operation in Clocked Synchronous Mode 1108
Figure 27.19 Sample Serial Reception Flowchart (1) 1109
Figure 27.19 Sample Serial Reception Flowchart (2) 1110
Figure 27.20 Sample SCIF Reception Operation in Clocked Synchronous Mode 1111
Figure 27.21 Sample Simultaneous Serial Transmission and Reception Flowchart 1112
Figure 27.22 Receive Data Sampling Timing in Asynchronous Mode 1116
Figure 27.23 Example of Synchronization Clock Transfer by DMAC 1117

Section 28 Serial Communication Interface with FIFO/IrDA Interface (SCIF/IrDA)

Figure 28.1 Block Diagram of SCIF/IrDA 1121
Figure 28.2 SCIF2_SCK Pin 1122
Figure 28.3 SCIF2_TXD Pin 1122
Figure 28.4 SCIF2_RXD Pin 1123
Figure 28.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits) 1155
Figure 28.6 Sample SCIF Initialization Flowchart 1158
Figure 28.7 Sample Serial Transmission Flowchart 1159
Figure 28.8 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit) 1161
Figure 28.9 Sample Serial Reception Flowchart (1) 1162
Figure 28.10 Sample Serial Reception Flowchart (2) 1163
Figure 28.11 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit) 1165
Figure 28.12 Data Format in Clocked Synchronous Communication 1165
Figure 28.13 Sample SCIF Initialization Flowchart 1167
Figure 28.14 Sample Serial Transmission Flowchart 1168
Figure 28.15 Sample SCIF Transmission Operation in Clocked Synchronous Mode 1169
Figure 28.16 Sample Serial Reception Flowchart (1) 1170
Figure 28.17 Sample Serial Reception Flowchart (2) 1171
Figure 28.18 Sample SCIF Reception Operation in Clocked Synchronous Mode 1172
Figure 28.19 Sample Simultaneous Serial Transmission and Reception Flowchart 1173
Figure 28.20 Receive Data Sampling Timing in Asynchronous Mode 1177
Figure 28.21 Infrared Communication Data Format 1179
Figure 28.22 Block Diagram of Infrared Data Communication Interface 1180
Figure 28.23 BRG Block Diagram 1181

Section 29 Serial I/O with FIFO (SIOF)

Figure 29.1 Block Diagram of SIOF 1186
Figure 29.2 Serial Clock Supply 1217
Figure 29.3 Serial Data Synchronization Timing 1219
Figure 29.4 SIOF Transmit/Receive Timing 1220
Figure 29.5 Transmit/Receive Data Bit Alignment 1223
Figure 29.6 Control Data Bit Alignment 1224
Figure 29.7 Control Data Interface (Slot Position) 1226
Figure 29.8 Control Data Interface (Secondary FS) 1227
Figure 29.9 Example of Transmit Operation in Master Mode 1229
Figure 29.10 Example of Receive Operation in Master Mode 1230
Figure 29.11 Example of Transmit Operation in Slave Mode 1231
Figure 29.12 Example of Receive Operation in Slave Mode 1232
Figure 29.13 Transmit and Receive Timing (8-Bit Monaural Data (1)) 1236
Figure 29.14 Transmit and Receive Timing (8-Bit Monaural Data (2)) 1237
Figure 29.15 Transmit and Receive Timing (16-Bit Monaural Data) 1237
Figure 29.16 Transmit and Receive Timing (16-Bit Stereo Data (1)) 1238
Figure 29.17 Transmit and Receive Timing (16-Bit Stereo Data (2)) 1238
Figure 29.18 Transmit and Receive Timing (16-Bit Stereo Data (3)) 1239
Figure 29.19 Transmit and Receive Timing (16-Bit Stereo Data (4)) 1239
Figure 29.20 Transmit and Receive Timing (16-Bit Stereo Data) 1240

Section 30 SIM Card Module (SIM)

Figure 30.1 Smart Card Interface 1242
Figure 30.2 Data Format Used by Smart Card Interface 1266
Figure 30.3 Examples of Start Character Waveforms 1269
Figure 30.4 Example of Initialization Flow 1271
Figure 30.5 Example of Transmit Processing 1273
Figure 30.6 Example of Receive Processing 1275
Figure 30.7 Receive Data Sampling Timing in Smart Card Mode 1278
Figure 30.8 Retransmission when Smart Card Interface is in Receive Mode 1279
Figure 30.9 Retransmit Standby Mode (Clock Stopped) when Smart Card Interface is in Transmit Mode 1280
Figure 30.10 Procedure for Stopping Clock and Restarting 1281
Figure 30.11 Example of Pin Connections in Smart Card Interface 1282
Figure 30.12 TEIE Set Timing 1283

Section 31 Multimedia Card Interface (MMCIF)

Figure 31.1 MMCIF Block Diagram 1286
Figure 31.2 Example of Command Sequence for Commands Not Requiring Command Response 1328

Figure 31.3 Example of Operational Flow for Commands Not Requiring Command Response 1329
Figure 31.4 Example of Command Sequence for Commands without Data Transfer (No Data Busy State) 1330
Figure 31.5 Example of Command Sequence for Commands without Data Transfer (with Data Busy State) 1331
Figure 31.6 Example of Operational Flow for Commands without Data Transfer 1332
Figure 31.7 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size) 1334
Figure 31.8 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size) 1335
Figure 31.9 Example of Command Sequence for Commands with Read Data (Multiblock Transfer) 1336
Figure 31.10 Example of Operational Flow for Commands with Read Data (Single Block Transfer) 1337
Figure 31.11 (1) Example of Operational Flow for Commands with Read Data (Open-ended Multiblock Transfer) 1338
Figure 31.11 (2) Example of Operational Flow for Commands with Read Data (Open-ended Multiblock Transfer) 1339
Figure 31.12 (1) Example of Operational Flow for Commands with Read Data (Pre-defined Multiblock Transfer) 1340
Figure 31.12 (2) Example of Operational Flow for Commands with Read Data (Pre-defined Multiblock Transfer) 1341
Figure 31.13 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size) 1343
Figure 31.14 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size) 1344
Figure 31.15 Example of Command Sequence for Commands with Write Data (Multiblock Transfer) 1345
Figure 31.16 Example of Operational Flow for Commands with Write Data (Single Block Transfer) 1346
Figure 31.17 (1) Example of Operational Flow for Commands with Write Data (Open-ended Multiblock Transfer) 1347
Figure 31.17 (2) Example of Operational Flow for Commands with Write Data (Open-ended Multiblock Transfer) 1348
Figure 31.18 (1) Example of Operational Flow for Commands with Write Data (Pre-defined Multiblock Transfer) 1349
Figure 31.18 (2) Example of Operational Flow for Commands with Write Data (Pre-defined Multiblock Transfer) 1350

Figure 31.19 (1) Example of Operational Flow for Auto-mode Pre-defined Multiblock Read Transfer 1352
Figure 31.19 (2) Example of Operational Flow for Auto-mode Pre-defined Multiblock Read Transfer 1353
Figure 31.20 (1) Example of Operational Flow for Auto-mode Pre-defined Multiblock Write Transfer 1355
Figure 31.20 (2) Example of Operational Flow for Auto-mode Pre-defined Multiblock Write Transfer 1356
Figure 31.21 Operation Flow to Apply the Card Identification Function 1358

Section 32 PC Card Controller (PCC)

Figure 32.1 PC Card Controller Block Diagram 1360
Figure 32.2 Continuous 32-Mbyte Area Mode 1362
Figure 32.3 Continuous 16-Mbyte Area Mode (Area 6) 1363
Figure 32.4 SH7763 Interface 1380
Figure 32.5 PCMCIA Memory Card Interface Basic Timing 1384
Figure 32.6 PCMCIA Memory Card Interface Wait Timing 1385
Figure 32.7 PCMCIA I/O Card Interface Basic Timing 1386
Figure 32.8 PCMCIA I/O Card Interface Wait Timing 1387
Figure 32.9 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface 1388

Section 33 Audio Codec Interface (HAC)

Figure 33.1 Block Diagram 1392
Figure 33.2 AC97 Frame Slot Structure 1409
Figure 33.3 Initialization Sequence 1412
Figure 33.4 Sample Flowchart for Off-Chip Codec Register Write 1413
Figure 33.5 Sample Flowchart for Off-Chip Codec Register Read (1) 1414
Figure 33.6 Sample Flowchart for Off-Chip Codec Register Read (2) 1415
Figure 33.7 Sample Flowchart for Off-Chip Codec Register Read (3) 1416

Section 34 Serial Sound Interface (SSI)

Figure 34.1 Block Diagram of SSI Module 1420
Figure 34.2 Philips Format (with no Padding) 1439
Figure 34.3 Philips Format (with Padding) 1439
Figure 34.4 Sony Format (with Serial Data First, Followed by Padding Bits) 1440
Figure 34.5 Matsushita Format (with Padding Bits First, Followed by Serial Data) 1440
Figure 34.6 Multichannel Format (2 Channels, No Padding) 1442
Figure 34.7 Multichannel Format (3 Channels with High Padding) 1442
Figure 34.8 Multichannel Format (4 Channels, with Padding Bits First, Followed by Serial Data, with Padding) 1443
Figure 34.9 Basic Sample Format (Transmit Mode with Example System/ Data Word Length) 1444

Figure 34.10 Inverted Clock 1445
Figure 34.11 Inverted Word Select 1445
Figure 34.12 Inverted Padding Polarity 1445
Figure 34.13 Padding Bits First, Followed by Serial Data, with Delay 1446
Figure 34.14 Padding Bits First, Followed by Serial Data, without Delay 1446
Figure 34.15 Serial Data First, Followed by Padding Bits, without Delay 1446
Figure 34.16 Parallel Right Aligned with Delay 1447
Figure 34.17 Mute Enabled 1447
Figure 34.18 Transition Diagram between Operation Modes 1448
Figure 34.19 Transmission Using DMA Controller 1450
Figure 34.20 Transmission using Interrupt Data Flow Control 1451
Figure 34.21 Reception using DMA Controller 1453
Figure 34.22 Reception using Interrupt Data Flow Control 1454

Section 35 USB Host Controller (USBH)

Figure 35.1 Block Diagram of USBH 1458
Figure 35.2 Connection Example of External Circuit 1493

Section 36 USB Function Controller (USBF)

Figure 36.1 Block Diagram of USBF 1496
Figure 36.2 Example of Endpoint Configuration 1554
Figure 36.3 Cable Connection Operation 1559
Figure 36.4 Cable Disconnection Operation 1560
Figure 36.5 Setup Stage Operation 1561
Figure 36.6 Data Stage (Control-In) Operation 1562
Figure 36.7 Data Stage (Control-Out) Operation 1563
Figure 36.8 Status Stage (Control-In) Operation 1564
Figure 36.9 Status Stage (Control-Out) Operation 1565
Figure 36.10 EP1 Bulk-Out Transfer Operation 1566
Figure 36.11 EP2 Bulk-In Transfer Operation 1567
Figure 36.12 EP3 Interrupt-In Transfer Operation 1569
Figure 36.13 EP4 Isochronous-Out Transfer Operation (SOF is Normal) 1570
Figure 36.14 EP4 Isochronous-Out Transfer Operation (SOF is Broken) 1571
Figure 36.15 EP5 Isochronous-In Transfer Operation (SOF is Normal) 1573
Figure 36.16 EP5 Isochronous-In Transfer Operation (SOF in Broken) 1574
Figure 36.17 Forcible Stall by Application 1578
Figure 36.18 Automatic Stall by USB Function Controller 1580
Figure 36.19 Example of Transceiver Connection for USB function Controller 1581
Figure 36.20 Set Timing of TR Interrupt Flag 1584

Section 37 LCD Controller (LCDC)

Figure 37.1 LCDC Block Diagram 1586
Figure 37.2 Valid Display and the Retrace Period 1624
Figure 37.3 Color-Palette Data Format 1628
Figure 37.4 Power-Supply Control Sequence and States of the LCD Module 1634
Figure 37.5 Power-Supply Control Sequence and States of the LCD Module 1634
Figure 37.6 Power-Supply Control Sequence and States of the LCD Module 1635
Figure 37.7 Power-Supply Control Sequence and States of the LCD Module 1635
Figure 37.8 Operation for Hardware Rotation (Normal Mode) 1639
Figure 37.9 Operation for Hardware Rotation (Rotation Mode) 1640
Figure 37.10 Clock and LCD Data Signal Example 1641
Figure 37.11 Clock and LCD Data Signal Example (STN Monochrome 8-Bit Data Bus Module) 1641
Figure 37.12 Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module) 1642
Figure 37.13 Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module) 1642
Figure 37.14 Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module) 1643
Figure 37.15 Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module) 1644
Figure 37.16 Clock and LCD Data Signal Example (DSTN Monochrome 8-Bit Data Bus Module) 1645
Figure 37.17 Clock and LCD Data Signal Example (DSTN Monochrome 16-Bit Data Bus Module) 1646
Figure 37.18 Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module) 1647
Figure 37.19 Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module) 1648
Figure 37.20 Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module) 1649
Figure 37.21 Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module) 1650
Figure 37.22 Clock and LCD Data Signal Example (8-Bit Interface Color 640 × 480) 1651
Figure 37.23 Clock and LCD Data Signal Example (16-Bit Interface Color 640 × 480) 1652

Section 38 A/D Converter

Figure 38.1 Block Diagram of A/D Converter 1656
Figure 38.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected) 1664
Figure 38.3 Example of A/D Converter Operation (Multi Mode, Three Channels AN0 to AN2 Selected) 1665
Figure 38.4 Example of A/D Converter Operation (Scan Mode, Three Channels AN0 to AN2 Selected) 1667
Figure 38.5 Definitions of A/D Conversion Accuracy 1670
Figure 38.6 Example of Analog Input Pin Protection Circuit 1671

Section 39 D/A Converter (DAC)
Figure 39.1 Block Diagram of D/A Converter 1673
Figure 39.2 D/A Converter Operation Example 1677

Section 41 User Break Controller (UBC)
Figure 41.1 Block Diagram of UBC 1760
Figure 41.2 Flowchart of User Break Debugging Support Function 1790

Section 42 User Debugging Interface (H-UDI)
Figure 42.1 H-UDI Block Diagram 1798
Figure 42.2 Sequence for switching from Boundary-Scan TAP Controller to H-UDI 1801
Figure 42.3 TAP Controller State Transitions 1823
Figure 42.4 H-UDI Reset 1824

Section 43 Electrical Characteristics
Figure 43.1 Power-On and Power-Off Timing 1827
Figure 43.2 EXTAL Clock Input Timing 1835
Figure 43.3 CLKOUT Clock Output Timing (1) 1835
Figure 43.4 CLKOUT Clock Output Timing (2) 1836
Figure 43.5 Power-On Oscillation Settling Time 1836
Figure 43.6 PLL Synchronization Settling Time 1837
Figure 43.7 Oscillation Settling Time on Return from Standby NMI or IRQ 1837
Figure 43.8 Reset Input Timing 1837
Figure 43.9 Control Signal Timing 1838
Figure 43.10 Pin Drive Timing in Standby Mode 1839
Figure 43.11 SRAM Bus Cycle: Basic Bus Cycle (No Wait) 1841
Figure 43.12 SRAM Bus Cycle: Basic Bus Cycle (One Wait only by Software) 1842
Figure 43.13 SRAM Bus Cycle: Basic Bus Cycle (One Wait by Software + One Wait by $\overline{\text{RDY}}$, $\overline{\text{RDY}}$ Signal is Synchronous Input) 1843
Figure 43.14 SRAM Bus Cycle: Basic Bus Cycle (No Wait, No Address Setup/ Hold Time Insertion, RDS = 1, RDH = 0, WTS = 1, WTH = 1) 1844
Figure 43.15 Burst ROM Bus Cycle (No Wait) 1845
Figure 43.16 Burst ROM Bus Cycle (1st Data: One Wait by Software + One Wait by $\overline{\text{RDY}}$; 2nd/3rd/4th Data: One Wait only by software) 1846
Figure 43.17 Burst ROM Bus Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0) 1847
Figure 43.18 Burst ROM Bus Cycle (One Wait by Software + One Wait by $\overline{\text{RDY}}$) 1848
Figure 43.19 PCMCIA Memory Bus Cycle 1849
Figure 43.20 PCMCIA I/O Bus Cycle 1850
Figure 43.21 PCMCIA I/O Bus Cycle (TEDA/TEDB = 1, TEHA/TEHB = 1, IW/PCIW = 1, Dynamic Bus Sizing) 1851
Figure 43.22 MPX Basic Bus Cycle: Read 1852

Figure 43.23 MPX Basic Bus Cycle: Write 1853
Figure 43.24 MPX Bus Cycle: Burst Read 1854
Figure 43.25 MPX Bus Cycle: Burst Write 1855
Figure 43.26 Byte Control SRAM Bus Cycle 1856
Figure 43.27 Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0) 1857
Figure 43.28 DDRIF MCLK Output Timing 1859
Figure 43.29 Read Timing of DDR-SDRAM (2 Burst Read) 1859
Figure 43.30 Write Timing of DDR-SDRAM (2 Burst Write) 1860
Figure 43.31 NMI Input Timing 1861
Figure 43.32 IRQ/$\overline{\text{IRL}}$, PINT Input and $\overline{\text{IRQOUT}}$ Output Timing 1862
Figure 43.33 External CPU Interface Read/Write Access Timing 1864
Figure 43.34 PCI Clock Input Timing 1866
Figure 46.35 Output Signal Timing 1866
Figure 43.36 Input Signal Timing 1866
Figure 43.37 $\overline{\text{DREQ}}$, $\overline{\text{TEND}}$, and $\overline{\text{DACK}}$ Timing 1867
Figure 43.38 TCLK Input Timing 1868
Figure 43.39 TPU Output Timing 1869
Figure 43.40 TPU Clock Input Timing 1869
Figure 43.41 MII Transmit Timing (normal operation) 1871
Figure 43.42 MII Receive Timing (normal operation) 1871
Figure 43.43 MII Receive Timing (When an Error is Detected) 1872
Figure 43.44 WOL Output Timing 1872
Figure 43.45 GMII Transmit Timing (normal operation) 1873
Figure 43.46 GMII Receive Timing (normal operation) 1873
Figure 43.47 GMII Receive Timing (When an Error is Detected) 1874
Figure 43.48 WOL Output Timing 1874
Figure 43.49 RMII Transmit Timing 1875
Figure 43.50 RMII Receive Timing (normal operation) 1875
Figure 43.51 RMII Receive Timing (When an Error is Detected) 1876
Figure 43.52 STIF Clock Valid Receive Timing 1877
Figure 43.53 STIF Clock Valid Transmit Timing 1878
Figure 43.54 STIF Strobe Receive Timing 1878
Figure 43.55 STIF Strobe Transmit Timing 1879
Figure 43.56 I^2C Bus Interface Input/Output Timing 1881
Figure 43.57 AC Characteristic Load Condition 1881
Figure 43.58 SCIFn_SCK Input Clock Timing 1882
Figure 43.59 SCIFn I/O Synchronous Mode Clock Timing 1883
Figure 43.60 SIOF_MCLK Input Timing 1884

Figure 43.61 SIOF Transmission/Reception Timing (Master Mode 1, Sampling at the Falling Edge) 1885
Figure 43.62 SIOF Transmission/Reception Timing (Master Mode 1, Sampling at the Rising Edge) 1885
Figure 43.63 SIOF Transmission/Reception Timing (Master Mode 2, Sampling at the Falling Edge) 1886
Figure 43.64 SIOF Transmission/Reception Timing (Master Mode 2, Sampling at the Rising Edge) 1886
Figure 43.65 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2) 1887
Figure 43.66 SIM Module Signal Timing 1888
Figure 43.67 MMCIF Transmit Timing 1889
Figure 43.68 MMCIF Receive Timing (Sampling at the Rising Edge) 1890
Figure 43.69 HAC Cold Reset Timing 1891
Figure 43.70 HAC SYNC Output Timing 1891
Figure 43.71 HAC Clock Input Timing 1892
Figure 43.72 HAC Interface Module Signal Timing 1892
Figure 43.73 SSI Clock Input/Output Timing 1893
Figure 43.74 SSI Transmit Timing (1) 1893
Figure 43.75 SSI Transmit Timing (2) 1894
Figure 43.76 SSI Receive Timing (1) 1894
Figure 43.77 SSI Receive Timing (2) 1894
Figure 43.78 USB Clock Timing 1895
Figure 43.79 LCDC Module Signal Timing 1897
Figure 43.80 GPIO Timing 1897
Figure 43.81 TCK Input Timing 1898
Figure 43.82 $\overline{\text{PRESET}}$ Hold Timing 1899
Figure 43.83 H-UDI Data Transfer Timing 1899
Figure 43.84 $\overline{\text{ASEBRK}}$ Pin Break Timing 1899
Figure 43.85 Output Load Circuit 1901
Figure 43.86 Load Capacitance - Delay Time 1902

Appendix

Figure B.1 Instruction Prefetch 1905
Figure D.1 Schematic Diagram of External Circuits 1908
Figure E.1 Connection Example of Bypass Capacitors for Analog Power Supply 1911
Figure F.1 Package Dimensions (449-Pin) 1912
Figure J.1 Overall View of Simulation Model (with heat sink) 1945
Figure J.2 Heat Sink Model 1946

Tables

Section 1 Overview
Table 1.1 Features of the SH7763 80
Table 1.2 Pin Configuration 94

Section 2 Programming Model
Table 2.1 Initial Register Values 40
Table 2.2 Bit Allocation for FPU Exception Handling 50

Section 3 Instruction Set
Table 3.1 Execution Order of Delayed Branch Instructions 57
Table 3.2 Addressing Modes and Effective Addresses 59
Table 3.3 Notation Used in Instruction List 64
Table 3.4 Fixed-Point Transfer Instructions 66
Table 3.5 Arithmetic Operation Instructions 68
Table 3.6 Logic Operation Instructions 70
Table 3.7 Shift Instructions 71
Table 3.8 Branch Instructions 72
Table 3.9 System Control Instructions 72
Table 3.10 Floating-Point Single-Precision Instructions 75
Table 3.11 Floating-Point Double-Precision Instructions 76
Table 3.12 Floating-Point Control Instructions 76
Table 3.13 Floating-Point Graphics Acceleration Instructions 77

Section 4 Pipelining
Table 4.1 Representations of Instruction Execution Patterns 80
Table 4.2 Instruction Groups 90
Table 4.3 Combination of Preceding and Following Instructions 93
Table 4.4 Issue Rates and Execution Cycles 95

Section 5 Exception Handling
Table 5.1 Register Configuration 105
Table 5.2 States of Register in Each Operating Mode 105
Table 5.3 Exceptions 110

Section 6 Memory Management Unit (MMU)
Table 6.1 Register Configuration 146
Table 6.2 Register States in Each Processing State 146

Section 7 Caches
Table 7.1 Cache Features ... 187
Table 7.2 Store Queue Features ... 187
Table 7.3 Register Configuration ... 190
Table 7.4 Register States in Each Processing State ... 190

Section 8 L Memory
Table 8.1 L Memory Addresses ... 217
Table 8.2 Register Configuration ... 218
Table 8.3 Register Status in Each Processing State ... 219
Table 8.4 Protective Function Exceptions to Access L Memory ... 231

Section 9 Interrupt Controller (INTC)
Table 9.1 Interrupt Types ... 82
Table 9.2 INTC Pin Configuration ... 86
Table 9.3 INTC Register Configuration ... 87
Table 9.4 Register States in Each Operating Mode ... 90
Table 9.5 Interrupt Request Sources and INT2PRI0 to INT2PRI13 ... 113
Table 9.6 RL[3:0], IRL[7:4] Pins and Interrupt Levels ... 140
Table 9.7 Interrupt Exception Handling and Priority ... 143
Table 9.8 Interrupt Response Time ... 154
Table 9.9 Switching Sequence of IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ Pin Function ... 156

Section 10 SuperHyway Bus Bridge (SBR)
Table 10.1 Register Configuration ... 316
Table 10.2 Register State in Each Operating Mode ... 316

Section 11 Local Bus State Controller (LBSC)
Table 11.1 Pin Configuration ... 324
Table 11.2 LBSC External Memory Space Map ... 327
Table 11.3 Setting of bus width for area 0 ... 330
Table 11.4 Correspondence between External Pin (MD5) and Endian ... 331
Table 11.5 PCMCIA Interface Features ... 331
Table 11.6 PCMCIA Support Interface ... 332
Table 11.7 Register Configuration ... 335
Table 11.8 Register State in Each Operating Made. ... 335
Table 11.9 32-Bit External Device/Big-Endian Access and Data Alignment ... 359
Table 11.10 16-Bit External Device/Big-Endian Access and Data Alignment ... 359
Table 11.11 8-Bit External Device/Big-Endian Access and Data Alignment ... 360
Table 11.12 32-Bit External Device/Little-Endian Access and Data Alignment ... 361
Table 11.13 16-Bit External Device/Little-Endian Access and Data Alignment ... 361
Table 11.14 8-Bit External Device/Little-Endian Access and Data Alignment ... 362

Table 11.15 Relationship between Address and CE When Using PCMCIA Interface 380

Section 12 DDR-SDRAM Interface (DDRIF)

Table 12.1 Pin Configuration 81
Table 12.2 Access and Data Alignment in Little Endian Mode (External Bus Width is 32 Bits) 82
Table 12.3 Access and Data Alignment in Big Endian Mode (External Bus Width is 32 Bits) 83
Table 12.4 Register Configuration 85
Table 12.5 Register State in Each Operating Mode 86
Table 12.6 DDR-SDRAM Commands Issued by DDRIF 102
Table 12.7 DDR-SDRAM Address Multiplexing (32-Bit Data Bus) 106

Section 13 PCI Controller (PCIC)

Table 13.1 Input/Output Pins 82
Table 13.2 List of PCIC Registers 85
Table 13.3 Register States in Each Operating Mode 88
Table 13.4 Supported Bus Commands 157
Table 13.5 PCIC Address Map 159
Table 13.6 Interrupt Priority 178

Section 14 Direct Memory Access Controller (DMAC)

Table 14.1 Pin Configuration 81
Table 14.2 Register Configuration of DMAC 83
Table 14.3 State of Registers in Each Operating Mode 85
Table 14.4 Transfer Request Sources 103
Table 14.5 Setting External Request Mode with RS bit 105
Table 14.6 Selecting External Request Detection with DL, DS Bits 106
Table 14.7 Selecting External Request Detection with DO Bit 106
Table 14.8 Selecting On-Chip Peripheral Module Request Modes with Bits RS[3:0] 107
Table 14.9 DMA Transfer Matrix in Auto-Request Mode (all channels) 117
Table 14.10 DMA Transfer Matrix in External Request Mode (only channels 0 to 3) 118
Table 14.11 DMA Transfer Matrix in On-Chip Peripheral module Request Mode 119
Table 14.12 Register Setting for SRAM, Burst ROM, Byte Control SRAM Interface. 131
Table 14.13 Register Setting for PCMCIA Interface 132
Table 14.14 Register Setting for MPX Interface (Read Access) 132
Table 14 15 Register Settings for MPX Interface (Write Access) 132

Section 15 External CPU Interface (EXCPU)

Table 15.1 Pin Configuration 612
Table 15.2 Register Configuration 613
Table 15.3 Register States in Each Operating Mode 613

Table 15.4 Access and Data Alignment for Little Endian 618
Table 15.5 Access and Data Alignment for Big Endian 619

Section 16 Clock Pulse Generator (CPG)
Table 16.1 Pin Configuration and Functions of CPG 82
Table 16.2 Clock Operating Modes 83
Table 16.3 Register Configuration 84
Table 16.4 Register States in Each Operating Mode 84

Section 17 Watchdog Timer and Reset (WDT)
Table 17.1 Pin Configuration 81
Table 17.2 Register Configuration 82
Table 17.3 Register State in Each Operating Mode 82

Section 18 Power-Down Mode
Table 18.1 States in Power-Down Modes 654
Table 18.2 Pin Configuration 655
Table 18.3 Register Configuration 656
Table 18.4 Register States in Each Operating Mode 656
Table 18.5 Pin Configuration 670

Section 19 Timer Unit (TMU)
Table 19.1 Pin Configuration 675
Table 19.2 Register Configuration 676
Table 19.3 Register States in Each Operating Mode 677
Table 19.4 TMU Interrupt Sources 690

Section 20 16-Bit Timer Pulse Unit (TPU)
Table 20.1 TPU Functions 694
Table 20.2 TPU Pin Configurations 696
Table 20.3 Register Configuration 697
Table 20.4 Register State in Each Operating Mode 699
Table 20.5 TPU Clock Sources 702
Table 20.6 TPSC[2:0] (1) 703
Table 20.6 TPSC[2:0] (2) 703
Table 20.6 TPSC[2:0] (3) 703
Table 20.6 TPSC[2:0] (4) 704
Table 20.7 IOA[2:0] 708
Table 20.8 Register Combinations in Buffer Operation 720
Table 20.9 Phase Counting Mode Clock Input Pins 726
Table 20.10 Up/Down-Count Conditions in Phase Counting Mode 1 728
Table 20.11 Up/Down-Count Conditions in Phase Counting Mode 2 729
Table 20.12 Up/Down-Count Conditions in Phase Counting Mode 3 730

Table 20.13 Up/Down-Count Conditions in Phase Counting Mode 4 731

Section 21 Compare Match Timer (CMT)
Table 21.1 Register Configuration 735
Table 21.2 Register State in Each Operating Mode 736

Section 22 Realtime Clock (RTC)
Table 22.1 RTC Pins 747
Table 22.2 Register Configuration 748
Table 22.3 Register State in Each Operating Mode 749
Table 22.4 Crystal Oscillator Circuit Constants (Recommended Values) 767
Table 22.5 Interrupt source and request generating order 768

Section 23 Gigabit Ethernet Controller (GETHER)
Table 23.1 Pin Configuration 81
Table 23.2 Register Configuration 86
Table 23.3 Register States in Each Operating Mode 94
Table 23.4 Relay Frame Process (Without CAM) 264
Table 23.5 Receive Frame Processing 266
Table 23.6 Relay Frame Process (With CAM) 266
Table 23.7 List of GETHER Interrupts 271

Section 25 Stream Interface (STIF)
Table 25.1 Pin Configuration 985
Table 25.2 Register Configuration 986
Table 25.3 Register States in Each Operating Mode 987

Section 26 I^2C Bus Interface (IIC)
Table 26.1 Pin Configuration 80
Table 26.2 Register Configuration 80
Table 26.3 Register State in Each Operating Mode 82
Table 26.4 Suggested Settings for CDF and SCGD 97
Table 26.5 Description on Symbols of I^2C Bus Data Format 100

Section 27 Serial Communication Interface with FIFO (SCIF)
Table 27.1 Pin Configuration 1051
Table 27.2 Register Configuration (1) 1052
Table 27.3 Register State in Each Operating Mode 1053
Table 27.4 SCSMR Settings 1069
Table 27.5 SCSMR Settings for Serial Transfer Format Selection 1079
Table 27.6 SCSMR and SCSCR Settings for SCIF Clock Source Selection 1079
Table 27.7 Serial Transfer Formats (Asynchronous Mode) 1081
Table 27.8 SCIF Interrupt Sources 1100

Section 28 Serial Communication Interface with FIFO/IrDA Interface (SCIF/IrDA)
Table 28.1 Pin Configuration ... 1109
Table 28.2 Register Configuration ... 1110
Table 28.3 Register States in each Operation Mode ... 1111
Table 28.4 SCSMR Settings ... 1127
Table 28.5 Baud Rate (3.6864 MHz Clock) ... 1134
Table 28.6 SCSMR Settings for Serial Transfer Format Selection ... 1138
Table 28.7 SCSMR and SCSCR Settings for SCIF Clock Source Selection ... 1139
Table 28.8 Serial Transfer Formats (Asynchronous Mode) ... 1142
Table 28.9 SCIF Interrupt Sources ... 1161

Section 29 Serial I/O with FIFO (SIOF)
Table 29.1 Pin Configuration ... 1173
Table 29.2 Register Configuration ... 1174
Table 29.3 Register State in Each Operating Mode ... 1176
Table 29.4 Operation in Each Transfer Mode ... 1180
Table 29.5 SIOF Serial Clock Frequency ... 1204
Table 29.6 Serial Transfer Modes ... 1207
Table 29.7 Frame Length ... 1207
Table 29.8 Audio Mode Specification for Transmit Data ... 1209
Table 29.9 Audio Mode Specification for Receive Data ... 1210
Table 29.10 Setting Number of Channels in Control Data ... 1211
Table 29.11 Conditions to Issue Transmit Request ... 1214
Table 29.12 Conditions to Issue Receive Request ... 1214
Table 29.13 Transmit and Receive Reset ... 1219
Table 29.14 SIOF Interrupt Sources ... 1220

Section 30 SIM Card Module (SIM)
Table 30.1 Pin Configuration ... 1229
Table 30.2 Register Configuration ... 1230
Table 30.3 Register State in Each Operating Mode ... 1231
Table 30.4 Register Settings for Smart Card Interface ... 1254
Table 30.5 Example of Bit Rates (bits/s) for SCBRR Settings (Pck0 = 66.6 MHz, SCSMPL = 371) ... 1256
Table 30.6 Interrupt Sources of Smart Card Interface ... 1262

Section 31 Multimedia Card Interface (MMCIF)
Table 31.1 Pin Configuration ... 1273
Table 31.2 Register Configuration ... 1274
Table 31.3 Register States in Each Processing Mode ... 1276
Table 31.4 Correspondence between Commands and CMDTYR and RSPTYR Settings ... 1281
Table 31.5 CMDR Configuration ... 1284

Table 31.6 Correspondence between Command Response Byte Number and RSPR 1286
Table 31.7 List of Chattering Elimination Pulse Cycles 1311
Table 31.8 MMCIF Interrupt Sources 1343

Section 32 PC Card Controller (PCC)
Table 32.1 Features of the PCMCIA Interface 1347
Table 32.2 PCC Pin Configuration 1350
Table 32.3 Register Configuration 1351
Table 32.4 Register State in Each Operating Mode 1351
Table 32.5 PCMCIA Support Interface 1367

Section 33 Audio Codec Interface (HAC)
Table 33.1 Pin Configuration 1378
Table 33.2 Register Configuration 1379
Table 33.3 Register State in Each Operating Mode 1379
Table 33.4 AC97 Transmit Frame Structure 1395
Table 33.5 AC97 Receive Frame Structure 1396

Section 34 Serial Sound Interface (SSI)
Table 34.1 Pin Configuration 81
Table 34.2 Register Configuration 82
Table 34.3 Register State in Each Operating Mode 83
Table 34.4 Bus Formats of SSI Module 97
Table 34.5 Number of Padding Bits for Each Valid Configuration 101

Section 35 USB Host Controller (USBH)
Table 35.1 USB Host Pin Assignment 1445
Table 35.2 Register Configuration 1446
Table 35.3 Register State in Each Operating Mode 1447

Section 36 USB Function Controller (USBF)
Table 36.1 Pin Configuration and Functions 81
Table 36.2 (1) Register Configuration (Access Size = 8 bits) 82
Table 36.2 (2) Register Configuration (Access Size = 32 bits) 84
Table 36.3 Register State in Each Operating Mode 86
Table 36.4 Restrictions of Settable Values 137
Table 36.5 Example of Endpoint Configuration 137
Table 36.6 Example of Setting of Endpoint Configuration Information 138
Table 36.7 Command Decoding on Application Side 160

Section 37 LCD Controller (LCDC)
Table 37.1 Pin Configuration 81
Table 37.2 Register Configuration 82

Table 37.3 Register State in Each Operating Mode 83
Table 37.4 I/O Clock Frequency and Clock Division Ratio 86
Table 37.5 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit SDRAM) 119
Table 37.6 Available Power-Supply Control-Sequence Periods at Typical Frame Rates 130
Table 37.7 LCDC Operating Modes 131
Table 37.8 LCD Module Power-Supply States 131

Section 38 A/D Converter
Table 38.1 Pin Configuration 1643
Table 38.2 Register Configuration 1644
Table 38.3 Register State in Each Operating Mode 1644
Table 38.4 Analog Input Channels and A/D Data Registers 1645
Table 38.5 A/D Conversion Time 1654
Table 38.6 Relationship between Clock Division Ratio and Usable Pck0 Clock Frequency 1658

Section 39 D/A Converter (DAC)
Table 39.1 Pin Configuration 1660
Table 39.2 Register Configuration 1660
Table 39.3 Register State in Each Operating Mode 1660

Section 40 General Purpose I/O (GPIO)
Table 40.1 Multiplexed Pins Controlled by Port Control Registers 80
Table 40.2 Register Configuration (1) 90
Table 40.3 Register States in Each Operating Mode 92

Section 41 User Break Controller (UBC)
Table 41.1 Register Configuration 1747
Table 41.2 Register Status in Each Processing State 1748
Table 41.3 Settings for Match Data Setting Register 1762
Table 41.4 Relation between Operand Sizes and Address Bits to be Compared 1771

Section 42 User Debugging Interface (H-UDI)
Table 42.1 Pin Configuration 1785
Table 42.2 Commands Supported by Boundary-Scan TAP Controller 1787
Table 42.3 Register Configuration (1) 1788
Table 42.4 Register Configuration (2) 1788
Table 42.5 Register Status in Each Processing State 1788
Table 42.6 SDBSR Configuration 1792

Section 43 Electrical Characteristics
Table 43.1 Absolute Maximum Ratings 79
Table 43.2 Power-On and Power-Off Timing 81

Table 43.3 DC Characteristics (1) [common] 83
Table 43.4 DC Characteristics (2-a) [Except of USB Transceiver and I2C Related Pins] 84
Table 43.5 DC Characteristics (2-b) [I2C Related Pins] 85
Table 43.6 DC Characteristics (2-c) [USB Transceiver Related Pins] 86
Table 43.7 Permissible Output Currents 86
Table 43.8 Maximum Operating Frequency 87
Table 43.9 Clock and Control Signal Timing 88
Table 43.10 Control Signal Timing 92
Table 43.11 Bus Timing 94
Table 43.12 DDRIF Signal Timing 112
Table 43.13 INTC Module Signal Timing 115
Table 43.14 External CPU Interface Access Timing 117
Table 43.15 PCIC Signal Timing 119
Table 43.16 DMAC Module Signal Timing 121
Table 43.17 TMU Module Signal Timing 122
Table 43.18 16-bit Timer Pulse Unit 123
Table 43.19 Ethernet Controller Signal Timing (MII) 124
Table 43.20 Ethernet Controller Signal Timing (GMII) 126
Table 43.21 Ethernet Controller Signal Timing (RMII) 128
Table 43.22 STIF Clock Valid Reception Signal Timing 130
Table 43.23 STIF Clock Valid Transmission Signal Timing 131
Table 43.24 STIF Strobe Reception Signal Timing 132
Table 43.25 STIF Strobe Transmission Signal Timing 133
Table 43.26 I2C Bus Interface Timing 134
Table 43.27 SCIF Module Signal Timing 136
Table 43.28 SIOF Module Signal Timing 138
Table 43.29 SIM Module Signal Timing 142
Table 43.30 MMCIF Module Signal Timing 143
Table 43.31 HAC Interface Module Signal Timing 145
Table 43.32 SSI Interface Module Signal Timing 147
Table 43.33 USB Module Clock Timing 149
Table 43.34 USB Electrical Characteristics (Full-Speed) 149
Table 43.35 USB Electrical Characteristics (Low-Speed) 150
Table 43.36 LCDC Module Signal Timing 150
Table 43.37 GPIO Signal Timing 151
Table 43.38 H-UDI Module Signal Timing 152
Table 43.39 A/D Converter Characteristics 154
Table 43.40 D/A Converter Characteristics 154

Appendix

Table D.1 Mode Control Pins 83

Table G.1 Pin States 89
Table H.1 Handling of Unused Pins 107
Table I.1 Register Configuration 119
Table J.1 Heat Resistance Simulation Results 121

Section 1 Overview

1.1 Features of the SH7763

This LSI is a single-chip multifunction CMOS microcomputer that integrates the Renesas Technology original RISC (reduced instruction set computer) CPU core with the peripheral functions required for a wide range of application systems such as high-speed Ethernet, display, and digital AV systems.

The SH-4A is upwardly compatible with the SH-1, SH-2, SH-3, and SH-4 microcompututers at the instruction set level. This microprocessor core integrates a cache memory and the MMU.

The CPU of this LSI has a RISC instruction set and basically operates on a one-cycle-per-instruction basis, which dramatically reduces the instruction execution time. The internal 32-bit configuration enhances the data processing capability. The CPU of this LSI allows high-performance, highly-functional systems to be built at lower cost even for applications that require high speeds which cannot be implemented with conventional microcomputers.

With this LSI, the on-chip DMAC (direct memory access controller) permits high-speed data transfer and the external memory access support function permits direct connection to various types of memory device. Furthermore, the LSI also incorporates powerful peripheral functions which are optimal for system configuration, such as the LCD controller, USB full-speed host controller, function controller, PCI controller, high-speed asynchronous serial communication interface, serial interface for a voice/audio codec, A/D converter, and D/A converter.

The LSI has a two-channel gigabit Ethernet controller that includes an IEEE802.3z-compliant media access controller (MAC) and a gigabit media-independent interface (GMII) standard unit, which allows implementation of 10/100/1000-Mbps LAN connections. In addition, the inclusion of the security accelerator allows efficient security control for the data on the network.

The external memory access support function permits direct connection to normal memory, DDR-SDRAM, and PCMCIA.

Because of the various built-in functions that can be applied to a wide range of fields, use of this LSI not only enables dramatic cost reduction of user-developed systems, but also enables downsizing and low power consumption.

Table 1.1 Features of the SH7763

Item	Features
Maximum operating frequency	• 266 MHz
Performance	• 478 MIPS (266 MHz), 1862 MFLOPS (266 MHz)
CPU	• Renesas Technology original architecture • 32-bit internal data bus • General-register files: — Sixteen 32-bit general registers (eight 32-bit shadow registers) — Seven 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatible with the SH-1, SH-2, SH-3 and SH-4 microcomputers) — Instruction length: 16-bit fixed length for improved code efficiency — Load/store architecture — Delayed branch instructions — Instructions executed with conditions — Instruction set based on the C language • Super scalar which executes two instructions simultaneously including the FPU • Instruction execution time: Two instruction per cycle (max) • Virtual address apace: 4 Gbytes • Space identifier ASID: 8 bits, 256 virtual address spaces • On-chip multiplier • Seven-stage pipeline

Item	Features
FPU	• On-chip floating-point coprocessor • Supports single-precision (32 bits) and double-precision (64 bits) • Supports IEEE754-compliant data types and exceptions • Two rounding modes: Round to Nearest and Round to Zero • Handling of denormalized numbers: Truncation to zero or interrupt generation for IEEE754 compliance • Floating-point registers: 32 bits × 16 words × 2 banks (single-precision × 16 words or double-precision × 8 words) × 2 banks • 32-bit CPU-FPU floating-point communication register (FPUL) • Supports FMAC (multiply-and-accumulate) instruction • Supports FDIV (divide) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution times — Latency (FADD/FSUB): 3 cycles (single-precision), 5 cycles (double-precision) — Latency (FMAC/ FMUL): 5 cycles (single-precision), 7 cycles (double-precision) — Pitch (FADD/FSUB): 1 cycle (single-precision/double-precision) — Pitch (FMAC/FMUL): 1 cycle (single-precision), 3 cycles (double-precision) Note: FMAC is supported for single-precision only. • 3-D graphics instructions (single-precision only): — 4-dimensional vector conversion and matrix operations (FTRV): 4 cycles (pitch), 8 cycles (latency) — 4-dimensional vector (FIPR) inner product: 1 cycle (pitch), 5 cycles (latency) • Ten-stage pipeline

Item	Features
Memory management unit (MMU)	• 4 Gbytes of physical address space, 256 address spaces (identified by an 8-bit ASID (address space identifier)) • Supports single virtual memory mode and multiple virtual memory mode • Supports multiple page sizes: 1 kbyte, 4 kbytes, 64 kbytes, or 1 Mbytes • 4-entry full associative TLB for instructions • 64-entry full associative TLB for instructions and operands • Supports software-controlled replacement and random-counter replacement algorithms • Contents of TLB are directly accessible through address mapping
Cache memory	• Instruction cache (IC) — 32-Kbyte 4-way set associative — 32-byte block length • Operand cache (OC) — 32-Kbyte 4-way set associative — 32-byte block length — Selectable write method (copy-back or write-through) • Storage queue (32 bytes × 2 entries)
LRAM	• High-speed memory (16 Kbytes) • Two independent read/write ports — 8-/16-/32-/64-bit access from the CPU/FPU — 8-/16-/32-/64-bit access from the DMAC • Supports memory protective mechanism
User break controller (UBC)	• Supports debugging by means of user break interrupts • Two break channels • Address, data value, access type, and data size are available as break condition settings • Supports sequential break functions

Item	Features
Clock pulse generator (CPG)	• Selectable CPU clock: 8 times EXTAL • Clock modes: — CPU frequency: 266 MHz (max.) — Local bus frequency: 1/4 times the CPU clock 66 MHz (max.) — DDR-SDRAM I/F frequency: 1/2 times the CPU clock 133 MHz (max.) — Peripheral bus 0 frequency: 1/4 times the CPU clock 66 MHz (max.) — Peripheral bus 1 frequency: 1/8 times the CPU clock 33 MHz (max.) • Support of power-down modes: — Sleep mode — Software standby mode — Module standby mode — RTC power supply backup mode — DDR-SDRAM power supply backup mode • Single-channel watchdog timer
Interrupt controller (INTC)	• Direct jump mode (SH4 compatible) • External interrupt pins: NMI, $\overline{\text{IRL7}}$ to $\overline{\text{IRL0}}$, IRQ7 to IRQ0, and PINT15 to PINT0 • On-chip peripheral module interrupts: Priority level can be set for each module

Item	Features
Local bus state controller (LBSC)	• Physical address space divided into seven areas (areas 0 to 6), each comprising up to 64 Mbytes — I/F configuration, bus width, and wait cycle insertion are settable for each area • SRAM interface — Wait cycle insertion by register setting — Wait cycle insertion by the $\overline{\text{RDY}}$ pin — Supported bus width: 8, 16, or 32 bits — Supported space: Areas 0 to 2 and areas 4 to 6 • Burst ROM interface — Wait cycle insertion by register setting — Number of bursts is specified by register setting — Supported bus width: 8, 16, or 32 bits — Supported space: Areas 0, 5, and 6 • Interface for SRAM with byte selection — Supports direct connection to SRAM with byte selection — Supported space: Areas 1 and 4 • PCMCIA interface (only supported in little endian mode) — Wait cycle insertion by register setting — Supports ATAPI interface (multi-word DMA supported) — Supports I/O bus-width sizing — Supported space: Areas 5 and 6

Item	Features
DDR-SDRAM controller (DDRIF)	• DDR-SDRAM interface: 32-bit data bus width • Supports the DDR266 or DDR200 SDRAM • DDR-SDRAM refreshing — Programmable refreshing intervals (auto-refresh mode) — Self-refresh mode • Supports a burst length of 2 • Switching of big/little endian for external memory access is possible at power-on reset • Capacity and bit width (bits) of connectable memory devices — 128-Mbit DDR-SDRAM (×16), two chips in parallel connection — 256-Mbit DDR-SDRAM (×16), two chips in parallel connection — 512-Mbit DDR-SDRAM (×16), two chips in parallel connection — 1-Gbit DDR-SDRAM (×16), two chips in parallel connection
PCI controller (PCIC)	• PCI controller (Rev.2.2-compatible) — 32-bit bus — 33 MHz/66 MHz • Supports PCI master/slave • Supports the PCI host function — Built-in bus arbiter • External input pin for clock exclusively used by the PCI bus • Interrupt requests can be sent to CPU
Direct memory access controller (DMAC)	• Six channels (four channels support external requests) • Transfer data size: Byte, word (2 bytes), longword (4 bytes), 16 or 32 bytes • Maximum number of transfers: 16,777,216 • Address mode: Dual address mode • Bus modes: Selectable from cycle-steal and burst modes • Transfer requests: Selectable from external request (channels 0 to 3 only), on-chip peripheral module request, and auto-request mode • Priority: Selectable from fixed channel priority mode and round-robin mode

Item	Features
Timer unit (TMU)	• 6-channel auto-reload 32-bit timer • Input-capture function (channels 2 and 5 only) • Choice of seven types of counter input clock for each channel — External clock (TCLK), five peripheral clocks (Pck0/4, Pck0/16, Pck0/64, Pck0/256, Pck0/1024), and RTC clock
Compare match timer (CMT)	• Includes 32-bit counters for four channels (16-bit/32-bit selectable) • An interrupt or DMA transfer request can be generated on compare match or an overflow
Realtime clock (RTC)	• On-chip clock, calendar function and alarm function • Built-in 32-kHz crystal oscillator with a maximum resolution of 1/256 seconds (interrupt cycle) • Alarm, periodic, and carry interrupt requests
Serial communication interface (SCIF)	• Each channel includes 64-byte transmit/receive FIFOs • Three channels (SCIF0, SCIF1, SCIF2) • Full-duplex communication • Modem control function (RTS/CTS) available in asynchronous mode (channels 0 and 1) • Transmit/receive clock source is selectable as either the internal clock from the baud-rate generator or the clock externally input from the SCK pin • Channel 2 includes an IrDA 1.0-compliant interface
Serial I/O with FIFO (SIOF)	• Each channel includes 64-byte transmit/receive FIFOs • Three channels (SIOF0, SIOF1, SIOF2) • Supports 8-/16-bit mono and 16-bit stereo audio input/output • Sampling rate clock input selectable from Pck0 and external pin • Includes a prescaler
Multimedia card interface (MMCIF)	• Supports MMC mode • A maximum bit rate of 16.7 Mbps at 33 MHz of peripheral clock 1 • Interface is through the MCCLK output pin for transfer clock output, MCCMD I/O pin for command output/response input, MCDAT I/O pin for data input/output • Four interrupt sources

Item	Features
Serial sound interface (SSI)	• Four channels (SSI0, SSI1, SSI2, SSI3) • Supports various serial audio formats • Supports master/slave functions • Programmable word clock and bit clock generation • Multi-channel format • Supports 8/16/18/20/22/24/32-bit data formats
Debug interface	• H-UDI (User Debugging Interface) • AUD (Advanced User Debugger)
USB host interface (OHCI USBH)	• Data transfer rates of 1.5 Mbps and 12 Mbps • OHCI version 1.0
USB function interface Version 2.0 (USBF)	• Includes a USB device controller (UDC) supporting USB2.0 — Automatic processing of standard USB commands (some commands are not included: Get Descriptor/Class/Vector commands are processed on the microcomputer's firmware) • Transfer rate: Full speed (12 Mbps only)
Audio codec interface (HAC)	• Digital interface for audio codec (single channel) • Supports transmission/reception for slot 1 to slot 4 • Choice of 16- or 20-bit DMA transfer in transmission/reception • Supports various sampling rates by adjusting slot data • Generates data ready, data request, overflow, and underflow interrupts
I^2C bus interface (IIC)	• Supports the I^2C bus interface protocol • Two channels (IIC0, IIC1) • Master/slave functions • Multi-master function • A maximum transfer rate of 400 Kbps • Programmable clock generation from the system clock
A/D converter (ADC)	• 10 bits ± 4LSB, four channels • Conversion time: 8.5 μs • Three conversion modes: single mode, multi-mode, and scan mode • Four data registers • Sample-and-hold function • Generates A/D conversion end interrupts • Input range: 0 to Avcc (max. 3.6 V)

Item	Features
D/A converter (DAC)	• 8 bits ± 4LSB, two channels • Conversion time: 10 μs • Two data registers • Output range: 0 to Avcc (max. 3.6 V)
LCD controller (LCDC)	• Display size: 16 × 1 pixels to 1024 × 1024 pixels • Color display of 4, 8, 15, or 16 bpp (bits per pixel) • Grayscale display of 1, 2, 4, or 6 bpp (bits per pixel) • 8-bit frame controller • Supported LCD panels: TFT, DSTN, and STN • Signal polarity selection • Hardware panel rotation • Power control function • Selectable clock source: Peripheral clock or external clock
PC card controller (PCC)	• Supports control signals for one slot • Compatible with the SH7709 when PCC operation is disabled (two slots)
SIM card interface (SIM)	• One channel. Conforms to the ISO 7816-3 data protocol. (T = 0, T = 1) • Asynchronous half-duplex character transmission protocol • Data length: 8 bits • Parity bit generation and check • Selectable output clock cycles per etu (elementary time unit) • Selectable direct convention/inverse convention • Built-in prescaler using Pck0 • Changeable clock polarity (high or low) in idle state • Interrupt request and DMAC request
16-bit timer pulse unit (TPU)	• Up to four pulse outputs • Up to four-phase PWM outputs
I/O ports (GPIO)	• Bidirectional port pins can be configured as input or output by setting the corresponding bits

Item	Features
Stream interface (STIF)	• Parallel connection available when MPEG2 TS stream is input — Parallel stream connection — Stream input: Master mode with clock-valid operation Byte transfer mode with strobe operation — Stream output: Master mode with clock-valid operation Byte transfer mode with strobe operation • Two channels (STIF0, STIF1)
Gigabit Ethernet controller (GETHER)	• E-DMAC (DMAC dedicated for Ethernet) — Four channels — Transfer between GETHER and external/internal memory — 32-byte burst transfer — Supports the one-descriptor-for-one-frame and multi-descriptors-for-one-frame (multi-buffer) transfer methods — Transfer data width: 32 bits — Includes FIFOs (2 Kbytes for transmission, 8 Kbytes for reception) • MAC (Media Access Control) — Two channels (GETHER0, GETHER1) — Data frame composition/decomposition (IEEE802.3, 2000 Edition-compliant frame format) — Changeable transfer rate: 10, 100, or 1000 Mbps — Full-duplex/half-duplex transmission/reception — IEEE802.3x-compliant flow control available. Automatic/manual transmission of PAUSE frames for flow control — Supports IEEE802.1Q (VLAN) — Supports IEEE802.3-compliant PHY interface GMII (Gigabit Media Independent Interface), MII (Media Independent Interface), and RMII (Reduced Media Independent Interface) — Support for higher-layer protocols (sum check) — Includes a switching unit for inter-channel transfer (transfer FIFO: 6 Kbytes)

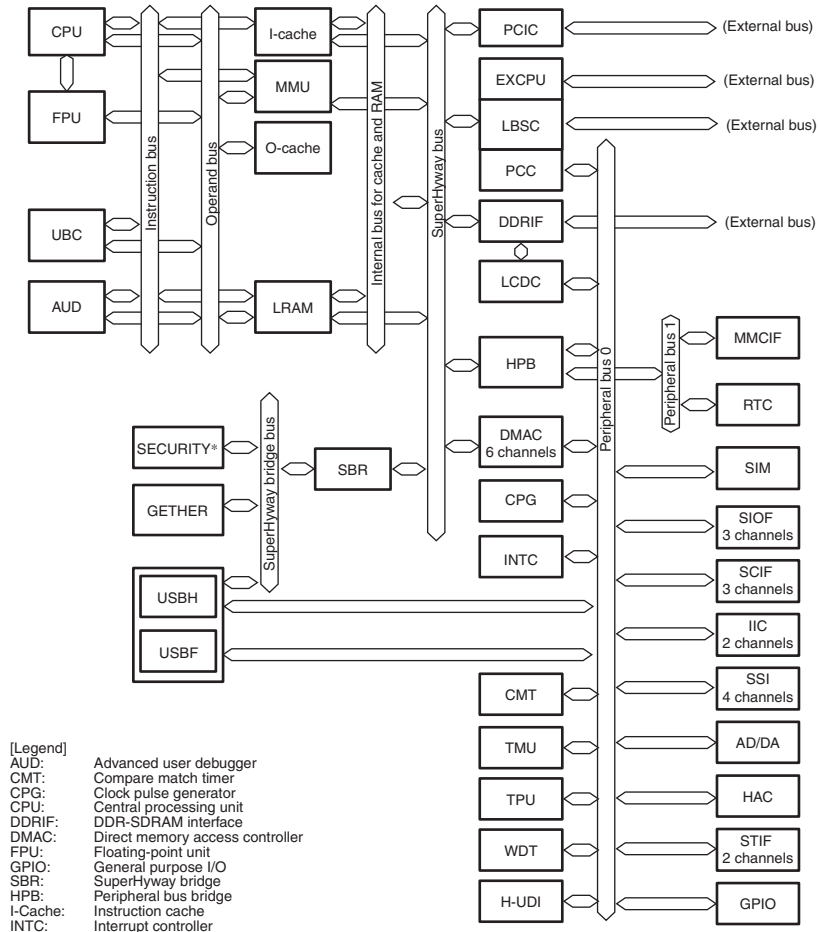
Item	Features
Security accelerator*1 (SECURITY)	• Encryption/decryption based on AES (Advanced Encryption Standard) (Key length: 128, 192, and 256 bits) • DES/Triple-DES encryption/decryption based on DES (Data Encryption Standard) • Hash function generation based on the MD5 (Message-Digest Algorithm) • Hash function generation based on sha-1 of the Source Hash Standard • Includes a dedicated DMAC for data transfer • Interrupt requests to the CPU
Package	• P-FBGA2121-449 (BGA – 449 pin (21 × 21 mm))
Power-supply voltage	• 3.3 V ±0.3 V, 1.25 V ±0.1 V, 2.5 V ±0.2 V (for DDR-SDRAM)
Temperature range	• -20 to +75°C*2
Process	• 0.13-μm CMOS, 5 metal layers

Product lineup

Abbrev.	Power Supply	Operating Frequency	Product Type	Package
R5S77630	3.3 V ±0.3 V 1.25 V ±0.1 V 2.5 V ±0.2 V	133 MHz	R5S77630Y266BGV	BGA-499 pin
R5S77631			R5S77631Y266BGV	

Notes: 1. The security accelerator is incorporated only in the R5S77630, not in the R5S77631.
2. Note that a heat radiation countermeasure, such as heat sinks, is required when the ambient temperature exceeds 60 degrees.

1.2 Block Diagram



[Legend]

AUD:	Advanced user debugger
CMT:	Compare match timer
CPG:	Clock pulse generator
CPU:	Central processing unit
DDRIF:	DDR-SDRAM interface
DMAC:	Direct memory access controller
FPU:	Floating-point unit
GPIO:	General purpose I/O
SBR:	SuperHyway bridge
HPB:	Peripheral bus bridge
I-Cache:	Instruction cache
INTC:	Interrupt controller
LBSC:	Local bus state controller
PCC:	PC card controller
LRAM:	L memory
MMCIF:	Multimedia card interface
MMU:	Memory management unit
O-Cache:	Operand (data) cache
PCIC:	PCI controller
EXCPU:	External CPU interface
RTC:	Realtime clock
UBC:	User break controller
H-UDI:	User debugging interface
GETHER:	Gigabit Ethernet controller
SECURITY*:	Security accelerator
USBH:	USB host controller
USBF:	USB function controller
SCIF:	Serial communication interface with FIFO
SIOF:	Serial I/O with FIFO
SSI:	Serial sound interface
STIF:	Stream interface
HAC:	Audio codec interface
AD/DA:	A/D converter, D/A converter
TMU:	Timer unit
IIC:	IIC bus interface
WDT:	Watchdog timer
SIM:	SIM card module
TPU:	16-bit pulse unit
LCDC:	LCD controller

Note: * SECURITY is incorporated only in the R5S77630, not in the R5S77631.

Figure 1.1 SH7763 Block Diagram

1.3 Pin Arrangement

Figure 1.2 shows the pin arrangement and table 1.2 lists the pin configuration of this LSI.

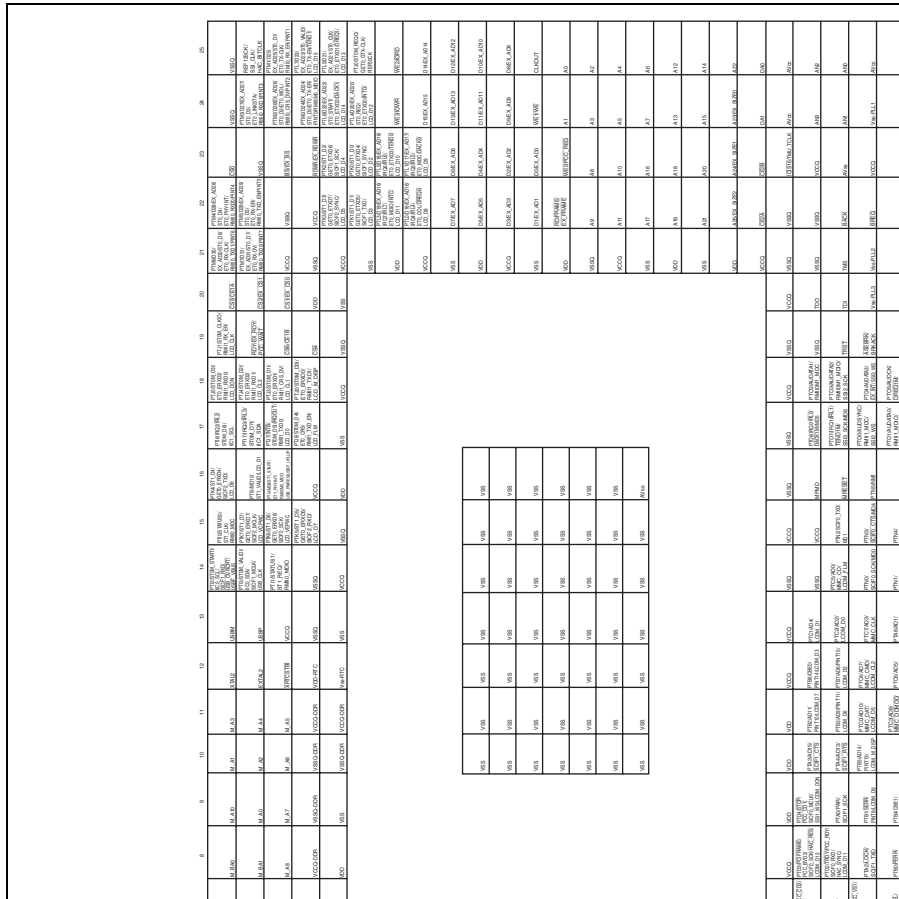


Figure 1.2 Pin Arrangement

Table 1.2 Pin Configuration

Pin No.	Pin Name	I/O	Function	Power Supply
A1	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
A2	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
A3	M_VREF	I	DDR-SDRAM VREF	VCCQ_DDR
A4	M_CLK0	O	DDR-SDRAM clock	VCCQ_DDR
A5	$\overline{\text{M_CLK1}}$	O	DDR-SDRAM clock	VCCQ_DDR
A6	$\overline{\text{M_WE}}$	O	DDR-SDRAM write enable	VCCQ_DDR
A7	$\overline{\text{M_RAS}}$	O	DDR-SDRAM RAS	VCCQ_DDR
A8	M_BA0	O	DDR-SDRAM bank active	VCCQ_DDR
A9	M_A10	O	DDR-SDRAM address bus	VCCQ_DDR
A10	M_A1	O	DDR-SDRAM address bus	VCCQ_DDR
A11	M_A3	O	DDR-SDRAM address bus	VCCQ_DDR
A12	XTAL2	O	Crystal resonator for RTC	VDD_RTC
A13	USBM	IO	D-	VCCQ
A14	PTI2/ST0M_STARTI/IIC0_SCL/ SIOF1_RXD/$\overline{\text{USB_OVRCRT}}$/ USBF_VBUS	I/I/IO/I/I/I	Port/ST data sync/IIC serial clock/SIOF receive data/USB over-current detection/USB cable connection monitor pin	VCCQ
A15	PTI0/STATUS0/ST1_CLK/ RMII0_MDC	IO/O/IO/O	Port/status 0/ST data clock/ RMII management data clock	VCCQ
A16	PTK4/ST1_D4/GET0_ERXD4/ SIOF2_TXD/LCD_D6	IO/IO/I/O/O	Port/ST data/ETHER receive data/ SIOF transmit data/LCD data	VCCQ
A17	PTI6/IRQ2/$\overline{\text{IRL2}}$/ST0M_D6I/ IIC1_SCL	I/I/I/I/IO	Port/external interrupt input/ST data (mirror pin)/IIC serial clock	VCCQ
A18	PTJ5/ST0M_D3I/ET0_ERXD3/ RMII1_RXD0/LCD_DON	IO/I/I/I/O	Port/ST data (mirror pin)/Ether receive data/RMII receive data/LCD display start	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
A19	PTJ1/ST0M_CLKIO/ RMII1_RX_ER/LCD_CLK	IO/IO/I/I	Port/ST data clock RMII receive error/LCD clock source	VCCQ
A20	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	O/O	Chip select/Card select	VCCQ
A21	PTM6/D30/EX_AD30/ST0_D6/ ET0_RX-CLK/RMII0_TXD1/ PINT6	IO/IO/IO/IO/I/O/I	Port/data bus/address-and-data bus/ST data/ETHER receive clock/ RMII transmit data/port interrupt input	VCCQ
A22	PTM4/D28/EX_AD28/ST0_D4/ ET0_PHY-INT/RMII0_RXD0/ PINT4	IO/IO/IO/IO/I/I/I	Port/data bus/address-and-data bus/ST data/ PHY interrupt/RMII receive data/port interrupt input	VCCQ
A23	$\overline{\text{CS0}}$	O	Chip select	VCCQ
A24	VSSQ	—	I/O GND	—
A25	VSSQ	—	I/O GND	—
B1	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
B2	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
B3	$\overline{\text{M_BKPRST}}$	I	DDR-SDRAM power supply backup reset	VCCQ_ DDR
B4	M_CKE	O	DDR-SDRAM clock enable	VCCQ_ DDR
B5	M_A13	O	DDR-SDRAM address bus	VCCQ_ DDR
B6	$\overline{\text{M_CAS}}$	O	DDR-SDRAM CAS	VCCQ_ DDR
B7	$\overline{\text{M_CS}}$	O	DDR-SDRAM chip select	VCCQ_ DDR
B8	M_BA1	O	DDR-SDRAM bank active	VCCQ_ DDR
B9	M_A0	O	DDR-SDRAM address bus	VCCQ_ DDR
B10	M_A2	O	DDR-SDRAM address bus	VCCQ_ DDR
B11	M_A4	O	DDR-SDRAM address bus	VCCQ_ DDR
B12	EXTAL2	I	Crystal resonator for RTC	VDD_RTC
B13	USBP	IO	D+	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
B14	PTI3/ST0M_VALIDI/IIC0_SDA/ SIOF1_MCLK/USB_CLK	I/I/IO/I/I	Port/ST data valid (mirror pin)/IIC serial data/ SIOF master clock/USB clock input	VCCQ
B15	PTK7/ST1_D7/GET0_ERXD7/ SIOF2_MCLK/LCD_VCPWC	IO/IO/I/I/O	Port/ST data /ETHER receive data/SIOF master clock/LCD power supply control	VCCQ
B16	PTI5/MD10/ST1_VALID/ LCD_D1	IO/I /IO/O	Port/mode control (external CPU connection select)/ST data valid/LCD data	VCCQ
B17	PTI7/IRQ3/$\overline{\text{IRL3}}$/ST0M_D7I/ IIC1_SDA	I/I/I/I/IO	Port/external interrupt input/ST data (mirror pin)/ IIC serial data	VCCQ
B18	PTJ4/ST0M_D2I/ET0_ERXD2/ RMII1_RXD1/LCD_CL2	IO/I/I/I/O	Port/ST data (mirror pin)/ETHER receive data/RMII receive data/LCD shift clock	VCCQ
B19	$\overline{\text{RDY}}$/$\overline{\text{EX_RDY}}$/$\overline{\text{PCC_WAIT}}$	I/O/I	Ready/external CPU ready/PCMCIA hardware wait request	VCCQ
B20	$\overline{\text{CS2}}$/$\overline{\text{EX_CS1}}$	O/I	Chip select	VCCQ
B21	PTM7/D31/EX_AD31/ST0_D7/ ET0_RX-DV/RMII0_TXD0/ PINT7	IO/IO/IO/IO/I/O/I	Port/data bus/address-and-data bus/ST data/ETHER receive data valid/RMII transmit data/port interrupt input	VCCQ
B22	PTM5/D29/EX_AD29/ST0_D5/ ET0_RX-ER/RMII0_TXD_EN/ PINT5	IO/IO/IO/IO/I/O/I	Port/data bus/address-and-data bus/ST data/ETHER receive error/RMII transmit enable/port interrupt input	VCCQ
B23	VSSQ	—	I/O GND	—
B24	PTM3/D27/EX_AD27/ST0_D3/ ET0_LINKSTA/RMII0_RXD1/ PINT3	IO/IO/IO/IO/I/I/I	Port/data bus/address-and-data bus/ST data/ETHER link status/RMII receive data/port interrupt input	VCCQ
B25	REF125CK/ SSI_CLK/HAC_BITCLK	I/I/I	125-MHz reference clock/SSI divider input clock/HAC clock	VCCQ
C1	M_D0	IO	DDR-SDRAM data bus	VCCQ_ DDR
C2	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
C3	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
C4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—

Pin No.	Pin Name	I/O	Function	Power Supply
C5	M_A12	O	DDR-SDRAM address bus	VCCQ_DDR
C6	M_A11	O	DDR-SDRAM address bus	VCCQ_DDR
C7	M_A9	O	DDR-SDRAM address bus	VCCQ_DDR
C8	M_A8	O	DDR-SDRAM address bus	VCCQ_DDR
C9	M_A7	O	DDR-SDRAM address bus	VCCQ_DDR
C10	M_A6	O	DDR-SDRAM address bus	VCCQ_DDR
C11	M_A5	O	DDR-SDRAM address bus	VCCQ_DDR
C12	$\overline{\text{XRTCSTBI}}$	I	RTC standby	VDD_RTC
C13	VCCQ	—	I/O VCC	—
C14	PTI1/STATUS1/ST1_REQ/RMII0_MDIO	IO/O/IO/IO	Port/status 1/ST data receive preparation request/RMII management data IO	VCCQ
C15	PTK6/ST1_D6/GET0_ERXD6/SIOF2_SCK/LCD_VEPWC	IO/IO/I/IO/O	Port/ST data/ETHER receive data/SIOF serial clock/LCD power supply control	VCCQ
C16	PTI4/MD8/ST1_START/ET1_PHY-INT/RMII0M0_MDC/USB_PWREN/USBF_UPLUP	IO/I /IO/I/O/O/O	Port/mode control (clock input mode)/ST data sync/PHY interrupt/RMII management data clock/USB power supply enable/USB Pull-up control output pin	VCCQ
C17	PTJ7/$\overline{\text{INTB}}$/ST0M_D5I/$\overline{\text{IRQOUT}}$/RMII1_TXD0/LCD_D0	IO/I/I/O/O/O	Port/PCI interrupt/ST data (mirror pin)/interrupt request output/RMII transmit data/LCD data	VCCQ
C18	PTJ3/ST0M_D1I/ET0_ERXD1/RMII1_CRS_DV/LCD_CL1	IO/I/I/I/O	Port/ST data (mirror pin)/ETHER receive data/RMII carrier detection/LCD shift clock	VCCQ
C19	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O/O	Chip select/Card select	VCCQ
C20	$\overline{\text{CS1}}$/$\overline{\text{EX_CS0}}$	O/I	Chip select	VCCQ
C21	VCCQ	—	I/O VCC	—

Pin No.	Pin Name	I/O	Function	Power Supply
C22	VSSQ	—	I/O GND	—
C23	$\overline{\text{BS}}$/EX_$\overline{\text{BS}}$	O/I	Bus cycle start	VCCQ
C24	PTM2/D26/EX_AD26/ST0_D2/ ET0_WOL/RMII0_CRS_DV/ PINT2	IO/IO/IO/IO/O/I/I	Port/data bus/address-and-data bus/ST data/ETHER wake on run/RMII carrier detection/port interrupt input	VCCQ
C25	PTM1/D25/EX_AD25/ST0_D1/ ET0_TX-CLK/RMII0_RX_ER/ PINT1	IO/IO/IO/IO/I/I/I	Port/data bus/address-and-data bus/ST data/ETHER transmit clock/RMII receive error/port interrupt input	VCCQ
D1	M_D1	IO	DDR-SDRAM data bus	VCCQ_ DDR
D2	M_D16	IO	DDR-SDRAM data bus	VCCQ_ DDR
D3	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
D4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
D5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
D6	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
D7	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
D8	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
D9	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
D10	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
D11	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
D12	VDD-RTC	—	RTC VDD	—
D13	VSSQ	—	I/O GND	—
D14	VSSQ	—	I/O GND	—
D15	PTK5/ST1_D5/GET0_ERXD5/ SIOF2_RXD/LCD_D7	IO/IO/I/I/O	Port/ST data/ETHER receive data/SIOF receive data/LCD data	VCCQ
D16	VCCQ	—	I/O VCC	—
D17	PTJ6/ST0M_D4I/ET0_CRS/ RMII1_TXD_EN/LCD_FLM	IO/I/I/O/O	Port/ST data (mirror pin)/ETHER carrier detection/RMII transmit enable/LCD line marker	VCCQ
D18	PTJ2/ST0M_D0I/ ET0_ERXD0/RMII1_TXD1/ LCD_M_DISP	IO/I/I/O/O	Port/ST data (mirror pin)/ETHER receive data/RMII transmit data/LCD liquid crystal AC signal	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
D19	$\overline{CS4}$	O	Chip select	VCCQ
D20	VDD	—	Internal VDD	—
D21	VSSQ	—	I/O GND	—
D22	VCCQ	—	I/O VCC	—
D23	$\overline{RDWR}$/$\overline{EX_RDWR}$	O/I	Read/write	VCCQ
D24	PTM0/D24/EX_AD24/ST0_D0/ ET0_TX-ER/PINT0/ RMII0M0_MDIO	IO/IO/IO/IO/O/I/I O	Port/data bus/address-and-data bus/ST data/ETHER transmit error/port interrupt input/RMII management data IO	VCCQ
D25	PTL7/D23/EX_AD23/ ST0_VALID/ET0_TX-EN/ $\overline{TEND1}$/LCD_D15	IO/IO/IO/IO/O/O/ O	Port/data bus/address-and-data bus/ST data valid/ETHER transmit enable/DMA transfer end/LCD data	VCCQ
E1	M_D2	IO	DDR-SDRAM data bus	VCCQ_ DDR
E2	M_D17	IO	DDR-SDRAM data bus	VCCQ_ DDR
E3	M_D18	IO	DDR-SDRAM data bus	VCCQ_ DDR
E4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
E5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
E6	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
E7	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
E8	VDD	—	Internal VDD	—
E9	VSS	—	Internal GND	—
E10	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
E11	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
E12	VSS-RTC	—	RTC GND	—
E13	VSS	—	Internal GND	—
E14	VCCQ	—	I/O VCC	—
E15	VSSQ	—	I/O GND	—
E16	VDD	—	Internal VDD	—
E17	VSS	—	Internal GND	—
E18	VCCQ	—	I/O VCC	—

Pin No.	Pin Name	I/O	Function	Power Supply
E19	VSSQ	—	I/O GND	—
E20	VSS	—	Internal GND	—
E21	VCCQ	—	I/O VCC	—
E22	PTK3/ST1_D3/GET0_ETXD7/ SIOF2_SYNC/LCD_D5	IO/IO/O/IO/O	Port/ST data/ETHER transmit data/SIOF frame sync/LCD data	VCCQ
E23	PTK2/ST1_D2/GET0_ETXD6/ SIOF1_SCK/LCD_D4	IO/IO/O/IO/O	Port/ST data/ETHER transmit data/SIOF serial clock/LCD data	VCCQ
E24	PTL6/D22/EX_AD22/ ST0_START/ET0_ETXD2/ $\overline{\text{DACK1}}$/LCD_D14	IO/IO/IO/IO/O/O/O	Port/data bus/address-and-data bus/ST data sync/ETHER transmit data/DMA transfer request acknowledge/LCD data	VCCQ
E25	PTL5/D21/EX_AD21/ST0_CLK/ ET0_ETXD1/$\overline{\text{DREQ1}}$/LCD_D13	IO/IO/IO/IO/O/I/O	Port/data bus/address-and-data bus/ST data clock/ETHER transmit data/DMA transfer request/LCD data	VCCQ
F1	M_D3	IO	DDR-SDRAM data bus	VCCQ_ DDR
F2	M_D19	IO	DDR-SDRAM data bus	VCCQ_ DDR
F3	M_D20	IO	DDR-SDRAM data bus	VCCQ_ DDR
F4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
F5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
F21	VSS	—	Internal GND	—
F22	PTK1/ST1_D1/GET0_ETXD5/ SIOF1_TXD/LCD_D3	IO/IO/O/O/O	Port/ST data/ETHER transmit data/SIOF transmit data/LCD data	VCCQ
F23	PTK0/ST1_D0/GET0_ETXD4/ SIOF1_SYNC/LCD_D2	IO/IO/O/IO/O	Port/ST data/ETHER transmit data/SIOF frame sync/LCD data	VCCQ
F24	PTL4/D20/EX_AD20/ ST0_REQ/ET0_ETXD0/$\overline{\text{INTD}}$/ LCD_D12	IO/IO/IO/IO/O/I/O	Port/data bus/address-and-data bus/ST data receive preparation request/ETHER transmit data/PCI interrupt/LCD data	VCCQ
F25	PTJ0/ST0M_REQO/ GET0_GTX-CLK/REF50CK	IO/O/O/I	Port/ST data receive preparation request (mirror pin)/GMII transmit clock/50-MHz reference clock	VCCQ
G1	M_D4	IO	DDR-SDRAM data bus	VCCQ_ DDR

Pin No.	Pin Name	I/O	Function	Power Supply
G2	M_D21	IO	DDR-SDRAM data bus	VCCQ_DDR
G3	M_D22	IO	DDR-SDRAM data bus	VCCQ_DDR
G4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
G5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
G21	VDD	—	Internal VDD	—
G22	PTL3/D19/EX_AD19/IRQ7/$\overline{\text{IRL7}}$/ET0_MDIO/$\overline{\text{INTC}}$/LCD_D11	IO/IO/IO/I/I/IO/I/O	Port/data bus/address-and-data bus/external interrupt input/ETHER management data IO/PCI interrupt/LCD data	VCCQ
G23	PTL2/D18/EX_AD18/IRQ6/$\overline{\text{IRL6}}$/ET0_ETXD3/$\overline{\text{TEND0}}$/LCD_D10	IO/IO/IO/I/I/O/O/O	Port/data bus/address-and-data bus/external interrupt input/ETHER transmit data/DMA transfer end/LCD data	VCCQ
G24	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	O/O	Data enable/PCMCIA IOWR	VCCQ
G25	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	O/O	Data enable/PCMCIA IORD	VCCQ
H1	M_D5	IO	DDR-SDRAM data bus	VCCQ_DDR
H2	M_D23	IO	DDR-SDRAM data bus	VCCQ_DDR
H3	M_DQS2	IO	DDR-SDRAM data strobe	VCCQ_DDR
H4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
H5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
H21	VCCQ	—	I/O VCC	—
H22	PTL0/D16/EX_AD16/IRQ4/$\overline{\text{IRL4}}$/ET0_COL/$\overline{\text{DREQ0}}$/LCD_D8	IO/IO/IO/I/I/I/O	Port/data bus/address-and-data bus/external interrupt input/ETHER collision detection/DMA transfer request/LCD data	VCCQ
H23	PTL1/D17/EX_AD17/IRQ5/$\overline{\text{IRL5}}$/ET0_MDC/$\overline{\text{DACK0}}$/LCD_D9	IO/IO/IO/I/I/O/O/O	Port/data bus/address-and-data bus/external interrupt input/ETHER management data clock/DMA transfer acknowledge/LCD data	VCCQ
H24	D15/EX_AD15	IO/IO	Data bus/address-and-data bus	VCCQ
H25	D14/EX_AD14	IO/IO	Data bus/address-and-data bus	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
J1	M_D7	IO	DDR-SDRAM data bus	VCCQ_DDR
J2	M_D6	IO	DDR-SDRAM data bus	VCCQ_DDR
J3	M_DQM2	O	DDR-SDRAM data mask	VCCQ_DDR
J4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
J5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
J21	VSS	—	Internal GND	—
J22	D7/EX_AD7	IO/IO	Data bus/address-and-data bus	VCCQ
J23	D6/EX_AD6	IO/IO	Data bus/address-and-data bus	VCCQ
J24	D13/EX_AD13	IO/IO	Data bus/address-and-data bus	VCCQ
J25	D12/EX_AD12	IO/IO	Data bus/address-and-data bus	VCCQ
K1	M_DQM0	O	DDR-SDRAM data mask	VCCQ_DDR
K2	M_DQS0	IO	DDR-SDRAM data strobe	VCCQ_DDR
K3	M_DQS3	IO	DDR-SDRAM data strobe	VCCQ_DDR
K4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
K5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
K10	VSS	—	Internal GND	—
K11	VSS	—	Internal GND	—
K12	VSS	—	Internal GND	—
K13	VSS	—	Internal GND	—
K14	VSS	—	Internal GND	—
K15	VSS	—	Internal GND	—
K16	VSS	—	Internal GND	—
K21	VDD	—	Internal VDD	—
K22	D5/EX_AD5	IO/IO	Data bus/address-and-data bus	VCCQ
K23	D4/EX_AD4	IO/IO	Data bus/address-and-data bus	VCCQ
K24	D11/EX_AD11	IO/IO	Data bus/address-and-data bus	VCCQ
K25	D10/EX_AD10	IO/IO	Data bus/address-and-data bus	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
L1	M_DQS1	IO	DDR-SDRAM data strobe	VCCQ_ DDR
L2	M_DQM1	O	DDR-SDRAM data mask	VCCQ_ DDR
L3	M_DQM3	O	DDR-SDRAM data mask	VCCQ_ DDR
L4	VSS-DLL1	—	DLL1 GND	—
L5	VSS-DLL2	—	DLL2 GND	—
L10	VSS	—	Internal GND	—
L11	VSS	—	Internal GND	—
L12	VSS	—	Internal GND	—
L13	VSS	—	Internal GND	—
L14	VSS	—	Internal GND	—
L15	VSS	—	Internal GND	—
L16	VSS	—	Internal GND	—
L21	VCCQ	—	I/O VCC	—
L22	D3/EX_AD3	IO/IO	Data bus/address-and-data bus	VCCQ
L23	D2/EX_AD2	IO/IO	Data bus/address-and-data bus	VCCQ
L24	D9/EX_AD9	IO/IO	Data bus/address-and-data bus	VCCQ
L25	D8/EX_AD8	IO/IO	Data bus/address-and-data bus	VCCQ
M1	M_D8	IO	DDR-SDRAM data bus	VCCQ_ DDR
M2	M_D24	IO	DDR-SDRAM data bus	VCCQ_ DDR
M3	M_D25	IO	DDR-SDRAM data bus	VCCQ_ DDR
M4	VDD-DLL1	—	DLL1 VDD	—
M5	VDD-DLL2	—	DLL2 VDD	—
M10	VSS	—	Internal GND	—
M11	VSS	—	Internal GND	—
M12	VSS	—	Internal GND	—
M13	VSS	—	Internal GND	—
M14	VSS	—	Internal GND	—

Pin No.	Pin Name	I/O	Function	Power Supply
M15	VSS	—	Internal GND	—
M16	VSS	—	Internal GND	—
M21	VSS	—	Internal GND	—
M22	D1/EX_AD1	IO/IO	Data bus/address-and-data bus	VCCQ
M23	D0/EX_AD0	IO/IO	Data bus/address-and-data bus	VCCQ
M24	$\overline{\text{WE1}}$/$\overline{\text{WE}}$	O/O	Data enable/PCMCIA WE	VCCQ
M25	CLKOUT	O	System clock output	VCCQ
N1	M_D9	IO	DDR-SDRAM data bus	VCCQ_DDR
N2	M_D26	IO	DDR-SDRAM data bus	VCCQ_DDR
N3	M_D27	IO	DDR-SDRAM data bus	VCCQ_DDR
N4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
N5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
N10	VSS	—	Internal GND	—
N11	VSS	—	Internal GND	—
N12	VSS	—	Internal GND	—
N13	VSS	—	Internal GND	—
N14	VSS	—	Internal GND	—
N15	VSS	—	Internal GND	—
N16	VSS	—	Internal GND	—
N21	VDD	—	Internal VDD	—
N22	$\overline{\text{RD}}$/$\overline{\text{FRAME}}$/EX_FRAME	O/O/I	Access cycle	VCCQ
N23	$\overline{\text{WE0}}$/$\overline{\text{PCC_REG}}$	O/O	Data enable/PCMCIA REG	VCCQ
N24	A1	O	Address bus	VCCQ
N25	A0	O	Address bus	VCCQ
P1	M_D10	IO	DDR-SDRAM data bus	VCCQ_DDR
P2	M_D28	IO	DDR-SDRAM data bus	VCCQ_DDR
P3	M_D29	IO	DDR-SDRAM data bus	VCCQ_DDR

Pin No.	Pin Name	I/O	Function	Power Supply
P4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
P5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
P10	VSS	—	Internal GND	—
P11	VSS	—	Internal GND	—
P12	VSS	—	Internal GND	—
P13	VSS	—	Internal GND	—
P14	VSS	—	Internal GND	—
P15	VSS	—	Internal GND	—
P16	VSS	—	Internal GND	—
P21	VSSQ	—	I/O GND	—
P22	A9	O	Address bus	VCCQ
P23	A8	O	Address bus	VCCQ
P24	A3	O	Address bus	VCCQ
P25	A2	O	Address bus	VCCQ
R1	M_D11	IO	DDR-SDRAM data bus	VCCQ_ DDR
R2	M_D30	IO	DDR-SDRAM data bus	VCCQ_ DDR
R3	M_D31	IO	DDR-SDRAM data bus	VCCQ_ DDR
R4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
R5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
R10	VSS	—	Internal GND	—
R11	VSS	—	Internal GND	—
R12	VSS	—	Internal GND	—
R13	VSS	—	Internal GND	—
R14	VSS	—	Internal GND	—
R15	VSS	—	Internal GND	—
R16	VSS	—	Internal GND	—
R21	VCCQ	—	I/O VCC	—
R22	A11	O	Address bus	VCCQ
R23	A10	O	Address bus	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
R24	A5	O	Address bus	VCCQ
R25	A4	O	Address bus	VCCQ
T1	M_D13	IO	DDR-SDRAM data bus	VCCQ_ DDR
T2	M_D12	IO	DDR-SDRAM data bus	VCCQ_ DDR
T3	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
T4	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
T5	VCCQ-DDR	—	DDR-SDRAM I/O VCC	—
T10	VSS	—	Internal GND	—
T11	VSS	—	Internal GND	—
T12	VSS	—	Internal GND	—
T13	VSS	—	Internal GND	—
T14	VSS	—	Internal GND	—
T15	VSS	—	Internal GND	—
T16	AVss	—	Analog GND	—
T21	VSS	—	Internal GND	—
T22	A17	O	Address bus	VCCQ
T23	A16	O	Address bus	VCCQ
T24	A7	O	Address bus	VCCQ
T25	A6	O	Address bus	VCCQ
U1	M_D15	IO	DDR-SDRAM data bus	VCCQ_ DDR
U2	M_D14	IO	DDR-SDRAM data bus	VCCQ_ DDR
U3	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
U4	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
U5	VSSQ-DDR	—	DDR-SDRAM I/O GND	—
U21	VDD	—	Internal VDD	—
U22	A19	O	Address bus	VCCQ
U23	A18	O	Address bus	VCCQ
U24	A13	O	Address bus	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
U25	A12	O	Address bus	VCCQ
V1	VDD	—	Internal VDD	—
V2	VDD	—	Internal VDD	—
V3	VDD	—	Internal VDD	—
V4	VDD	—	Internal VDD	—
V5	VDD	—	Internal VDD	—
V21	VSS	—	Internal GND	—
V22	A21	O	Address bus	VCCQ
V23	A20	O	Address bus	VCCQ
V24	A15	O	Address bus	VCCQ
V25	A14	O	Address bus	VCCQ
W1	PTG1/$\overline{\text{GNT2}}$/ET1_ETXD0	IO/O/O	Port/PCI bus grant/ETHER transmit data	VCCQ
W2	PTG2/$\overline{\text{REQ1}}$/ET1_ETXD1	IO/I/O	Port/PCI bus request/ETHER transmit data	VCCQ
W3	PTG3/$\overline{\text{REQ3}}$/ET1_ETXD2	IO/I/O	Port/PCI bus request/ETHER transmit data	VCCQ
W4	PTF0/$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$/SIM_D/ ET1_ETXD3/$\overline{\text{DREQ3}}$	IO/IO/I/IO/O/I	Port/PCI bus grant/SIM data/ETHER transmit data/DMA transfer request	VCCQ
W5	VDD	—	Internal VDD	—
W21	VDD	—	Internal VDD	—
W22	A25/EX_SIZE2	O/I	Address bus/access size	VCCQ
W23	A24/EX_SIZE1	O/I	Address bus/access size	VCCQ
W24	A23/EX_SIZE0	O/I	Address bus/access size	VCCQ
W25	A22	O	Address bus	VCCQ
Y1	PTE1/PCICLK/GET1_ETXD4/ $\overline{\text{DACK2}}$	IO/I/O/O	Port/PCI input clock/GMII transmit data/DMA transfer request acknowledge	VCCQ
Y2	PTD6/$\overline{\text{REQ2}}$/PCC_BVD1/ GET1_ETXD5/SSI1_SCK/ LCDM_VCPWC	IO/I/I/O/IO/O	Port/PCI bus mastership request (host)/PCMCIA BVD1/GMII transmit data/SSI serial bit clock/LCD power supply control (mirror pin)	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
Y3	PTE0/$\overline{\text{INTA}}$/$\overline{\text{PCC_DRV}}$/ GET1_ETXD6/$\overline{\text{DREQ2}}$	IO/IO/O/O/I	Port/PCI interrupt/PCMCIA buffer control/GMII transmit data/DMA transfer request	VCCQ
Y4	PTD7/$\overline{\text{PCIRESET}}$/ PCC_RESET/GET1_ETXD7/ LCDM_VEPWC	O/O/O/O/O	Port/PCI reset/PCMCIA reset/GMII transmit data/LCD power supply control (mirror pin)	VCCQ
Y5	VSS	—	Internal GND	—
Y21	VCCQ	—	I/O VCC	—
Y22	$\overline{\text{CE2A}}$	O	PCMCIA select	VCCQ
Y23	$\overline{\text{CE2B}}$	O	PCMCIA select	VCCQ
Y24	DA1	O	Analog output	AVcc
Y25	DA0	O	Analog output	AVcc
AA1	PTF1/$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$/ SIM_CLK/ET1_MDC/$\overline{\text{DACK3}}$	IO/IO/O/O/O/O	Port/PCI bus mastership request (host)/PCI bus mastership request output/SIM clock output/ETHER management data clock/DMA transfer request acknowledge	VCCQ
AA2	PTF2/AD31/SIM_RST/ ET1_MDIO/$\overline{\text{TEND3}}$	IO/IO/O/IO/O	Port/PCI address-and-data bus/SIM reset/ETHER management data IO/DMA transfer end	VCCQ
AA3	PTG0/$\overline{\text{GNT1}}$/ET1_WOL	IO/O/O	Port/PCI bus GND/ETHER wake on LAN	VCCQ
AA4	PTG4/AD30/ET1_LINKSTA	IO/IO/I	Port/PCI address-and-data bus/ ETHER link status	VCCQ
AA5	VSSQ	—	I/O GND	—
AA6	VCCQ	—	I/O VCC	—
AA7	VSSQ	—	I/O GND	—
AA8	VCCQ	—	I/O VCC	—
AA9	VDD	—	Internal VDD	—
AA10	VDD	—	Internal VDD	—
AA11	VDD	—	Internal VDD	—
AA12	VCCQ	—	I/O VCC	—
AA13	VCCQ	—	I/O VCC	—
AA14	VSSQ	—	I/O GND	—

Pin No.	Pin Name	I/O	Function	Power Supply
AA15	VCCQ	—	I/O VCC	—
AA16	VSSQ	—	I/O GND	—
AA17	VSSQ	—	I/O GND	—
AA18	VCCQ	—	I/O VCC	—
AA19	VSSQ	—	I/O GND	—
AA20	VCCQ	—	I/O VCC	—
AA21	VSSQ	—	I/O GND	—
AA22	VSSQ	—	I/O GND	—
AA23	$\overline{\text{IOIS16}}$/TMU_TCLK	I/I	PCMCIA16-bit IO/TMU clock	VCCQ
AA24	AVcc	—	Analog VCC	—
AA25	AVcc	—	Analog VCC	—
AB1	PTE5/AD29/SCIF2_TXD/ GET1_GTX-CLK/SSI0_SCK	IO/IO/O/O/IO	Port/PCI address-and-data bus/SCIF transmit data/GMII transmit clock/SSI serial bit clock	VCCQ
AB2	PTG7/AD28/ET1_TX-EN	IO/IO/O	Port/PCI address-and-data bus/ ETHER transmit enable	VCCQ
AB3	PTG6/AD26/ET1_TX-ER	IO/IO/O	Port/PCI address-and-data bus/ ETHER transmit error	VCCQ
AB4	VSSQ	—	I/O GND	—
AB5	VCCQ	—	I/O VCC	—
AB6	PTE4/AD22/SCIF2_RXD/ GET1_ERXD4/SSI0_SDATA	IO/IO/I/I/IO	Port/PCI address-and-data bus/SCIF receive data/GMII receive data/SSI serial data IO	VCCQ
AB7	PTD5/AD18/$\overline{\text{PCC_CD2}}$/ GET1_ERXD6/SSI1_SDATA/ LCDM_D14	IO/IO/I/I/IO/O	Port/PCI address-and-data bus/PCMCIA CD2/GMII receive data/SSI serial data IO/LCD data (mirror pin)	VCCQ
AB8	PTD3/$\overline{\text{PCIFRAME}}$/PCC_BVD2/ SIOF0_SCK/$\overline{\text{HAC_RES}}$/ LCDM_D12	IO/IO/I/IO/O/O	Port/PCI cycle frame/PCMCIA BVD2/SIOF serial clock/HAC reset/LCD data (mirror pin)	VCCQ
AB9	PTD4/$\overline{\text{STOP}}$/$\overline{\text{PCC_CD1}}$/ SIOF0_MCLK/SSI1_WS/ LCDM_DON	IO/IO/I/I/IO/O	Port/PCI stop/PCMCIA CD1/SIOF master clock/SSI word select/LCD display start (mirror pin)	VCCQ
AB10	PTA3/AD15/$\overline{\text{SCIF1_CTS}}$	IO/IO/IO	Port/PCI address-and-data bus/SCIF modem control (CTS)	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
AB11	PTB2/AD11/PINT10/LCDM_D7	IO/IO/I/O	Port/PCI address-and-data bus/port interrupt input/LCD data (mirror pin)	VCCQ
AB12	PTB6/CBE0/PINT14/ LCDM_D3	IO/IO/I/O	Port/PCI command and byte enable/port interrupt input/LCD data (mirror pin)	VCCQ
AB13	PTC1/AD4/LCDM_D1	IO/IO/O	Port/PCI address-and-data bus/LCD data (mirror pin)	VCCQ
AB14	VSSQ	—	I/O GND	—
AB15	VCCQ	—	I/O VCC	—
AB16	MPMD	I	Chip mode specification	VCCQ
AB17	PTO6/IRQ0/$\overline{\text{IRL0}}$/ $\overline{\text{DACK1M}}$/MD5	IO/I/I/O/I	Port/external interrupt input/DMA transfer request acknowledge (mirror pin)/mode control (endian switching)	VCCQ
AB18	PTO2/AUDATA1/ RMII0M1_MDC	IO/O/O	Port/AUD data/RMII management data clock	VCCQ
AB19	VSSQ	—	I/O GND	—
AB20	TDO	O	H-UDI data output	VCCQ
AB21	VSSQ	—	I/O GND	—
AB22	VSSQ	—	I/O GND	—
AB23	VCCQ	—	I/O VCC	—
AB24	AN3	I	Analog input	AVcc
AB25	AN2	I	Analog input	AVcc
AC1	PTH6/AD27/TPU_TO2/ ET1_CRS/RMII1M_TXD_EN	IO/IO/O/I/O	Port/PCI address-and-data bus/TPU clock output/ETHER carrier detection/RMII transmit enable (mirror pin)	VCCQ
AC2	PTH0/AD25/TPU_TI3A/ ET1_COL/RMII1M_RX_ER	IO/IO/I/I/I	Port/PCI address-and-data bus/TPU clock input/ETHER collision detection/RMII receive error (mirror pin)	VCCQ
AC3	VSSQ	—	I/O GND	—
AC4	VCCQ	—	I/O VCC	—
AC5	PTH1/IDSEL/TPU_TI3B/ ET1_RX-ER/RMII1M_CRS_DV	IO/I/I/I/I	Port/PCI configuration device select/TPU clock input/ETHER receive error/RMII carrier detection (mirror pin)	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
AC6	PTE3/AD20/SCIF2_SCK/ GET1_ERXD5/SSI0_WS	IO/IO/IO/I/IO	Port/PCI address-and-data bus/IrDA serial clock /GMII receive data/SSI word select	VCCQ
AC7	PTE2/AD16/$\overline{\text{PCC_IOIS16}}$/ GET1_ERXD7/$\overline{\text{TEND2}}$	IO/IO/I/I/O	Port/PCI address-and-data bus/PCMCIA 16-bit IO/GMII receive data/DMA transmit end	VCCQ
AC8	PTD2/$\overline{\text{TRDY}}$/PCC_RDY/ SIOF0_RXD/HAC_SYNC/ LCDM_D11	IO/IO/I/I/O/O	Port/PCI target ready/PCMCIA ready/SIOF receive data/HAC frame sync/LCD data (mirror pin)	VCCQ
AC9	PTA0/PAR/SCIF1_SCK	IO/IO/IO	Port/PCI parity signal/SCIF serial clock	VCCQ
AC10	PTA4/AD13/$\overline{\text{SCIF1_RTS}}$	IO/IO/IO	Port/PCI address-and-data bus/SCIF modem control (RTS)	VCCQ
AC11	PTB3/AD9/PINT11/LCDM_D6	IO/IO/I/O	Port/PCI address-and-data bus/port interrupt input/LCD data (mirror pin)	VCCQ
AC12	PTB7/AD6/PINT15/LCDM_D2	IO/IO/I/O	Port/PCI address-and-data bus/port interrupt input/LCD data (mirror pin)	VCCQ
AC13	PTC2/AD2/LCDM_D0	IO/IO/O	Port/PCI address-and-data bus/LCD data (mirror pin)	VCCQ
AC14	PTC5/AD0/MMC_CD/ LCDM_FLM	IO/IO/I/O	Port/PCI address-and-data bus/MMC card detection/LCD line marker (mirror pin)	VCCQ
AC15	PTN2/SCIF0_TXD/MD1	IO/O/I	Port/SCIF transmit data/mode control (clock operating mode)	VCCQ
AC16	$\overline{\text{MRESET}}$	I	Manual rest input	VCCQ
AC17	PTO7/IRQ1/$\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/ SSI3_SCK/MD6	IO/I/I/O/IO/I	Port/external interrupt input/DMA transfer end (mirror pin)/SSI serial bit clock/mode control (PCI operating mode select)	VCCQ
AC18	PTO3/AUDATA2/ RMII0M1_MDIO/SSI2_SCK	IO/O/IO/IO	Port/AUD data/RMII management data IO/SSI serial bit clock	VCCQ
AC19	$\overline{\text{TRST}}$	I	H-UDI reset input	VCCQ
AC20	TDI	I	H-UDI data input	VCCQ
AC21	TMS	I	H-UDI mode input	VCCQ
AC22	$\overline{\text{BACK}}$	O	Bus usage permission	VCCQ
AC23	AVss	—	Analog GND	—

Pin No.	Pin Name	I/O	Function	Power Supply
AC24	AN1	I	Analog input	AVcc
AC25	AN0	I	Analog input	AVcc
AD1	PTF3/CBE3/ET1_TX-CLK	IO/IO/I	Port/PCI command and byte enable/transmit clock	VCCQ
AD2	VSSQ	—	I/O GND	—
AD3	VCCQ	—	I/O VCC	—
AD4	PTH2/AD24/TPU_TI2A/ ET1_ERXD0/RMII1M_TXD1	IO/IO/I/I/O	Port/PCI address-and-data bus/TPU clock input/ETHER receive data/RMII transmit data (mirror pin)	VCCQ
AD5	PTH3/AD21/TPU_TI2B/ ET1_ERXD2/RMII1M_RXD1	IO/IO/I/I/I	Port/PCI address-and-data bus/TPU clock input/ETHER receive data/RMII receive data (mirror pin)	VCCQ
AD6	PTH7/AD17/TPU_TO3/ ET1_RX-DV	IO/IO/O/I	Port/PCI address-and-data bus/TPU clock output/ETHER receive data valid	VCCQ
AD7	PTD0/$\overline{\text{IRDY}}$/$\overline{\text{PCC_VS1}}$/ SIOF0_SYNC/HAC_SD_IN/ LCDM_D13	IO/IO/I/IO/I/O	Port/PCI initiator ready/PCMCIA VS1/SIOF frame sync/HAC serial data input/LCD data (mirror pin)	VCCQ
AD8	PTA2/$\overline{\text{LOCK}}$/SCIF1_TXD	IO/IO/O	Port/PCI lock/SCIF transmit data	VCCQ
AD9	PTB1/$\overline{\text{SERR}}$/PINT9/ LCDM_D9	IO/IO/I/O	Port/PCI parity error/port interrupt input/ LCD data (mirror pin)	VCCQ
AD10	PTB5/AD14/PINT13/ LCDM_M_DISP	IO/IO/I/O	Port/PCI address-and-data bus/port interrupt input/LCD liquid crystal AC signal (mirror pin)	VCCQ
AD11	PTC0/AD10/MMC_DAT/ LCDM_D5	IO/IO/IO/O	Port/PCI address-and-data bus/MMC DAT/LCD data (mirror pin)	VCCQ
AD12	PTC4/AD7/MMC_CMD LCDM_CL2	IO/IO/IO/O	Port/PCI address-and-data bus/MMC CMD/LCD shift clock (mirror pin)	VCCQ
AD13	PTC7/AD3/MMC_CLK	IO/IO/O	Port/PCI address-and-data bus/MMC clock output	VCCQ
AD14	PTN0/SCIF0_SCK/MD0	IO/IO/I	Port/SCIF serial clock/mode control (clock operating mode)	VCCQ
AD15	PTN3/$\overline{\text{SCIF0_CTS}}$/MD4	IO/IO/I	Port/SCIF modem control (CTS)/mode control (bus width for area 0)	VCCQ
AD16	PTN5/NMI	IO/I	Port/non-maskable interrupt input	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
AD17	PTO0/AUDSYNC/ RMII1_MDC/SSI2_WS	IO/O/O/IO	Port/AUD sync signal/RMII management data clock/SSI word select	VCCQ
AD18	PTO4/AUDATA3/$\overline{\text{EX_INT}}$/ SSI3_WS	IO/O/O/IO	Port/AUD data/external CPU interrupt/SSI word select	VCCQ
AD19	$\overline{\text{ASEBRK}}$/BRKACK	IO	Break mode acknowledge	VCCQ
AD20	VSS-PLL3	—	PLL3 GND	—
AD21	VSS-PLL2	—	PLL2 GND	—
AD22	$\overline{\text{BREQ}}$	I	Bus release request	VCCQ
AD23	VCCQ	—	I/O VCC	—
AD24	VSS-PLL1	—	PLL1 GND	—
AD25	AVcc	—	Analog VCC	—
AE1	VSSQ	—	I/O GND	—
AE2	VCCQ	—	I/O VCC	—
AE3	PTG5/$\overline{\text{GNT3}}$/ET1_RX-CLK	IO/O/I	Port/PCI bus grant/ETHER receive clock	VCCQ
AE4	PTH5/AD23/TPU_TO1/ ET1_ERXD1/RMII1M_TXD0	IO/IO/O/I/O	Port/PCI address-and-data bus/TPU clock output/ETHER receive data/RMII transmit data (mirror pin)	VCCQ
AE5	PTH4/AD19/TPU_TO0/ ET1_ERXD3/RMII1M_RXD0	IO/IO/O/I/I	Port/PCI address-and-data bus/TPU clock output/ETHER receive data/RMII receive data (mirror pin)	VCCQ
AE6	PTD1/CBE2/$\overline{\text{PCC_VS2}}$/ SIOF0_TXD/HAC_SD_OUT/ LCDM_D15	IO/IO/I/O/O/O	Port/PCI command-and-byte enable/PCMCIA VS2/SIOF transmit data/HAC serial data output/LCD data (mirror pin)	VCCQ
AE7	PTA1/$\overline{\text{DEVSEL}}$/SCIF1_RXD	IO/IO/I	Port/PCI device select/SCIF receive data	VCCQ
AE8	PTB0/$\overline{\text{PERR}}$/PINT8/ LCDM_D10	IO/IO/I/O	Port/PCI system error/port interrupt input/LCD data (mirror pin)	VCCQ
AE9	PTB4/CBE1/PINT12/ LCDM_D8	IO/IO/I/O	Port/PCI command-and-byte enable/port interrupt input/LCD data (mirror pin)	VCCQ
AE10	PTA5/AD12	IO/IO	Port/PCI address-and-data bus	VCCQ

Pin No.	Pin Name	I/O	Function	Power Supply
AE11	PTC3/AD8/$\overline{\text{MMC_ODMOD}}$/ LCDM_D4	IO/IO/O/O	Port/PCI address-and-data bus/MMC open-drain control/LCD data (mirror pin)	VCCQ
AE12	PTC6/AD5/LCDM_CL1	IO/IO/O	Port/PCI address-and-data bus/LCD shift clock (mirror pin)	VCCQ
AE13	PTA6/AD1/MMC_VDDON	IO/IO/O	Port/PCI address-and-data bus/MMC card power supply control	VCCQ
AE14	PTN1/SCIF0_RXD/MD3	IO/I/I	Port/SCIF receive data/mode control (bus width for area 0)	VCCQ
AE15	PTN4/$\overline{\text{SCIF0_RTS}}$/MD2	IO/IO/I	Port/SCIF modem control (RTS)/mode control (clock operating mode)	VCCQ
AE16	$\overline{\text{PRESET}}$	I	Power-on reset	VCCQ
AE17	PTO1/AUDATA0/ RMII1_MDIO/SSI2_SDATA	IO/O/IO/IO	Port/AUD data/RMII management data IO/SSI serial data IO	VCCQ
AE18	PTO5/AUDCK/$\overline{\text{DREQ1M}}$/ SSI3_SDATA	IO/O/I/IO	Port/AUD clock/DMA transfer request (mirror pin)/SSI serial data IO	VCCQ
AE19	TCK	I	H-UDI clock	VCCQ
AE20	VDD-PLL3	—	PLL3 VDD	—
AE21	VDD-PLL2	—	PLL2 VDD	—
AE22	EXTAL	I	External clock input/crystal resonator	VCCQ
AE23	XTAL	O	Crystal resonator	VCCQ
AE24	VDD-PLL1	—	PLL1 VDD	—
AE25	VSSQ	—	I/O GND	—

Section 2 Programming Model

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

2.1 Data Formats

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

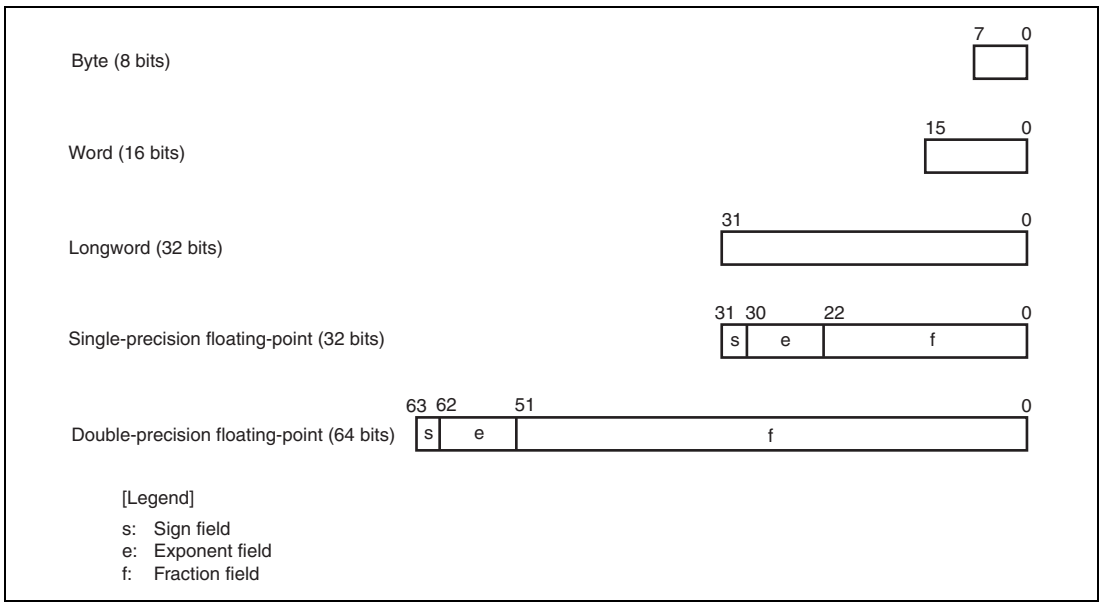


Figure 2.1 Data Formats

2.2 Register Descriptions

2.2.1 Privileged Mode and Banks

(1) Processing Modes

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

(2) General Registers

There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers which are switched by a processing mode change.

- Privileged mode

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

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

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

(3) Control Registers

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

(4) System Registers

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

(5) Floating-Point Registers and System Registers Related to FPU

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

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

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

Register values after a reset are shown in table 2.1.

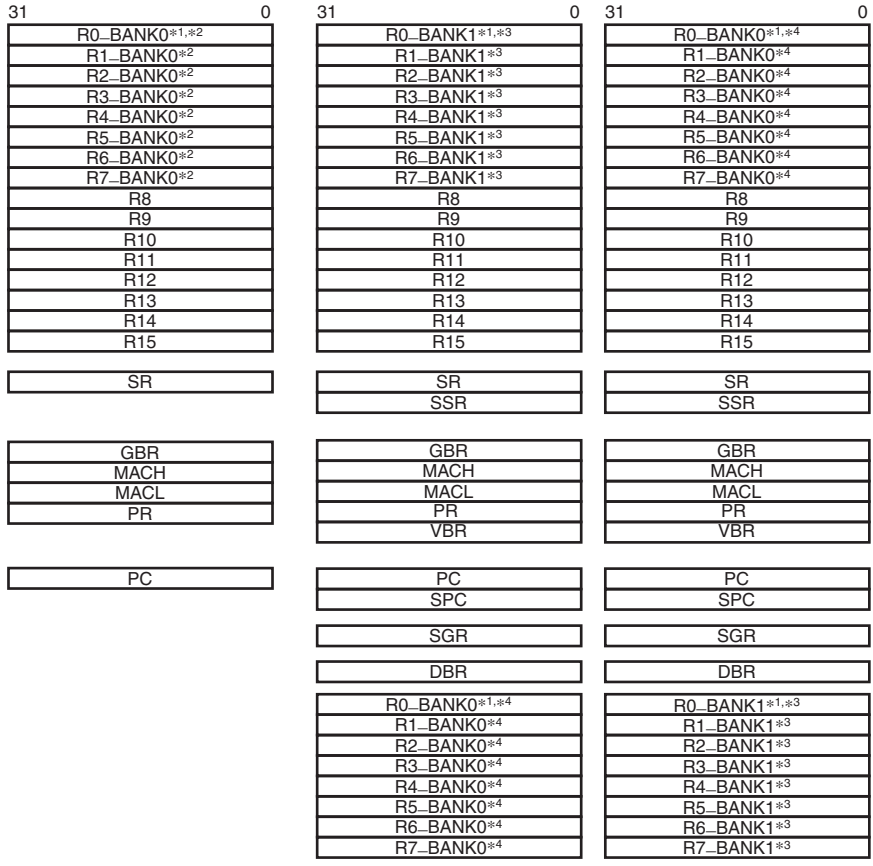
Table 2.1 Initial Register Values

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

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

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

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



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

Figure 2.2 CPU Register Configuration in Each Processing Mode

2.2.2 General Registers

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

- R0_BANK0 to R7_BANK0
  Allocated to R0 to R7 in user mode (SR.MD = 0)
  Allocated to R0 to R7 when SR.RB = 0 in privileged mode (SR.MD = 1).
- R0_BANK1 to R7_BANK1
  Cannot be accessed in user mode.
  Allocated to R0 to R7 when SR.RB = 1 in privileged mode.

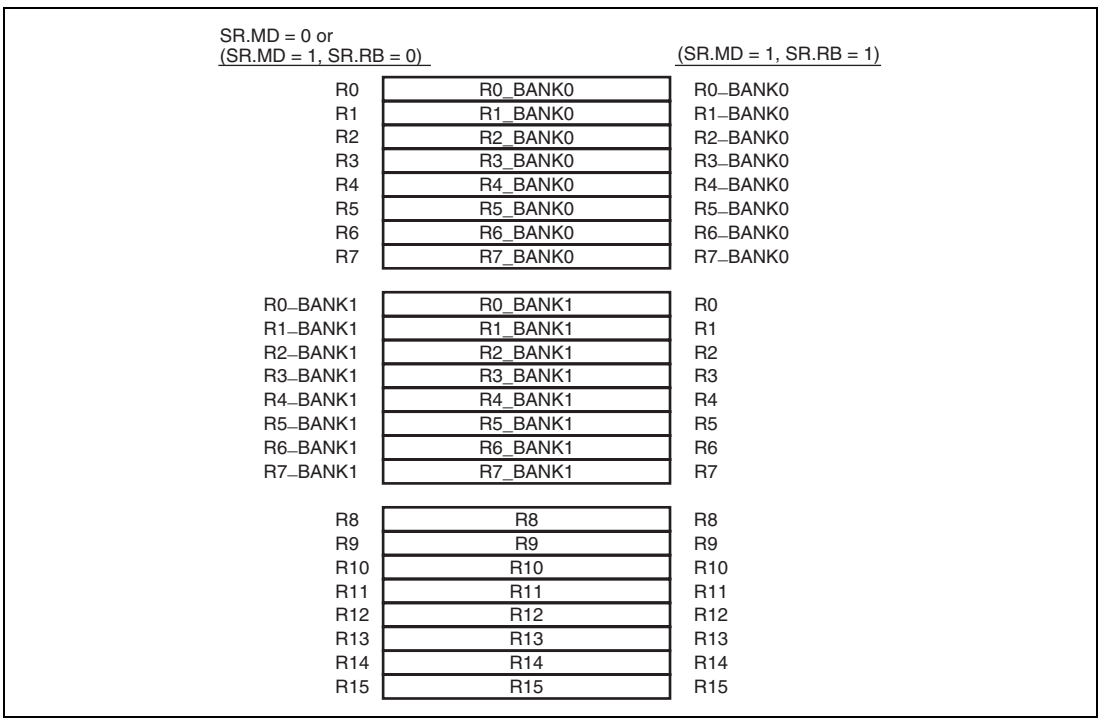


Figure 2.3 General Registers

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

2.2.3 Floating-Point Registers

Figure 2.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers, FPR0_BANK0 to FPR15_BANK0, AND FPR0_BANK1 to FPR15_BANK1, comprising two banks. These registers are referenced as FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, or XMTRX. Reference names of each register are defined depending on the state of the FR bit in FPSCR (see figure 2.4).

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

2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are assigned to FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 are assigned to FPR0_BANK1 to FPR15_BANK1.

3. Double-precision floating-point registers or single-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

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

5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are assigned to FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 are assigned to FPR0_BANK0 to FPR15_BANK0.

6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}

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

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

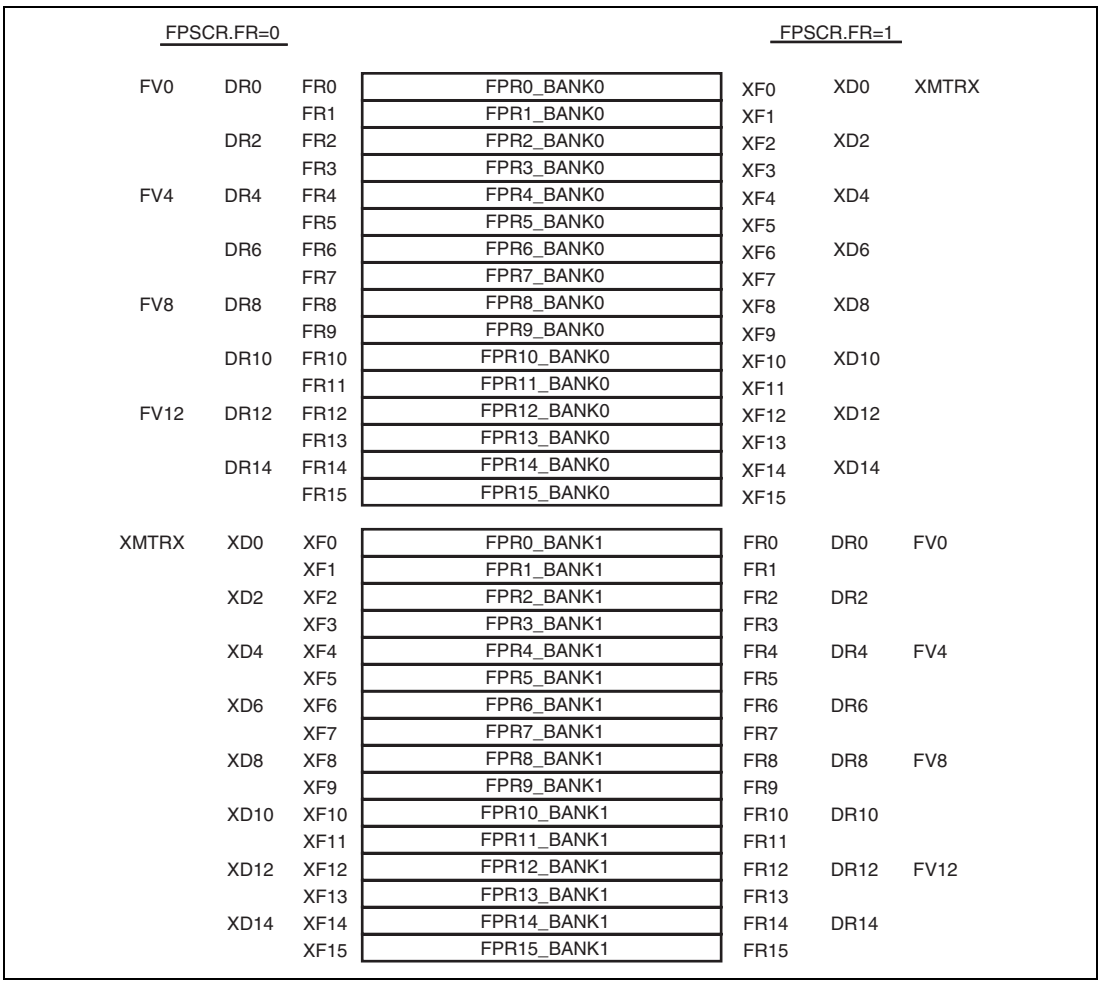


Figure 2.4 Floating-Point Registers

2.2.4 Control Registers

(1) Status Register (SR)

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD	—	—	—	—	—	M	Q	IMASK				—	—	S	T
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

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

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

(2) Saved Status Register (SSR) (32 bits, Privileged Mode, Initial Value = Undefined)

The contents of SR are saved to SSR in the event of an exception or interrupt.

(3) Saved Program Counter (SPC) (32 bits, Privileged Mode, Initial Value = Undefined)

The address of an instruction at which an interrupt or exception occurs is saved to SPC.

(4) Global Base Register (GBR) (32 bits, Initial Value = Undefined)

GBR is referenced as the base address of addressing @(disp,GBR) and @(R0,GBR).

(5) Vector Base Register (VBR) (32 bits, Privileged Mode, Initial Value = H'00000000)

VBR is referenced as the branch destination base address in the event of an exception or interrupt. For details, see section 5, Exception Handling.

(6) Saved General Register 15 (SGR) (32 bits, Privileged Mode, Initial Value = Undefined)

The contents of R15 are saved to SGR in the event of an exception or interrupt.

(7) Debug Base Register (DBR) (32 bits, Privileged Mode, Initial Value = Undefined)

When the user break debugging function is enabled (CBCR.UBDE = 1), DBR is referenced as the branch destination address of the user break handler instead of VBR.

2.2.5 System Registers

(1) Multiply-and-Accumulate Registers (MACH and MACL) (32 bits, Initial Value = Undefined)

MACH and MACL are used for the added value in a MAC instruction, and to store the operation result of a MAC or MUL instruction.

(2) Procedure Register (PR) (32 bits, Initial Value = Undefined)

The return address is stored in PR in a subroutine call using a BSR, BSRF, or JSR instruction. PR is referenced by the subroutine return instruction (RTS).

(3) Program Counter (PC) (32 bits, Initial Value = H'A0000000)

PC indicates the address of the instruction currently being executed.

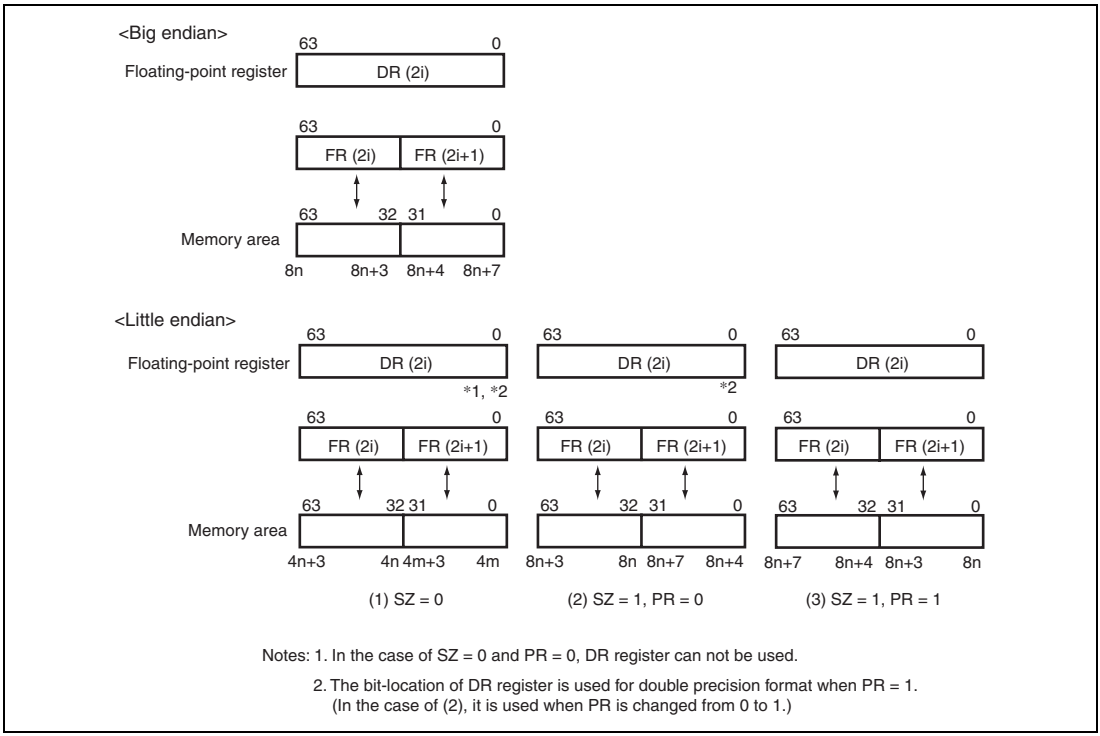
(4) Floating-Point Status/Control Register (FPSCR)

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable (EN)					Flag					RM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 are assigned to FR0 to FR15
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits) For relationship between the SZ bit, PR bit, and endian, see figure 2.5.
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined) For relationship between the SZ bit, PR bit, and endian, see figure 2.5
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable (EN)	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. When an FPU exception occurs, the bits corresponding to FPU exception cause field and flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. For bit allocations of each field, see table 2.2.
1, 0	RM	01	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved



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

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

Figure 2.5 Relationship between SZ bit and Endian

Table 2.2 Bit Allocation for FPU Exception Handling

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

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

Information is transferred between the FPU and CPU via FPUL.

2.3 Memory-Mapped Registers

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

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

These two areas are used as follows.

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

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

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

  The operation of an access to this area without using the address translation function of the MMU is not guaranteed.
- H'FC00 0000 to H'FFFF FFFF

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

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

2.4 Data Formats in Registers

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

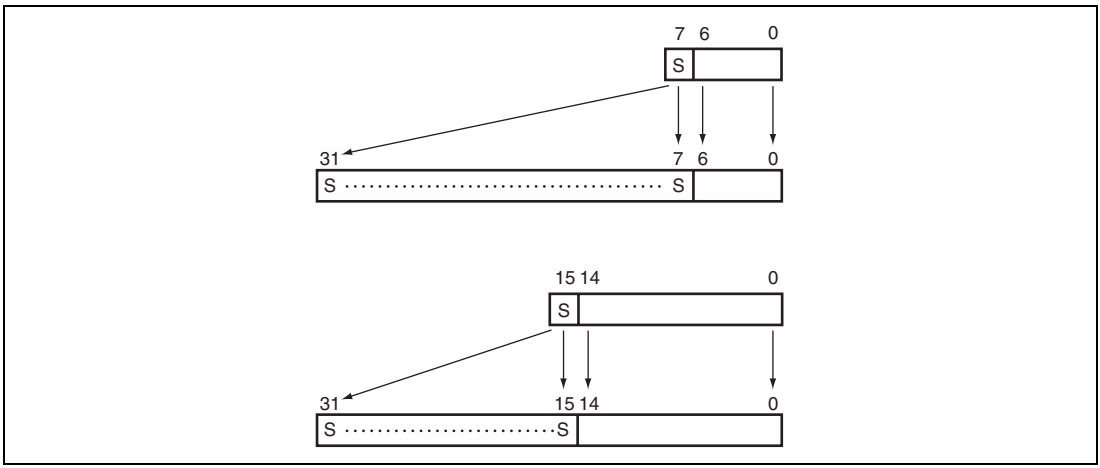


Figure 2.6 Formats of Byte Data and Word Data in Register

2.5 Data Formats in Memory

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

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

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

The data format in memory is shown in figure 2.7.

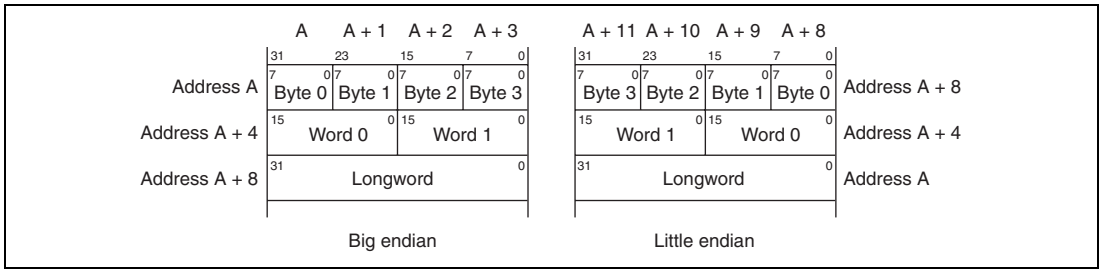


Figure 2.7 Data Formats in Memory

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

2.6 Processing States

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

(1) Reset State

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

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

(2) Instruction Execution State

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

(3) Power-Down State

In a power-down state, CPU halts operation and power consumption is reduced. The power-down state is entered by executing a SLEEP instruction. There are two modes in the power-down state: sleep mode and standby mode.

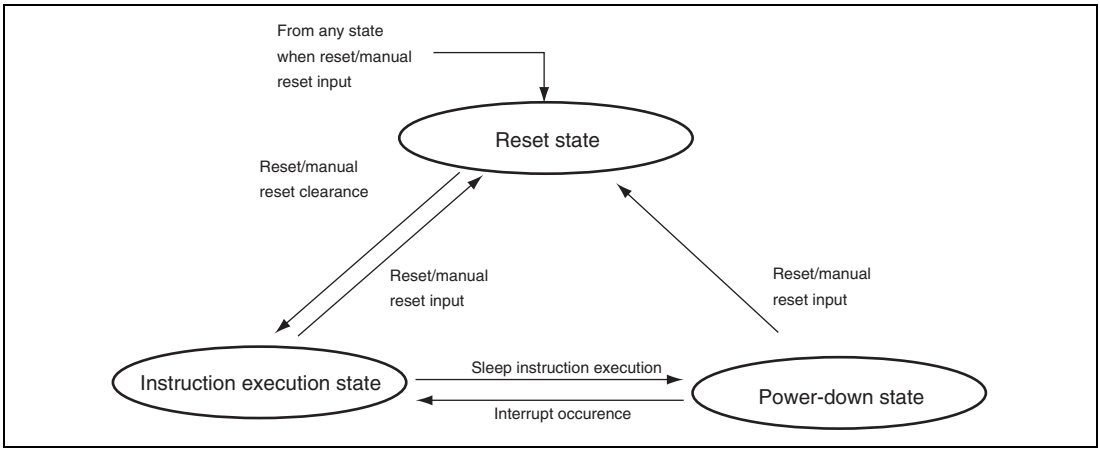


Figure 2.8 Processing State Transitions

2.7 Usage Note

2.7.1 Notes on Self-Modified Codes*

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

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

```
SYNCO
ICBI  @Rn
```

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

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

```
SYNCO
ICBI  @Rn
```

The all instruction cache area corresponding to the modified codes should be invalidated by the ICBI instruction. The ICBI instruction should be issued to each cache line. One cache line is 32 bytes.

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

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

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

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

Section 3 Instruction Set

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

3.1 Execution Environment

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

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

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

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

Table 3.1 Execution Order of Delayed Branch Instructions

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

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

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

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

In an RTE delay slot, the SR bits are referenced as follows. In instruction access, the MD bit is used before modification, and in data access, the MD bit is accessed after modification. The other bits—S, T, M, Q, FD, BL, and RB—after modification are used for delay slot instruction execution. The STC and STC.L SR instructions access all SR bits after modification.

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

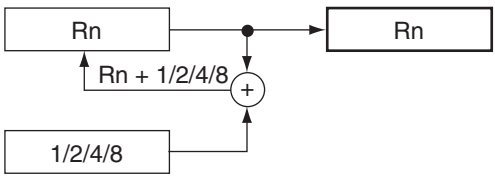
```
MOV.W   @(disp, PC), Rn
MOV.L   @(disp, PC), Rn
```

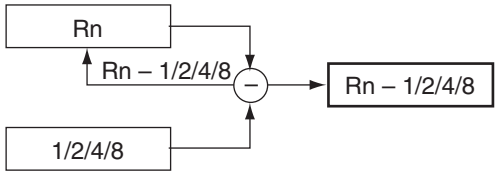
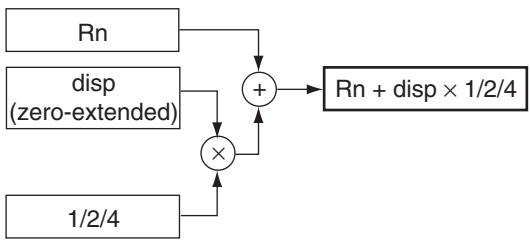
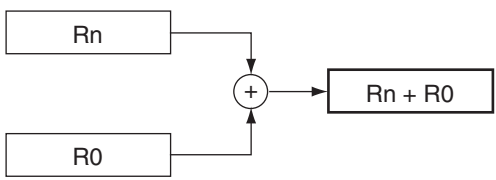
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

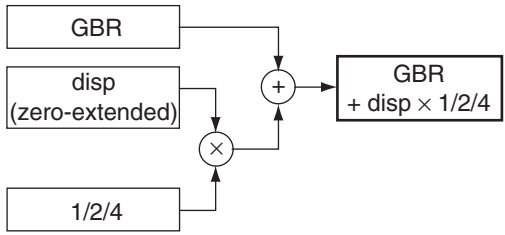
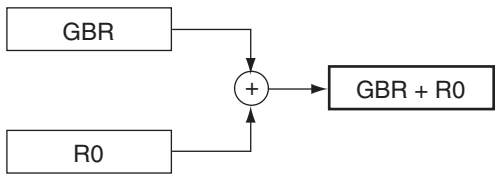
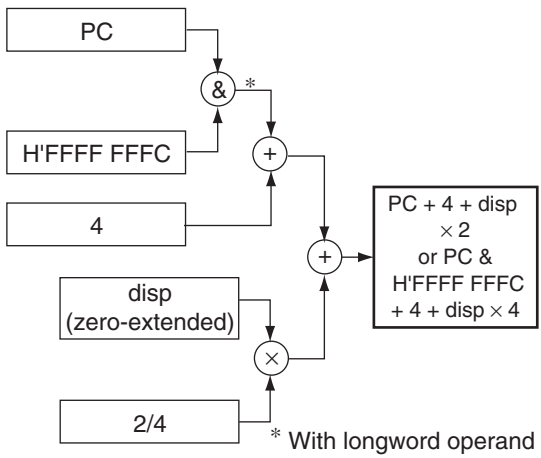
3.2 Addressing Modes

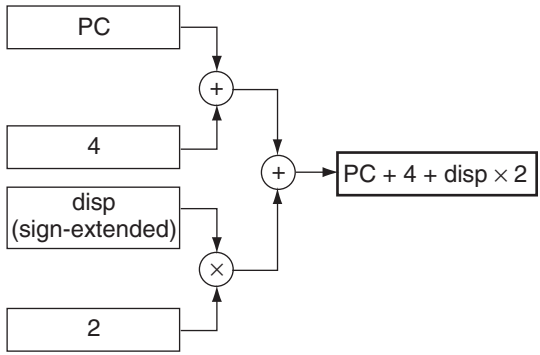
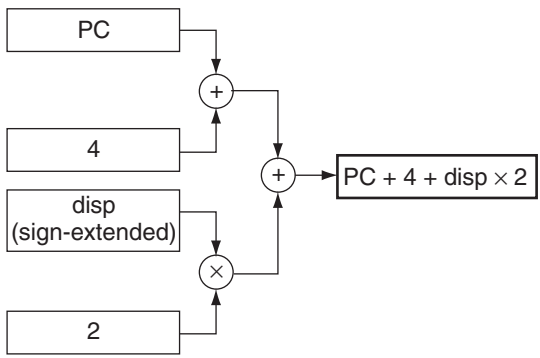
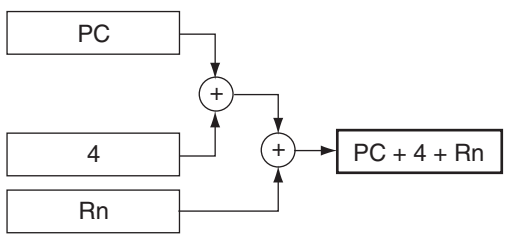
Addressing modes and effective address calculation methods are shown in table 3.2. When a location in virtual memory space is accessed (AT in MMUCR = 1), the effective address is translated into a physical memory address. If multiple virtual memory space systems are selected (SV in MMUCR = 0), the least significant bit of PTEH is also referenced as the access ASID. For details, see section 6, Memory Management Unit (MMU).

Table 3.2 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn → EA (EA: effective address)
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand, 8 for a quadword operand. 	Rn → EA After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn Quadword: Rn + 8 → Rn

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

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: GBR + disp → EA Word: GBR + disp × 2 → EA Longword: GBR + disp × 4 → EA
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0 → EA
PC-relative with displacement	@(disp:8, PC)	Effective address is PC + 4 with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: PC + 4 + disp × 2 → EA Longword: PC & H'FFFF FFFC + 4 + disp × 4 → EA

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative	disp:8	Effective address is PC + 4 with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
PC-relative	disp:12	Effective address is PC + 4 with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
	Rn	Effective address is sum of PC + 4 and Rn. 	PC + 4 + Rn → Branch-Target

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

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

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

3.3 Instruction Set

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

Table 3.3 Notation Used in Instruction List

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

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

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

Table 3.4 Fixed-Point Transfer Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV	#imm,Rn	imm → sign extension → Rn	1110nnnniiiiiiii	—	—	—
MOV.W	@(disp*,PC),Rn	(disp × 2 + PC + 4) → sign extension → Rn	1001nnnndddddddd	—	—	—
MOV.L	@(disp*,PC),Rn	(disp × 4 + PC & H'FFFF FFFC + 4) → Rn	1101nnnndddddddd	—	—	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	—	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	—	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	—	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	—	—
MOV.B	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0000	—	—	—
MOV.W	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0001	—	—	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	—	—
MOV.B	Rm,@-Rn	Rn-1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	—	—
MOV.W	Rm,@-Rn	Rn-2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	—	—
MOV.L	Rm,@-Rn	Rn-4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	—	—
MOV.B	@Rm+,Rn	(Rm)→ sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	—	—
MOV.W	@Rm+,Rn	(Rm) → sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	—	—
MOV.L	@Rm+,Rn	(Rm) → Rn, Rm + 4 → Rm	0110nnnnmmmm0110	—	—	—
MOV.B	R0,@(disp*,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	—	—
MOV.W	R0,@(disp*,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	—	—
MOV.L	Rm,@(disp*,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	—	—
MOV.B	@(disp*,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmdddd	—	—	—
MOV.W	@(disp*,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmdddd	—	—	—
MOV.L	@(disp*,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—	—
MOV.B	R0,@(disp*,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	—	—
MOV.W	R0,@(disp*,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	—	—
MOV.L	R0,@(disp*,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	—	—
MOV.B	@(disp*,GBR),R0	(disp + GBR) → sign extension → R0	11000100dddddddd	—	—	—
MOV.W	@(disp*,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101dddddddd	—	—	—
MOV.L	@(disp*,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	—	—
MOVA	@(disp*,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111dddddddd	—	—	—
MOVCO.L	R0,@Rn	LDST → T If (T == 1) R0 → (Rn) 0 → LDST	0000nnnn01110011	—	LDST	New
MOVLI.L	@Rm,R0	1 → LDST (Rm) → R0 When interrupt/exception occurred 0 → LDST	0000mmmm01100011	—	—	New
MOVUA.L	@Rm,R0	(Rm) → R0 Load non-boundary alignment data	0100mmmm10101001	—	—	New
MOVUA.L	@Rm+,R0	(Rm) → R0, Rm + 4 → Rm Load non-boundary alignment data	0100mmmm11101001	—	—	New
MOVT	Rn	T → Rn	0000nnnn00101001	—	—	—
SWAP.B	Rm,Rn	Rm → swap lower 2 bytes → Rn	0110nnnnmmmm1000	—	—	—
SWAP.W	Rm,Rn	Rm → swap upper/lower words → Rn	0110nnnnmmmm1001	—	—	—
XTRCT	Rm,Rn	Rm:Rn middle 32 bits → Rn	0010nnnnmmmm1101	—	—	—

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

Table 3.5 Arithmetic Operation Instructions

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

Instruction		Operation	Instruction Code	Privileged	T Bit	New
DMULU.L	Rm,Rn	Unsigned, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm0101	—	—	—
DT	Rn	Rn – 1 → Rn; when Rn = 0, 1 → T When Rn ≠ 0, 0 → T	0100nnnn00010000	—	Comparison result	—
EXTS.B	Rm,Rn	Rm sign-extended from byte → Rn	0110nnnnmmmm1110	—	—	—
EXTS.W	Rm,Rn	Rm sign-extended from word → Rn	0110nnnnmmmm1111	—	—	—
EXTU.B	Rm,Rn	Rm zero-extended from byte → Rn	0110nnnnmmmm1100	—	—	—
EXTU.W	Rm,Rn	Rm zero-extended from word → Rn	0110nnnnmmmm1101	—	—	—
MAC.L	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	—	—
MAC.W	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	—	—
MUL.L	Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	—	—	—
MULS.W	Rm,Rn	Signed, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1111	—	—	—
MULU.W	Rm,Rn	Unsigned, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1110	—	—	—
NEG	Rm,Rn	0 – Rm → Rn	0110nnnnmmmm1011	—	—	—
NEGC	Rm,Rn	0 – Rm – T → Rn, borrow → T	0110nnnnmmmm1010	—	Borrow	—
SUB	Rm,Rn	Rn – Rm → Rn	0011nnnnmmmm1000	—	—	—
SUBC	Rm,Rn	Rn – Rm – T → Rn, borrow → T	0011nnnnmmmm1010	—	Borrow	—
SUBV	Rm,Rn	Rn – Rm → Rn, underflow → T	0011nnnnmmmm1011	—	Underflow	—

Table 3.6 Logic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
AND	Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	—	—
AND	#imm,R0	R0 & imm → R0	11001001iiiiiiii	—	—	—
AND.B	#imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	—	—
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	—	—
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	—	—
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	—	—
OR.B	#imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	—	—
TAS.B	@Rn	When (Rn) = 0, 1 → T Otherwise, 0 → T In both cases, 1 → MSB of (Rn)	0100nnnn00011011	—	Test result	—
TST	Rm,Rn	Rn & Rm; when result = 0, 1 → T Otherwise, 0 → T	0010nnnnmmmm1000	—	Test result	—
TST	#imm,R0	R0 & imm; when result = 0, 1 → T Otherwise, 0 → T	11001000iiiiiiii	—	Test result	—
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; when result = 0, 1 → T Otherwise, 0 → T	11001100iiiiiiii	—	Test result	—
XOR	Rm,Rn	Rn ∧ Rm → Rn	0010nnnnmmmm1010	—	—	—
XOR	#imm,R0	R0 ∧ imm → R0	11001010iiiiiiii	—	—	—
XOR.B	#imm,@(R0,GBR)	(R0 + GBR) ∧ imm → (R0 + GBR)	11001110iiiiiiii	—	—	—

Table 3.7 Shift Instructions

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

Table 3.8 Branch Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
BF	label	When T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001011dddddddd	—	—	—
BF/S	label	Delayed branch; when T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001111dddddddd	—	—	—
BT	label	When T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001001dddddddd	—	—	—
BT/S	label	Delayed branch; when T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001101dddddddd	—	—	—
BRA	label	Delayed branch, disp × 2 + PC + 4 → PC	1010dddddddddddd	—	—	—
BRAF	Rn	Delayed branch, Rn + PC + 4 → PC	0000nnnn00100011	—	—	—
BSR	label	Delayed branch, PC + 4 → PR, disp × 2 + PC + 4 → PC	1011dddddddddddd	—	—	—
BSRF	Rn	Delayed branch, PC + 4 → PR, Rn + PC + 4 → PC	0000nnnn00000011	—	—	—
JMP	@Rn	Delayed branch, Rn → PC	0100nnnn00101011	—	—	—
JSR	@Rn	Delayed branch, PC + 4 → PR, Rn → PC	0100nnnn00001011	—	—	—
RTS		Delayed branch, PR → PC	0000000000001011	—	—	—

Table 3.9 System Control Instructions

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

Instruction		Operation	Instruction Code	Privileged	T Bit	New
LDC	Rm,SGR	Rm → SGR	0100mmmm00111010	Privileged	—	—
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	Privileged	—	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	Privileged	—	—
LDC	Rm,DBR	Rm → DBR	0100mmmm11111010	Privileged	—	—
LDC	Rm,Rn_BANK	Rm → Rn_BANK (n = 0 to 7)	0100mmmm1nnn1110	Privileged	—	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	Privileged	LSB	—
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	—	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	Privileged	—	—
LDC.L	@Rm+,SGR	(Rm) → SGR, Rm + 4 → Rm	0100mmmm00110110	Privileged	—	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	—	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	—	—
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	—	—
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	—	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	—	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	—	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	—	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	—	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	—	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	—	—
LDTLB		PTEH/PTEL → TLB	0000000000111000	Privileged	—	—
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	—	—
NOP		No operation	0000000000001001	—	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—	—
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—	—
PREFI	@Rn	Reads 32-byte instruction block into instruction cache	0000nnnn11010011	—	—	New
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	Privileged	—	—
SETS		1 → S	0000000001011000	—	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit	New
SETT		1 → T	0000000000011000	—	1	—
SLEEP		Sleep or standby	0000000000011011	Privileged	—	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	Privileged	—	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	—	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	Privileged	—	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	Privileged	—	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	Privileged	—	—
STC	SGR,Rn	SGR → Rn	0000nnnn00111010	Privileged	—	—
STC	DBR,Rn	DBR → Rn	0000nnnn11111010	Privileged	—	—
STC	Rm_BANK,Rn	Rm_BANK → Rn (m = 0 to 7)	0000nnnn1mmm0010	Privileged	—	—
STC.L	SR,@-Rn	Rn – 4 → Rn, SR → (Rn)	0100nnnn00000011	Privileged	—	—
STC.L	GBR,@-Rn	Rn – 4 → Rn, GBR → (Rn)	0100nnnn00010011	—	—	—
STC.L	VBR,@-Rn	Rn – 4 → Rn, VBR → (Rn)	0100nnnn00100011	Privileged	—	—
STC.L	SSR,@-Rn	Rn – 4 → Rn, SSR → (Rn)	0100nnnn00110011	Privileged	—	—
STC.L	SPC,@-Rn	Rn – 4 → Rn, SPC → (Rn)	0100nnnn01000011	Privileged	—	—
STC.L	SGR,@-Rn	Rn – 4 → Rn, SGR → (Rn)	0100nnnn00110010	Privileged	—	—
STC.L	DBR,@-Rn	Rn – 4 → Rn, DBR → (Rn)	0100nnnn11110010	Privileged	—	—
STC.L	Rm_BANK,@-Rn	Rn – 4 → Rn, Rm_BANK → (Rn) (m = 0 to 7)	0100nnnn1mmm0011	Privileged	—	—
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—	—	—
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—	—	—
STS	PR,Rn	PR → Rn	0000nnnn00101010	—	—	—
STS.L	MACH,@-Rn	Rn – 4 → Rn, MACH → (Rn)	0100nnnn00000010	—	—	—
STS.L	MACL,@-Rn	Rn – 4 → Rn, MACL → (Rn)	0100nnnn00010010	—	—	—
STS.L	PR,@-Rn	Rn – 4 → Rn, PR → (Rn)	0100nnnn00100010	—	—	—
SYNCO		Prevents the next instruction from being issued until instructions issued before this instruction have been completed.	0000000010101011	—	—	New
TRAPA	#imm	PC + 2 → SPC, SR → SSR, #imm << 2 → TRA, H'160 → EXPEVT, VBR + H'0100 → PC	11000011iiiiiiii	—	—	—

Table 3.10 Floating-Point Single-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FLDI0	FRn	H'0000 0000 → FRn	1111nnnn10001101	—	—	—
FLDI1	FRn	H'3F80 0000 → FRn	1111nnnn10011101	—	—	—
FMOV	FRm,FRn	FRm → FRn	1111nnnnmmmm1100	—	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnmmmm1000	—	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnmmmm0110	—	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnmmmm1001	—	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnmmmm1010	—	—	—
FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnmmmm1011	—	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnmmmm0111	—	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0mmm01100	—	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0mmmm1000	—	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0mmmm0110	—	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0mmmm1001	—	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnmmm01010	—	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnmmm01011	—	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnmmm00111	—	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111mmmm00011101	—	—	—
FSTS	FPUL,FRn	FPUL → FRn	1111nnnn00001101	—	—	—
FABS	FRn	FRn & H'7FFF FFFF → FRn	1111nnnn01011101	—	—	—
FADD	FRm,FRn	FRn + FRm → FRn	1111nnnnmmmm0000	—	—	—
FCMP/EQ	FRm,FRn	When FRn = FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0100	—	Comparison result	—
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Comparison result	—
FDIV	FRm,FRn	FRn/FRm → FRn	1111nnnnmmmm0011	—	—	—
FLOAT	FPUL,FRn	(float) FPUL → FRn	1111nnnn00101101	—	—	—
FMAC	FR0,FRm,FRn	FR0*FRm + FRn → FRn	1111nnnnmmmm1110	—	—	—
FMUL	FRm,FRn	FRn*FRm → FRn	1111nnnnmmmm0010	—	—	—
FNEG	FRn	FRn ∧ H'8000 0000 → FRn	1111nnnn01001101	—	—	—
FSQRT	FRn	√FRn → FRn	1111nnnn01101101	—	—	—
FSUB	FRm,FRn	FRn – FRm → FRn	1111nnnnmmmm0001	—	—	—
FTRC	FRm,FPUL	(long) FRm → FPUL	1111mmmm00111101	—	—	—

Table 3.11 Floating-Point Double-Precision Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FABS	DRn	DRn & H'7FFF FFFF FFFF FFFF → DRn	1111nnn001011101	—	—	—
FADD	DRm,DRn	DRn + DRm → DRn	1111nnn0mmm00000	—	—	—
FCMP/EQ	DRm,DRn	When DRn = DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00100	—	Comparison result	—
FCMP/GT	DRm,DRn	When DRn > DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00101	—	Comparison result	—
FDIV	DRm,DRn	DRn /DRm → DRn	1111nnn0mmm00011	—	—	—
FCNVDS	DRm,FPUL	double_to_ float(DRm) → FPUL	1111mmm010111101	—	—	—
FCNVSD	FPUL,DRn	float_to_ double (FPUL) → DRn	1111nnn010101101	—	—	—
FLOAT	FPUL,DRn	(float)FPUL → DRn	1111nnn000101101	—	—	—
FMUL	DRm,DRn	DRn *DRm → DRn	1111nnn0mmm00010	—	—	—
FNEG	DRn	DRn ^ H'8000 0000 0000 0000 → DRn	1111nnn001001101	—	—	—
FSQRT	DRn	√DRn → DRn	1111nnn001101101	—	—	—
FSUB	DRm,DRn	DRn – DRm → DRn	1111nnn0mmm00001	—	—	—
FTRC	DRm,FPUL	(long) DRm → FPUL	1111mmm000111101	—	—	—

Table 3.12 Floating-Point Control Instructions

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

Table 3.13 Floating-Point Graphics Acceleration Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit	New
FMOV	DRm,XDn	DRm → XDn	1111nnn1mmm01100	—	—	—
FMOV	XDm,DRn	XDm → DRn	1111nnn0mmm11100	—	—	—
FMOV	XDm,XDn	XDm → XDn	1111nnn1mmm11100	—	—	—
FMOV	@Rm,XDn	(Rm) → XDn	1111nnn1mmmm1000	—	—	—
FMOV	@Rm+,XDn	(Rm) → XDn, Rm + 8 → Rm	1111nnn1mmmm1001	—	—	—
FMOV	@(R0,Rm),XDn	(R0 + Rm) → XDn	1111nnn1mmmm0110	—	—	—
FMOV	XDm,@Rn	XDm → (Rn)	1111nnnnmmm11010	—	—	—
FMOV	XDm,@-Rn	Rn – 8 → Rn, XDm → (Rn)	1111nnnnmmm11011	—	—	—
FMOV	XDm,@(R0,Rn)	XDm → (R0 + Rn)	1111nnnnmmm10111	—	—	—
FIPR	FVm,FVn	inner_product (FVm, FVn) → FR[n+3]	1111nnmm11101101	—	—	—
FTRV	XMTRX,FVn	transform_vector (XMTRX, FVn) → FVn	1111nn0111111101	—	—	—
FRCHG		~FPSCR.FR → FPSCR.FR	1111101111111101	—	—	—
FSCHG		~FPSCR.SZ → FPSCR.SZ	1111001111111101	—	—	—
FPCHG		~FPSCR.PR → FPSCR.PR	1111011111111101	—	—	New
FSRRA	FRn	1/sqrt (FRn)* → FRn	1111nnnn01111101	—	—	New
FSCA	FPUL,DRn	sin(FPUL) → FRn cos(FPUL) → FR[n + 1]	1111nnn011111101	—	—	New

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

Section 4 Pipelining

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

4.1 Pipelines

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

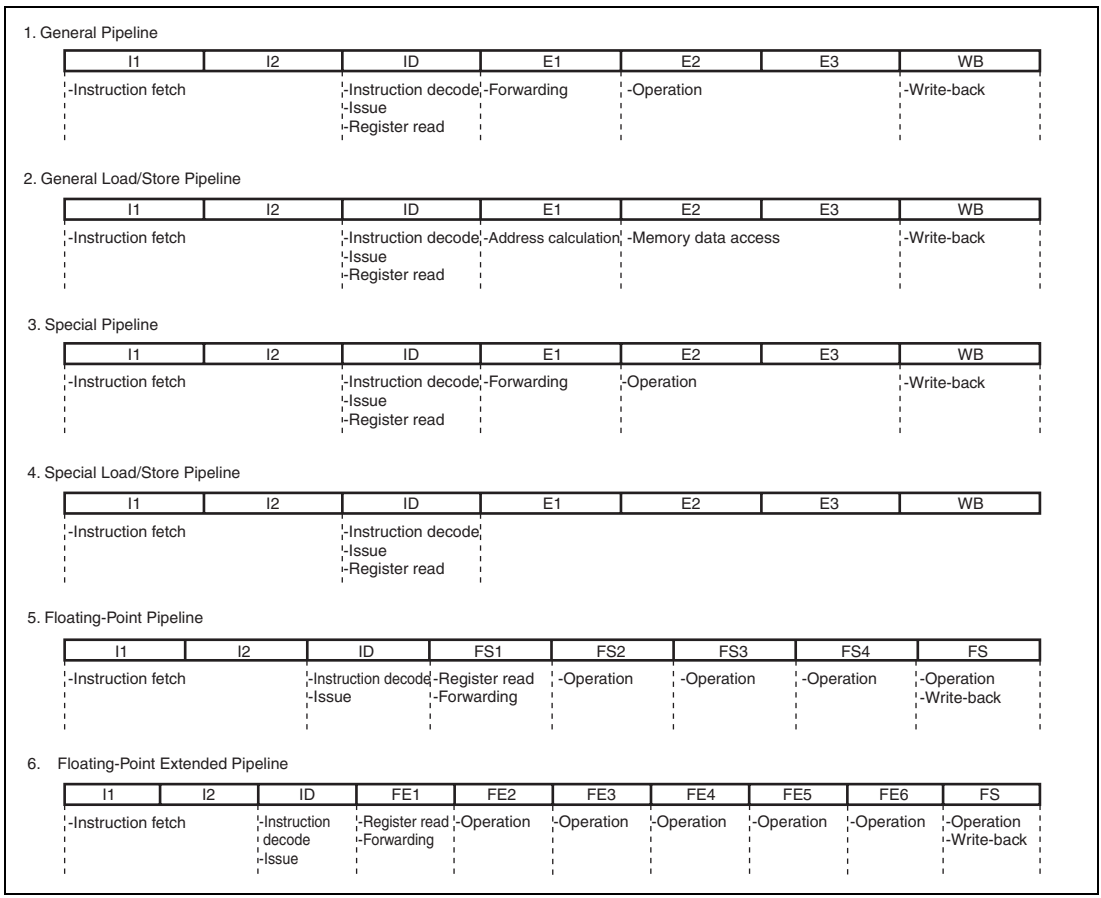


Figure 4.1 Basic Pipelines

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

Table 4.1 Representations of Instruction Execution Patterns

Representation	Description
E1 E2 E3 WB	CPU EX pipe is occupied
S1 S2 S3 WB	CPU LS pipe is occupied (with memory access)
s1 s2 s3 WB	CPU LS pipe is occupied (without memory access)
E1/S1	Either CPU EX pipe or CPU LS pipe is occupied
E1S1 , E1s1	Both CPU EX pipe and CPU LS pipe are occupied
M2 M3 MS	CPU MULT operation unit is occupied
FE1 FE2 FE3 FE4 FE5 FE6 FS	FPU-EX pipe is occupied
FS1 FS2 FS3 FS4 FS	FPU-LS pipe is occupied
ID	ID stage is locked
	Both CPU and FPU pipes are occupied

(1-1) BF, BF/S, BT, BT/S, BRA, BSR:1 issue cycle + 0 to 2 branch cycles



(Branch destination instruction)

Note: In branch instructions that are categorized as (1-1), the number of branch cycles may be reduced by prefetching.

(1-2) JSR, JMP, BRAF, BSRF: 1 issue cycle + 3 branch cycles



(Branch destination instruction)

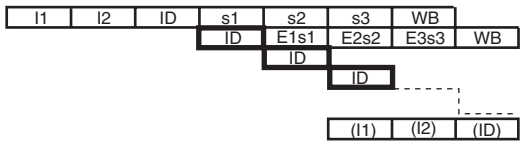
(1-3) RTS: 1 issue cycle + 0 to 3 branch cycles



(Branch destination instruction)

Note: The number of branch cycles may be 0 by prefetching instruction.

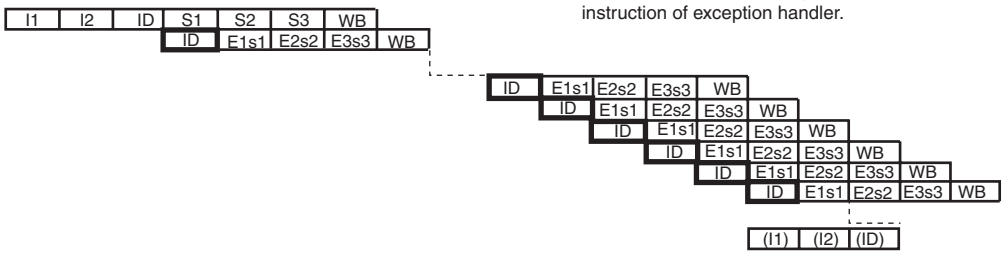
(1-4) RTE: 4 issue cycles + 1 branch cycles



(Branch destination instruction)

(1-5) TRAPA: 8 issue cycles + 5 cycles + 1 branch cycle

Note: It is 14 cycles to the ID stage in the first instruction of exception handler.



(1-6) SLEEP: 2 issue cycles



Note: It is not constant cycles to the clock halted period.

Figure 4.2 Instruction Execution Patterns (1)

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

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

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

I1	I2	ID	E1	E2	E3	WB

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

MOVA

I1	I2	ID	s1	s2	s3	WB

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

MOV#, NOP

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

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

MOV

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

Figure 4.2 Instruction Execution Patterns (2)

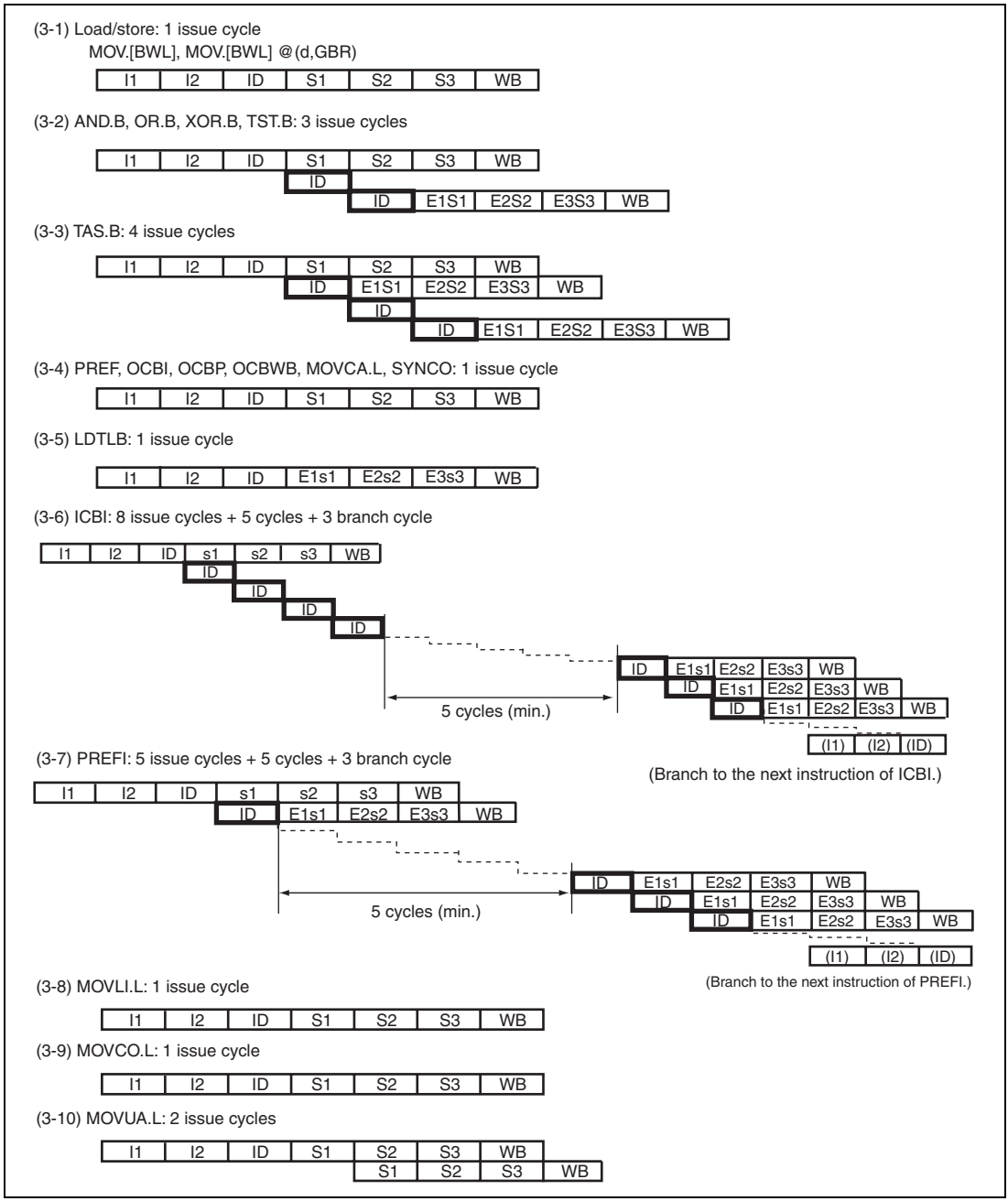


Figure 4.2 Instruction Execution Patterns (3)

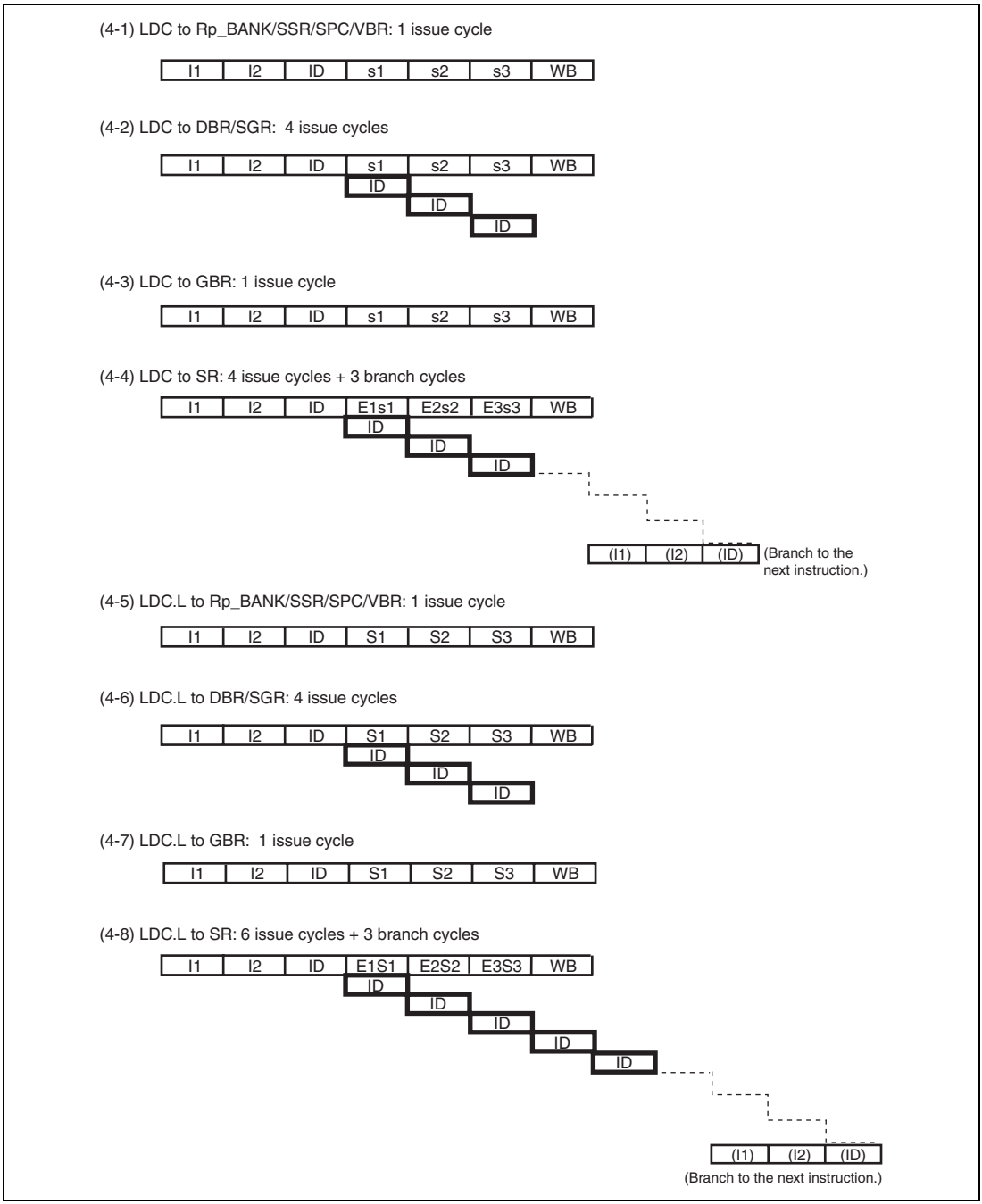


Figure 4.2 Instruction Execution Patterns (4)

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

I1	I2	ID	s1	s2	s3	WB

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

I1	I2	ID	E1s1	E2s2	E3s3	WB

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

I1	I2	ID	S1	S2	S3	WB

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

I1	I2	ID	E1S1	E2S2	E3S3	WB

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

I1	I2	ID	s1	s2	s3	WB

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

I1	I2	ID	S1	S2	S3	WB

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

I1	I2	ID	s1	s2	s3	WB

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

I1	I2	ID	S1	S2	S3	WB

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

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

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

Figure 4.2 Instruction Execution Patterns (5)

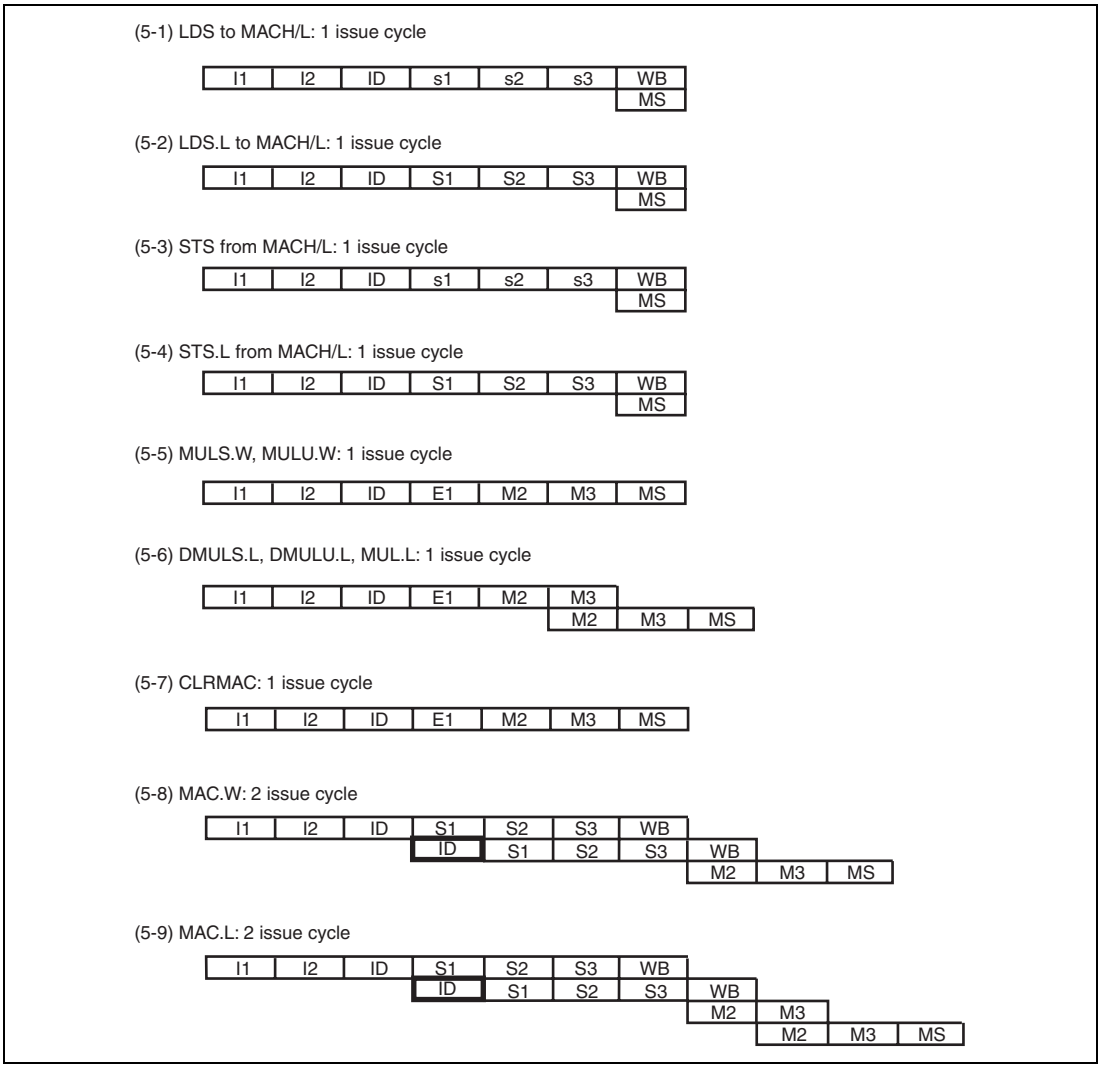
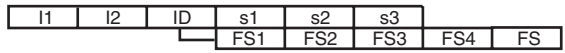
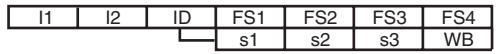


Figure 4.2 Instruction Execution Patterns (6)

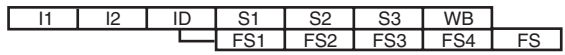
(6-1) LDS to FPUL: 1 issue cycle



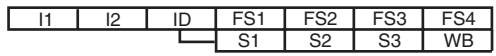
(6-2) STS from FPUL: 1 issue cycle



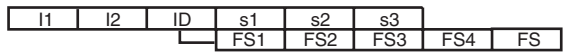
(6-3) LDS.L to FPUL: 1 issue cycle



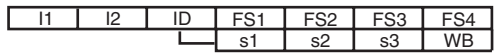
(6-4) STS.L from FPUL: 1 issue cycle



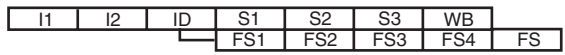
(6-5) LDS to FPSCR: 1 issue cycle



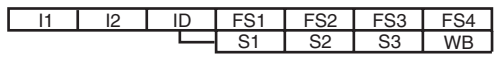
(6-6) STS from FPSCR: 1 issue cycle



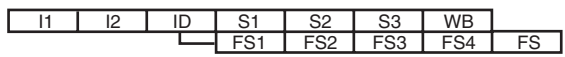
(6-7) LDS.L to FPSCR: 1 issue cycle



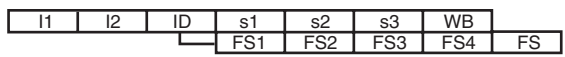
(6-8) STS.L from FPSCR: 1 issue cycle



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



(6-10) FLDS: 1 issue cycle



(6-11) FSTS: 1 issue cycle

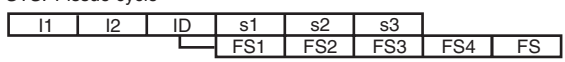
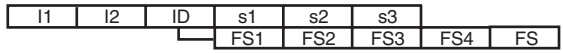
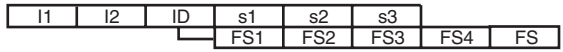


Figure 4.2 Instruction Execution Patterns (7)

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

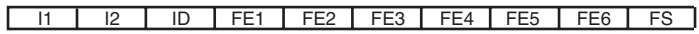


(6-13) FLDI0, FLDI1: 1 issue cycle

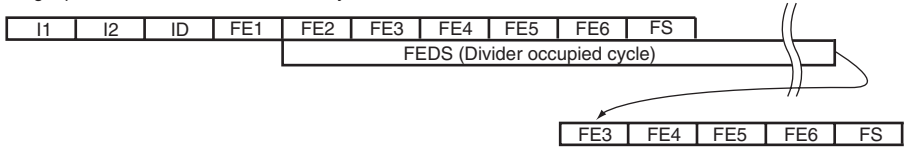


(6-14) Single-precision floating-point computation: 1 issue cycle

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

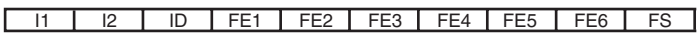


(6-15) Single-precision FDIV/FSQRT: 1 issue cycle



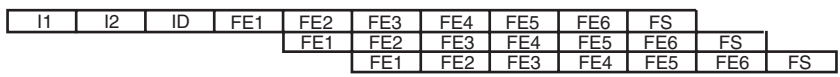
(6-16) Double-precision floating-point computation: 1 issue cycle

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



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

FMUL



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

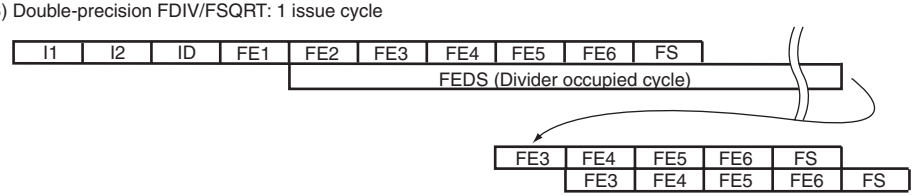


Figure 4.2 Instruction Execution Patterns (8)

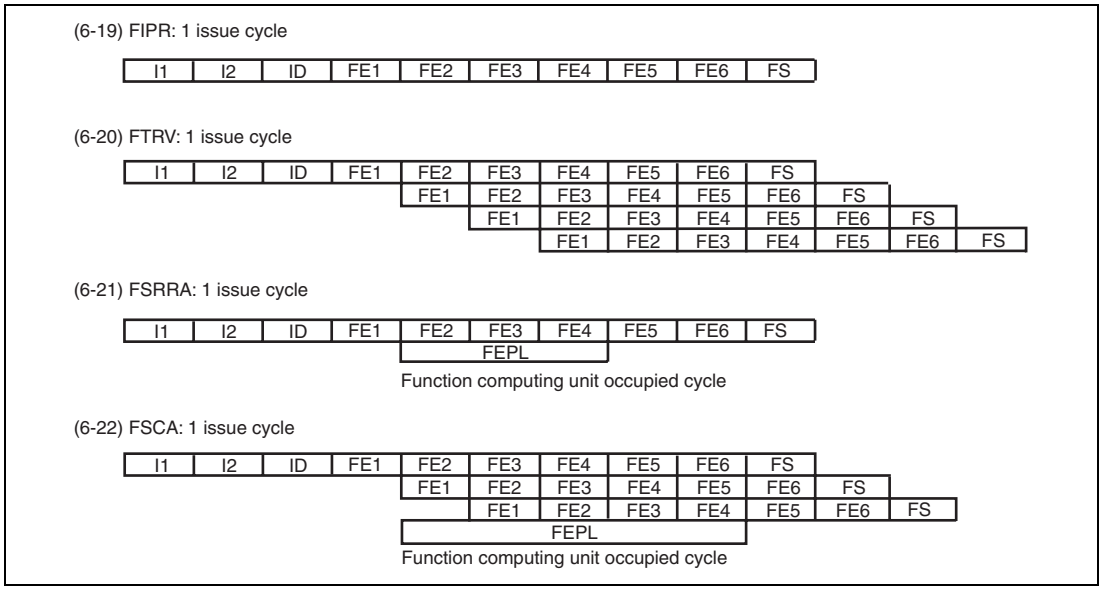


Figure 4.2 Instruction Execution Patterns (9)

4.2 Parallel-Executability

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

Table 4.2 Instruction Groups

Instruction Group	Instruction			
EX	ADD	DT	ROTL	SHLR8
	ADDC	EXTS	ROTR	SHLR16
	ADDV	EXTU	SETS	SUB
	AND #imm,R0	MOVT	SETT	SUBC
	AND Rm,Rn	MUL.L	SHAD	SUBV
	CLRMAC	MULS.W	SHAL	SWAP
	CLRS	MULU.W	SHAR	TST #imm,R0
	CLRT	NEG	SHLD	TST Rm,Rn
	CMP	NEGC	SHLL	XOR #imm,R0
	DIV0S	NOT	SHLL2	XOR Rm,Rn
	DIV0U	OR #imm,R0	SHLL8	XTRCT
	DIV1	OR Rm,Rn	SHLL16	
	DMUS.L	ROTCL	SHLR	
	DMULU.L	ROTCR	SHLR2	
MT	MOV #imm,Rn	MOV Rm,Rn	NOP	
BR	BF	BRAF	BT	JSR
	BF/S	BSR	BT/S	RTS
	BRA	BSRF	JMP	

Instruction Group	Instruction			
LS	FABS	FMOV.S FR,@adr	MOV.[BWL] @adr,R	STC CR2,Rn
	FNEG	FSTS	MOV.[BWL] R,@adr	STC.L CR2,@-Rn
	FLDI0	LDC Rm,CR1	MOVA	STS SR2,Rn
	FLDI1	LDC.L @Rm+,CR1	MOVCA.L	STS.L SR2,@-Rn
	FLDS	LDS Rm,SR1	MOVUA	STS SR1,Rn
	FMOV @adr,FR	LDS Rm,SR2	OCBI	STS.L SR1,@-Rn
	FMOV FR,@adr	LDS.L @adr,SR2	OCBP	
	FMOV FR,FR	LDS.L @Rm+,SR1	OCBWB	
	FMOV.S @adr,FR	LDS.L @Rm+,SR2	PREF	
FE	FADD	FDIV	FRCHG	FSCA
	FSUB	FIPR	FSCHG	FSRRA
	FCMP (S/D)	FLOAT	FSQRT	FPCHG
	FCNVDS	FMAC	FTRC	
	FCNVSD	FMUL	FTRV	
CO	AND.B #imm,@(R0,GBR)	LDC.L @Rm+,SR	PREFI	TRAPA
	ICBI	LDTLB	RTE	TST.B #imm,@(R0,GBR)
	LDC Rm,DBR	MAC.L	SLEEP	XOR.B #imm,@(R0,GBR)
	LDC Rm, SGR	MAC.W	STC SR,Rn	
	LDC Rm,SR	MOVCO	STC.L SR,@-Rn	
	LDC.L @Rm+,DBR	MOVLI	SYNCO	
	LDC.L @Rm+,SGR	OR.B #imm,@(R0,GBR)	TAS.B	

[Legend]

R: Rm/Rn

@adr: Address

SR1: MACH/MACL/PR

SR2: FPUL/FPSCR

CR1: GBR/Rp_BANK/SPC/SSR/VBR

CR2: CR1/DBR/SGR

FR: FRm/FRn/DRm/DRn/XDm/XDn

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

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

Table 4.3 Combination of Preceding and Following Instructions

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

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

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

[Legend]

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

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

LS: Original LS instructions except FLSR and FLSM

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

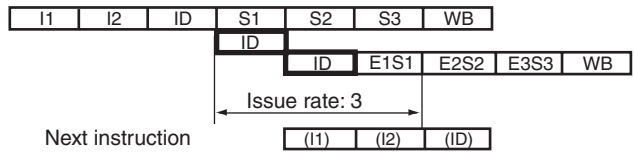
4.3 Issue Rates and Execution Cycles

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

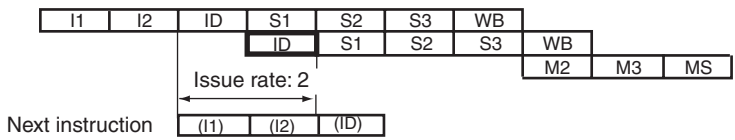
1. Issue Rate

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

E.g. AND.B instruction



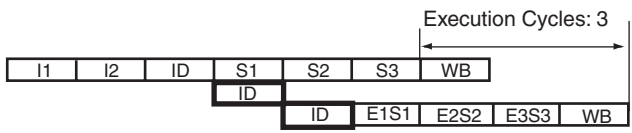
E.g. MAC.W instruction



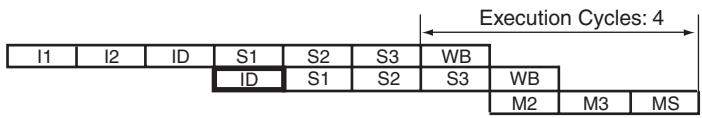
2. Execution Cycles

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

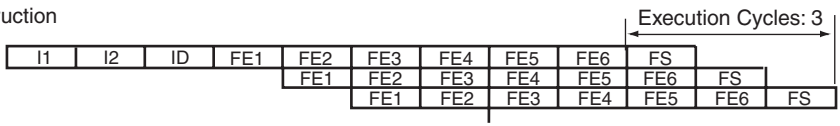
CPU instruction
E.g. AND.B instruction



E.g. MAC.W instruction



FPU instruction
E.g. FMUL instruction



E.g. FDIV instruction

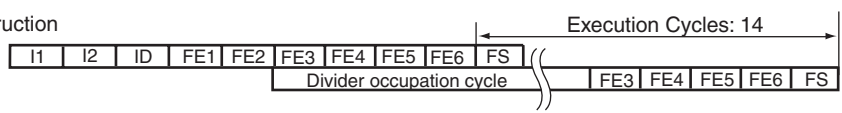


Table 4.4 Issue Rates and Execution Cycles

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Data transfer instructions	1	EXTS.B	Rm,Rn	EX	1	1	2-1
	2	EXTS.W	Rm,Rn	EX	1	1	2-1
	3	EXTU.B	Rm,Rn	EX	1	1	2-1
	4	EXTU.W	Rm,Rn	EX	1	1	2-1
	5	MOV	Rm,Rn	MT	1	1	2-4
	6	MOV	#imm,Rn	MT	1	1	2-3
	7	MOVA	@(disp,PC),R0	LS	1	1	2-2
	8	MOV.W	@(disp,PC),Rn	LS	1	1	3-1
	9	MOV.L	@(disp,PC),Rn	LS	1	1	3-1
	10	MOV.B	@Rm,Rn	LS	1	1	3-1
	11	MOV.W	@Rm,Rn	LS	1	1	3-1
	12	MOV.L	@Rm,Rn	LS	1	1	3-1
	13	MOV.B	@Rm+,Rn	LS	1	1	3-1
	14	MOV.W	@Rm+,Rn	LS	1	1	3-1
	15	MOV.L	@Rm+,Rn	LS	1	1	3-1
	16	MOV.B	@(disp,Rm),R0	LS	1	1	3-1
	17	MOV.W	@(disp,Rm),R0	LS	1	1	3-1
	18	MOV.L	@(disp,Rm),Rn	LS	1	1	3-1
	19	MOV.B	@(R0,Rm),Rn	LS	1	1	3-1
	20	MOV.W	@(R0,Rm),Rn	LS	1	1	3-1
	21	MOV.L	@(R0,Rm),Rn	LS	1	1	3-1
	22	MOV.B	@(disp,GBR),R0	LS	1	1	3-1
	23	MOV.W	@(disp, GBR),R0	LS	1	1	3-1
	24	MOV.L	@(disp, GBR),R0	LS	1	1	3-1
	25	MOV.B	Rm,@Rn	LS	1	1	3-1
	26	MOV.W	Rm,@Rn	LS	1	1	3-1
	27	MOV.L	Rm,@Rn	LS	1	1	3-1
	28	MOV.B	Rm,@-Rn	LS	1	1	3-1
	29	MOV.W	Rm,@-Rn	LS	1	1	3-1

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

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Fixed-point arithmetic instructions	60	CMP/GT	Rm,Rn	EX	1	1	2-1
	61	CMP/HI	Rm,Rn	EX	1	1	2-1
	62	CMP/HS	Rm,Rn	EX	1	1	2-1
	63	CMP/PL	Rn	EX	1	1	2-1
	64	CMP/PZ	Rn	EX	1	1	2-1
	65	CMP/STR	Rm,Rn	EX	1	1	2-1
	66	DIV0S	Rm,Rn	EX	1	1	2-1
	67	DIV0U		EX	1	1	2-1
	68	DIV1	Rm,Rn	EX	1	1	2-1
	69	DMULS.L	Rm,Rn	EX	1	2	5-6
	70	DMULU.L	Rm,Rn	EX	1	2	5-6
	71	DT	Rn	EX	1	1	2-1
	72	MAC.L	@Rm+,@Rn+	CO	2	5	5-9
	73	MAC.W	@Rm+,@Rn+	CO	2	4	5-8
	74	MUL.L	Rm,Rn	EX	1	2	5-6
	75	MULS.W	Rm,Rn	EX	1	1	5-5
	76	MULU.W	Rm,Rn	EX	1	1	5-5
	77	NEG	Rm,Rn	EX	1	1	2-1
	78	NEGC	Rm,Rn	EX	1	1	2-1
	79	SUB	Rm,Rn	EX	1	1	2-1
	80	SUBC	Rm,Rn	EX	1	1	2-1
	81	SUBV	Rm,Rn	EX	1	1	2-1
Logical instructions	82	AND	Rm,Rn	EX	1	1	2-1
	83	AND	#imm,R0	EX	1	1	2-1
	84	AND.B	#imm,@(R0,GBR)	CO	3	3	3-2
	85	NOT	Rm,Rn	EX	1	1	2-1
	86	OR	Rm,Rn	EX	1	1	2-1
	87	OR	#imm,R0	EX	1	1	2-1
	88	OR.B	#imm,@(R0,GBR)	CO	3	3	3-2
	89	TAS.B	@Rn	CO	4	4	3-3
	90	TST	Rm,Rn	EX	1	1	2-1

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Logical instructions	91	TST	#imm,R0	EX	1	1	2-1
	92	TST.B	#imm,@(R0,GBR)	CO	3	3	3-2
	93	XOR	Rm,Rn	EX	1	1	2-1
	94	XOR	#imm,R0	EX	1	1	2-1
	95	XOR.B	#imm,@(R0,GBR)	CO	3	3	3-2
Shift instructions	96	ROTL	Rn	EX	1	1	2-1
	97	ROTR	Rn	EX	1	1	2-1
	98	ROTCL	Rn	EX	1	1	2-1
	99	ROTCR	Rn	EX	1	1	2-1
	100	SHAD	Rm,Rn	EX	1	1	2-1
	101	SHAL	Rn	EX	1	1	2-1
	102	SHAR	Rn	EX	1	1	2-1
	103	SHLD	Rm,Rn	EX	1	1	2-1
	104	SHLL	Rn	EX	1	1	2-1
	105	SHLL2	Rn	EX	1	1	2-1
	106	SHLL8	Rn	EX	1	1	2-1
	107	SHLL16	Rn	EX	1	1	2-1
	108	SHLR	Rn	EX	1	1	2-1
	109	SHLR2	Rn	EX	1	1	2-1
	110	SHLR8	Rn	EX	1	1	2-1
	111	SHLR16	Rn	EX	1	1	2-1
Branch instructions	112	BF	disp	BR	1+0 to 2	1	1-1
	113	BF/S	disp	BR	1+0 to 2	1	1-1
	114	BT	disp	BR	1+0 to 2	1	1-1
	115	BT/S	disp	BR	1+0 to 2	1	1-1
	116	BRA	disp	BR	1+0 to 2	1	1-1
	117	BRAF	Rm	BR	1+3	1	1-2
	118	BSR	disp	BR	1+0 to 2	1	1-1
	119	BSRF	Rm	BR	1+3	1	1-2
	120	JMP	@Rn	BR	1+3	1	1-2

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Branch instructions	121	JSR	@Rn	BR	1+3	1	1-2
	122	RTS		BR	1+0 to 3	1	1-3
System control instructions	123	NOP		MT	1	1	2-3
	124	CLRMAC		EX	1	1	5-7
	125	CLRS		EX	1	1	2-1
	126	CLRT		EX	1	1	2-1
	127	ICBI	@Rn	CO	8+5+3	13	3-6
	128	SETS		EX	1	1	2-1
	129	SETT		EX	1	1	2-1
	130	PREFI		CO	5+5+3	10	3-7
	131	SYNCO	@Rn	CO	Undefined	Undefined	3-4
	132	TRAPA	#imm	CO	8+5+1	13	1-5
	133	RTE		CO	4+1	4	1-4
	134	SLEEP		CO	Undefined	Undefined	1-6
	135	LDTLB		CO	1	1	3-5
	136	LDC	Rm,DBR	CO	4	4	4-2
	137	LDC	Rm,SGR	CO	4	4	4-2
	138	LDC	Rm,GBR	LS	1	1	4-3
	139	LDC	Rm,Rp_BANK	LS	1	1	4-1
	140	LDC	Rm,SR	CO	4+3	4	4-4
	141	LDC	Rm,SSR	LS	1	1	4-1
	142	LDC	Rm,SPC	LS	1	1	4-1
	143	LDC	Rm,VBR	LS	1	1	4-1
	144	LDC.L	@Rm+,DBR	CO	4	4	4-6
	145	LDC.L	@Rm+,SGR	CO	4	4	4-6
	146	LDC.L	@Rm+,GBR	LS	1	1	4-7
	147	LDC.L	@Rm+,Rp_BANK	LS	1	1	4-5
	148	LDC.L	@Rm+,SR	CO	6+3	4	4-8
	149	LDC.L	@Rm+,SSR	LS	1	1	4-5
	150	LDC.L	@Rm+,SPC	LS	1	1	4-5

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
System control instructions	151	LDC.L	@Rm+,VBR	LS	1	1	4-5
	152	LDS	Rm,MACH	LS	1	1	5-1
	153	LDS	Rm,MACL	LS	1	1	5-1
	154	LDS	Rm,PR	LS	1	1	4-13
	155	LDS.L	@Rm+,MACH	LS	1	1	5-2
	156	LDS.L	@Rm+,MACL	LS	1	1	5-2
	157	LDS.L	@Rm+,PR	LS	1	1	4-14
	158	STC	DBR,Rn	LS	1	1	4-9
	159	STC	SGR,Rn	LS	1	1	4-9
	160	STC	GBR,Rn	LS	1	1	4-9
	161	STC	Rp_BANK,Rn	LS	1	1	4-9
	162	STC	SR,Rn	CO	1	1	4-10
	163	STC	SSR,Rn	LS	1	1	4-9
	164	STC	SPC,Rn	LS	1	1	4-9
	165	STC	VBR,Rn	LS	1	1	4-9
	166	STC.L	DBR,@-Rn	LS	1	1	4-11
	167	STC.L	SGR,@-Rn	LS	1	1	4-11
	168	STC.L	GBR,@-Rn	LS	1	1	4-11
	169	STC.L	Rp_BANK,@-Rn	LS	1	1	4-11
	170	STC.L	SR,@-Rn	CO	1	1	4-12
	171	STC.L	SSR,@-Rn	LS	1	1	4-11
	172	STC.L	SPC,@-Rn	LS	1	1	4-11
	173	STC.L	VBR,@-Rn	LS	1	1	4-11
	174	STS	MACH,Rn	LS	1	1	5-3
	175	STS	MACL,Rn	LS	1	1	5-3
	176	STS	PR,Rn	LS	1	1	4-15
	177	STS.L	MACH,@-Rn	LS	1	1	5-4
	178	STS.L	MACL,@-Rn	LS	1	1	5-4
	179	STS.L	PR,@-Rn	LS	1	1	4-16

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Single-precision floating-point instructions	180	FLDI0	FRn	LS	1	1	6-13
	181	FLDI1	FRn	LS	1	1	6-13
	182	FMOV	FRm,FRn	LS	1	1	6-9
	183	FMOV.S	@Rm,FRn	LS	1	1	6-9
	184	FMOV.S	@Rm+,FRn	LS	1	1	6-9
	185	FMOV.S	@(R0,Rm),FRn	LS	1	1	6-9
	186	FMOV.S	FRm,@Rn	LS	1	1	6-9
	187	FMOV.S	FRm,@-Rn	LS	1	1	6-9
	188	FMOV.S	FRm,@(R0,Rn)	LS	1	1	6-9
	189	FLDS	FRm,FPUL	LS	1	1	6-10
	190	FSTS	FPUL,FRn	LS	1	1	6-11
	191	FABS	FRn	LS	1	1	6-12
	192	FADD	FRm,FRn	FE	1	1	6-14
	193	FCMP/EQ	FRm,FRn	FE	1	1	6-14
	194	FCMP/GT	FRm,FRn	FE	1	1	6-14
	195	FDIV	FRm,FRn	FE	1	14	6-15
	196	FLOAT	FPUL,FRn	FE	1	1	6-14
	197	FMAC	FR0,FRm,FRn	FE	1	1	6-14
	198	FMUL	FRm,FRn	FE	1	1	6-14
	199	FNEG	FRn	LS	1	1	6-12
	200	FSQRT	FRn	FE	1	30	6-15
	201	FSUB	FRm,FRn	FE	1	1	6-14
	202	FTRC	FRm,FPUL	FE	1	1	6-14
	203	FMOV	DRm,DRn	LS	1	1	6-9
	204	FMOV	@Rm,DRn	LS	1	1	6-9
	205	FMOV	@Rm+,DRn	LS	1	1	6-9
	206	FMOV	@(R0,Rm),DRn	LS	1	1	6-9
	207	FMOV	DRm,@Rn	LS	1	1	6-9
	208	FMOV	DRm,@-Rn	LS	1	1	6-9
	209	FMOV	DRm,@(R0,Rn)	LS	1	1	6-9

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Double-precision floating-point instructions	210	FABS	DRn	LS	1	1	6-12
	211	FADD	DRm,DRn	FE	1	1	6-16
	212	FCMP/EQ	DRm,DRn	FE	1	1	6-16
	213	FCMP/GT	DRm,DRn	FE	1	1	6-16
	214	FCNVDS	DRm,FPUL	FE	1	1	6-16
	215	FCNVSD	FPUL,DRn	FE	1	1	6-16
	216	FDIV	DRm,DRn	FE	1	14	6-18
	217	FLOAT	FPUL,DRn	FE	1	1	6-16
	218	FMUL	DRm,DRn	FE	1	3	6-17
	219	FNEG	DRn	LS	1	1	6-12
	220	FSQRT	DRn	FE	1	30	6-18
	221	FSUB	DRm,DRn	FE	1	1	6-16
	222	FTRC	DRm,FPUL	FE	1	1	6-16
FPU system control instructions	223	LDS	Rm,FPUL	LS	1	1	6-1
	224	LDS	Rm,FPSCR	LS	1	1	6-5
	225	LDS.L	@Rm+,FPUL	LS	1	1	6-3
	226	LDS.L	@Rm+,FPSCR	LS	1	1	6-7
	227	STS	FPUL,Rn	LS	1	1	6-2
	228	STS	FPSCR,Rn	LS	1	1	6-6
	229	STS.L	FPUL,@-Rn	LS	1	1	6-4
	230	STS.L	FPSCR,@-Rn	LS	1	1	6-8
Graphics acceleration instructions	231	FMOV	DRm,XDn	LS	1	1	6-9
	232	FMOV	XDm,DRn	LS	1	1	6-9
	233	FMOV	XDm,XDn	LS	1	1	6-9
	234	FMOV	@Rm,XDn	LS	1	1	6-9
	235	FMOV	@Rm+,XDn	LS	1	1	6-9
	236	FMOV	@(R0,Rm),XDn	LS	1	1	6-9
	237	FMOV	XDm,@Rn	LS	1	1	6-9
	238	FMOV	XDm,@-Rn	LS	1	1	6-9
	239	FMOV	XDm,@(R0,Rn)	LS	1	1	6-9
	240	FIPR	FVm,FVn	FE	1	1	6-19

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Execution Cycles	Execution Pattern
Graphics acceleration instructions	241	FRCHG		FE	1	1	6-14
	242	FSCHG		FE	1	1	6-14
	243	FPCHG		FE	1	1	6-14
	244	FSRRA	FRn	FE	1	1	6-21
	245	FSCA	FPUL,DRn	FE	1	3	6-22
	246	FTRV	XMTRX,FVn	FE	1	4	6-20

Section 5 Exception Handling

5.1 Summary of Exception Handling

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

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

5.2 Register Descriptions

Table 5.1 lists the configuration of registers related exception handling.

Table 5.1 Register Configuration

Register Name	Abbr.	R/W	P4 Address*	Area 7 Address*	Access Size
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32

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

Table 5.2 States of Register in Each Operating Mode

Register Name	Abbr.	Power-on Reset	Manual Reset	Sleep	Standby
TRAPA exception register	TRA	Undefined	Undefined	Retained	Retained
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Retained	Retained
Interrupt event register	INTEVT	Undefined	Undefined	Retained	Retained

5.2.1 TRAPA Exception Register (TRA)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
9 to 2	TRACODE	Undefined	R/W	TRAPA Code 8-bit immediate data of TRAPA instruction is set
1, 0	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.

5.2.2 Exception Event Register (EXPEVT)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
11 to 0	EXPCODE	H'000 or H'020	R/W	Exception Code The exception code for a reset or general exception is set. For details, see table 5.3.

5.2.3 Interrupt Event Register (INTEVT)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For details on reading/writing this bit, see General Precautions on Handling of Product.
13 to 0	INTCODE	Undefined	R/W	Exception Code The exception code for an interrupt is set. For details, see table 5.3.

5.3 Exception Handling Functions

5.3.1 Exception Handling Flow

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

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

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

5.3.2 Exception Handling Vector Addresses

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

5.4 Exception Types and Priorities

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

Table 5.3 Exceptions

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

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

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

5.5 Exception Flow

5.5.1 Exception Flow

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

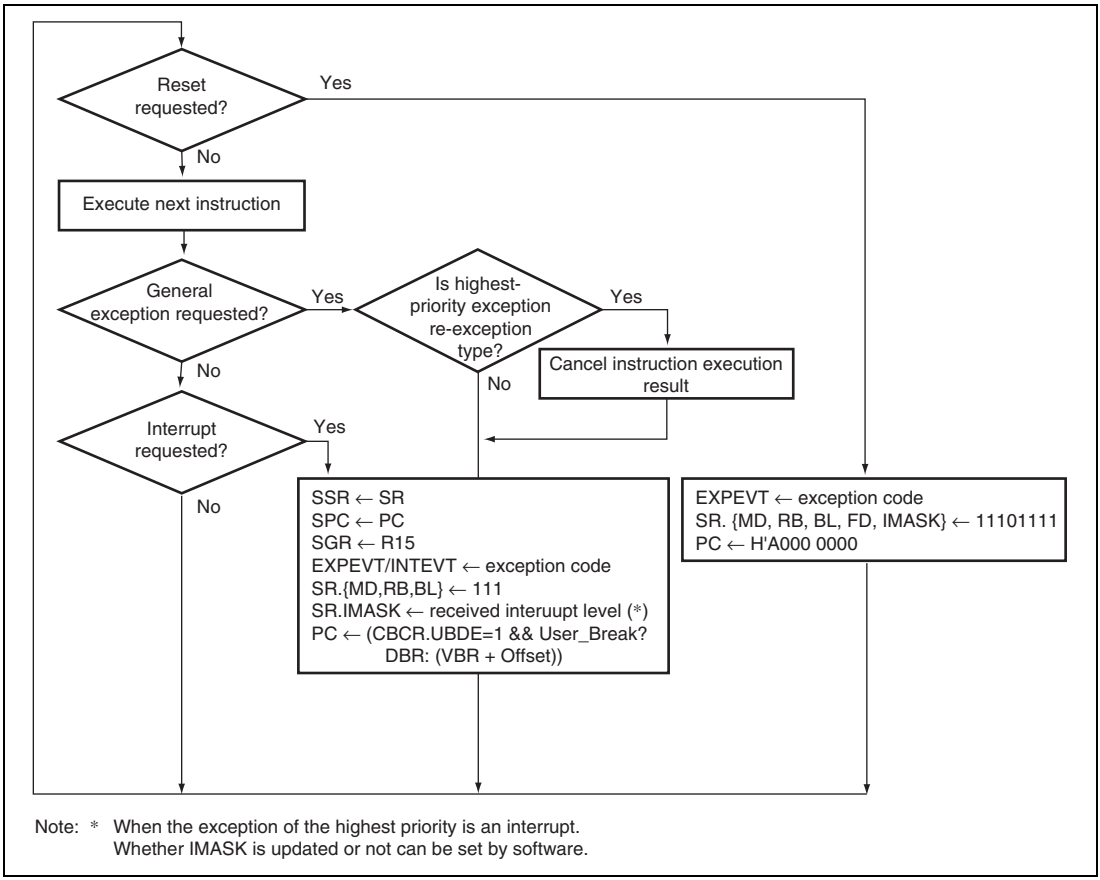


Figure 5.1 Instruction Execution and Exception Handling

5.5.2 Exception Source Acceptance

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

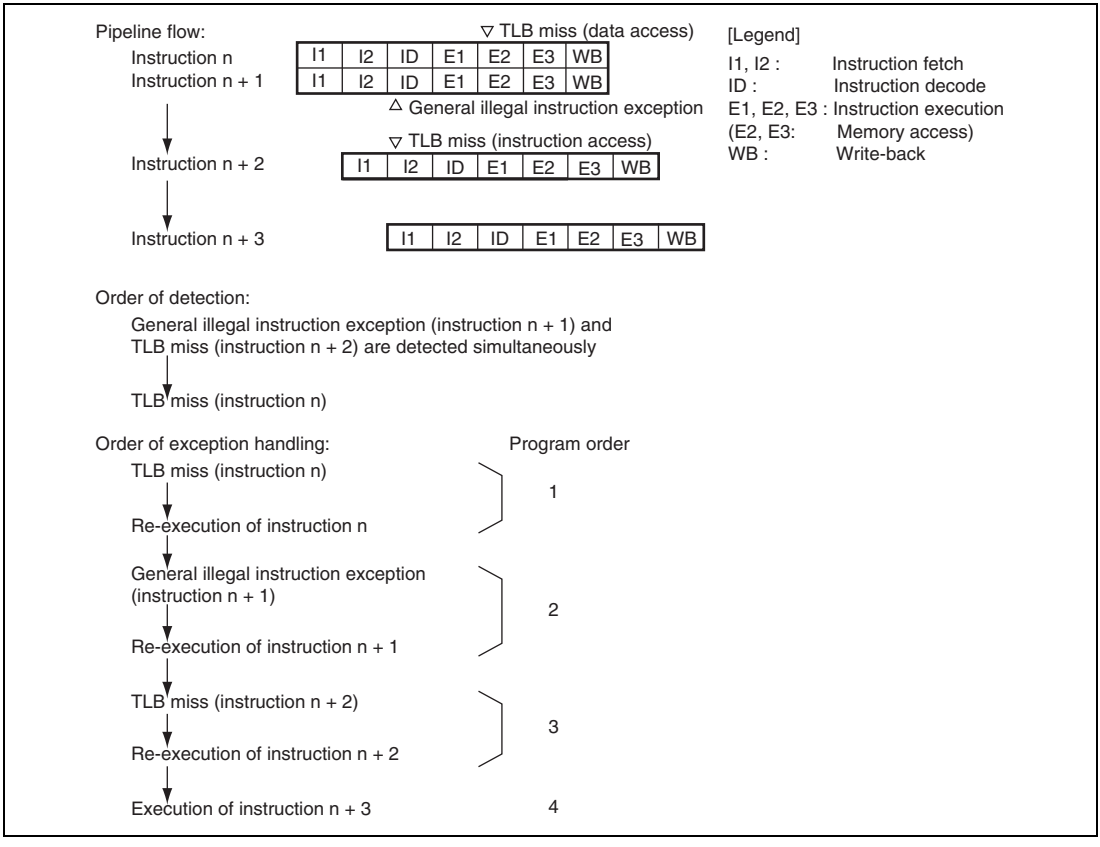


Figure 5.2 Example of General Exception Acceptance Order

5.5.3 Exception Requests and BL Bit

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

When the BL bit in SR is 1 and an exception other than a user break is generated, the CPU's internal registers and the registers of the other modules are set to their states following a manual reset, and the CPU branches to the same address as in a reset (H'A0000000). For the operation in the event of a user break, see section 41, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

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

5.5.4 Return from Exception Handling

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

5.6 Description of Exceptions

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

5.6.1 Resets

(1) Power-On Reset

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

(2) Manual Reset

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

(3) H-UDI Reset

- Source: SDIR.TI[7:4] = B'0110 (negation) or B'0111 (assertion)
- Transition address: H'A0000000
- Transition operations:
  Exception code H'000 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A0000000.
  CPU and on-chip peripheral module initialization is performed. For details, see the register descriptions in the relevant sections.

(4) Instruction TLB Multiple-Hit Exception

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

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

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

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

(5) Data TLB Multiple-Hit Exception

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

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

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

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

5.6.2 General Exceptions

(1) Data TLB Miss Exception

- Source: Address mismatch in UTLB address comparison
- Transition address: VBR + H'00000400
- Transition operations:

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

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

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

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

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

(2) Instruction TLB Miss Exception

- Source: Address mismatch in ITLB address comparison
- Transition address: VBR + H'00000400
- Transition operations:

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

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

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

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

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

(3) Initial Page Write Exception

- Source: TLB is hit in a store access, but dirty bit D = 0
- Transition address: VBR + H'00000100
- Transition operations:

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

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

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

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

(4) Data TLB Protection Violation Exception

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

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

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

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

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

  Exception code H'0A0 (for a read access) or H'0C0 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

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

(5) Instruction TLB Protection Violation Exception

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

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

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

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

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

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

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

(6) Data Address Error

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

    Areas H'E0000000 to H'E3FFFFFF and H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 6, Memory Management Unit (MMU) and section 8, L Memory.
- Transition address: VBR + H'0000100
- Transition operations:

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

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

  Exception code H'0E0 (for a read access) or H'100 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 6, Memory Management Unit (MMU).

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

(7) Instruction Address Error

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

    Area H'E5000000 to H'E5FFFFFF can be accessed in user mode. For details, see section 8, L Memory.
- Transition address: VBR + H'00000100
- Transition operations:

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

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

  Exception code H'0E0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 6, Memory Management Unit (MMU).

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

(8) Unconditional Trap

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

  As this is a processing-completion-type exception, the PC contents for the instruction following the TRAPA instruction are saved in SPC. The value of SR and R15 when the TRAPA instruction is executed are saved in SSR and SGR. The 8-bit immediate value in the TRAPA instruction is multiplied by 4, and the result is set in TRA [9:0]. Exception code H'160 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

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

(9) General Illegal Instruction Exception

- Sources:
  - — Decoding of an undefined instruction not in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - — Decoding in user mode of a privileged instruction not in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
- Transition address: VBR + H'00000100
- Transition operations:
  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.
  Exception code H'180 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

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

(10) Slot Illegal Instruction Exception

- Sources:
  - — Decoding of an undefined instruction in a delay slot
    Delayed branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instruction: H'FFFD
  - — Decoding of an instruction that modifies PC in a delay slot
    Instructions that modify PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm,SR, LDC.L @Rm+,SR, ICBI, PREFI
  - — Decoding in user mode of a privileged instruction in a delay slot
    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
  - — Decoding of a PC-relative MOV instruction or MOVA instruction in a delay slot
- Transition address: VBR + H'000 0100
- Transition operations:
  The PC contents for the preceding delayed branch instruction are saved in SPC. The SR and R15 contents when this exception occurred are saved in SSR and SGR.
  Exception code H'1A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

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

(11) General FPU Disable Exception

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

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

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

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

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

(12) Slot FPU Disable Exception

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

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

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

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

(13) Pre-Execution User Break/Post-Execution User Break

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

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

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

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

  For details of PC, etc., when a data break is set, see section 41, User Break Controller (UBC).

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

(14) FPU Exception

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

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

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

5.6.3 Interrupts

(1) NMI (Nonmaskable Interrupt)

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

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

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

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

(2) General Interrupt Request

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

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

  The code corresponding to the each interrupt source is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'0600.

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

5.6.4 Priority Order with Multiple Exceptions

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

- Instructions that make two accesses to memory

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

1. Data address error in first data transfer
2. TLB miss in first data transfer
3. TLB protection violation in first data transfer
4. Initial page write exception in first data transfer
5. Data address error in second data transfer
6. TLB miss in second data transfer
7. TLB protection violation in second data transfer
8. Initial page write exception in second data transfer

- Indivisible delayed branch instruction and delay slot instruction

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

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

  If the delay slot instruction has a second data transfer, two checks are performed in step 2, as in the above case (Instructions that make two accesses to memory).

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

5.7 Usage Notes

(1) Return from exception handling

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

(2) If an exception or interrupt occurs when BL bit in SR = 1

1. Exception

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

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

   In sleep or standby mode, however, an interrupt is accepted even if the BL bit in SR is set to 1.

(3) SPC when an exception occurs

1. Re-execution type exception

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

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

(4) RTE instruction delay slot

1. The instruction in the delay slot of the RTE instruction is executed only after the value saved in SSR has been restored to SR. The acceptance of the exception related to the instruction access is determined depending on SR before restoring, while the acceptance of other exceptions is determined depending on the processing mode by SR after restoring or the BL

bit. The completion type exception is accepted before branching to the destination of RTE instruction. However, if the re-execution type exception is occurred, the operation cannot be guaranteed.

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

(5) Changing the SR register value and accepting exception

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

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

Section 6 Memory Management Unit (MMU)

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

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

6.1 Overview of MMU

The MMU was conceived as a means of making efficient use of physical memory. As shown in (0) in figure 6.1, when a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory, but if the process increases in size to the point where it does not fit into physical memory, it becomes necessary to divide the process into smaller parts, and map the parts requiring execution onto physical memory as occasion arises ((1) in figure 6.1). Having this mapping onto physical memory executed consciously by the process itself imposes a heavy burden on the process. The virtual memory system was devised as a means of handling all physical memory mapping to reduce this burden ((2) in figure 6.1). With a virtual memory system, the size of the available virtual memory is much larger than the actual physical memory, and processes are mapped onto this virtual memory. Thus processes only have to consider their operation in virtual memory, and mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally managed by the OS, and physical memory switching is carried out so as to enable the virtual memory required by a process to be mapped smoothly onto physical memory. Physical memory switching is performed via secondary storage, etc.

The virtual memory system that came into being in this way works to best effect in a time sharing system (TSS) that allows a number of processes to run simultaneously ((3) in figure 6.1). Running a number of processes in a TSS did not increase efficiency since each process had to take account of physical memory mapping. Efficiency is improved and the load on each process reduced by the use of a virtual memory system ((4) in figure 6.1). In this virtual memory system, virtual memory is allocated to each process. The task of the MMU is to map a number of virtual memory areas onto physical memory in an efficient manner. It is also provided with memory protection functions to prevent a process from inadvertently accessing another process's physical memory.

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

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

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

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

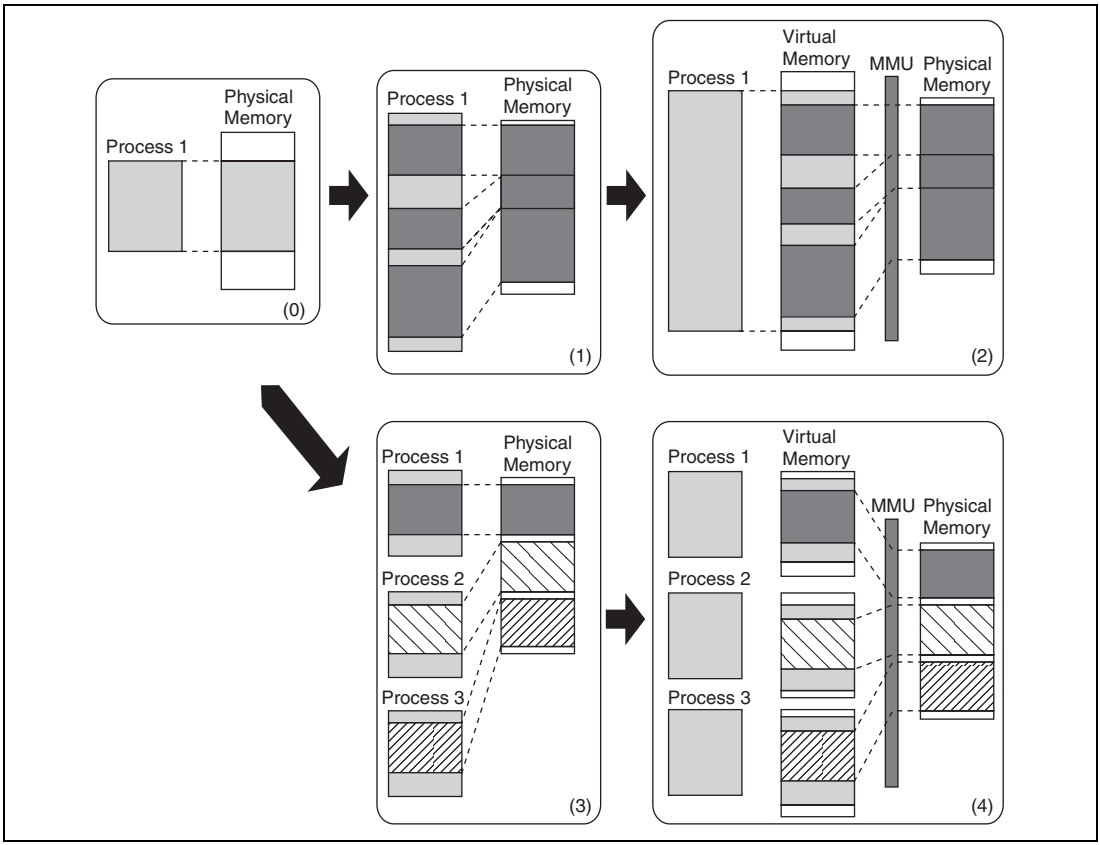


Figure 6.1 Role of MMU

6.1.1 Address Spaces

Virtual Address Space: This LSI supports a 32-bit virtual address space, and can access a 4-Gbyte address space. The virtual address space is divided into a number of areas, as shown in figures 6.2 and 6.3. In privileged mode, the 4-Gbyte space from the P0 area to the P4 area can be accessed. In user mode, a 2-Gbyte space in the U0 area can be accessed. When the SQMD bit in the MMU control register (MMUCR) is 0, a 64-Mbyte space in the store queue area can be accessed. When the RMD bit in the on-chip memory control register (RAMCR) is 1, a 16-Mbyte space in on-chip memory area can be accessed. Accessing areas other than the U0 area, store queue area, and on-chip memory area in user mode will cause an address error.

When the AT bit in MMUCR is set to 1 and the MMU is enabled, the P0, P3, and U0 areas can be mapped onto any physical address space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units. By using an 8-bit address space identifier, the P0, P3, and U0 areas can be increased to a maximum of 256. Mapping from the virtual address space to the 29-bit physical address space is carried out using the TLB.

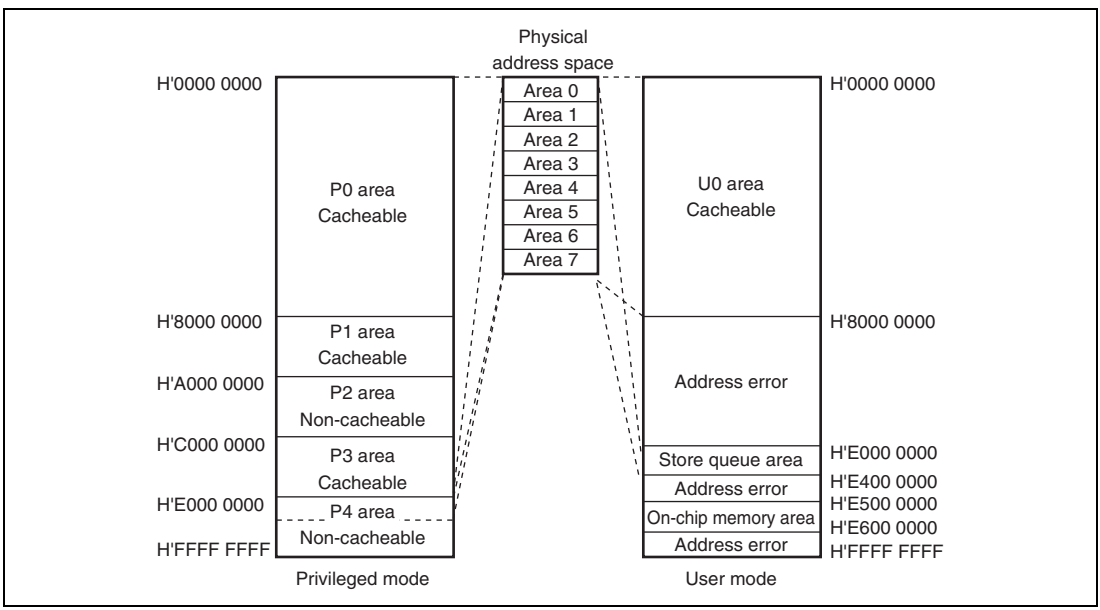


Figure 6.2 Virtual Address Space (AT in MMUCR = 0)

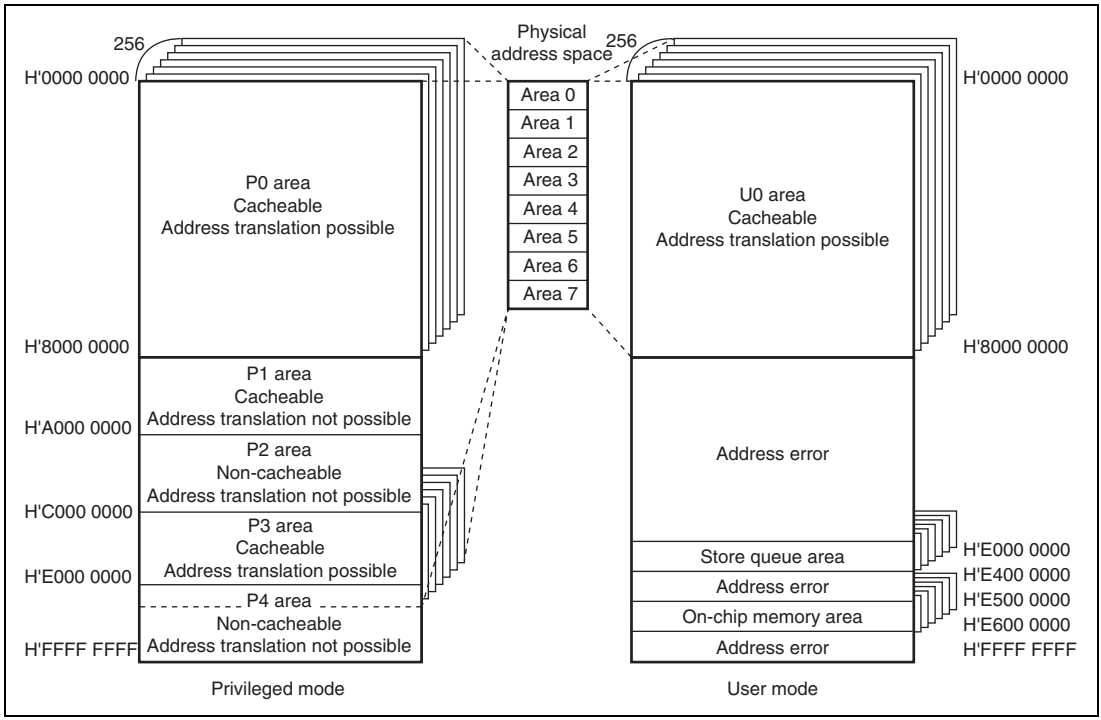


Figure 6.3 Virtual Address Space (AT in MMUCR = 1)

- P0, P3, and U0 Areas:

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

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

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

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

- P1 Area:

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

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

- P2 Area:

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

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

- P4 Area:

  The P4 area is mapped onto the internal resource of this LSI. This area except the store queue and on-chip memory areas does not allow address translation using the TLB. This area cannot be accessed using the cache. The P4 area is shown in detail in figure 6.4.

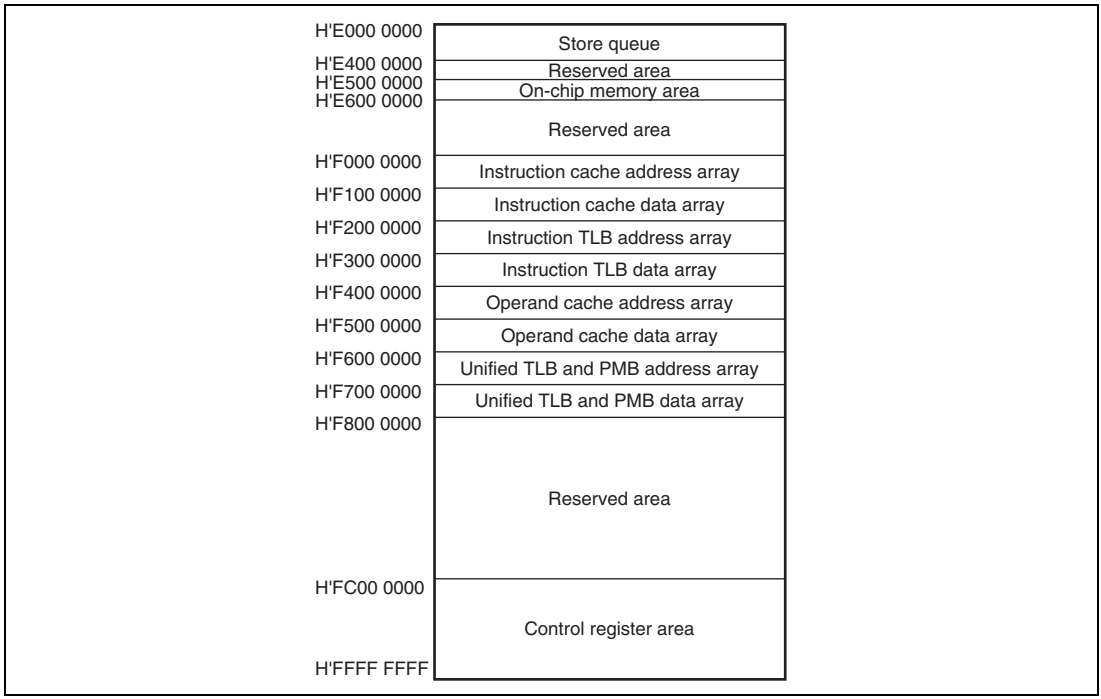


Figure 6.4 P4 Area

The area from H'E000 0000 to H'E3FF FFFF comprises addresses for accessing the store queues (SQs). In user mode, the access right is specified by the SQMD bit in MMUCR. For details, see section 7.7, Store Queues.

The area from H'E500 0000 to H'E5FF FFFF comprises addresses for accessing the on-chip memory. In user mode, the access right is specified by the RMD bit in RAMCR. For details, see section 8, L Memory.

The area from H'F000 0000 to H'F0FF FFFF is used for direct access to the instruction cache address array. For details, see section 7.6.1, IC Address Array.

The area from H'F100 0000 to H'F1FF FFFF is used for direct access to the instruction cache data array. For details, see section 7.6.2, IC Data Array.

The area from H'F200 0000 to H'F2FF FFFF is used for direct access to the instruction TLB address array. For details, see section 6.6.1, ITLB Address Array.

The area from H'F300 0000 to H'F37F FFFF is used for direct access to instruction TLB data array. For details, see section 6.6.2, ITLB Data Array.

The area from H'F400 0000 to H'F4FF FFFF is used for direct access to the operand cache address array. For details, see section 7.6.3, OC Address Array.

The area from H'F500 0000 to H'F5FF FFFF is used for direct access to the operand cache data array. For details, see section 7.6.4, OC Data Array.

The area from H'F600 0000 to H'F60F FFFF is used for direct access to the unified TLB address array. For details, see section 6.6.3, UTLB Address Array.

The area from H'F700 0000 to H'F70F FFFF is used for direct access to unified TLB data array. For details, see section 6.6.4, UTLB Data Array.

The area from H'F610 0000 to H'F61F FFFF is used for direct access to the PMB address array. For details, see section 6.7.5, Memory-Mapped PMB Configuration.

The area from H'F710 0000 to H'F71F FFFF is used for direct access to the PMB data array. For details, see section 6.7.5, Memory-Mapped PMB Configuration.

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

Physical Address Space: This LSI supports a 29-bit physical address space. The physical address space is divided into eight areas as shown in figure 6.5. Area 7 is a reserved area.

Only when area 7 in the physical address space is accessed using the TLB, addresses H'1C00 0000 to H'1FFF FFFF of area 7 are not designated as a reserved area, but are equivalent to the control register area in the P4 area in the virtual address space.

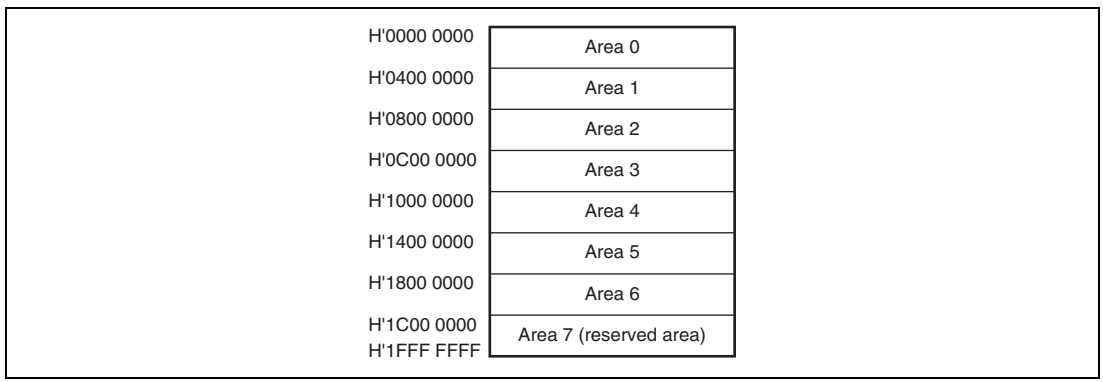


Figure 6.5 Physical Address Space

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

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two virtual memory systems, single virtual memory and multiple virtual memory, either of which can be selected with the SV bit in MMUCR. In the single virtual memory system, a number of processes run simultaneously, using virtual address space on an exclusive basis, and the physical address corresponding to a particular virtual address is uniquely determined. In the multiple virtual memory system, a number of processes run while sharing the virtual address space, and particular virtual addresses may be translated into different physical addresses depending on the process. The only difference between the single virtual memory and multiple virtual memory systems in terms of operation is in the TLB address comparison method (see section 6.3.3, Address Translation Method).

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

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

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

6.2 Register Descriptions

The following registers are related to MMU processing.

Table 6.1 Register Configuration

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

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

Table 6.2 Register States in Each Processing State

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

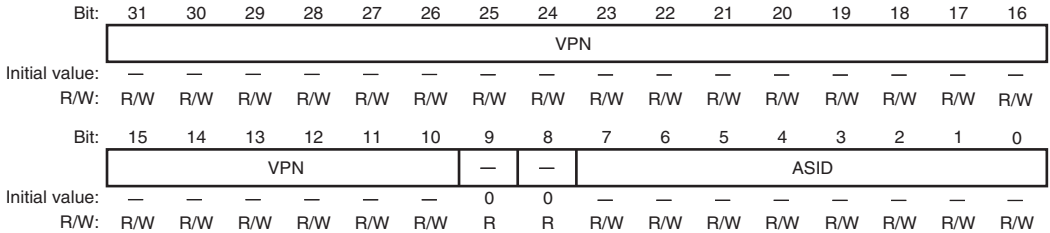
6.2.1 Page Table Entry High Register (PTEH)

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

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

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating the ASID field, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after the ASID field has been updated.

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



Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	—	R/W	Virtual Page Number
9, 8	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
7 to 0	ASID	—	R/W	Address Space Identifier

6.2.2 Page Table Entry Low Register (PTEL)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
28 to 10	PPN	—	R/W	Physical Page Number
9	—	0	R	Reserved For details on reading from or writing to this bit, see description in General Precautions on Handling of Product.
8	V	—	R/W	Page Management Information The meaning of each bit is same as that of corresponding bit in Common TLB (UTLB). For details, see section 6.3, TLB Functions.
7	SZ1	—	R/W	
6	PR1	—	R/W	
5	PR0	—	R/W	
4	SZ0	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
0	WT	—	R/W	

6.2.3 Translation Table Base Register (TTB)

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

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

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

6.2.4 TLB Exception Address Register (TEA)

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEA				Virtual address at which MMU exception or address error occurred											
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEA				Virtual address at which MMU exception or address error occurred											
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6.2.5 MMU Control Register (MMUCR)

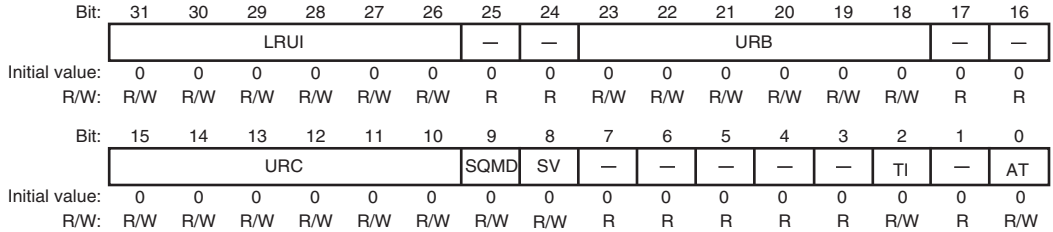
The individual bits perform MMU settings as shown below. Therefore, MMUCR rewriting should be performed by a program in the P1 or P2 area.

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

1. Execute a branch using the RTE instruction. In this case, the branch destination may be the P0, P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating MMUCR, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after MMUCR has been updated.

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

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



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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	URC						SQMD	SV	—	—	—	—	—	TI	—	AT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	LRUI	All 0	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. x means that updating is not performed. 000xxx: ITLB entry 0 is used 1xx00x: ITLB entry 1 is used x1x1x0: ITLB entry 2 is used xx1x11: ITLB entry 3 is used xxxxxx: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update. x means "don't care". 111xxx: ITLB entry 0 is updated 0xx11x: ITLB entry 1 is updated x0x0x1: ITLB entry 2 is updated xx0x00: ITLB entry 3 is updated Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
25, 24	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
23 to 18	URB	All 0	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary at which replacement is to be performed. Valid only when URB ≠ 0.
17, 16	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
15 to 10	URC	All 0	R/W	UTLB Replace Counter These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction. This bit is incremented each time the UTLB is accessed. If URB > 0, URC is cleared to 0 when the condition URC = URB is satisfied. Also note that if a value is written to URC by software which results in the condition of URC > URB, incrementing is first performed in excess of URB until URC = H'3F. URC is not incremented by an LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Bit Specifies the right of access to the store queues. 0: User/privileged access possible 1: Privileged access possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching Bit When this bit is changed, ensure that 1 is also written to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode
7 to 3	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.
1	—	0	R	Reserved For details on reading from or writing to this bit, see description in General Precautions on Handling of Product.
0	AT	0	R/W	Address Translation Enable Bit These bits enable or disable the MMU. 0: MMU disabled 1: MMU enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

6.2.6 Physical Address Space Control Register (PASCR)

PASCR controls the operation in the physical address space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mbytes) When writing is performed without using the cache or in the cache write-through mode, these bits specify whether the next bus access from the CPU waits for the end of writing for each area. 0: The CPU does not wait for the end of writing bus access and starts the next bus access 1: The CPU waits for the end of writing bus access and starts the next bus access UB[7]: Corresponding to the control register area UB[6]: Corresponding to area 6 UB[5]: Corresponding to area 5 UB[4]: Corresponding to area 4 UB[3]: Corresponding to area 3 UB[2]: Corresponding to area 2 UB[1]: Corresponding to area 1 UB[0]: Corresponding to area 0

6.2.7 Instruction Re-Fetch Inhibit Control Register (IRMCR)

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

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

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

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT	MT	MC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4	R2	0	R/W	Re-Fetch Inhibit 2 after Register Change When MMUCR, PASCR, CCR, PTEH, or RAMCR is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed
3	R1	0	R/W	Re-Fetch Inhibit 1 after Register Change When a register allocated in addresses H'FF200000 to H'FF2FFFFF is changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed
2	LT	0	R/W	Re-Fetch Inhibit after LDTLB Execution This bit controls whether re-fetch is performed for the next instruction after the LDTLB instruction has been executed. 0: Re-fetch is performed 1: Re-fetch is not performed
1	MT	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped TLB This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped ITLB/UTLB while the AT bit in MMUCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

Bit	Bit Name	Initial Value	R/W	Description
0	MC	0	R/W	Re-Fetch Inhibit after Writing Memory-Mapped IC This bit controls whether re-fetch is performed for the next instruction after writing memory-mapped IC while the ICE bit in CCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

6.3 TLB Functions

6.3.1 Unified TLB (UTLB) Configuration

The UTLB is used for the following two purposes:

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

The UTLB is so called because of its use for the above two purposes. Information in the address translation table located in external memory is cached into the UTLB. The address translation table contains virtual page numbers and address space identifiers, and corresponding physical page numbers and page management information. Figure 6.6 shows the UTLB configuration. The UTLB consists of 64 fully-associative type entries. Figure 6.7 shows the relationship between the page size and address format.

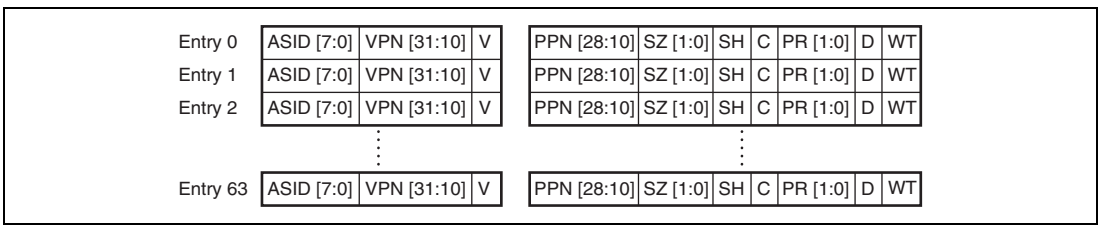


Figure 6.6 UTLB Configuration

[Legend]

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

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

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

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

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

- PPN: Physical page number
  Upper 22 bits of the physical address of the physical page number.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  The synonym problem must be taken into account when setting the PPN (see section 6.4.5, Avoiding Synonym Problems).

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

- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable
  1: Cacheable
  When the control register area is mapped, this bit must be cleared to 0.

- D: Dirty bit
  Indicates whether a write has been performed to a page.
  0: Write has not been performed
  1: Write has been performed

- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  1: Write-through mode

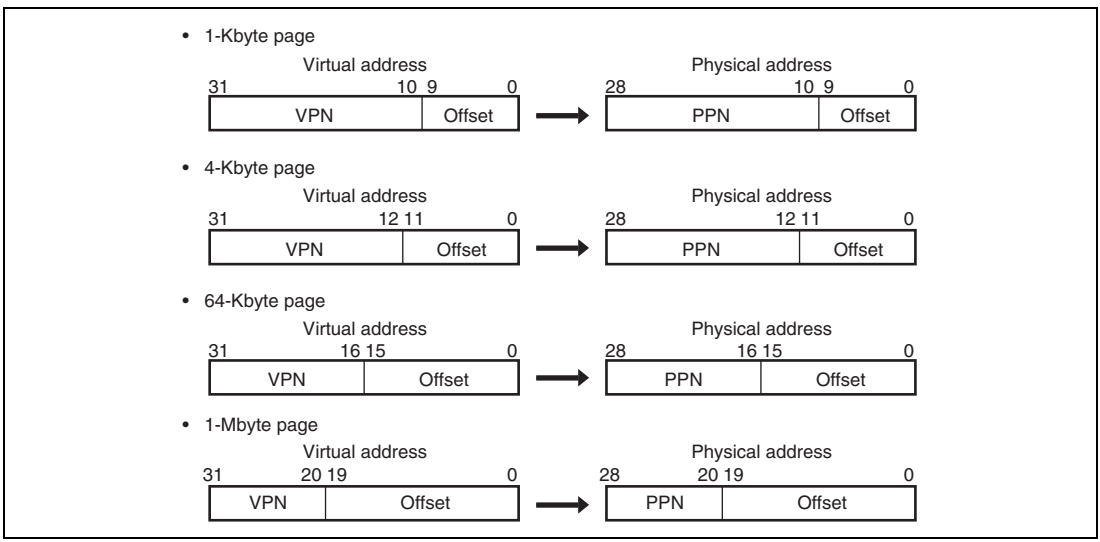


Figure 6.7 Relationship between Page Size and Address Format

6.3.2 Instruction TLB (ITLB) Configuration

The ITLB is used to translate a virtual address to a physical address in an instruction access. Information in the address translation table located in the UTLB is cached into the ITLB. Figure 6.8 shows the ITLB configuration. The ITLB consists of four fully-associative type entries.

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

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

Figure 6.8 ITLB Configuration

6.3.3 Address Translation Method

Figure 6.9 shows a flowchart of a memory access using the UTLB.

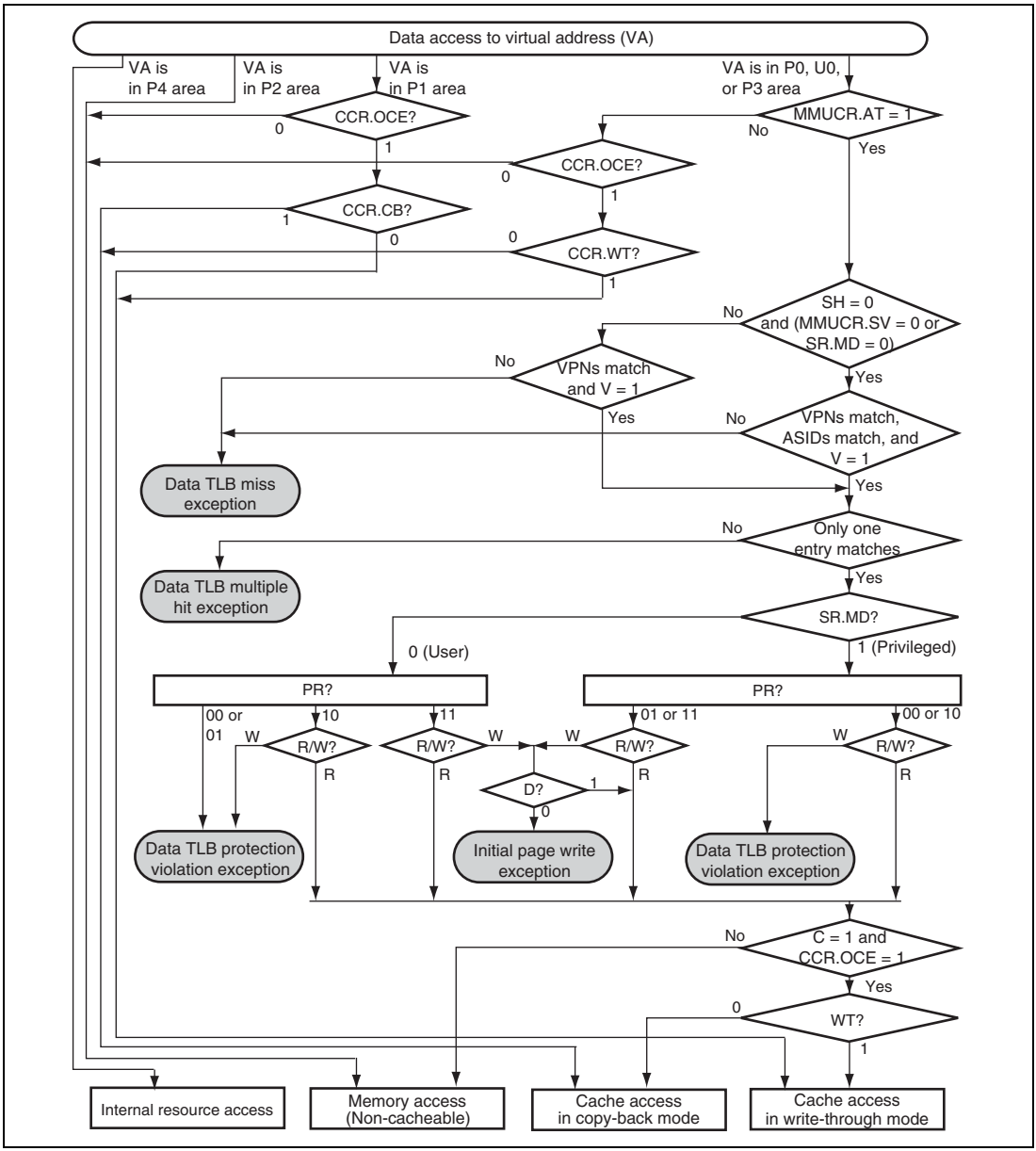


Figure 6.9 Flowchart of Memory Access Using UTLB

Figure 6.10 shows a flowchart of a memory access using the ITLB.

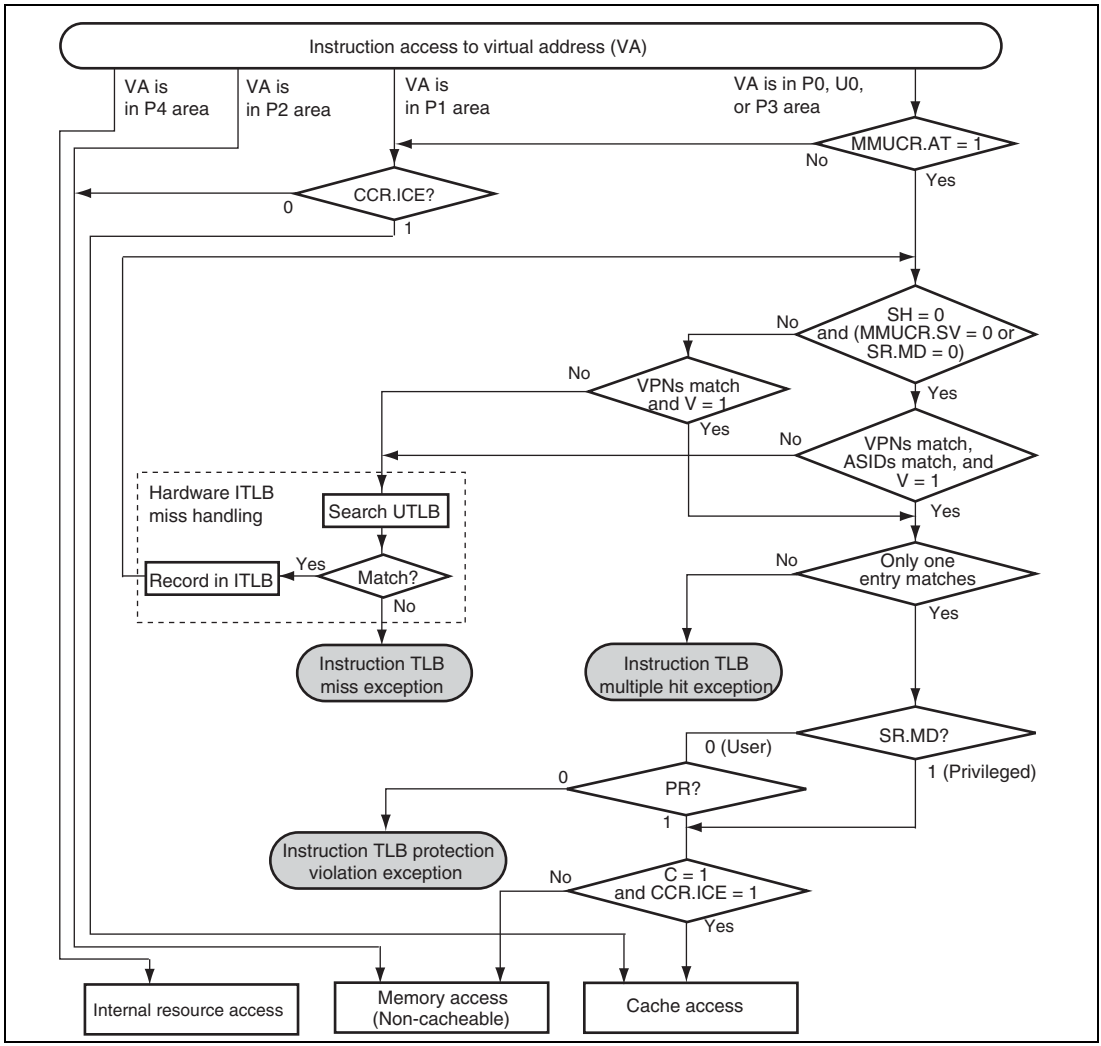


Figure 6.10 Flowchart of Memory Access Using ITLB

6.4 MMU Functions

6.4.1 MMU Hardware Management

This LSI supports the following MMU functions.

1. The MMU decodes the virtual address to be accessed by software, and performs address translation by controlling the UTLB/ITLB in accordance with the MMUCR settings.
2. The MMU determines the cache access status on the basis of the page management information read during address translation (C and WT bits).
3. If address translation cannot be performed normally in a data access or instruction access, the MMU notifies software by means of an MMU exception.
4. If address translation information is not recorded in the ITLB in an instruction access, the MMU searches the UTLB. If the necessary address translation information is recorded in the UTLB, the MMU copies this information into the ITLB in accordance with the LRUI bit setting in MMUCR.

6.4.2 MMU Software Management

Software processing for the MMU consists of the following:

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

6.4.3 MMU Instruction (LDTLB)

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

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

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

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

The operation of the LDTLB instruction is shown in figure 6.11.

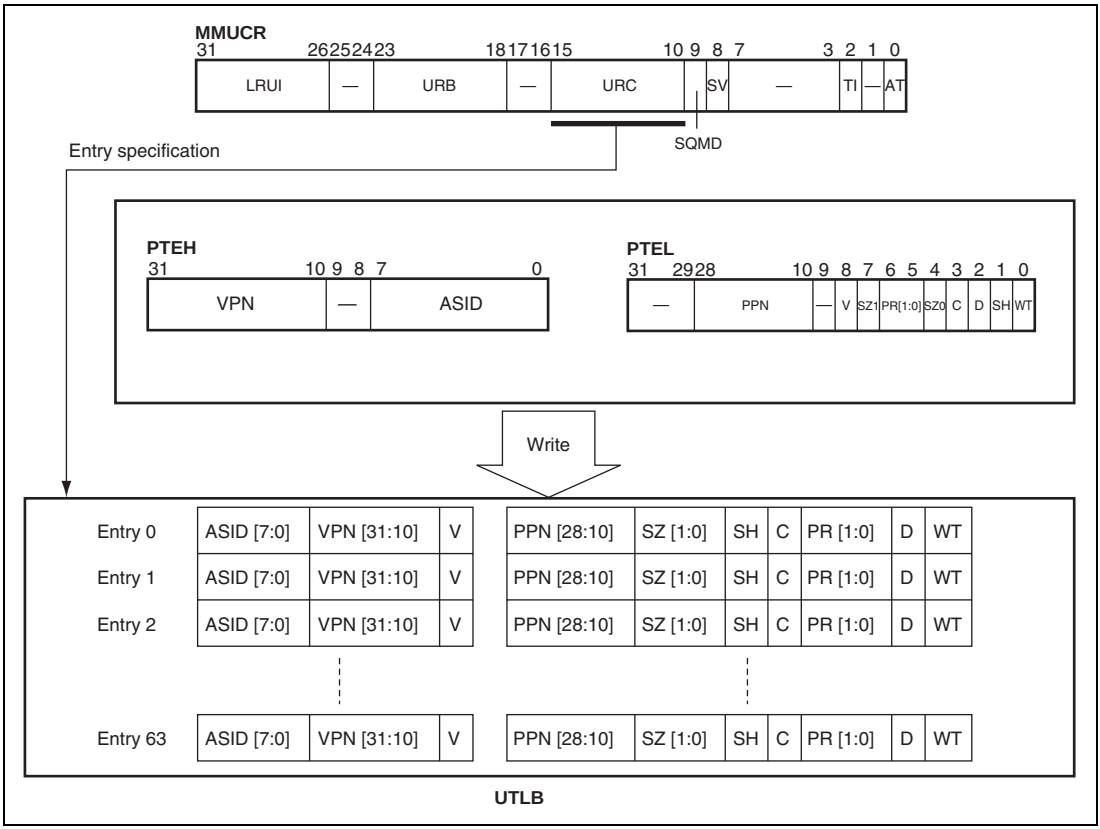


Figure 6.11 Operation of LDTLB Instruction

6.4.4 Hardware ITLB Miss Handling

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

6.4.5 Avoiding Synonym Problems

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

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

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

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

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

6.5 MMU Exceptions

There are seven MMU exceptions: instruction TLB multiple hit exception, instruction TLB miss exception, instruction TLB protection violation exception, data TLB multiple hit exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception. Refer to figures 6.9 and 6.10 for the conditions under which each of these exceptions occurs.

6.5.1 Instruction TLB Multiple Hit Exception

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

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

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

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

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

6.5.2 Instruction TLB Miss Exception

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

Hardware Processing: In the event of an instruction TLB miss exception, hardware carries out the following processing:

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

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

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

6.5.3 Instruction TLB Protection Violation Exception

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

Hardware Processing: In the event of an instruction TLB protection violation exception, hardware carries out the following processing:

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

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

6.5.4 Data TLB Multiple Hit Exception

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

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

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

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

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

6.5.5 Data TLB Miss Exception

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

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

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

9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the data TLB miss exception handling routine.

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

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

6.5.6 Data TLB Protection Violation Exception

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

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

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

8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the data TLB protection violation exception handling routine.

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

6.5.7 Initial Page Write Exception

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

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

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

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

1. Retrieve the necessary page table entry from external memory.
2. Write 1 to the D bit in the external memory page table entry.

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

6.6 Memory-Mapped TLB Configuration

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

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

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

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

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

6.6.1 ITLB Address Array

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

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

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

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

1. ITLB address array read

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

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

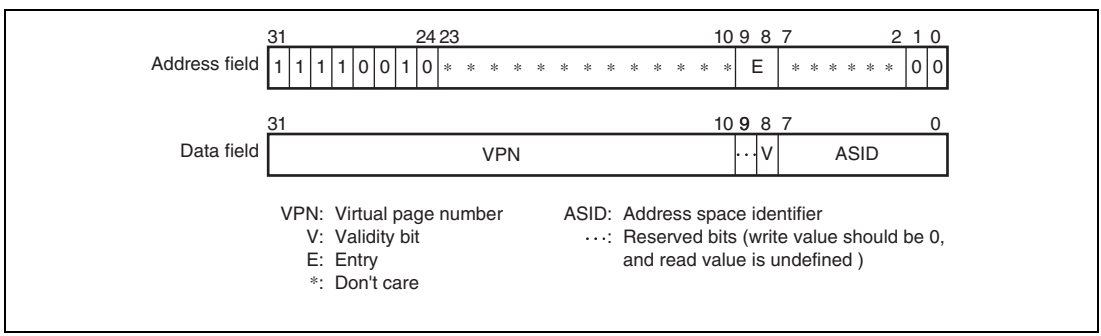


Figure 6.12 Memory-Mapped ITLB Address Array

6.6.2 ITLB Data Array

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

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

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bit [6] indicates PR, bit [3] indicates C, and bit [1] indicates SH.

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

1. ITLB data array read

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

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

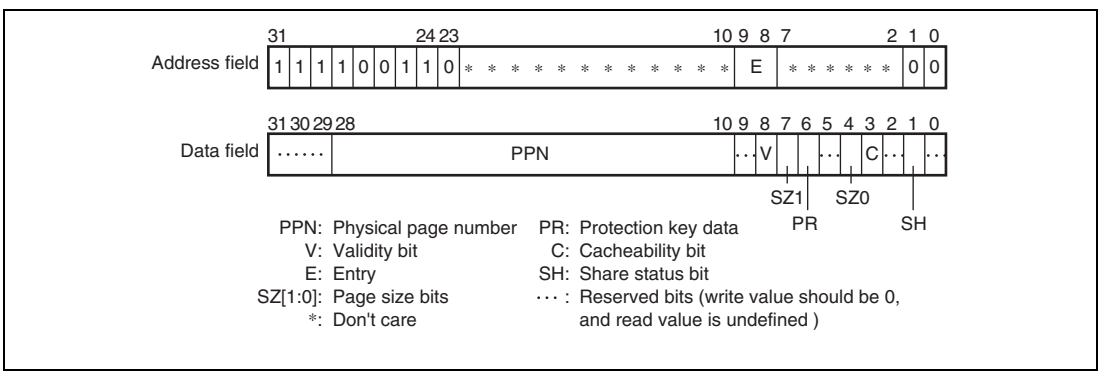


Figure 6.13 Memory-Mapped ITLB Data Array

6.6.3 UTLB Address Array

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

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

In the data field, bits [31:10] indicate VPN, bit [9] indicates D, bit [8] indicates V, and bits [7:0] indicate ASID.

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

1. UTLB address array read

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

   VPN, D, V, and ASID specified in the data field are written to the UTLB entry corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.
3. UTLB address array write (associative)

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

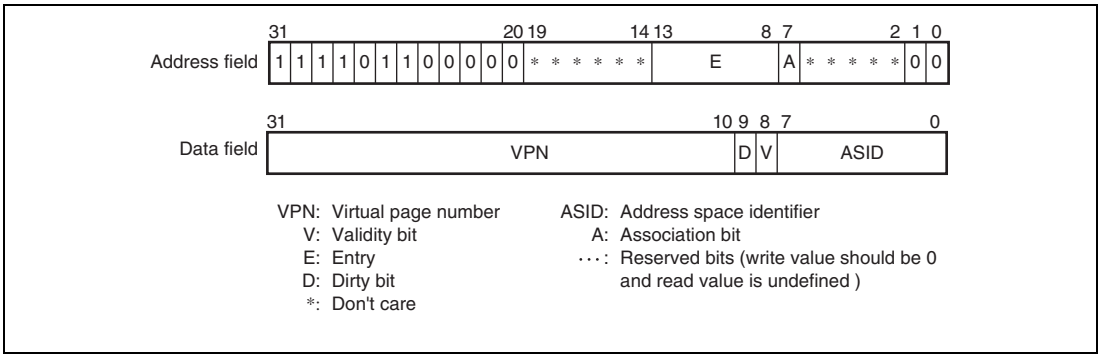


Figure 6.14 Memory-Mapped UTLB Address Array

6.6.4 UTLB Data Array

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

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

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

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

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

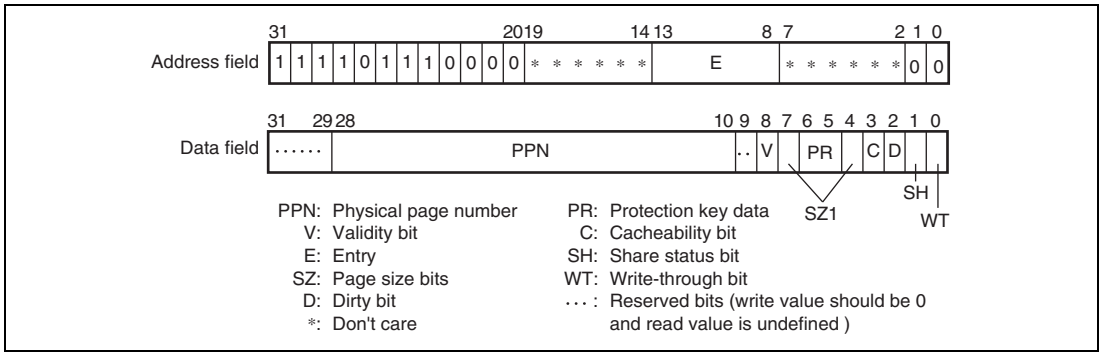


Figure 6.15 Memory-Mapped UTLB Data Array

6.7 32-Bit Address Extended Mode

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

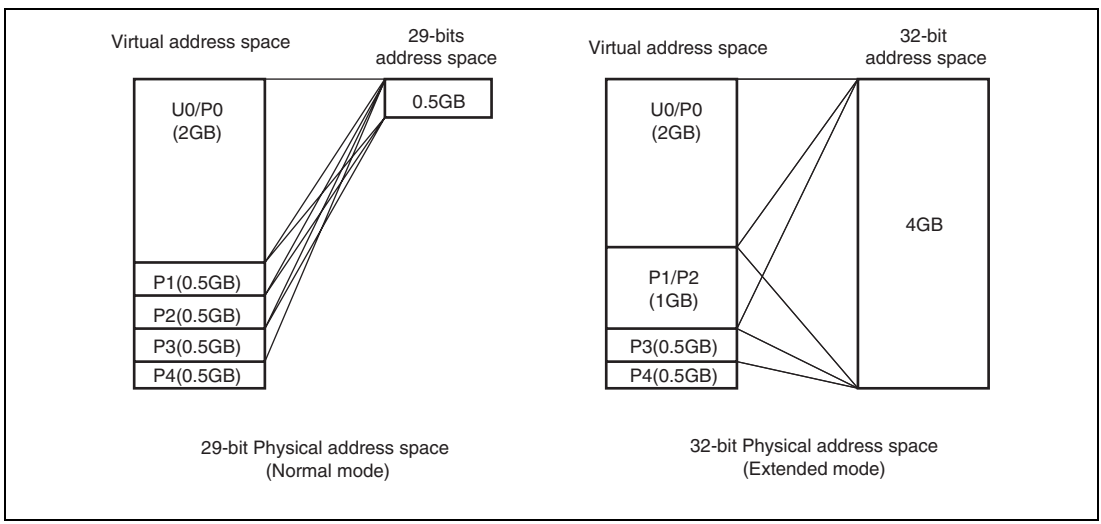


Figure 6.16 Physical Address Space (32-Bit Address Extended Mode)

6.7.1 Overview of 32-Bit Address Extended Mode

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

As for the cache operation, P1 area is cacheable and P2 area is non-cacheable in the case of 29-bit address mode, but the cache operation of both P1 and P2 area are determined by the C bit and WT bit in the PMB in the case of 32-bit address mode.

6.7.2 Transition to 32-Bit Address Extended Mode

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

1. When the AT bit in MMUCR is 0, virtual addresses in the U0, P0, or P3 area become 32-bit physical addresses. Addresses in the P1 or P2 area are translated according to the PMB mapping information.

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

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

6.7.3 Privileged Space Mapping Buffer (PMB) Configuration

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

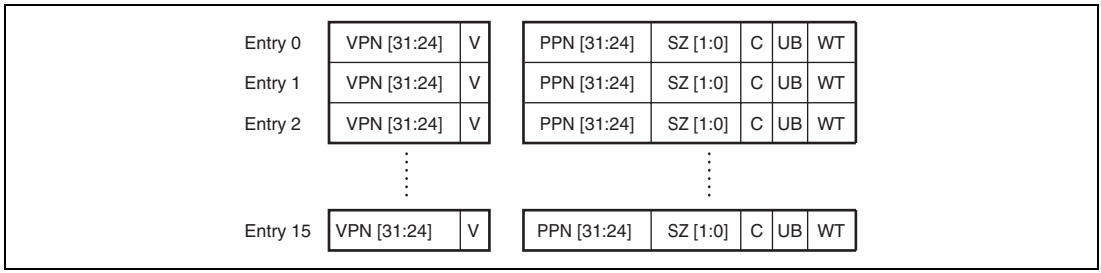


Figure 6.17 PMB Configuration

[Legend]

- VPN: Virtual page number
  For 16-Mbyte page: Upper 8 bits of virtual address
  For 64-Mbyte page: Upper 6 bits of virtual address
  For 128-Mbyte page: Upper 5 bits of virtual address
  For 512-Mbyte page: Upper 3 bits of virtual address
  Note: B'10 should be set to the upper 2 bits of VPN in order to indicate P1 or P2 area.

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

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

- PPN: Physical page number
  Upper 8 bits of the physical address of the physical page number.
  With a 16-Mbyte page, PPN[31:24] are valid.
  With a 64-Mbyte page, PPN[31:26] are valid.
  With a 128-Mbyte page, PPN[31:27] are valid.
  With a 512-Mbyte page, PPN[31:29] are valid.

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

- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  1: Write-through mode

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

6.7.4 PMB Function

This LSI supports the following PMB functions.

1. Only memory-mapped write can be used for writing to the PMB. The LDTLB instruction cannot be used to write to the PMB.
2. Software must ensure that every accessed P1 or P2 address has a corresponding PMB entry before the access occurs. When an access to an address in the P1 or P2 area which is not recorded in the PMB is made, this LSI is reset by the TLB. In this case, the accessed address in the P1 or P2 area which causes the TLB reset is stored in the TEA and code H′140 in the EXPEVT.
3. This LSI does not guarantee the operation when multiple hit occurs in the PMB. Special care should be taken when the PMB mapping information is recorded by software.
4. The PMB does not have an associative write function.
5. Since there is no PR field in the PMB, read/write protection cannot be preformed. The address translation target of the PMB is the P1 or P2 address. In user mode access, an address error exception occurs.
6. Both entries from the UTLB and PMB are mixed and recorded in the ITLB by means of the hardware ITLB miss handling. However, these entries can be identified by checking whether VPN[31:30] is 10 or not. When an entry from the PMB is recorded in the ITLB, H′00, 01, and 1 are recorded in the ASID, PR, and SH fields which do not exist in the PMB, respectively.

6.7.5 Memory-Mapped PMB Configuration

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

1. PMB address array read

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

2. PMB address array write

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

3. PMB data array read

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

4. PMB data array write

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

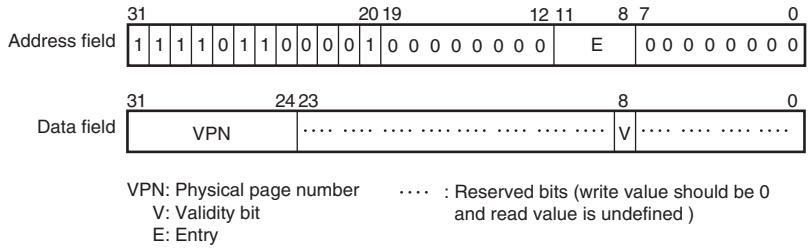


Figure 6.18 Memory-Mapped PMB Address Array

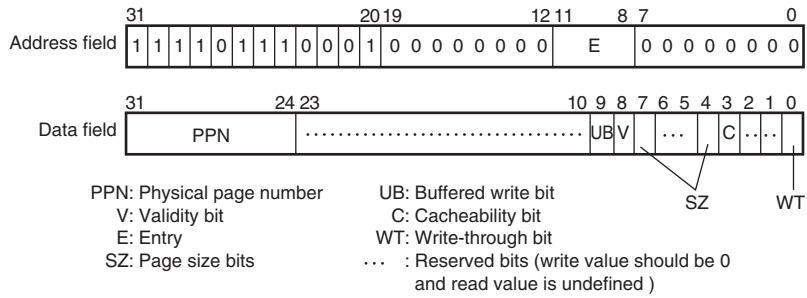


Figure 6.19 Memory-Mapped PMB Data Array

6.7.6 Notes on Using 32-Bit Address Extended Mode

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

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

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

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

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

- UB: Buffered write bit

  Specifies whether a buffered write is performed.

  0: Buffered write (Subsequent processing proceeds without waiting for the write to complete.)

  1: Unbuffered write (Subsequent processing is stalled until the write has completed.)

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

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

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

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

QACR0, QACR1: AREA0[4:2]/AREA1[4:2] fields of QACR0/QACR1 are extended to AREA0[7:2]/AREA1[7:2] corresponding to physical address [31:26]. See section 7.2.2, Queue Address Control Register 0 (QACR0) and section 7.2.3, Queue Address Control Register 1 (QACR1).

LSA0, LSA1, LDA0, LDA1: L0SADR, L1SADR, L0DADR and L1DADR fields are extended to bits 31 to 10. See section 8.2.2, L Memory Transfer Source Address Register 0 (LSA0), section 8.2.3, L Memory Transfer Source Address Register 1 (LSA1), section 8.2.4, L Memory Transfer Destination Address Register 0 (LDA0), and section 8.2.5, L Memory Transfer Destination Address Register 1 (LDA1).

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

1. For the SE bit switching, only switching from 0 to 1 is supported in Cache and MMU disabled boot routine after a power-on reset or manual reset.
2. After switching the SE bit, an area in which the program is allocated becomes the target of the PMB address translation. Therefore, the area should be recorded in the PMB before switching the SE bit. An address which may be accessed in the P1 or P2 area such as the exception handler should also be recorded in the PMB.
3. When an external memory access occurs by an operand memory access located before the MOV.L instruction which switches the SE bit, external memory space addresses accessed in both address modes should be the same.
4. Note that the V bit is mapped to both address array and data array in PMB registration. That is, first write 0 to the V bit in one of arrays and then write 1 to the V bit in another array.

6.8 Usage Notes

When using an LDTLB instruction instead of software to a value to the MMUCR. URC, execute 1 or 2 below.

1. In 29-bit address mode, follow A. and B. below. In 32-bit address mode, follow A. through B. below.
   A. Place the TLB miss exception handling routine*[1] only in the P1 or P2 area so that all the instruction accesses*[3] in the TLB miss exception handling routine should occur solely in the P1 or P2 area.
   B. Use only one page of the PMB for instruction accesses*[3] in the TLB miss exception handling routine*[1]. In 32-bit address mode, do not place them in the last 64 bytes of a page of the PMB.
   C. In 32-bit address mode, obey 1 and 2 below when recording information in the UTLB in the MMU-related exception*[2] handling routine.
      a. If a TLB miss exception occurs, do not record the page, in which the exception has occurred, in the UTLB using the following two operations.
         - Specifies the protection key data that causes a protection violation exception upon re-execution of the instruction that has caused the TLB miss exception and records the page, in which the TLB miss exception has occurred, in the UTLB.
         - Specifies the protection key data that does not cause a protection violation exception in the protection violation exception handling routine to record the page in the UTLB and re-executes the instruction that has caused the protection violation exception.
      b. Exclude the pages for which software has once set 1 to the dirty bit upon occurrence of an initial page write exception and intentionally deleted from the TLB or set 0 to the dirty bit.
   D. Do not make an attempt to execute the FDIV or FSQRT instruction in the MMU-related exception handling routine.
2. If a TLB miss exception occurs, add 1 to MMUCR.URC before executing an LDTLB instruction.

Notes: 1. An exception handling routine is an entire set of instructions that are executed from the address (VBR + offset) upon occurrence of an exception to the RTE for returning to the original program or to the RTE delay slot.
2. MMU-related exceptions are: instruction TLB miss exception, instruction TLB miss protection violation exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception.
3. Instruction accesses include the PREFI and ICBI instructions.

Section 7 Caches

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

7.1 Features

The features of the cache are given in table 7.1.

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory. The features of the store queues are given in table 7.2.

Table 7.1 Cache Features

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

Table 7.2 Store Queue Features

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

The operand cache of this LSI uses the 4-way set-associative, each way comprising 256 cache lines. Figure 7.1 shows the configuration of the operand cache.

The instruction cache is 4-way set-associative, each way is comprising 256 cache lines. Figure 7.2 shows the configuration of the instruction cache.

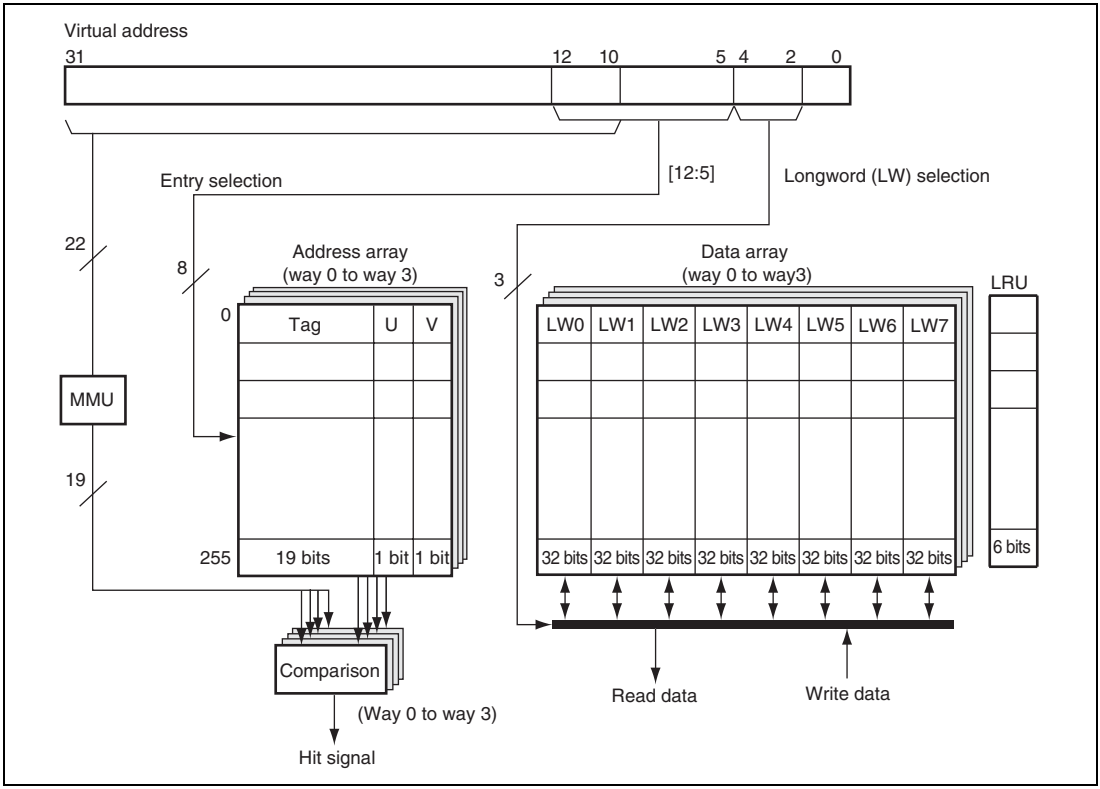


Figure 7.1 Configuration of Operand Cache (OC)

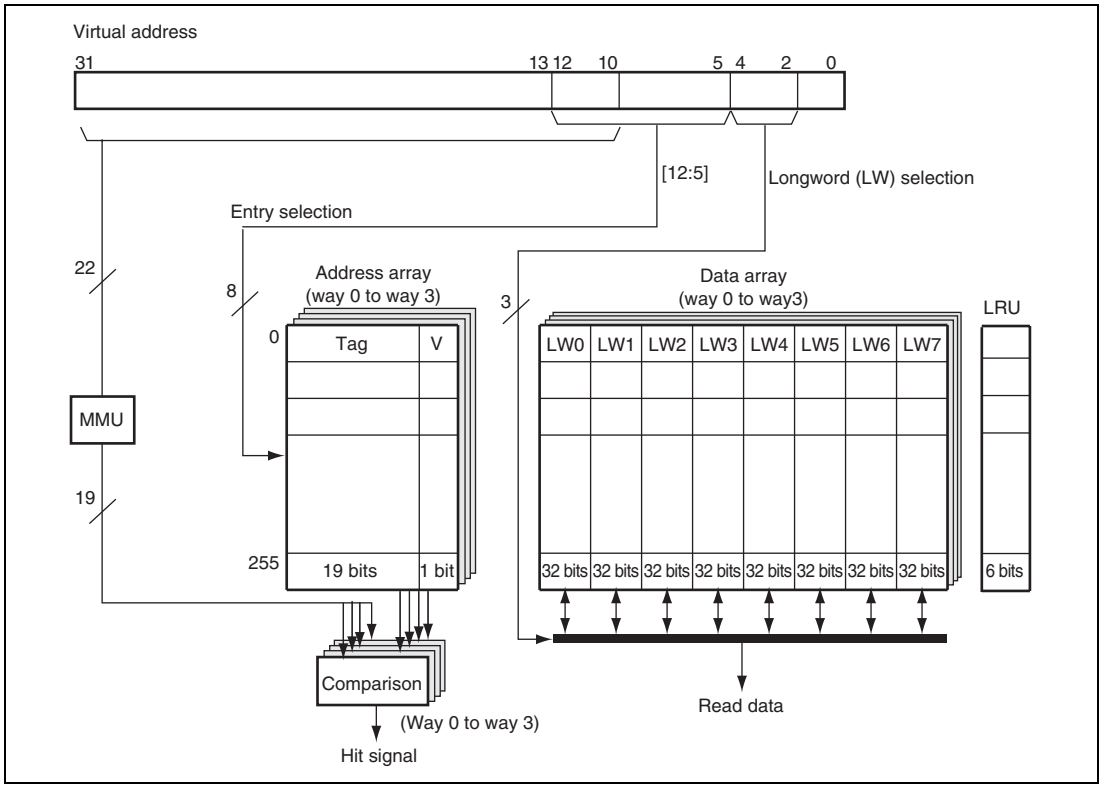


Figure 7.2 Configuration of Instruction Cache (IC)

- Tag

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

- V bit (validity bit)

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

- U bit (dirty bit)

  The U bit is set to 1 if data is written to the cache line while the cache is being used in copy-back mode. That is, the U bit indicates a mismatch between the data in the cache line and the data in external memory. The U bit is never set to 1 while the cache is being used in write-through mode, unless it is modified by accessing the memory-mapped cache (see section 7.6, Memory-Mapped Cache Configuration). The U bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- Data array

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

- LRU

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

7.2 Register Descriptions

The following registers are related to cache.

Table 7.3 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Size
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32
Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32
Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

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

Table 7.4 Register States in Each Processing State

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

7.2.1 Cache Control Register (CCR)

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

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

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

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

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
8	ICE	0	R/W	IC Enable Bit Selects whether the IC is used. Note however when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC not used 1: IC used
7 to 4	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of all OC entries are cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Copy-Back Bit Indicates the P1 area cache write mode. 0: Write-through mode 1: Copy-back mode
1	WT	0	R/W	Write-Through Mode Indicates the P0, U0, and P3 area cache write mode. When address translation is performed, the value of the WT bit in the page management information has priority. 0: Copy-back mode 1: Write-through mode

Bit	Bit Name	Initial Value	R/W	Description
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note however when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC not used 1: OC used

7.2.2 Queue Address Control Register 0 (QACR0)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4 to 2	AREA0	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ0.
1, 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

7.2.3 Queue Address Control Register 1 (QACR1)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
4 to 2	AREA1	Undefined	R/W	When the MMU is disabled, these bits generate physical address bits [28:26] for SQ1.
1, 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

7.2.4 On-Chip Memory Control Register (RAMCR)

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

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

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

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.
9	RMD	0	R/W	On-Chip Memory Access Mode Bit For details, see section 8.4, L Memory Protective Functions.

Bit	Bit Name	Initial Value	R/W	Description
8	RP	0	R/W	On-Chip Memory Protection Enable Bit For details, see section 8.4, L Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode bit 0: IC is a four-way operation 1: IC is a two-way operation For details, see section 7.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode bit 0: OC is a four-way operation 1: OC is a two-way operation For details, see section 7.3.6, OC Two-Way Mode.
5 to 0	—	All 0	R	Reserved For details on reading from or writing to these bits, see description in General Precautions on Handling of Product.

7.3 Operand Cache Operation

7.3.1 Read Operation

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

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

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

3. Cache hit

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

4. Cache miss (no write-back)

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

5. Cache miss (with write-back)

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

7.3.2 Prefetch Operation

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

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

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

3. Cache hit (copy-back)

   Then the LRU bits are updated to indicate the hitted way is the latest one.

4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the physical address space corresponding to the virtual address. Data reading is performed, using the wraparound method, in order from the quad-word data (8 bytes) including the cache-missed data. In the prefetch operation the CPU doesn't wait the data arrives. While the one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and 0 is written to the U bit on the way. And the LRU bit is updated to indicate the way is latest one.

5. Cache miss (with write-back)

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

7.3.3 Write Operation

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

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

- If there is a way whose tag matches and its V bit is 1, see No. 3 for copy-back and No. 4 for write-through.
- I If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for write-through.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way which is selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for write-through.

3. Cache hit (copy-back)

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

4. Cache hit (write-through)

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

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

   A data write in accordance with the access size is performed for the data of the data field on the hit way which is indexed by virtual address bits [4:0]. Then, the data, excluding the cache-missed data which is written already, is read into the cache line on the way which is selected to replace from the physical address space corresponding to the virtual address.

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

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

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

7. Cache miss (write-through)

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

7.3.4 Write-Back Buffer

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

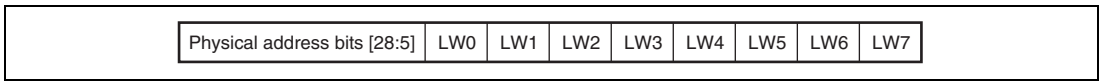


Figure 7.3 Configuration of Write-Back Buffer

7.3.5 Write-Through Buffer

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

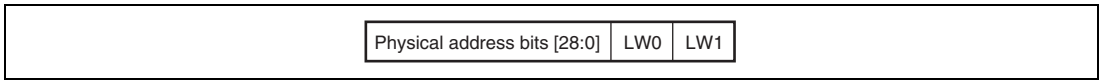


Figure 7.4 Configuration of Write-Through Buffer

7.3.6 OC Two-Way Mode

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

The OC2W bit should be modified by a program in the P2 area. At that time, if the valid line has already been recorded in the OC, data should be written back by software, if necessary, 1 should be written to the OCI bit in CCR, and all entries in the OC should be invalid before modifying the OC2W bit.

7.4 Instruction Cache Operation

7.4.1 Read Operation

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

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

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

3. Cache hit

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

4. Cache miss

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

7.4.2 Prefetch Operation

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

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

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

3. Cache hit

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

4. Cache miss

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

7.4.3 IC Two-Way Mode

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

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

7.5 Cache Operation Instruction

7.5.1 Coherency between Cache and External Memory

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

- Operand cache invalidate instruction: OCBI @Rn
  Operand cache invalidation (no write-back)
- Operand cache purge instruction: OCBP @Rn
  Operand cache invalidation (with write-back)
- Operand cache write-back instruction: OCBWB @Rn
  Operand cache write-back
- Operand cache allocate instruction: MOVCA.L R0,@Rn
  Operand cache allocation
- Instruction cache invalidate instruction: ICBI @Rn
  Instruction cache invalidation
- Operand access synchronization instruction: SYNCO
  Wait for data transfer completion

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

- 1 Kbyte page size cannot be used.

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

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

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

7.5.2 Prefetch Operation

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

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

7.6 Memory-Mapped Cache Configuration

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

1. Execute a branch using the RTE instruction.
2. Execute a branch to the P0, U0, P1, or P3 area after executing the ICBI instruction for any address (including non-cacheable area).
3. If the MC bit in IRMCR is 0 (initial value) before making an access to the memory-mapped IC, the specific instruction does not need to be executed. However, note that the CPU processing performance will be lowered because the instruction fetch is performed again for the next instruction after making an access to the memory-mapped IC.

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

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

7.6.1 IC Address Array

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

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

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

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

1. IC address array read

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

   The tag and V bit specified in the data field are written to the IC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.
3. IC address array write (associative)

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

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

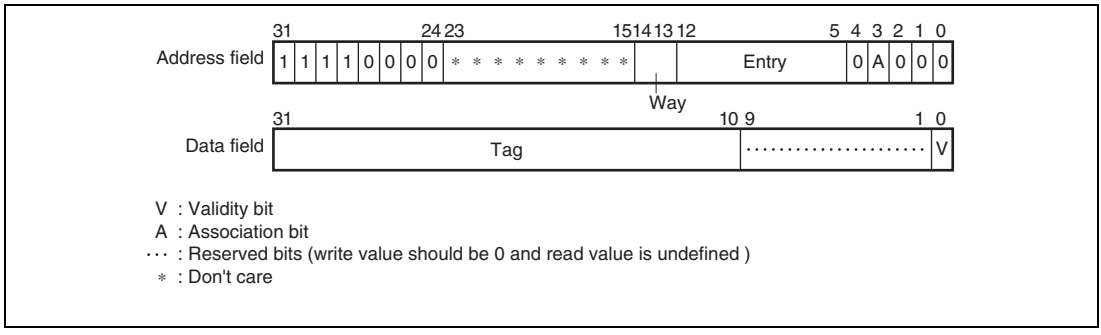


Figure 7.5 Memory-Mapped IC Address Array

7.6.2 IC Data Array

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

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

The data field is used for the longword data specification.

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

1. IC data array read

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

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

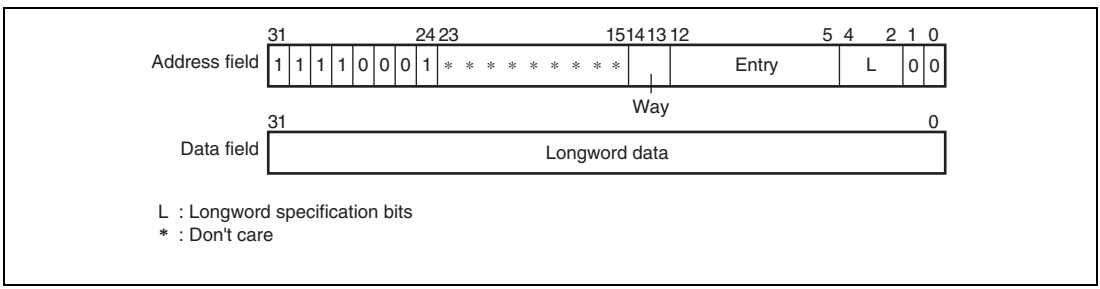


Figure 7.6 Memory-Mapped IC Data Array

7.6.3 OC Address Array

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

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

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

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

1. OC address array read

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

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-back of that cache line, the tag, U bit, and V bit specified in the data field are written.

3. OC address array write (associative)

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

Note: This function may not be supported in the future SuperH Series. Therefore, it is recommended that the OCBI, OCBP, or OCBWB instruction should be used to operate the OC definitely by reporting data TLB miss exception.

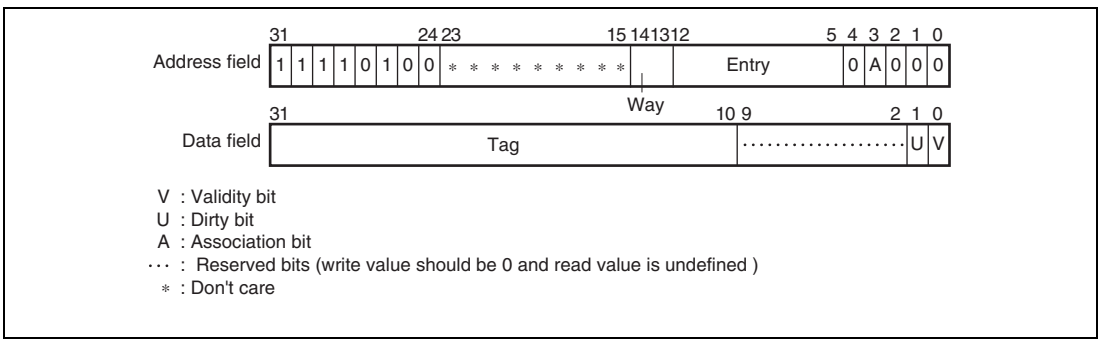


Figure 7.7 Memory-Mapped OC Address Array

7.6.4 OC Data Array

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

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

The data field is used for the longword data specification.

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

1. OC data array read

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

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

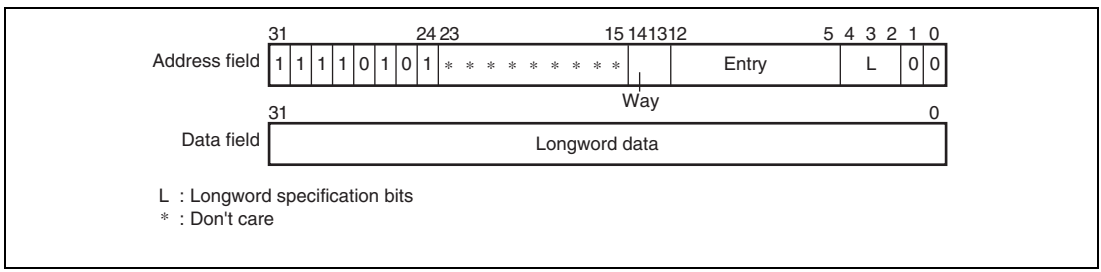


Figure 7.8 Memory-Mapped OC Data Array

7.7 Store Queues

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

7.7.1 SQ Configuration

There are two 32-byte store queues, SQ0 and SQ1, as shown in figure 7.9. These two store queues can be set independently.

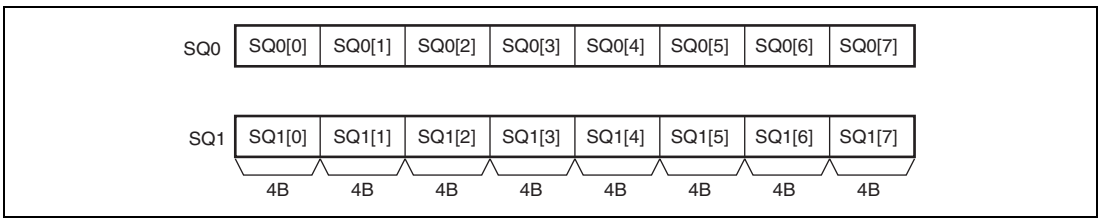


Figure 7.9 Store Queue Configuration

7.7.2 Writing to SQ

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

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

7.7.3 Transfer to External Memory

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

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

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

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

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

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

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

  QACR0[4:2] : Physical address bits [28:26] corresponding to SQ0
  QACR1[4:2] : Physical address bits [28:26] corresponding to SQ1

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

7.7.4 Determination of SQ Access Exception

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

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

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

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

  Operation is in accordance with the SQMD bit in MMUCR.

  0: Privileged/user mode access possible

  1: Privileged mode access possible

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

7.7.5 Reading from SQ

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

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

7.8 Notes on Using 32-Bit Address Extended Mode

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

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

Section 8 L Memory

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

8.1 Features

- Capacity

  Total L memory capacity is 16 Kbytes.
- Page

  The L memory is divided into two pages (pages 0 and 1).
- Memory map

  The L memory is allocated in the addresses shown in table 8.1 in both the virtual address space and the physical address space.

Table 8.1 L Memory Addresses

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

- Ports

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

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

8.2 Register Descriptions

The following registers are related to L memory.

Table 8.2 Register Configuration

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

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

Table 8.3 Register Status in Each Processing State

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

8.2.1 On-Chip Memory Control Register (RAMCR)

RAMCR controls the protective functions in the L memory.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31to10	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.
9	RMD	0	R/W	On-Chip Memory Access Mode Specifies the right of access to the L memory from the virtual address space. 0: An access in privileged mode is allowed. (An address error exception occurs in user mode.) 1: An access user/privileged mode is allowed.
8	RP	0	R/W	On-Chip Memory Protection Enable Selects whether or not to use the protective functions using ITLB and UTLB for accessing the L memory from the virtual address space. 0: Protective functions are not used. 1: Protective functions are used. For further details, refer to section 8.4, L Memory Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode For further details, refer to section 7.4.3, IC Two-Way Mode.
6	OC2W	0	R/W	OC Two-Way Mode For further details, refer to section 7.3.6, OC Two-Way Mode.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

8.2.2 L Memory Transfer Source Address Register 0 (LSA0)

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

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

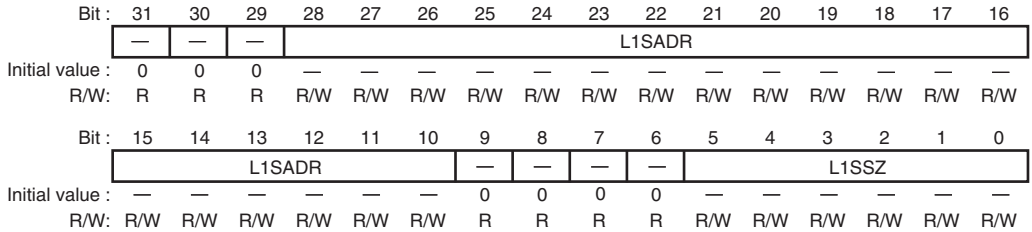
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0SADR						—	—	—	—	L0SSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For read/write in these bits, refer General Precautions on Handling of Product.
28 to 10	L0SADR	Undefined	R/W	L Memory Page 0 Block Transfer Source Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify the transfer source physical address for block transfer to page 0 in the L memory.
9 to 6	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

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

8.2.3 L Memory Transfer Source Address Register 1 (LSA1)

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

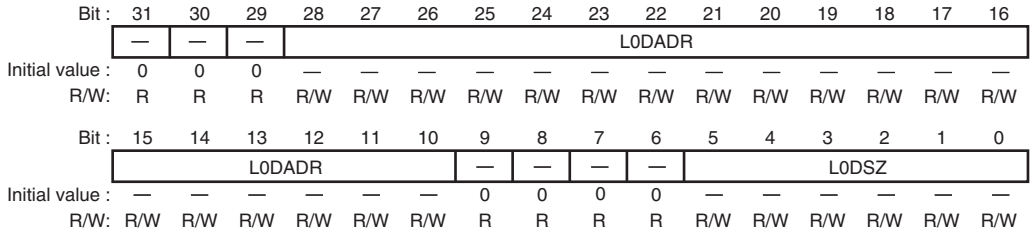


Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.
28 to 10	L1SADR	Undefined	R/W	L Memory Page 1 Block Transfer Source Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer source physical address for block transfer to page 1 in the L memory.
9 to 6	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

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

8.2.4 L Memory Transfer Destination Address Register 0 (LDA0)

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



Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	L0DADR												
Initial value :	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

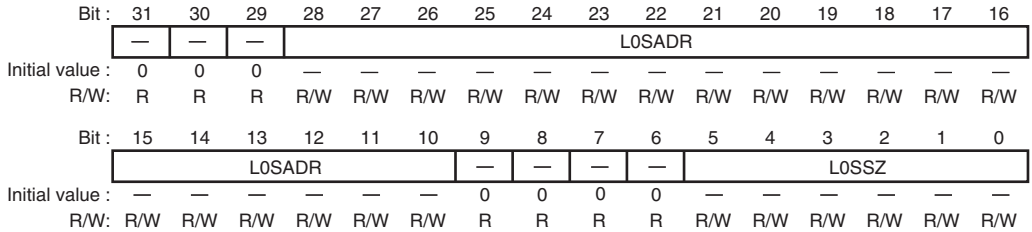
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L0DADR						—	—	—	—	L0DSZ					
Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.
28 to 10	L0DADR	Undefined	R/W	L Memory Page 0 Block Transfer Destination Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer destination physical address for block transfer to page 0 in the L memory.
9 to 6	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

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

8.2.5 L Memory Transfer Destination Address Register 1 (LDA1)

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



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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.
28 to 10	L1DADR	Undefined	R/W	L Memory Page 1 Block Transfer Destination Address When MMUCR.AT = 0 or RAMCR.RP = 0, these bits specify transfer destination physical address for block transfer to page 1 in the L memory.
9 to 6	—	All 0	R	Reserved For read/write in these bits, refer to General Precautions on Handling of Product.

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

8.3 Operation

8.3.1 Access from the CPU and FPU

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

8.3.2 Access from the SuperHyway Bus Master Module

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

8.3.3 Block Transfer

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

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

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

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

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

(1) When MMU is Enabled (MMUCR.AT = 1) and RAMCR.RP = 1

An address of the L memory area is specified to the UTLB VPN field, and to the physical address of the transfer source (in the case of the PREF instruction) or the transfer destination (in the case

of the OCBWB instruction) to the PPN field. The ASID, V, SZ, SH, PR, and D bits have the same meaning as normal address conversion; however, the C and WT bits have no meaning in this page.

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

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

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

(2) When MMU is Disabled (MMUCR.AT = 0) or RAMCR.RP = 0

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

The transfer destination physical address in block transfer from page 0 in the L memory is set in the L0DADR bits of the LDA0 register. And the L0DSZ bits in the LDA0 register choose either the virtual addresses specified through the OCBWB instruction or the L0DADR values as bits 15 to 10 of the transfer destination physical address. In other words, the transfer source area can be specified in units of 1 Kbyte to 64 Kbytes.

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

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

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

8.4 L Memory Protective Functions

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

- Protective functions for access from the CPU and FPU

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

  When MMUCR.AT = 1 and RAMCR.RP = 1, MMU exception and address error exception are checked in the L memory area which is a part of P4 area as with the area P0/P3/U0.

The above descriptions are summarized in table 8.4.

Table 8.4 Protective Function Exceptions to Access L Memory

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

Note: * Don't care

8.5 Usage Notes

8.5.1 Page Conflict

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

8.5.2 L Memory Coherency

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

- SYNCO
- ICBI @Rn

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

8.5.3 Sleep Mode

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

8.6 Note on Using 32-Bit Address Extended Mode

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

Section 9 Interrupt Controller (INTC)

The interrupt controller (INTC) determines the priority of interrupt sources and controls interrupt requests to the CPU (SH-4A). The INTC has a register that sets the priority of each interrupt and interrupt requests are processed according to the priority set in this register by the user.

9.1 Features

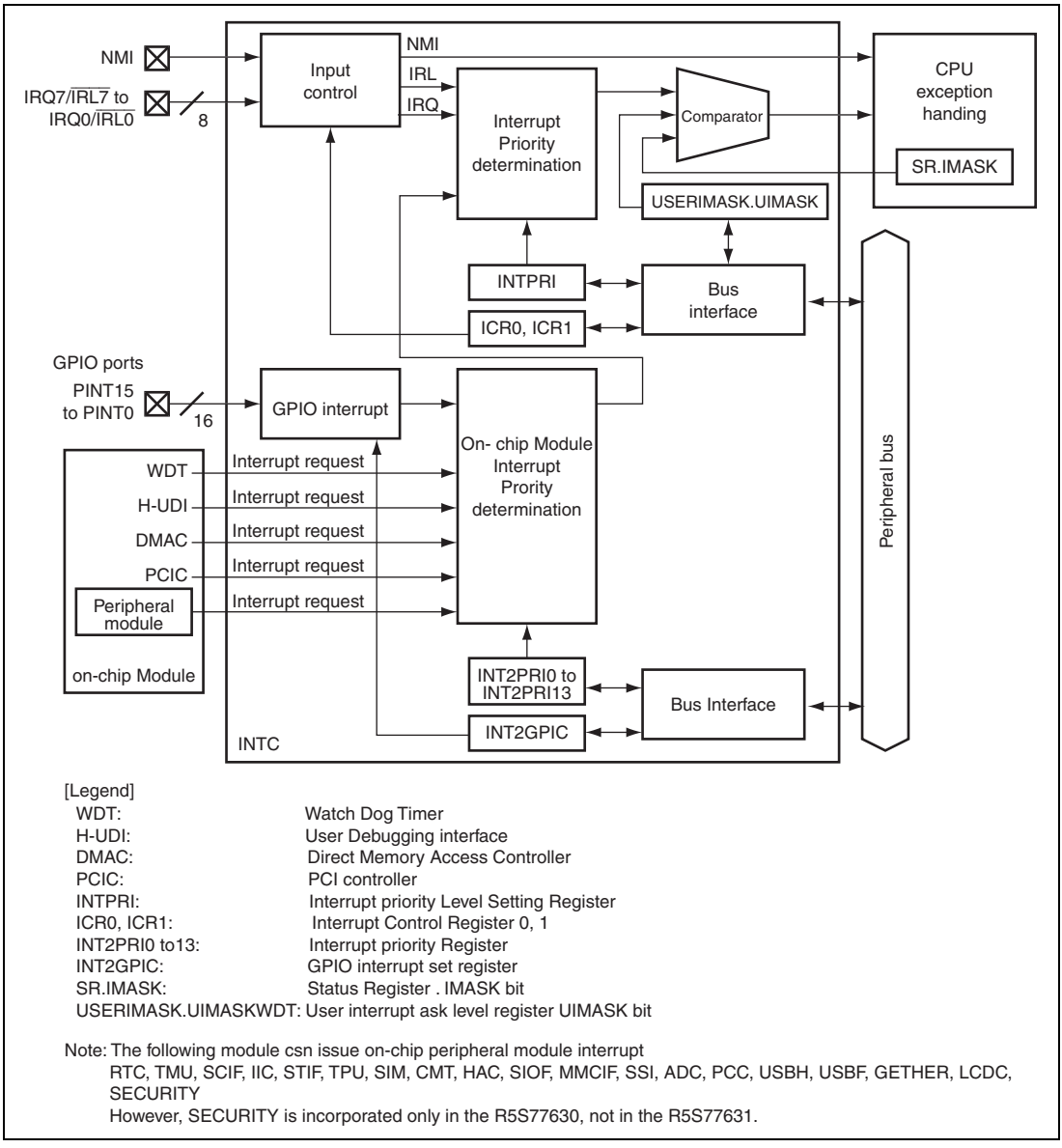
SH-4 compatible specifications

- Fifteen levels of external interrupt priority can be set
  By setting the interrupt priority registers, the priorities of external interrupts can be selected from 15 levels for individual request sources.
- NMI noise canceller function
  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception handling routine, the pin state can be checked, enabling it to be used as a noise canceller.
- NMI request masking when the block bit (BL) in the status register (SR) is set to 1
  Whether to mask NMI requests when the BL bit in SR is set to 1 can be selected.

Extended function for SH-4A

- Automatically updates the IMASK bit in SR according to the accepted interrupt level
- Thirteen levels of on-chip module interrupt priority can be set
  By setting thirteen interrupt priority registers, the priorities of on-chip module interrupts can be selected from 30 levels for individual request sources.
- User-mode interrupt disabling function
  Specifying an interrupt mask level in the user interrupt mask level register (USERIMASK) disables interrupts which are not higher in priority than the specified mask level in user mode.

Figure 9.1 shows a block diagram of the INTC.



[Legend]

WDT:	Watch Dog Timer
H-UDI:	User Debugging interface
DMAC:	Direct Memory Access Controller
PCIC:	PCI controller
INTPRI:	Interrupt priority Level Setting Register
ICR0, ICR1:	Interrupt Control Register 0, 1
INT2PRI0 to13:	Interrupt priority Register
INT2GPIC:	GPIO interrupt set register
SR.IMASK:	Status Register . IMASK bit
USERIMASK.UIMASKWDT:	User interrupt ask level register UIMASK bit

Note: The following module csn issue on-chip peripheral module interrupt
RTC, TMU, SCIF, IIC, STIF, TPU, SIM, CMT, HAC, SIOF, MMCIF, SSI, ADC, PCC, USBH, USBF, GETHER, LCDC, SECURITY
However, SECURITY is incorporated only in the R5S77630, not in the R5S77631.

Figure 9.1 Block Diagram of INTC

9.1.1 Interrupt Method

The basic exception handling flow for the interrupt is as follows.

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

1. The PC, SR and R15 contents are saved to SPC, SSR and SGR, respectively.
2. The block (BL) bit in SR is set to 1.
3. The mode (MD) bit in SR is set to 1.
4. The register bank (RB) bit in SR is set to 1.
5. In a reset, the FPU disable (FD) bit in SR is cleared to 0.
6. The exception code is written to bits 13 to 0 in the interrupt event register (INTEVT) of the exception source.
7. The processing is jumped to the start address of the interrupt exception handling routine, vector base register (VBR) + H'600.
8. The processing is branched to the vector address of the determined interrupt exception handling and the interrupt exception handling routine is started.

9.1.2 Interrupt Types in INTC

Table 9.1 shows an example of the interrupt types. The INTC supports both the external interrupts and on-chip module interrupts.

The external interrupts is the interrupt input from external pins, NMI, IRL, and IRQ.

The IRQ and IRL are assigned to the same pin in the SH7763. The pin function is selected according to the system.

IRQ input can be selected as a level, or a rising or falling edge.

Table 9.1 Interrupt Types

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks	
External interrupts	NMI	1	—	H'1C0		
	IRL interrupt*[1]	2	Inversion values of input pin values (because of negative pins) For example $\overline{IRL[7:4]}$ pin = H'0 means external pin input level is, $\overline{IRL[7]}$ pin = Low $\overline{IRL[6]}$ pin = Low $\overline{IRL[5]}$ pin = Low $\overline{IRL[4]}$ pin = Low then priority level is,15 (H'F) (See table 9.6)	H'200	$\overline{IRL[7:4]}$ pin = H'0	High
					$\overline{IRL[3:0]}$ pin = H'0	
				H'220	$\overline{IRL[7:4]}$ pin = H'1	
					$\overline{IRL[3:0]}$ pin = H'1	
				H'240	$\overline{IRL[7:4]}$ pin = H'2	
					$\overline{IRL[3:0]}$ pin = H'2	
				H'260	$\overline{IRL[7:4]}$ pin = H'3	
					$\overline{IRL[3:0]}$ pin = H'3	
				H'280	$\overline{IRL[7:4]}$ pin = H'4	
					$\overline{IRL[3:0]}$ pin = H'4	
				H'2A0	$\overline{IRL[7:4]}$ pin = H'5	
					$\overline{IRL[3:0]}$ pin = H'5	
				H'2C0	$\overline{IRL[7:4]}$ pin = H'6	
					$\overline{IRL[3:0]}$ pin = H'6	
				H'2E0	$\overline{IRL[7:4]}$ pin = H'7	
					$\overline{IRL[3:0]}$ pin = H'7	
				H'300	$\overline{IRL[7:4]}$ pin = H'8	
					$\overline{IRL[3:0]}$ pin = H'8	Low

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks	
External interrupts	IRL interrupt*1	2	Inversion values of input pin values (because of negative pins) For example $\overline{IRL}[7:4]$ pin = H'0 means external pin input level is, $\overline{IRL}[7]$ pin = Low $\overline{IRL}[6]$ pin = Low $\overline{IRL}[5]$ pin = Low $\overline{IRL}[4]$ pin = Low then priority level is,15 (H'F) (See table 9.6)	H'320	$\overline{IRL}[7:4]$ pin = H'9	High
					$\overline{IRL}[3:0]$ pin = H'9	
				H'340	$\overline{IRL}[7:4]$ pin = H'A	
					$\overline{IRL}[3:0]$ pin = H'A	
				H'360	$\overline{IRL}[7:4]$ pin = H'B	
					$\overline{IRL}[3:0]$ pin = H'B	
				H'380	$\overline{IRL}[7:4]$ pin = H'C	
					$\overline{IRL}[3:0]$ pin = H'C	
				H'3A0	$\overline{IRL}[7:4]$ pin = H'D	
					$\overline{IRL}[3:0]$ pin = H'D	
				H'3C0	$\overline{IRL}[7:4]$ pin = H'E	
					$\overline{IRL}[3:0]$ pin = H'E	
	IRQ interrupt	8	Values set in INTPRI	H'240	IRQ[0]	
				H'280	IRQ[1]	
				H'2C0	IRQ[2]	
				H'300	IRQ[3]	
				H'340	IRQ[4]	
				H'380	IRQ[5]	
				H'3C0	IRQ[6]	
				H'200	IRQ[7]	Low
On-chip module interrupts*2	RTC	3	Setting value of INT2PRI0 to INT2PRI13	H'480	ATI	
				H'4A0	PRI	
				H'4C0	CUI	
	SECURITY *3	1		H'4E0	SECI	
	WDT	1		H'560	ITI	
	TMU0	1		H'580	TUNI0	
	TMU1	1		H'5A0	TUNI1	
	TMU2	2		H'5C0	TUNI2	
				H'5E0	TICPI2	
	H-UDI	1		H'600	H-UDI	

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts*2	LCDC	1	Setting value of INT2PRI0 to INT2PRI13	H'620	LCDCI
	DMAC	7(5/7)		H'640	DMTE0
				H'660	DMTE1
				H'680	DMTE2
				H'6A0	DMTE3
				H'6C0	DMAE
	SCIF0	4		H'700	ERI0
				H'720	RXI0
				H'740	BRI0
				H'760	TXI0
	DMAC	7(2/7)		H'780	DMTE4
				H'7A0	DMTE5
	IIC0	1		H'8A0	IICI0
	IIC1	1		H'8C0	IICI1
	CMT	1		H'900	CMTI
	GETHER	3		H'920	GEINT0
				H'940	GEINT1
				H'960	GEINT2
	HAC	1		H'980	HACI
	PCIC0	1		H'A00	SERR
	PCIC1	1		H'A20	INTA
	PCIC2	1		H'A40	INTB
	PCIC3	1		H'A60	INTC
	PCIC4	1		H'A80	INTD
	PCIC5	5		H'AA0	ERR
				H'AC0	PWD3
				H'AE0	PWD2
				H'B00	PWD1
				H'B20	PWD0
	STIF0	1		H'B40	STIFI0
	STIF1	1		H'B60	STIFI1

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts※2	SCIF1	4	Setting value of INT2PRI0 to INT2PRI13	H'B80	ERI1
				H'BA0	RXI1
				H'BC0	BRI1
				H'BE0	TXI1
	SIOF0	1		H'C00	SIOFI0
	SIOF1	1		H'C20	SIOFI1
	SIOF2	1		H'C40	SIOFI2
	USBH	1		H'C60	USBHI
	USBF	2		H'C80	USBFI0
				H'CA0	USBFI1
	TPU	1		H'CC0	TPI
	PCC	1		H'CE0	PCCI
	MMCIF	4		H'D00	FSTAT
				H'D20	TRAN
				H'D40	ERR
				H'D60	FRDY
	SIM	4		H'D80	ERI
				H'DA0	RXI
				H'DC0	TXI
				H'DE0	TEND
	TMU3	1		H'E00	TUNI3
	TMU4	1		H'E20	TUNI4
	TMU5	1		H'E40	TUNI5
	ADC	1		H'E60	ADI
	SSI0	1		H'E80	SSII0
	SSI1	1		H'EA0	SSII1
	SSI2	1		H'EC0	SSII2
	SSI3	1		H'EE0	SSII3
	SCIF2	4		H'F00	ERI2
				H'F20	RXI2
				H'F40	BRI2
				H'F60	TXI2

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
On-chip module interrupts*2	GPIO	4	Setting value of INT2PRI0 to INT2PRI13	H'F80	CH0
				H'FA0	CH1
				H'FC0	CH2
				H'FE0	CH3

Notes: 1. Since the IRL interrupt request by IRL[3:0] (IRQ3/IRL3 to IRQ0/IRL0 pins) and IRL interrupt request by IRL[7:4] (IRQ7/IRL7 to IRQ4/IRL4 pins) have the same INTEVT codes, it is impossible to distinguish the former from the latter. Note that there is no flags in this LSI for distinguishing between them.

2. ITI: Interval timer interrupt
   TUNI0 to TUNI5: TMU channel 0 to 5 under flow interrupt
   TICPI2: TMU channel 2 input capture interrupt
   DMINT0 to DMINT11: DMAC channel 0 to 5 transfer end interrupt
   DMAE: DMAC address error interrupt (channel 0 to 5)
   ERI0, ERI1: SCIF channel 0, 1 receive error interrupt
   RXI0, RXI1: SCIF channel 0, 1 receive data full interrupt
   BRI0, BRI1: SCIF channel 0, 1 break interrupt
   TXI0, TXI1: SCIF channel 0, 1 transmission data empty interrupt
3. The SECURITY is not incorporated in the R5S77631. Therefore, the INTEVT code is reserved in the R5S77631.

9.2 Input/Output Pins

Table 9.2 shows the pin configuration.

Table 9.2 INTC Pin Configuration

Pin Name	Function	I/O	Description
NMI	Nonmaskable interrupt input pin	Input	Nonmaskable interrupt request signal input
IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$	External interrupt input pin	Input	Interrupt request signal input IRL [3:0] 4-bit level-encoded interrupt input when ICR0.IRLM0 = 0 IRQ3 to IRQ0 individual interrupt input when ICR0.IRLM0 = 1

Pin Name	Function	I/O	Description
IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$	External interrupt input pin	Input	Interrupt request signal input IRL [7:4] 4-bit level-encoded interrupt input when ICR0.IRLM1 = 0 IRQ7 to IRQ4 individual interrupt input when ICR0.IRLM1 = 1
$\overline{\text{IRQOUT}}$	Interrupt request output	Output	Notifies that an interrupt request has generated This pin is asserted even if the CPU does not accept the interrupt request, but not asserted when the interrupt is masked.
PINT15 to PINT0	Port interrupt input pins	Input	Port interrupt request signal input

9.3 Register Descriptions

Table 9.3 shows the INTC register configuration. These registers maintain software interfaces with the CPU (SH-4A) and are initialized by a power-on reset and a manual reset.

Table 9.3 shows the INTC register configuration. Table 9.4 shows the register states in each operating mode.

Table 9.3 INTC Register Configuration

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Interrupt control register 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32
Interrupt control register 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32
Interrupt priority register	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32
Interrupt source register	INTREQ	R/(W)	H'FFD0 0024	H'1FD0 0024	32
Interrupt mask register 0	INTMSK0	R/W	H'FFD0 0044	H'1FD0 0044	32
Interrupt mask register 1	INTMSK1	R/W	H'FFD0 0048	H'1FD0 0048	32
Interrupt mask register 2	INTMSK2	R/W	H'FFD4 0080	H'1FD4 0080	32
Interrupt mask clear register 0	INTMSKCLR0	R/W	H'FFD0 0064	H'1FD0 0064	32
Interrupt mask clear register 1	INTMSKCLR1	R/W	H'FFD0 0068	H'1FD0 0068	32
Interrupt mask clear register 2	INTMSKCLR2	R/W	H'FFD4 0084	H'1FD4 0084	32

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
NMI flag control register	NMIFCR	R/(W)	H'FFD0 00C0	H'1FD0 00C0	32
User interrupt mask level register	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32
Interrupt priority register 0	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32
Interrupt priority register 1	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32
Interrupt priority register 2	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32
Interrupt priority register 3	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32
Interrupt priority register 4	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32
Interrupt priority register 5	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32
Interrupt priority register 6	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32
Interrupt priority register 7	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32
Interrupt priority register 8	INT2PRI8	R/W	H'FFD4 00A0	H'1FD4 00A0	32
Interrupt priority register 9	INT2PRI9	R/W	H'FFD4 00A4	H'1FD4 00A4	32
Interrupt priority register 10	INT2PRI10	R/W	H'FFD4 00A8	H'1FD4 00A8	32
Interrupt priority register 11	INT2PRI11	R/W	H'FFD4 00AC	H'1FD4 00AC	32
Interrupt priority register 12	INT2PRI12	R/W	H'FFD4 00B0	H'1FD4 00B0	32
Interrupt priority register 13	INT2PRI13	R/W	H'FFD4 00B4	H'1FD4 00B4	32
Interrupt source register 0 (mask state is not affected)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32
Interrupt source register 01 (mask state is not affected)	INT2A01	R	H'FFD4 00C0	H'1FD4 00C0	32
Interrupt source register 1 (mask state is affected)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32
Interrupt source register 11 (mask state is affected)	INT2A11	R	H'FFD4 00C4	H'1FD4 00C4	32
Interrupt mask register	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32
Interrupt mask register 1	INT2MSKR1	R/W	H'FFD4 00D0	H'1FD4 00D0	32
Interrupt mask clear register	INT2MSKCR	W	H'FFD4 003C	H'1FD4 003C	32
Interrupt mask clear register 1	INT2MSKCR1	W	H'FFD4 00D4	H'1FD4 00D4	32
Individual module interrupt source register 0	INT2B0	R	H'FFD4 0040	H'1FD4 0040	32
Individual module interrupt source register 1	INT2B1	R	H'FFD4 0044	H'1FD4 0044	32

Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Individual module interrupt source register 2	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32
Individual module interrupt source register 3	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32
Individual module interrupt source register 4	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32
Individual module interrupt source register 5	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32
Individual module interrupt source register 6	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32
Individual module interrupt source register 7	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32
Individual module interrupt source register 9	INT2B9	R	H'FFD4 0064	H'1FD4 0064	32
Individual module interrupt source register 10	INT2B10	R	H'FFD4 0068	H'1FD4 0068	32
Individual module interrupt source register 11	INT2B11	R	H'FFD4 006C	H'1FD4 006C	32
GPIO interrupt set register	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32

Table 9.4 Register States in Each Operating Mode

Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Interrupt control register 0	ICR0	H'x000 0000	H'x000 0000	Retained	Retained
Interrupt control register 1	ICR1	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register	INTPRI	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt source register	INTREQ	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask register 0	INTMSK0	H'FF00 0000	H'FF00 0000	Retained	Retained
Interrupt mask register 1	INTMSK1	H'FF00 0000	H'FF00 0000	Retained	Retained
Interrupt mask register 2	INTMSK2	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask clear register 0	INTMSKCLR0	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask clear register 1	INTMSKCLR1	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask clear register 2	INTMSKCLR2	H'0000 0000	H'0000 0000	Retained	Retained
NMI flag control register	NMIFCR	H'x000 0000	H'x000 0000	Retained	Retained
User interrupt mask level register	USERIMASK	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 0	INT2PRI0	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 1	INT2PRI1	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 2	INT2PRI2	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 3	INT2PRI3	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 4	INT2PRI4	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 5	INT2PRI5	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 6	INT2PRI6	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 7	INT2PRI7	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 8	INT2PRI8	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 9	INT2PRI9	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 10	INT2PRI10	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 11	INT2PRI11	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 12	INT2PRI12	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt priority register 13	INT2PRI13	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt source register 0 (mask state is not affected)	INT2A0	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt source register 01 (mask state is affected)	INT2A01	H'0000 0000	H'0000 0000	Retained	Retained

Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Interrupt source register 1 (mask state is affected)	INT2A1	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt source register 11 (mask state is affected)	INT2A11	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask register	INT2MSKR	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Interrupt mask register 1	INT2MSKR1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Interrupt mask clear register	INT2MSKCR	H'0000 0000	H'0000 0000	Retained	Retained
Interrupt mask clear register 1	INT2MSKCR1	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 0	INT2B0	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 1	INT2B1	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 2	INT2B2	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 3	INT2B3	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 4	INT2B4	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 5	INT2B5	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 6	INT2B6	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 7	INT2B7	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 9	INT2B9	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 10	INT2B10	H'0000 0000	H'0000 0000	Retained	Retained
Individual module interrupt source registers 11	INT2B11	H'0000 0000	H'0000 0000	Retained	Retained
GPIO interrupt set register	INT2GPIC	H'0000 0000	H'0000 0000	Retained	Retained

9.3.1 Interrupt Control Register 0 (ICR0)

ICR0 is a 32-bit readable and partially writable register that sets the input signal detection mode of the external interrupt input pins (IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$) and NMI pin, and indicates the input level to the NMI pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMIL	MAI	—	—	—	—	NMIB	NMIE	IRLM0	IRLM1	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	NMIL	Undefined	R	NMI Input Level Sets the signal level input to the NMI pin. Reading this bit allows the user to know the NMI pin level, and writing is invalid. 0: Low level is input to the NMI pin 1: High level is input to the NMI pin
30	MAI	0	R/W	MAI Interrupt Mask Specifies whether all interrupts are masked during the low level period of the NMI pin level regardless of the BL bit in SR of the CPU. 0: Interrupts are enabled even if the NMI pin goes low 1: Interrupts are disabled if the NMI pin goes low
29 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
25	NMIB	0	R/W	NMI Block Mode Selects whether an NMI interrupt is held until the BL bit in SR is cleared to 0 or detected immediately when the BL bit in SR of the CPU is set to 1. 0: An NMI interrupt is held when the BL bit in SR is set to 1 (initial value) 1: An NMI interrupt is not held when the BL bit in SR is set to 1 Note: If interrupts are accepted with the BL bit in SR set to 1, previous exception information (SSR, SPC, SGR, and INTEVT) is lost.
24	NMIE	0	R/W	NMI Edge Select Selects whether an interrupt request signal to the NMI pin is detected at the rising edge or the falling edge. 0: An interrupt request is detected at the falling edge of NMI input (initial value) 1: An interrupt request is detected at the rising edge of NMI input
23	IRLM0	0	R/W	IRL Pin Mode 0 Selects whether IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ are used as the 4-bit encoded interrupt requests or as four independent interrupts. 0: IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ are used as the 4-bit level-encoded interrupt requests (IRL [3:0] interrupt; initial value) 1: IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ are used as four independent interrupt requests (IRQ [n] interrupt; n = 3 to 0) Note: The level-encoded IRL interrupt is not detected unless the pin levels sampled at every bus clock cycle remain unchanged for four consecutive cycles.

Bit	Bit Name	Initial Value	R/W	Description
22	IRLM1	0	R/W	IRL Pin Mode 1 Selects whether IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ are used as the 4-bit encoded interrupt requests or as four independent interrupts. 0: IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ are used as the 4-bit level-encoded interrupt requests (IRL [7:4] interrupt; initial value) 1: IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ are used as four independent interrupt requests (IRQ [n] interrupt; n = 7 to 4) Note: The level-encoded IRL interrupt is not detected unless the pin levels sampled at every bus clock cycle remain unchanged for four consecutive cycles.
21 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.2 Interrupt Control Register 1 (ICR1)

ICR1 is a 32-bit readable/writable register that specifies the individual input signal detection modes of external interrupt input pins IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$. This setting is valid only when using $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ as IRQ independent interrupts input to set the IRLM0 and IRLM1 bits to 1 in ICR0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0S		IRQ1S		IRQ2S		IRQ3S		IRQ4S		IRQ5S		IRQ6S		IRQ7S	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	IRQ0S	0	R/W	IRQn Sense Select (n = 0 to 7) Selects whether interrupt signals to the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pins are detected at the rising edge, falling edge, high level, or low level. 00: Interrupt requests are detected at the falling edge of IRQn input 01: Interrupt requests are detected at the rising edge of IRQn input 10: Interrupt requests are detected at the low level of IRQn input 11: Interrupt requests are detected at the high level of IRQn input
29, 28	IRQ1S	0	R/W	
27, 26	IRQ2S	0	R/W	
25, 24	IRQ3S	0	R/W	
23, 22	IRQ4S	0	R/W	
21, 20	IRQ5S	0	R/W	
19, 18	IRQ6S	0	R/W	
17, 16	IRQ7S	0	R/W	
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When the IRQ is set as level input (IRQnS1 = 1), an interrupt source is held until the CPU accepts an interrupt (not always IRQ). Therefore even if an interrupt source is disabled before this LSI returns from sleep mode, it is guaranteed that processing is branched to the interrupt handler when this LSI returns from sleep mode. The held interrupt can be cleared by setting the corresponding interrupt mask bit (the IM bit in the interrupt mask register) to 1.

9.3.3 Interrupt Priority Register (INTPRI)

INTPRI is a 32-bit readable/writable register that sets the IRQ[7:0] interrupt priorities (levels 15 to 0). This setting is valid only when using IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ and IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ as IRQ independent interrupts input to set the IRLM0 and IRLM1 bits to 1 in ICR0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IP0				IP1				IP2				IP3			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IP4				IP5				IP6				IP7			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

9.3.4 Interrupt Source Register (INTREQ)

INTREQ is a 32-bit readable and writable with conditions register that indicates which IRQ [n] (n = 0 to 7) interrupt is requested to the INTC.

Even if interrupts are masked by INTPRI and INTMSK0, the INTREQ bits are not affected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description	
				At Edge Detection (IRQnS = 00 or 01, n = 0 to 7)	At Level Detection (IRQnS = 10 or 11, n = 0 to 7)
31	IR0	0	R/W	[When reading]	[When reading]
30	IR1	0	R/W	0: A corresponding IRQ interrupt request is not detected	0: A corresponding IRQ interrupt pin is not asserted
29	IR2	0	R/W		
28	IR3	0	R/W	1: A corresponding IRQ interrupt request is detected	1: A corresponding IRQ interrupt pin has asserted, but the CPU does not accept it yet
27	IR4	0	R/W		
26	IR5	0	R/W	[When writing]*	Writing is ignored.
25	IR6	0	R/W	0: Each bit is cleared by writing 0 after reading 1	
24	IR7	0	R/W	1: Holds detected interrupt request	
				Note: Write 1 to the corresponding bit read as 0.	
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	

9.3.5 Interrupt Mask Register 0 (INTMSK0)

INTMSK0 is 32-bit readable and writable with conditions registers that control mask settings for each interrupt request. To clear the mask settings for interrupts, write 1 to the corresponding bits in INTMSKCLR0. Writing 0 to bits in INTMSK0 is invalid.

Note: Write B'1111 to the IM [03:00] or IM [07:04] bits when IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ or IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ is set to the 4-bit encoded interrupt input.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM00	IM01	IM02	IM03	IM04	IM05	IM06	IM07	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IM00	1	R/W	Sets masking of an independent interrupt request of IRQ0.
30	IM01	1	R/W	Sets masking of an independent interrupt request of IRQ1.
29	IM02	1	R/W	Sets masking of an independent interrupt request of IRQ2.
28	IM03	1	R/W	Sets masking of an independent interrupt request of IRQ3.
27	IM04	1	R/W	Sets masking of an independent interrupt request of IRQ4.
26	IM05	1	R/W	Sets masking of an independent interrupt request of IRQ5.
25	IM06	1	R/W	Sets masking of an independent interrupt request of IRQ6.
24	IM07	1	R/W	Sets masking of an independent interrupt request of IRQ7.
				[When reading] 0: Interrupts are accepted 1: Interrupts are masked [When writing] 0: Invalid 1: Interrupts are masked
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.6 Interrupt mask register 1 (INTMSK1)

INTMSK1 is 32-bit readable and writable with conditions registers that control mask settings for each interrupt request. To clear the mask settings for interrupts, write 1 to the corresponding bits in INTMSKCLR1. Writing 0 to bits in INTMSK1 is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM10	IM11	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	IM10	1	R/W	Sets masking of $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ interrupt requests when IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ are encoded interrupt input. [When reading] 0: Interrupts are accepted 1: Interrupts are masked [When writing] 0: Invalid 1: Interrupts are masked
30	IM11	1	R/W	Sets masking of IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ interrupt requests when IRL [7:4] are encoded interrupt input.
29 to 24	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.7 Interrupt mask register 2 (INTMSK2)

INTMSK2 is 32-bit readable and writable with conditions registers that control mask settings for each interrupt request. To clear the mask settings for interrupts, write 1 to the corresponding bits in INTMSKCLR2. Writing 0 to bits in INTMSK2 is invalid.

INTMSK2 is valid when an IRL interrupt, which is generated by encoding input signals on pins $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ or $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, is requested while an IRL interrupt is not masked by INTMSK1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM015	IM014	IM013	IM012	IM011	IM010	IM009	IM008	IM007	IM006	IM005	IM004	IM003	IM002	IM001	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IM115	IM114	IM113	IM112	IM111	IM110	IM109	IM108	IM107	IM106	IM105	IM104	IM103	IM102	IM101	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description	
31	IM015	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LLLL (H'0).	[When reading] 0: Interrupts are accepted 1: Interrupts are masked [When writing] 0: Invalid 1: Interrupts are masked
30	IM014	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LLLH (H'1).	
29	IM013	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LLHL (H'2).	
28	IM012	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LLHH (H'3).	
27	IM011	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LHLL (H'4)..	
26	IM010	0	R/W	Sets masking of an interrupt request when $\overline{\text{IRL}}$ [3:0] = LHLH (H'5).	

Bit	Bit Name	Initial Value	R/W	Description
25	IM009	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = LHHL (H'6).
24	IM008	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = LHHH (H'7).
23	IM007	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HLLL (H'8).
22	IM006	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HLLH (H'9).
21	IM005	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HLHL (H'A).
20	IM004	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HLHH (H'B).
19	IM003	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HHLL (H'C).
18	IM002	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HHLH (H'D).
17	IM001	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [3:0] = HHHL (H'E).
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

[When reading]
0: Interrupts are accepted
1: Interrupts are masked
[When writing]
0: Invalid
1: Interrupts are masked
Initial value: 0

Bit	Bit Name	Initial Value	R/W	Description	
15	IM115	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LLLL (H'0).	[When reading] 0: Interrupts are accepted 1: Interrupts are masked [When writing] 0: Invalid 1: Interrupts are masked
14	IM114	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LLLH (H'1).	
13	IM113	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LLHL (H'2).	
12	IM112	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LLHH (H'3).	
11	IM111	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LHLL (H'4).	
10	IM110	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LHLH (H'5).	
9	IM109	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LHHL (H'6).	
8	IM108	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = LHHH (H'7).	
7	IM107	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HLLL (H'8).	
6	IM106	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HLLH (H'9).	
5	IM105	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HLHL (H'A).	
4	IM104	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HLHH (H'B).	

Bit	Bit Name	Initial Value	R/W	Description	Bit
3	IM103	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HHLL (H'C).	[When reading] 0: Interrupts are accepted 1: Interrupts are masked [When writing] 0: Invalid 1: Interrupts are masked
2	IM102	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HHLH (H'D).	
1	IM101	0	R/W	Sets masking of an interrupt request when $\overline{IRL}$ [7:4] = HHHL (H'E).	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.	

9.3.8 Interrupt Mask Clear Register 0 (INTMSKCLR0)

INTMSKCLR0 is 32-bit write-only registers that clear the mask settings for IRQn (n = 0 to 7) interrupt requests. An undefined value is read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC00	IC01	IC02	IC03	IC04	IC05	IC06	IC07	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description	
31	IC00	0	R/W	Clears masking of an independent interrupt request of IRQ0.	[When reading] An undefined value is read. [When writing] 0: Invalid 1: Clears the corresponding interrupt mask (Interrupts are enabled)
30	IC01	0	R/W	Clears masking of an independent interrupt request of IRQ1.	
29	IC02	0	R/W	Clears masking of an independent interrupt request of IRQ2.	
28	IC03	0	R/W	Clears masking of an independent interrupt request of IRQ3.	
27	IC04	0	R/W	Clears masking of an independent interrupt request of IRQ4.	
26	IC05	0	R/W	Clears masking of an independent interrupt request of IRQ5.	
25	IC06	0	R/W	Clears masking of an independent interrupt request of IRQ6.	
24	IC07	0	R/W	Clears masking of an independent interrupt request of IRQ7.	
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	

9.3.9 Interrupt mask clear register 1 (INTMSKCLR1)

INTMSKCLR1 is 32-bit write-only registers that clear the mask settings for IRL interrupt requests. An undefined value is read.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31	IC10	0	R/W	Clears masking of IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ interrupt requests when $\overline{\text{IRL}}$[3:0] are encoded interrupt input. [When reading] An undefined value is read. [When writing] 0: Invalid 1: Clears the corresponding interrupt mask (Interrupts are enabled)
30	IC11	0	R/W	Clears masking of IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ interrupt requests when $\overline{\text{IRL}}$[7:4] are encoded interrupt input.
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.10 Interrupt mask clear register 2 (INTMSKCLR2)

INTMSKCLR2 is 32-bit write-only registers that clear the mask settings for IRL interrupt requests. An undefined value is read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC015	IC014	IC013	IC012	IC011	IC010	IC009	IC008	IC007	IC006	IC005	IC004	IC003	IC002	IC001	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC115	IC114	IC113	IC112	IC111	IC110	IC109	IC108	IC107	IC106	IC105	IC104	IC103	IC102	IC101	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	IC015	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LLLL (H'0).
30	IC014	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LLLH (H'1).
29	IC013	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LLHL (H'2).
28	IC012	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LLHH (H'3).
27	IC011	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LHLL (H'4).
26	IC010	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LHLH (H'5).
25	IC009	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LHHL (H'6).
24	IC008	0	R/W	Clears masking of an interrupt request when $\overline{\text{IRL}}$[3:0] = LHHH (H'7).

[When reading]

An undefined value is read.

[When writing]

0: Invalid

1: Clears the corresponding interrupt mask (Interrupts are enabled)

Bit	Bit Name	Initial Value	R/W	Description	
23	IC007	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HLLL (H'8).	[When reading] An undefined value is read. [When writing] 0: Invalid 1: Clears the corresponding interrupt mask (Interrupts are enabled)
22	IC006	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HLLH (H'9).	
21	IC005	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HLHL (H'A).	
20	IC004	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HLHH (H'B).	
19	IC003	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HHLL (H'C).	
18	IC002	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HHLH (H'D).	
17	IC001	0	R/W	Clears masking of an interrupt request when $\overline{IRL[3:0]}$ = HHHL (H'E).	
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.	
15	IC115	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LLLL (H'0).	[When reading] An undefined value is read. [When writing] 0: Invalid 1: Clears the corresponding interrupt mask (Interrupts are enabled)
14	IC114	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LLLH (H'1).	
13	IC113	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LLHL (H'2).	
12	IC112	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LLHH (H'3).	

Bit	Bit Name	Initial Value	R/W	Description	
11	IC111	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LHLL (H'4).	[When reading] An undefined value is read. [When writing] 0: Invalid 1: Clears the corresponding interrupt mask (Interrupts are enabled)
10	IC110	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LHLH (H'5).	
9	IC109	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LHHL (H'6).	
8	IC108	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = LHHH (H'7).	
7	IC107	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HLLL (H'8).	
6	IC106	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HLLH (H'9).	
5	IC105	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HLHL (H'A).	
4	IC104	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HLHH (H'B).	
3	IC103	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HHLL (H'C).	
2	IC102	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HHLH (H'D).	
1	IC101	0	R/W	Clears masking of an interrupt request when $\overline{IRL[7:4]}$ = HHHL (H'E).	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.	

9.3.11 NMI Flag Control Register (NMIFCR)

NMIFCR is a 32-bit readable and partially writable with conditions register that has an NMI flag (NMIFL bit) that can be read or cleared by software. The NMIFL bit is automatically set to 1 by hardware when an NMI interrupt is detected by the INTC. To clear the NMIFL bit, write 0 to the bit by software.

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31	NMIL	Undefined	R	NMI Input Level Indicates the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: Low level is input to the NMI pin 1: High level is input to the NMI pin
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
16	NMIFL	0	R/W	NMI Interrupt Request Signal Detection Indicates whether an NMI interrupt request signal has been detected. This bit is automatically set to 1 when the INTC detects an NMI interrupt request. Write 0 to clear the bit. Writing 1 is ignored. [When reading] 1: NMI is detected 0: NMI is not detected [When writing] 0: The NMI flag is cleared 1: Writing 1 is ignored
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.12 User Interrupt Mask Level Register (USERIMASK)

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

Interrupts whose priority levels are lower than the level set in the UIMASK bit are masked. If the value of H'F is set to the UIMASK bit, all interrupts other than the NMI are masked.

Interrupts whose priority levels are higher than the level set in the UIMASK bit are accepted under the following conditions:

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

Even if interrupts are accepted, the UIMASK value is not changed.

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

To prevent incorrect writing, this register can only be written to with bits 31 to 24 set to H'A5.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	H'00	R/W	To write a value to bits 7 to 4, write H'A5 to them. These bits are always read as 0.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 4	UIMASK	H'0	R/W	Interrupt Mask Level Masks interrupts whose priority levels are lower than the level set in the UIMASK bit.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Procedure for Using User Interrupt Mask Level Register

This function is used to save time by disabling interrupts whose priorities are low when a high priority interrupt is processed in the device driver.

Setting the interrupt mask level in USERIMASK disables interrupts having an equal or lower priority level than the specified mask level. This function can disable less-urgent interrupts in a task (such as device driver) operating in user mode to accelerate urgent processing.

USERIMASK is allocated to a different 64-Kbyte page than where the other INTC registers are allocated. When accessing this register in user mode, translate the address through the MMU. In the system that uses a multitasking OS, processes that can access USERIMASK must be controlled by using memory protection functions of the MMU. When terminating the task or switching to another task, be sure to clear USERIMASK to 0 before quitting the task. If the UIMASK bits are left set to a non-zero value, interrupts which are not higher in priority than the

UIMASK level are held disabled, and correct operation may not be performed (for example, the OS cannot switch tasks).

An example of the usage procedure is shown below.

1. Classify interrupts to A and B as described below and set the A priority higher than the B priority.
   A. Interrupts to be accepted in the device driver (interrupts to be used by the operating system: a timer interrupt etc.)
   B. Interrupts to be disabled in the device driver
2. Make the MMU settings so that the address space including USERIMASK can only be accessed by the device driver in which interrupts should be disabled.
3. Branch to the device driver.
4. Set the UIMASK bit to mask B interrupts in the device driver that is operating in user mode.
5. Process interrupts with high priority in the device driver.
6. Clear the UIMASK bit to 0 to return from processing in the device driver.

9.3.13 On-chip module Interrupt Priority Registers (INT2PRI0 to INT2PRI13)

INT2PRI0 to INT2PRI13 are 32-bit readable/writable registers that set priorities (levels 31 to 0) of the on-chip peripheral module interrupts. INT2PRI0 to INT2PRI13 are initialized to H'0000 0000 by a reset.

INT2PRI0 to INT2PRI13 can set 30 priority levels (32 types of interrupt requests) to individual interrupt sources with five bits (interrupt requests are masked at H'00 and H'01).

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

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

Table 9.5 shows the correspondence between interrupt request sources and bits in INT2PRI0 to INT2PRI13.

Table 9.5 Interrupt Request Sources and INT2PRI0 to INT2PRI13

	Bit			
Register	**28 to 24**	**20 to 16**	**12 to 8**	**4 to 0**
INT2PRI0	TMU0 (TUNI0)	TMU0 (TUNI1)	TMU0 (TUNI2)	TMU0 (TICPI2)
INT2PRI1	TMU1 (TUNI3)	TMU1 (TUNI4)	TMU1 (TUNI5)	RTC
INT2PRI2	SCIF0	SCIF1	WDT	Reserved
INT2PRI3	H-UDI	DMAC	ADC	Reserved
INT2PRI4	CMT	HAC	PCIC0	PCIC1
INT2PRI5	PCIC2	PCIC3	PCIC4	PCIC5
INT2PRI6	SIOF0	USBF	MMCIF	SSI0
INT2PRI7	SCIF2	GPIO	Reserved	Reserved
INT2PRI8	SSI3	SSI2	SSI1	SECURITY*
INT2PRI9	LCDC	Reserved	IIC1	IIC0
INT2PRI10	TPU	SIM	SIOF2	SIOF1
INT2PRI11	PCC	Reserved	Reserved	Reserved
INT2PRI12	Reserved	Reserved	USBH	GETHER
INT2PRI13	Reserved	Reserved	STIF1	STIF0

Note: If the value is larger, the priority is higher. Interrupt requests are masked at H'00 and H'01.
* These bits are reserved in the R5S77631.

9.3.14 Interrupt Source Register 0 (Mask State is not affected) (INT2A0)

INT2A0 (mask state is not affected) is a 32-bit read-only register that indicates interrupt source modules. Even if interrupt masking is set in the interrupt mask register, INT2A0 indicates a source module in a corresponding bit (the corresponding interrupt is not generated). If source indication is not necessary depending on the state of the interrupt mask register, use INT2A1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SSIO	MMCIF	—	SIOF0	PCIC5	PCIC4	PCIC3	PCIC2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCIC1	PCIC0	HAC	CMT	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	RTC	TMU1	TMU0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	These bits are always read as 0. The write value should always be 0.	Indicates interrupt sources for each peripheral module (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A0 is not necessary.
25	GPIO	0	R	Indicates GPIO interrupt source	
24	—	0	R	This bit is always read as 0. The write value should always be 0.	
23	SSI0	0	R	Indicates SSI0 interrupt source	
22	MMCIF	0	R	Indicates MMCIF interrupt source	
21	—	0	R	This bit is always read as 0. The write value should always be 0.	
20	SIOF0	0	R	Indicates SIOF0 interrupt source	
19	PCIC5	0	R	Indicates PCIC5 interrupt source	
18	PCIC4	0	R	Indicates PCIC4 interrupt source	
17	PCIC3	0	R	Indicates PCIC3 interrupt source	
16	PCIC2	0	R	Indicates PCIC2 interrupt source	
15	PCIC1	0	R	Indicates PCIC1 interrupt source	
14	PCIC0	0	R	Indicates PCIC0 interrupt source	
13	HAC	0	R	Indicates HAC interrupt source	

Bit	Bit Name	Initial Value	R/W	Function	Description
12	CMT	0	R	Indicates CMT interrupt source	Indicates interrupt sources for each peripheral module (INT2A0 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A0 is not necessary.
11 to 9	—	All 0	R	This bit is always read as 0. The write value should always be 0.	
8	DMAC	0	R	Indicates DMAC interrupt source	
7	H-UDI	0	R	Indicates H-UDI interrupt source	
6	—	0	R	This bit is always read as 0. The write value should always be 0.	
5	WDT	0	R	Indicates WDT interrupt source	
4	SCIF1	0	R	Indicates SCIF1 interrupt source	
3	SCIF0	0	R	Indicates SCIF0 interrupt source	
2	RTC	0	R	Indicates RTC interrupt source	
1	TMU1	0	R	Indicates TMU1 interrupt source	
0	TMU0	0	R	Indicates TMU0 interrupt source	

9.3.15 Interrupt Source Register 01 (Mask State is not affected) (INT2A01)

INT2A01 (mask state is not affected) is a 32-bit read-only register that indicates interrupt source modules. Even if interrupt masking is set in the interrupt mask register, INT2A01 indicates a source module in a corresponding bit (the corresponding interrupt is not generated). If source indication is not necessary depending on the state of the interrupt mask register, use INT2A11.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	USBF	—	—	STIF1	STIF0	—	—	USBH	GETHER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCC	—	—	ADC	TPU	SIM	SIOF2	SIOF1	LCDC	—	IIC1	IIC0	SSI3	SSI2	SSI1	SECURITY*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * This bit is reserved in the R5S77631.

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	This bit is always read as 0. The write value should always be 0.	Indicates interrupt sources for each peripheral module (INT2A01 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A01 is not necessary.
25	SCIF2	0	R	Indicates SCIF2 interrupt source	
24	USBF	0	R	Indicates USBF interrupt source	
23, 22	—	All 0	R	These bits are always read as 0. The write value should always be 0.	
21	STIF1	0	R	Indicates STIF1 interrupt source	
20	STIF0	0	R	Indicates STIF0 interrupt source	
19, 18	—	All 0	R	Undefined values are read from these bits. The write value should always be 0	
17	USBH	0	R	Indicates USBH interrupt source	
16	GETHER	0	R	Indicates GETHER interrupt source	
15	PCC	0	R	Indicates PCC interrupt source	
14	—	0	R	This bit is always read as 0. The write value should always be 0.	
13	—	0	R	Undefined value is read from this bit. The write value should always be 0.	
12	ADC	0	R	Indicates ADC interrupt source	
11	TPU	0	R	Indicates TPU interrupt source	
10	SIM	0	R	Indicates SIM interrupt source	

Bit	Bit Name	Initial Value	R/W	Function	Description
9	SIOF2	0	R	Indicates SIOF2 interrupt source	Indicates interrupt sources for each peripheral module (INT2A01 is not affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A01 is not necessary.
8	SIOF1	0	R	Indicates SIOF1 interrupt source	
7	LCDC	0	R	Indicates LCDC interrupt source	
6	—	0	R	This bit is always read as 0. The write value should always be 0.	
5	IIC1	0	R	Indicates IIC1 interrupt source	
4	IIC0	0	R	Indicates IIC0 interrupt source	
3	SSI3	0	R	Indicates SSI3 interrupt source	
2	SSI2	0	R	Indicates SSI2 interrupt source	
1	SSI1	0	R	Indicates SSI1 interrupt source	
0	SECURITY*	0	R	Indicates SECURITY interrupt source	

Note: * This bit is reserved in the R5S77631.

9.3.16 Interrupt Source Register (Mask State is affected) (INT2A1)

INT2A1 (mask state is affected) is a 32-bit read-only register that indicates interrupt source modules. Note that if interrupt masking is set in the interrupt mask register, INT2A1 does not indicate a source module in a corresponding bit. To check whether interrupts are generated, regardless of the state of the interrupt mask register, use INT2A0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SSI0	MMCIF	—	SIOF0	PCIC5	PCIC4	PCIC3	PCIC2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCIC1	PCIC0	HAC	CMT	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	RTC	TMU1	TMU0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	These bits are always read as 0. The write value should always be 0.	Indicates interrupt sources for each peripheral module (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A1 is not necessary.
25	GPIO	0	R	Indicates GPIO interrupt source	
24	—	0	R	This bit is always read as 0. The write value should always be 0.	
23	SSI0	0	R	Indicates SSI0 interrupt source	
22	MMCIF	0	R	Indicates MMCIF interrupt source	
21	—	0	R	This bit is always read as 0. The write value should always be 0.	
20	SIOF0	0	R	Indicates SIOF0 interrupt source	
19	PCIC5	0	R	Indicates PCIC5 interrupt source	
18	PCIC4	0	R	Indicates PCIC4 interrupt source	

Bit	Bit Name	Initial Value	R/W	Function	Description
17	PCIC3	0	R	Indicates PCIC3 interrupt source	Indicates interrupt sources for each peripheral module (INT2A1 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A1 is not necessary.
16	PCIC2	0	R	Indicates PCIC2 interrupt source	
15	PCIC1	0	R	Indicates PCIC1 interrupt source	
14	PCIC0	0	R	Indicates PCIC0 interrupt source	
13	HAC	0	R	Indicates HAC interrupt source	
12	CMT	0	R	Indicates CMT interrupt source	
11 to 9	—	All 0	R	These bits are always read as 0. The write value should always be 0.	
8	DMAC	0	R	Indicates DMAC interrupt source	
7	H-UDI	0	R	Indicates H-UDI interrupt source	
6	—	0	R	This bit is always read as 0. The write value should always be 0.	
5	WDT	0	R	Indicates WDT interrupt source	
4	SCIF1	0	R	Indicates SCIF1 interrupt source	
3	SCIF0	0	R	Indicates SCIF0 interrupt source	
2	RTC	0	R	Indicates RTC interrupt source	
1	TMU1	0	R	Indicates TMU1 interrupt source	
0	TMU0	0	R	Indicates TMU0 interrupt source	

9.3.17 Interrupt Source Register 11 (Mask State is affected) (INT2A11)

INT2A11 (mask state is affected) is a 32-bit read-only register that indicates interrupt source modules. Note that if interrupt masking is set in the interrupt mask register, INT2A11 does not indicate a source module in a corresponding bit. To check whether interrupts are generated, regardless of the state of the interrupt mask register, use INT2A01.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	USBF	—	—	STIF1	STIF0	—	—	USBH	GETHER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCC	—	—	ADC	TPU	SIM	SIOF2	SIOF1	LCDC	—	IIC1	IIC0	SSI3	SSI2	SSI1	SECURITY*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * This bit is reserved in the R5S77631.

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	These bits are always read as 0. The write value should always be 0.	Indicates interrupt sources for each peripheral module (INT2A11 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A11 is not necessary.
25	SCIF2	0	R	Indicates SCIF2 interrupt source	
24	USBF	0	R	Indicates USBF interrupt source	
23, 22	—	All 0	R	These bits are always read as 0. The write value should always be 0.	
21	STIF1	0	R	Indicates STIF1 interrupt source	
20	STIF0	0	R	Indicates STIF0 interrupt source	
19, 18	—	All 0	R	These bits are always read as 0. The write value should always be 0.	

Bit	Bit Name	Initial Value	R/W	Function	Description
17	USBH	0	R	Indicates USBH interrupt source	Indicates interrupt sources for each peripheral module (INT2A11 is affected by the state of the interrupt mask register). 0: No interrupts 1: Interrupts are generated Note: Reading the INTEVT code notified to the CPU directly can identify interrupt sources. In this case, reading INT2A11 is not necessary.
16	GETHER	0	R	Indicates GETHER interrupt source	
15	PCC	0	R	Indicates PCC interrupt source	
14, 13	—	All 0	R	These bits are always read as 0. The write value should always be 0.	
12	ADC	0	R	Indicates ADC interrupt source	
11	TPU	0	R	Indicates TPU interrupt source	
10	SIM	0	R	Indicates SIM interrupt source	
9	SIOF2	0	R	Indicates SIOF2 interrupt source	
8	SIOF1	0	R	Indicates SIOF1 interrupt source	
7	LCDC	0	R	Indicates LCDC interrupt source	
6	—	0	R	This bit is always read as 0. The write value should always be 0.	
5	IIC1	0	R	Indicates IIC1 interrupt source	
4	IIC0	0	R	Indicates IIC0 interrupt source	
3	SSI3	0	R	Indicates SSI3 interrupt source	
2	SSI2	0	R	Indicates SSI2 interrupt source	
1	SSI1	0	R	Indicates SSI1 interrupt source	
0	SECURITY*	0	R	Indicates SECURITY interrupt source	

Note: * This bit is reserved in the R5S77631.

9.3.18 Interrupt Mask Register (INT2MSKR)

INT2MSKR is a 32-bit readable/writable register that sets masking for each source indicated in the interrupt source register. Interrupts whose corresponding bits in INT2MSKRG are set to 1 are not notified to the CPU.

INT2MSKR is initialized to H'FFFF FFFF (mask state) by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SSI0	MMCIF	—	SIOF0	PCIC5	PCIC4	PCIC3	PCIC2
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCIC1	PCIC0	HAC	CMT	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	RTC	TMU1	TMU0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 1	R	These bits are always read as 1. The write value should always be 1.	Masks interrupts for each peripheral module. [When writing] 0: Invalid 1: Interrupts are masked [When reading] 0: No mask setting 1: Mask setting
25	GPIO	1	R/W	Masks GPIO interrupts	
24	—	1	R	This bit is always read as 1. The write value should always be 1.	
23	SSI0	1	R/W	Masks SSI0 interrupts	
22	MMCIF	1	R/W	Masks MMCIF interrupts	
21	—	1	R	This bit is always read as 1. The write value should always be 1.	
20	SIOF0	1	R/W	Masks SIOF0 interrupts	
19	PCIC5	1	R/W	Masks PCIC5 interrupts	
18	PCIC4	1	R/W	Masks PCIC4 interrupts	
17	PCIC3	1	R/W	Masks PCIC3 interrupts	
16	PCIC2	1	R/W	Masks PCIC2 interrupts	
15	PCIC1	1	R/W	Masks PCIC1 interrupts	
14	PCIC0	1	R/W	Masks PCIC0 interrupts	

Bit	Bit Name	Initial Value	R/W	Function	Description
13	HAC	1	R/W	Masks HAC interrupts	Masks interrupts for each peripheral module. [When writing] 0: Invalid 1: Interrupts are masked [When reading] 0: No mask setting 1: Mask setting
12	CMT	1	R/W	Masks CMT interrupts	
11 to 9	—	All 1	R	These bits are always read as 1. The write value should always be 1.	
8	DMAC	1	R/W	Masks DMAC interrupts	
7	H-UDI	1	R/W	Masks H-UDI interrupts	
6	—	1	R	This bit is always read as 1. The write value should always be 1.	
5	WDT	1	R/W	Masks WDT interrupts	
4	SCIF1	1	R/W	Masks SCIF1 interrupts	
3	SCIF0	1	R/W	Masks SCIF0 interrupts	
2	RTC	1	R/W	Masks RTC interrupts	
1	TMU1	1	R/W	Masks TMU1 interrupts	
0	TMU0	1	R/W	Masks TMU0 interrupts	

9.3.19 Interrupt Mask Register 1 (INT2MSKR1)

INT2MSKR1 is a 32-bit readable/writable register that sets masking for each source indicated in the interrupt source register. Interrupts whose corresponding bits in INT2MSKR1 are set to 1 are not notified to the CPU.

INT2MSK1 is initialized to H'FFFF FFFF (mask state) by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	USBF	—	—	STIF1	STIF0	—	—	USBH	GETHER
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCC	—	—	ADC	TPU	SIM	SIOF2	SIOF1	LCDC	—	IIC1	IIC0	SSI3	SSI2	SSI1	SECURITY*
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * This bit is reserved in the R5S77631.

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 1	R	These bits are always read as 1. The write value should always be 1.	Masks interrupts for each peripheral module. [When writing] 0: Invalid 1: Interrupts are masked [When reading] 0: No mask setting 1: Mask setting
25	SCIF2	1	R/W	Masks SCIF2 interrupts	
24	USBF	1	R/W	Masks USBF interrupts	
23, 22	—	All 1	R	These bits are always read as 1. The write value should always be 1.	
21	STIF1	1	R/W	Masks STIF1 interrupts	
20	STIF0	1	R/W	Masks STIF0 interrupts	
19, 18	—	All 1	R	These bits are always read as 1. The write value should always be 1.	
17	USBH	1	R/W	Masks USBH interrupts	
16	GETHER	1	R/W	Masks GETHER interrupts	
15	PCC	1	R/W	Masks PCC interrupts	
14, 13	—	All 1	R	These bits are always read as 1. The write value should always be 1.	
12	ADC	1	R/W	Masks ADC interrupts	
11	TPU	1	R/W	Masks TPU interrupts	
10	SIM	1	R/W	Masks SIM interrupts	
9	SIOF2	1	R/W	Masks SIOF2 interrupts	
8	SIOF1	1	R/W	Masks SIOF1 interrupts	
7	LCDC	1	R/W	Masks LCDC interrupts	
6	—	1	R	This bit is always read as 1. The write value should always be 1.	
5	IIC1	1	R/W	Masks IIC1 interrupts	
4	IIC0	1	R/W	Masks IIC0 interrupts	
3	SSI3	1	R/W	Masks SSI3 interrupts	

Bit	Bit Name	Initial Value	R/W	Function	Description
2	SSI2	1	R/W	Masks SSI2 interrupts	Masks interrupts for each peripheral module. [When writing] 0: Invalid 1: Interrupts are masked [When reading] 0: No mask setting 1: Mask setting
1	SSI1	1	R/W	Masks SSI1 interrupts	
0	SECURITY*	1	R/W	Masks SECURITY interrupts	

Note: * This bit is reserved in the R5S77631.

9.3.20 Interrupt Mask Clear Register (INT2MSKCR)

INT2MSKCR is a 32-bit write-only register that clears any masking set in the interrupt mask register. Setting bits in this register to 1 clears the masking of the corresponding interrupt sources. Reading bits in this register is always 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SSI0	MMC	—	SIOF0	PCIC5	PCIC4	PCIC3	PCIC2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCIC1	PCIC0	HAC	CMT	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	RTC	TMU1	TMU0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	These bits are always read as 0. The write value should always be 0	Clears interrupt masking for each peripheral module. [When writing] 0: Invalid 1: Interrupt mask is cleared [When reading] Always 0
25	GPIO	0	R/W	Clears GPIO interrupt masking	
24	—	0	R	This bit is always read as 0. The write value should always be 0	
23	SSI0	0	R/W	Clears SSI0 interrupt masking	
22	MMC	0	R/W	Clears MMC interrupt masking	
21	—	0	R	This bit is always read as 0. The write value should always be 0	
20	SIOF0	0	R/W	Clears SIOF0 interrupt masking	
19	PCIC5	0	R/W	Clears PCIC5 interrupt masking	
18	PCIC4	0	R/W	Clears PCIC4 interrupt masking	
17	PCIC3	0	R/W	Clears PCIC3 interrupt masking	
16	PCIC2	0	R/W	Clears PCIC2 interrupt masking	
15	PCIC1	0	R/W	Clears PCIC1 interrupt masking	
14	PCIC0	0	R/W	Clears PCIC0 interrupt masking	
13	HAC	0	R/W	Clears HAC interrupt masking	
12	CMT	0	R/W	Clears CMT interrupt masking	
11 to 9	—	All 0	R	These bits are always read as 0. The write value should always be 0	
8	DMAC	0	R/W	Clears DMAC interrupt masking	

Bit	Bit Name	Initial Value	R/W	Function	Description
7	H-UDI	0	R/W	Clears H-UDI interrupt masking	Clears interrupt masking for each peripheral module. [When writing] 0: Invalid 1: Interrupt mask is cleared [When reading] Always 0
6	—	0	R	This bit is always read as 0. The write value should always be 0	
5	WDT	0	R/W	Clears WDT interrupt masking	
4	SCIF1	0	R/W	Clears SCIF1 interrupt masking	
3	SCIF0	0	R/W	Clears SCIF0 interrupt masking	
2	RTC	0	R/W	Clears RTC interrupt masking	
1	TMU1	0	R/W	Clears TMU1 interrupt masking	
0	TMU0	0	R/W	Clears TMU0 interrupt masking	

9.3.21 Interrupt Mask Clear Register 1 (INT2MSKCR1)

INT2MSKCR1 is a 32-bit write-only register that clears any masking set in the interrupt mask register. Setting bits in this register to 1 clears the masking of the corresponding interrupt sources. Reading bits in this register is always 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	USBF	—	—	STIF1	STIF0	—	—	USBH	GETHER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCC	—	—	ADC	TPU	SIM	SIOF2	SIOF1	LCDC	—	IIC1	IIC0	SSI3	SSI2	SSI1	SECURITY*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * This bit is reserved in the R5S77631.

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 26	—	All 0	R	These bits are always read as 0. The write value should always be 0	Clears interrupt masking for each peripheral module. [When writing] 0: Invalid 1: Interrupt mask is cleared [When reading] Always 0
25	SCIF2	0	R/W	Clears SCIF2 interrupt masking	
24	USBF	0	R/W	Clears USBF interrupt masking	
23, 22	—	All 0	R	These bits are always read as 0. The write value should always be 0	
21	STIF1	0	R/W	Clears STIF1 interrupt masking	
20	STIF0	0	R/W	Clears STIF0 interrupt masking	
19, 18	—	All 0	R	These bits are always read as 0. The write value should always be 0	
17	USBH	0	R/W	Clears USBH interrupt masking	
16	GETHER	0	R/W	Clears GETHER interrupt masking	
15	PCC	0	R/W	Clears PCC interrupt masking	
14, 13	—	All 0	R	These bits are always read as 0. The write value should always be 0	
12	ADC	0	R/W	Clears ADC interrupt masking	
11	TPU	0	R/W	Clears TPU interrupt masking	
10	SIM	0	R/W	Clears SIM interrupt masking	
9	SIOF2	0	R/W	Clears SIOF2 interrupt masking	
8	SIOF1	0	R/W	Clears SIOF1 interrupt masking	
7	LCDC	0	R/W	Clears LCDC interrupt masking	
6	—	0	R	This bit is always read as 0. The write value should always be 0	
5	IIC1	0	R/W	Clears IIC1 interrupt masking	
4	IIC0	0	R/W	Clears IIC0 interrupt masking	
3	SSI3	0	R/W	Clears SSI3 interrupt masking	
2	SSI2	0	R/W	Clears SSI2 interrupt masking	

Bit	Bit Name	Initial Value	R/W	Function	Description
1	SSI1	0	R/W	Clears SSI1 interrupt masking	Clears interrupt masking for each peripheral module. [When writing] 0: Invalid 1: Interrupt mask is cleared [When reading] Always 0
0	SECURITY*	0	R/W	Clears SECURITY interrupt masking	

Note: * This bit is reserved in the R5S77631.

9.3.22 On-chip Module Interrupt Source Registers (INT2B0 to INT2B7 and INT2B9 to INT2B11)

INT2B0 to INT2B7 and INT2B9 to INT2B11 are 32-bit read-only registers that indicate detailed sources for interrupt source modules indicated in the interrupt source register. INT2B0 to INT2B7 and INT2B9 to INT2B11 are not affected by the mask state of the interrupt mask register. When mask setting is made for individual detailed sources, set the interrupt mask register or interrupt enable register in the corresponding modules.

The initial value of these registers is undefined (reserve bit is always read as 0).

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

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

INT2B0: Indicates detailed interrupt sources for the TMU.

Module	Bit	Source	Function	Description
TMU	31 to 7	—	These bits are always read as 0. The write value should always be 0.	Indicates TMU interrupt sources. This register indicates the TMU interrupt sources even if mask setting is made in the interrupt mask register for them.
	6	TUNI5	TMU channel 5 underflow interrupt	
	5	TUNI4	TMU channel 4 underflow interrupt	
	4	TUNI3	TMU channel 3 underflow interrupt	
	3	TICPI2	TMU channel 2 input capture interrupt	
	2	TUNI2	TMU channel 2 underflow interrupt	
	1	TUNI1	TMU channel 1 underflow interrupt	
	0	TUNI0	TMU channel 0 underflow interrupt	

INT2B1: Indicates detailed interrupt sources for the RTC.

Module	Bit	Source	Function	Description
RTC	31 to 3	—	These bits are always read as 0. The write value should always be 0.	Indicates RTC interrupt sources. This register indicates the RTC interrupt sources even if mask setting is made in the interrupt mask register for them.
	2	CUI	RTC carry interrupt	
	1	PRI	RTC period interrupt	
	0	ATI	RTC alarm interrupt	

INT2B2: Indicates detailed interrupt sources for the SCIF.

Module	Bit	Source	Function	Description
SCIF1	31 to 8	—	These bits are always read as 0. The write value should always be 0.	Indicates SCIF interrupt sources. This register indicates the SCIF interrupt sources even if mask setting is made in the interrupt mask register for them.
	7	TXI1	SCIF channel 1 transmit FIFO data empty interrupt	
	6	BRI1	SCIF channel 1 break interrupt or overrun error interrupt	
	5	RXI1	SCIF channel 1 receive FIFO data full interrupt or receive data ready interrupt	
	4	ERI1	SCIF channel 1 receive error interrupt	

Module	Bit	Source	Function	Description
SCIF0	3	TXI0	SCIF channel 0 transmit FIFO data empty interrupt	Indicates SCIF interrupt sources. This register indicates the SCIF interrupt sources even if mask setting is made in the interrupt mask register for them.
	2	BRI0	SCIF channel 0 break interrupt or overrun error interrupt	
	1	RXI0	SCIF channel 0 receive FIFO data full interrupt or receive data ready interrupt	
	0	ERI0	SCIF channel 0 receive error interrupt	

INT2B3: Indicates detailed interrupt sources for the DMAC.

Module	Bit	Source	Function	Description
DMAC	31 to 13	—	These bits are always read as 0. The write value should always be 0.	Indicates DMAC interrupt sources. This register indicates DMAC interrupt sources even if mask setting is made in the interrupt mask register for them.
	12	DMAE	DMA channels 0 to 5 address error interrupt	
	11 to 6	—	These bits are always read as 0. The write value should always be 0.	
	5	DMTE5	Channel 5 DMA transfer end interrupt	
	4	DMTE4	Channel 4 DMA transfer end interrupt	
	3	DMTE3	Channel 3 DMA transfer end interrupt	
	2	DMTE2	Channel 2 DMA transfer end interrupt	
	1	DMTE1	Channel 1 DMA transfer end interrupt	
	0	DMTE0	Channel 0 DMA transfer end interrupt	

INT2B4: Indicates detailed interrupt sources for the PCIC.

Module	Bit	Source	Function	Description
PCIC	31 to 10	—	These bits are always read as 0. The write value should always be 0.	Indicates PCIC interrupt sources. This register indicates the PCIC interrupt sources even if mask setting is made in the interrupt mask register for them.
	9	PWD0	PCIC power state D0 state interrupt	
	8	PWD1	PCIC power state D1 state interrupt	
	7	PWD2	PCIC power state D2 state interrupt	
	6	PWD3	PCIC power state D3 state interrupt	
	5	ERR	PCIC error interrupt	
	4	INTD	PCIC INTD interrupt	
	3	INTC	PCIC INTC interrupt	
	2	INTB	PCIC INTB interrupt	
	1	INTA	PCIC INTA interrupt	
	0	SERR	PCIC SERR interrupt	

INT2B5: Indicates detailed interrupt sources for the MMCIF.

Module	Bit	Source	Function	Description
MMCIF	31 to 4	—	These bits are always read as 0. The write value should always be 0.	Indicates MMC interrupt sources. This register indicates MMC interrupt sources even if mask setting is made in the interrupt mask register for them.
	3	FRDY	FIFO ready interrupt	
	2	ERR	CRC error interrupt, data timeout error interrupt, or command timeout error interrupt	
	1	TRAN	Data response interrupt, data transfer end interrupt, command response receive end interrupt, command transmit end interrupt, or data busy end interrupt	
	0	FSTAT	MMC FIFO empty interrupt or FIFO full interrupt	

INT2B6: Indicates detailed interrupt sources for the SCIF2.

Module	Bit	Source	Function	Description
SCIF2	31 to 4	—	These bits are always read as 0. The write value should always be 0.	Indicates SCIF2 interrupt sources. This register indicates the SCIF2 interrupt sources even if mask setting is made in the interrupt mask register for them.
	3	TXI2	SCIF channel 2 transmit FIFO data empty interrupt	
	2	BRI2	SCIF channel 2 break interrupt or overrun error interrupt	
	1	RXI2	SCIF channel 2 receive FIFO data full interrupt or receive data ready interrupt	
	0	ERI2	SCIF channel 2 receive error interrupt	

INT2B7: Indicates detailed interrupt sources for the GPIO.

Module	Bit	Source	Function	Description
GPIO	31 to 28	—	These bits are always read as 0. The write value should always be 0.	Indicates GPIO interrupt sources. This register indicates GPIO interrupt sources even if mask setting is made in the interrupt mask register for them.
	27	PINT15I	GPIO interrupt from PINT15 pin	
	26	PINT14I	GPIO interrupt from PINT14 pin	
	25	PINT13I	GPIO interrupt from PINT13pin	
	24	PINT12I	GPIO interrupt from PINT12 pin	
	23 to 20	—	These bits are always read as 0. The write value should always be 0.	
	19	PINT11I	GPIO interrupt from PINT11 pin	
	18	PINT10I	GPIO interrupt from PINT10 pin	
	17	PINT9I	GPIO interrupt from PINT9 pin	
	16	PINT8I	GPIO interrupt from PINT8 pin	
	15 to 12	—	These bits are always read as 0. The write value should always be 0.	
	11	PINT7I	GPIO interrupt from PINT7 pin	

Module	Bit	Source	Function	Description
GPIO	10	PINT6I	GPIO interrupt from PINT6 pin	Indicates GPIO interrupt sources. This register indicates GPIO interrupt sources even if mask setting is made in the interrupt mask register for them.
	9	PINT5I	GPIO interrupt from PINT5 pin	
	8	PINT4I	GPIO interrupt from PINT4 pin	
	7 to 4	—	These bits are always read as 0. The write value should always be 0.	
	3	PINT3I	GPIO interrupt from PINT3 pin	
	2	PINT2I	GPIO interrupt from PINT2 pin	
	1	PINT1I	GPIO interrupt from PINT1 pin	
	0	PINT0I	GPIO interrupt from PINT0 pin	

INT2B9: Indicates detailed interrupt sources for the GETHER.

Module	Bit	Source	Function	Description
GETHER	31 to 3	—	These bits are always read as 0. The write value should always be 0.	Indicates GETHER interrupt sources. This register indicates the GETHER interrupt sources even if mask setting is made in the interrupt mask register for them.
	2	GEINT2	GEINT2 interrupt	
	1	GEINT1	GEINT1 interrupt	
	0	GEINT0	GEINT0 interrupt	

INT2B10: Indicates detailed interrupt sources for the USBF.

Module	Bit	Source	Function	Description
USBF	31 to 2	—	These bits are always read as 0. The write value should always be 0.	Indicates USBF interrupt sources. This register indicates the USBF interrupt sources even if mask setting is made in the interrupt mask register for them.
	1	USBFI1	USBFI1 interrupt	
	0	USBFI0	USBFI0 interrupt	

INT2B11: Indicates detailed interrupt sources for the SIM.

Module	Bit	Source	Function	Description
SIM	31 to 4	—	These bits are always read as 0. The write value should always be 0.	Indicates SIM interrupt sources. This register indicates the SIM interrupt sources even if mask setting is made in the interrupt mask register for them.
	3	TEND	Transmit end interrupt	
	2	TXI	Transmit data empty interrupt	
	1	RXI	Receive data full interrupt	
	0	ERI	Transmit/receive error interrupt	

9.3.23 GPIO Interrupt Set Register (INT2GPIC)

INT2GPIC enables interrupt requests input from the following pins: PTB0 to PTB7 and PTM0 to PTM7.

A GPIO interrupt is a low active and level-sense signal. Before enabling an interrupt request, set the corresponding pin as an input with the corresponding port control register (PBCR, PMCR). For the port control registers, see section 40, General Purpose I/O (GPIO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PINT15E	PINT14E	PINT13E	PINT12E	—	—	—	—	PINT11E	PINT10E	PINT9E	PINT8E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PINT7E	PINT6E	PINT5E	PINT4E	—	—	—	—	PINT3E	PINT2E	PINT1E	PINT0E
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Function	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	Enables a GPIO interrupt request for each pin. 0: Disables an interrupt request 1: Enables an interrupt request
27	PINT15E	0	R/W	Enables a GPIO interrupt request from PINT15 pin	
26	PINT14E	0	R/W	Enables a GPIO interrupt request from PINT14 pin	
25	PINT13E	0	R/W	Enables a GPIO interrupt request from PINT13 pin	
24	PINT12E	0	R/W	Enables a GPIO interrupt request from PINT12 pin	
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	
19	PINT11E	0	R/W	Enables a GPIO interrupt request from PINT11 pin	
18	PINT10E	0	R/W	Enables a GPIO interrupt request from PINT10 pin	
17	PINT9E	0	R/W	Enables a GPIO interrupt request from PINT9 pin	
16	PINT8E	0	R/W	Enables a GPIO interrupt request from PINT8 pin	
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	

Bit	Bit Name	Initial Value	R/W	Function	Description
11	PINT7E	0	R/W	Enables a GPIO interrupt request from PINT7 pin	Enables a GPIO interrupt request for each pin. 0: Disables an interrupt request 1: Enables an interrupt request
10	PINT6E	0	R/W	Enables a GPIO interrupt request from PINT6 pin	
9	PINT5E	0	R/W	Enables a GPIO interrupt request from PINT5 pin	
8	PINT4E	0	R/W	Enables a GPIO interrupt request from PINT4 pin	
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	
3	PINT3E	0	R/W	Enables a GPIO interrupt request from PINT3 pin	
2	PINT2E	0	R/W	Enables a GPIO interrupt request from PINT2 pin	
1	PINT1E	0	R/W	Enables a GPIO interrupt request from PINT1 pin	
0	PINT0E	0	R/W	Enables a GPIO interrupt request from PINT0 pin	

When GPIO ports are used as interrupt ports, if the GPIO detects an interrupt, the interrupt is notified to the INTC from the GPIO. However, it is indicated as a one-bit source in the INT2A0 or INT2A1 register of the INTC. Referring to the on-chip module interrupt source register INT2B7. Referring to the INTEVT code in the CPU can specify from which port group an interrupt is generated.

9.4 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, IRL, and on-chip modules. Each interrupt has a priority level (16 to 0), with level 16 as the highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and interrupt requests are ignored.

9.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. It is always accepted unless the BL bit in SR of the CPU is set to 1. In sleep mode, the interrupt is accepted even if the BL bit is set to 1.

A setting can also be made to have the NMI interrupt accepted even if the BL bit is set to 1. Input from the NMI pin is edge-detected. The NMI edge select bit (NMIE) in ICR0 is used to select either rising or falling edge as the detection edge. When the NMIE bit in ICR0 is modified, the NMI interrupt is not detected for a maximum of six bus clock cycles after the modification. If the INTMU bit in CPUOPM is set to 1, the IMASK value in SR is automatically set to 15 by the accepted NMI interrupt. If the INTMU bit in CPUOPM is set to 0, the IMASK value in SR is not affected by the NMI interrupt exception handling.

9.4.2 IRQ Interrupts

IRQ interrupts are available when using the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pin for IRQn (n = 7 to 0) independent interrupt inputs by setting the IRLM0 and IRLM1 bits to 1 in ICR0. The IRQnS1 and IRQnS0 bits in ICR1 are used to select either rising edge, falling edge, low level, or high level detection. A priority level can be set for each input by using the interrupt priority level setting register (INTPRI).

When level detection is selected for IRQ interrupt requests, the IRQ interrupt pin input level should be held until the CPU accepts the interrupt and starts interrupt handling. In level detection mode, after an interrupt request is accepted, the interrupt request held in the detection circuit should be cleared. For the specific clearing procedure, see section 9.7.3, To Clear IRQ and IRL Interrupt Requests.

Note: In level detection mode, once an IRQ interrupt request has been detected, the INTC holds the interrupt request as the interrupt source in INTREQ even if the corresponding IRQ interrupt pin level is changed to cancel the request before the CPU accepts the request. The interrupt source will be held until the CPU accepts any other interrupt request (IRQ or not) or the corresponding interrupt mask bit is set to 1. For details, see section 9.7, Usage Notes.

When the INTMU bit in CPUOPM is 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit is 0, the IMASK value in SR is not affected by the accepted interrupt.

9.4.3 IRL Interrupts

IRL interrupts are input by level at pins IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ or IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$. The priority level is the level indicated by pins IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ or IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$. An IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ or IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ pins input are all low level indicates the highest-level interrupt request (interrupt priority level 15), and all high level indicates no interrupt request (interrupt priority level 0). Figure 9.2 shows an example of IRL interrupt connection, and table 9.6 shows the correspondence between the IRL pins and interrupt levels.

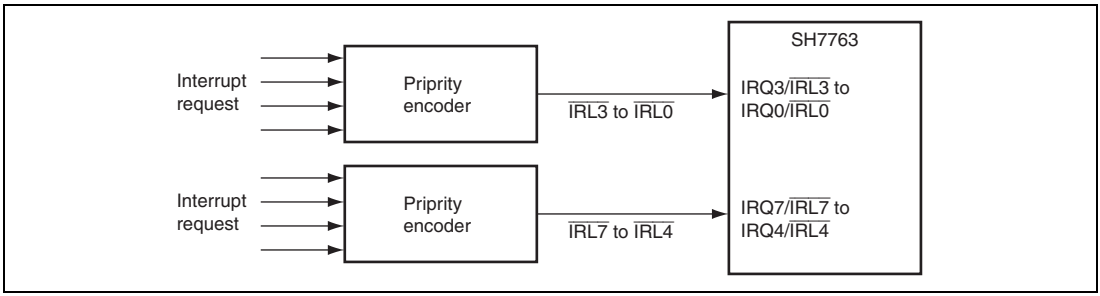


Figure 9.2 Example of IRL Interrupt Connection

Table 9.6 RL[3:0], IRL[7:4] Pins and Interrupt Levels

$\overline{\text{IRL3}}$ or $\overline{\text{IRL7}}$	$\overline{\text{IRL2}}$ or $\overline{\text{IRL6}}$	$\overline{\text{IRL1}}$ or $\overline{\text{IRL5}}$	$\overline{\text{IRL0}}$ or $\overline{\text{IRL4}}$	Interrupt Priority Level	Interrupt Request
Low	Low	Low	Low	15	Level 15 Interrupt Request
Low	Low	Low	High	14	Level 14 Interrupt Request
Low	Low	High	Low	13	Level 13 Interrupt Request
Low	Low	High	High	12	Level 12 Interrupt Request
Low	High	Low	Low	11	Level 11 Interrupt Request
Low	High	Low	High	10	Level 10 Interrupt Request
Low	High	High	Low	9	Level 9 Interrupt Request
Low	High	High	High	8	Level 8 Interrupt Request
High	Low	Low	Low	7	Level 7 Interrupt Request
High	Low	Low	High	6	Level 6 Interrupt Request
High	Low	High	Low	5	Level 5 Interrupt Request
High	Low	High	High	4	Level 4 Interrupt Request
High	High	Low	Low	3	Level 3 Interrupt Request
High	High	Low	High	2	Level 2 Interrupt Request
High	High	High	Low	1	Level 1 Interrupt Request
High	High	High	High	0	No Interrupt Request

After an interrupt request is accepted, the interrupt request held in the detection circuit should be cleared. For the specific clearing procedure, see section 9.7.3, To Clear IRQ and IRL Interrupt Requests.

Note: Although there is no interrupt source register for the IRL interrupt requests, once an IRL interrupt request has been detected, the INTC holds the IRL interrupt request with the highest priority level as the interrupt source in the detection circuit even if the corresponding IRL interrupt pin level is changed to cancel the request before the CPU accepts the request. The interrupt source will be held until the CPU accepts any other interrupt request (IRL or not) or the corresponding interrupt mask bit is set to 1. In this case, the IRL interrupt request should be cleared in the interrupt handling routine. For details, see section 9.7, Usage Notes.

When the INTMU bit in CPUOPM is 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit is 0, the IMASK value in SR is not affected by the accepted interrupt.

9.4.4 On-chip Module Interrupts

On-chip module interrupts are interrupts generated by on-chip modules. Not every interrupt source is assigned a different interrupt vector, but sources are reflected in the interrupt event register (INTEVT), so it is easy to identify sources by using the INTEVT value as a branch offset in the exception handling routine.

A priority level from 31 to 0 can be set for each module by means of INT2PRI0 to INT2PRI13. The INTC rounds off the lowest one bit and sends 4-bit code to the CPU. In detail, see section 9.4.5, Interrupt Priority Level of On-chip Module Interrupts.

When the INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level 15 of the accepted NMI interrupt. When the INTMU bit in CPUOPM is cleared to 0, the IMASK value in SR is not affected by the accepted NMI interrupt.

Updating of the interrupt source flag and interrupt enable flag of a peripheral module should only be carried out when the BL bit in SR is set to 1 or while the corresponding interrupt does not occur by setting its mask bit. To prevent erroneous interrupt acceptance from an interrupt source that should have been updated, first read the on-chip peripheral register containing the relevant flag and wait the priority determination time shown in table 9.8, then clear the BL bit to 0. This will secure the necessary timing internally. When updating a number of flags, there is no problem if only the register containing the last flag updated is read from.

If flag updating is performed while the BL bit is cleared to 0, the program may jump to the interrupt handling routine when the INTEVT value is 0. In this case, interrupt processing is initiated due to the timing relationship between the flag update and interrupt request recognition within this LSI. Processing can be continued without any problem by executing an RTE instruction.

9.4.5 Interrupt Priority Level of On-chip Module Interrupts

When an on-chip module interrupt is generated, the INTC outputs its interrupt exception code (INTEVT code) as individual source identification to the CPU. When the CPU accepts an interrupt, the corresponding INTEVT code is indicated in INTEVT. Even if the interrupt source register of the INTC is not read, the interrupt source can be identified by reading INTEVT of the CPU in the interrupt handler. Table 9.1 lists the source of on-chip module interrupt and the interrupt exception codes.

On-chip module interrupt, it can be set individual interrupt sources to 30 (5-bit) priority levels (see figure 9.1). The interrupt level receive interface consists of four bits and there are 15 priority levels (H'0 is interrupt request mask). The INTC consists of five bits in which one bit is extended

and determines the priorities of individual interrupt sources. The lowest one bit is then rounded off, the data is converted to 4-bit data, and the priority levels are notified. For example, two interrupt sources whose priority levels are set to H'1A and H'1B are both output as 4-bit priority level H'D. That is, the two interrupt sources have the same value. However, in terms of the INTEVT code that is notified when a conflict occurs between two interrupt sources, the INTEVT code that corresponds to the interrupt with a priority level of H'1B has priority. This is because the priority level of H'1B is higher than that of H'1A when comparing 5-bit data. When a conflict occurs between interrupts with the same priority level, the INTEVT code is notified according to the priority level shown in table 9.1.

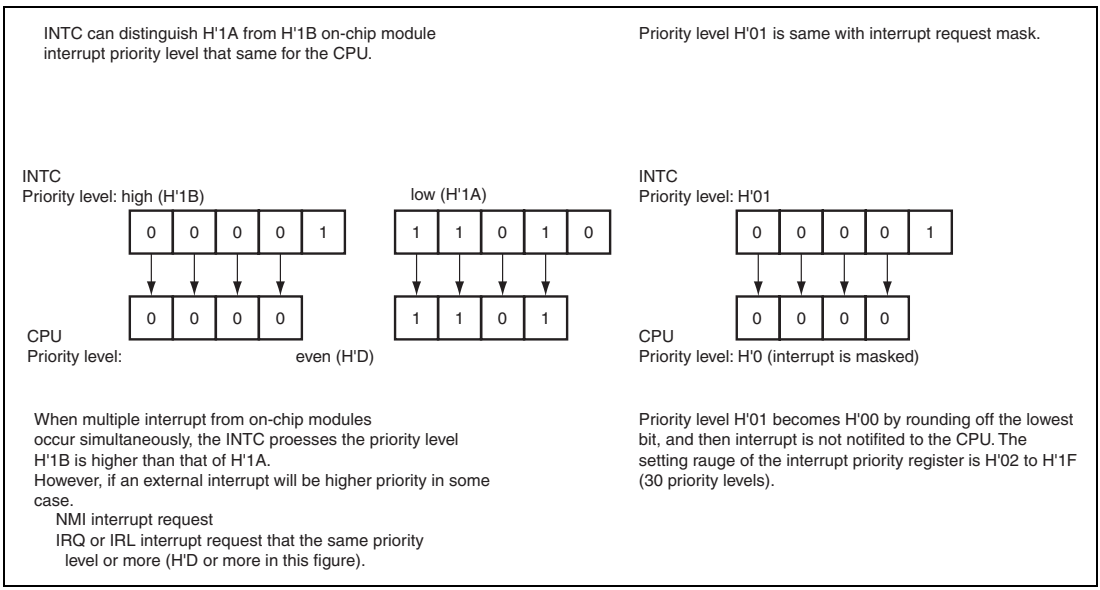


Figure 9.3 On-chip Module Interrupt Priority

9.4.6 Interrupt Exception Handling and Priority

Table 9.7 lists the codes for the interrupt event register (INTEVT), and the order of interrupt priority.

Each interrupt source is assigned a unique INTEVT code. The start address of the exception handling routine is common to each interrupt source. Therefore, to identify the interrupt source, branching is performed at the start of the exception handling routine using the INTEVT value. For instance, the INTEVT value is used as a branch offset.

The priority order of the on-chip modules is specified as desired by setting priority levels from 31 to 0 in INT2PRI0 to INT2PRI14. The priority order of the on-chip modules is set to 0 by a reset.

When the priorities for multiple interrupt sources are set to the same level and such interrupts are generated simultaneously, they are handled according to the default priority order shown in table 9.7.

Updating of INTPRI and INT2PRI0 to INT2PRI14 should only be carried out when the BL bit in SR is set to 1. To prevent erroneous interrupt acceptance, first read one of the interrupt priority level setting registers, then clear the BL bit to 0. This will secure the necessary timing internally.

Table 9.7 Interrupt Exception Handling and Priority

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
NMI	—	H'1C0	16	—	—	—	—	High
IRL L: Low level input H: High level input (See table 9.6)	$\overline{IRL[7:4]}$ = LLLL (H'0)	H'200	15	INTMSK2[15] INTMSKCLR2[15]	—	—		
	$\overline{IRL[3:0]}$ = LLLL (H'0)			INTMSK2[31] INTMSKCLR2[31]	—	—		
	$\overline{IRL[7:4]}$ = LLLH (H'1)	H'220	14	INTMSK2[14] INTMSKCLR2[14]	—	—		
	$\overline{IRL[3:0]}$ = LLLH (H'1)			INTMSK2[30] INTMSKCLR2[30]	—	—		
	$\overline{IRL[7:4]}$ = LLHL (H'2)	H'240	13	INTMSK2[13] INTMSKCLR2[13]	—	—		
	$\overline{IRL[3:0]}$ = LLHL (H'2)			INTMSK2[29] INTMSKCLR2[29]	—	—		
	$\overline{IRL[7:4]}$ = LLHH (H'3)	H'260	12	INTMSK2[12] INTMSKCLR2[12]	—	—		
	$\overline{IRL[3:0]}$ = LLHH (H'3)			INTMSK2[28] INTMSKCLR2[28]	—	—		
	$\overline{IRL[7:4]}$ = LHLL (H'4)	H'280	11	INTMSK2[11] INTMSKCLR2[11]	—	—		
	$\overline{IRL[3:0]}$ = LHLL (H'4)			INTMSK2[27] INTMSKCLR2[27]	—	—		Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
IRL L: Low level input H: High level input (See table 9.6)	$\overline{IRL[7:4]}$ = LHLH (H'5)	H'2A0	10	INTMSK2[10] INTMSKCLR2[10]	—	—		High
	$\overline{IRL[3:0]}$ = LHLH (H'5)			INTMSK2[26] INTMSKCLR2[26]	—	—		
	$\overline{IRL[7:4]}$ = LHHL (H'6)	H'2C0	9	INTMSK2[9] INTMSKCLR2[9]	—	—		
	$\overline{IRL[3:0]}$ = LHHL (H'6)			INTMSK2[25] INTMSKCLR2[25]	—	—		
	$\overline{IRL[7:4]}$ = LHHH (H'7)	H'2E0	8	INTMSK2[8] INTMSKCLR2[8]	—	—		
	$\overline{IRL[3:0]}$ = LHHH (H'7)			INTMSK2[24] INTMSKCLR2[24]	—	—		
	$\overline{IRL[7:4]}$ = HLLL (H'8)	H'300	7	INTMSK2[7] INTMSKCLR2[7]	—	—		
	$\overline{IRL[3:0]}$ = HLLL (H'8)			INTMSK2[23] INTMSKCLR2[23]	—	—		
	$\overline{IRL[7:4]}$ = HLLH (H'9)	H'320	6	INTMSK2[6] INTMSKCLR2[6]	—	—		
	$\overline{IRL[3:0]}$ = HLLH (H'9)			INTMSK2[22] INTMSKCLR2[22]	—	—		
	$\overline{IRL[7:4]}$ = HLHL (H'A)	H'340	5	INTMSK2[5] INTMSKCLR2[5]	—	—		
	$\overline{IRL[3:0]}$ = HLHL (H'A)			INTMSK2[21] INTMSKCLR2[21]	—	—		
	$\overline{IRL[7:4]}$ = HLHH (H'B)	H'360	4	INTMSK2[4] INTMSKCLR2[4]	—	—		
	$\overline{IRL[3:0]}$ = HLHH (H'B)			INTMSK2[20] INTMSKCLR2[20]	—	—		Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
IRL L: Low level input H: High level input (See table 9.6)	$\overline{IRL[7:4]}$ = HHLL (H'C)	H'380	3	INTMSK2[3] INTMSKCLR2[3]	—	—		High
	$\overline{IRL[3:0]}$ = HHLL (H'C)			INTMSK2[19] INTMSKCLR2[19]	—	—		
	$\overline{IRL[7:4]}$ = HHLH (H'D)	H'3A0	2	INTMSK2[2] INTMSKCLR2[2]	—	—		
	$\overline{IRL[3:0]}$ = HHLH (H'D)			INTMSK2[18] INTMSKCLR2[18]	—	—		
	$\overline{IRL[7:4]}$ = HHHL (H'E)	H'3C0	1	INTMSK2[1] INTMSKCLR2[1]	—	—		
	$\overline{IRL[3:0]}$ = HHHL (H'E)			INTMSK2[17] INTMSKCLR2[17]	—	—		
IRQ	IRQ[0]	H'240	INTPRI [31:28]	INTMSK0[31] INTMSKCLR0[31]	INTREQ [31]	—	High	
	IRQ[1]	H'280	INTPRI [27:24]	INTMSK0[30] INTMSKCLR0[30]	INTREQ [30]	—		
	IRQ[2]	H'2C0	INTPRI [23:20]	INTMSK0[29] INTMSKCLR0[29]	INTREQ [29]	—		
	IRQ[3]	H'300	INTPRI [19:16]	INTMSK0[28] INTMSKCLR0[28]	INTREQ [28]	—		
	IRQ[4]	H'340	INTPRI [15:12]	INTMSK0[27] INTMSKCLR0[27]	INTREQ [27]	—		
	IRQ[5]	H'380	INTPRI [11:8]	INTMSK0[26] INTMSKCLR0[26]	INTREQ [26]	—		
	IRQ[6]	H'3C0	INTPRI [7:4]	INTMSK0[25] INTMSKCLR0[25]	INTREQ [25]	—		
	IRQ[7]	H'200	INTPRI [3:0]	INTMSK0[24] INTMSKCLR0[24]	INTREQ [24]	—	Low	Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
RTC	ATI	H'480	INT2PRI1[4:0]	INT2MSKR[2]	INT2A0[2]	INT2B1[0]	High	High
	PRI	H'4A0		INT2MSKCR[2]	INT2A1[2]	INT2B1[1]		
	CUI	H'4C0				INT2B1[2]	Low	
SECURITY*2	SECI	H'4E0	INT2PRI8[4:0]	INT2MSKR1[0] INT2MSKCR1[0]	INT2A01[0] INT2A11[0]	—		
WDT	ITI*1	H'560	INT2PRI2[12:8]	INT2MSKR[5] INT2MSKCR[5]	INT2A0[5] INT2A1[5]	—		
TMU0	TUNI0*1	H'580	INT2PRI0[28:24]	INT2MSKR[0] INT2MSKCR[0]	INT2A0[0] INT2A1[0]	INT2B0[0]		
TMU1	TUNI1*1	H'5A0	INT2PRI0[20:16]			INT2B0[1]		
TMU2	TUNI2*1	H'5C0	INT2PRI0[12:8]			INT2B0[2]		
	TICPI2*1	H'5E0	INT2PRI0[4:0]			INT2B0[3]		
H-UDI	H-UDI	H'600	INT2PRI3[28:24]	INT2MSKR[7] INT2MSKCR[7]	INT2A0[7] INT2A1[7]	—		
LCDC	LCDCI	H'620	INT2PRI9[28:24]	INT2MSKR1[7] INT2MSKCR1[7]	INT2A01[7] INT2A11[7]	—		
DMAC (0)	DMINT0*1	H'640	INT2PRI3[20:16]	INT2MSKR[8] INT2MSKCR[8]	INT2A0[8] INT2A1[8]	INT2B3[0]	High	
	DMINT1*1	H'660				INT2B3[1]		
	DMINT2*1	H'680				INT2B3[2]		
	DMINT3*1	H'6A0				INT2B3[3]		
	DMAE (ch0 to 5)*1	H'6C0				INT2B3[12]		
						INT2B3[13]	Low	Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
SCIF0	ERI0*	H'700	INT2PRI2 [28:24]	INT2MSKR[3]	INT2A0[3]	INT2B2[0]	High	High
	RXI0*[1]	H'720		INT2MSKCR[3]	INT2A1[3]	INT2B2[1]		
	BRI0*[1]	H'740				INT2B2[2]		
	TXI0*[1]	H'760				INT2B2[3]		
							Low	
DMAC (0)	DMINT4*[1]	H'780	INT2PRI3 [20:16]	INT2MSKR[8]	INT2A0[8]	INT2B3[4]	High	
	DMINT5*[1]	H'7A0		INT2MSKCR[8]	INT2A1[8]	INT2B3[5]	Low	
IIC0	IICI0	H'8A0	INT2PRI9 [4:0]	INT2MSKR1[4]	INT2A01[4]	—		
				INT2MSKCR1[4]	INT2A11[4]			
IIC1	IICI1	H'8C0	INT2PRI9 [12:8]	INT2MSKR1[5]	INT2A01[5]	—		
				INT2MSKCR1[5]	INT2A11[5]			
CMT	CMTI	H'900	INT2PRI4 [28:24]	INT2MSKR[12]	INT2A0[12]	—		
				INT2MSKCR[12]	INT2A1[12]			
GEther	GEINT0	H'920	INT2PRI 12[4:0]	INT2MSKR1[16]	INT2A01[16]	INT2B9[0]		
	GEINT1	H'940		INT2MSKCR1[16]	INT2A11[16]	INT2B9[1]		
	GEINT2	H'960				INT2B9[2]		
HAC	HACI	H'980	INT2PRI4 [20:16]	INT2MSKR[13]	INT2A0[13]	—		
				INT2MSKCR[13]	INT2A1[13]			
PCIC0	PCISERR	H'A00	INT2PRI4 [12:8]	INT2MSKR[14]	INT2A0[14]	INT2B4[0]		
				INT2MSKCR[14]	INT2A1[14]			
PCIC1	PCIINTA	H'A20	INT2PRI4 [4:0]	INT2MSKR[15]	INT2A0[15]	INT2B4[1]		
				INT2MSKCR[15]	INT2A1[15]			
PCIC2	PCIINTB	H'A40	INT2PRI5 [28:24]	INT2MSKR[16]	INT2A0[16]	INT2B4[2]		
				INT2MSKCR[16]	INT2A1[16]			
PCIC3	PCIINTC	H'A60	INT2PRI5 [20:16]	INT2MSKR[17]	INT2A0[17]	INT2B4[3]		
				INT2MSKCR[17]	INT2A1[17]			
PCIC4	PCIINTD	H'A80	INT2PRI5 [12:8]	INT2MSKR[18]	INT2A0[18]	INT2B4[4]		
				INT2MSKCR[18]	INT2A1[18]			Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
PCIC5	PCIERR	H'AA0	INT2PRI5 [4:0]	INT2MSKR[19]	INT2A0[19]	INT2B4[5]	High	High
	PCIPWD3	H'AC0		INT2MSKCR[19]	INT2A1[19]	INT2B4[6]		
	PCIPWD2	H'AE0				INT2B4[7]		
	PCIPWD1	H'B00				INT2B4[8]		
	PCIPWD0	H'B20				INT2B4[9]	Low	
STIF0	STIFI0	H'B40	INT2PRI 13[4:0]	INT2MSKR1[20]	INT2A01[20]	—		
				INT2MSKCR1[20]	INT2A11[20]			
STIF1	STIFI1	H'B60	INT2PRI 13[12:8]	INT2MSKR1[21]	INT2A01[21]	—		
				INT2MSKCR1[21]	INT2A11[21]			
SCIF1	ERI1*[1]	H'B80	INT2PRI2 [20:16]	INT2MSKR[4]	INT2A0[4]	INT2B2[4]	High	
	RXI1*[1]	H'BA0		INT2MSKCR[4]	INT2A1[4]	INT2B2[5]		
	BRI1*[1]	H'BC0				INT2B2[6]		
	TXI1*[1]	H'BE0				INT2B2[7]	Low	
SIOF0	SIOFI0	H'C00	INT2PRI6 [28:24]	INT2MSKR[14]	INT2A0[14]	—		
				INT2MSKCR[14]	INT2A1[14]			
SIOF1	SIOFI1	H'C20	INT2PRI1 0[4:0]	INT2MSKR1[8]	INT2A01[8]	—		
				INT2MSKCR1[8]	INT2A11[8]			
SIOF2	SIOFI2	H'C40	INT2PRI1 0[12:8]	INT2MSKR1[9]	INT2A01[9]	—		
				INT2MSKCR1[9]	INT2A11[9]			
USBH	USBHI	H'C60	INT2PRI1 2[12:8]	INT2MSKR1[17]	INT2A01 [17]	—		
				INT2MSKCR1 [17]	INT2A11 [17]			
USBF	USBFI0	H'C80	INT2PRI6 [20:16]	INT2MSKR1[24]	INT2A01 [24]	INT2B10 [0]		
	USBFI1	H'CA0		INT2MSKCR1 [24]	INT2A11 [24]	INT2B10 [1]		Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
TPU	TPI	H'CC0	INT2PRI10 [28:24]	INT2MSKR1[11] INT2MSKCR1 [11]	INT2A01 [11] INT2A11 [11]	—		High
PCC	PCCI	H'CE0	INT2PRI11 [28:24]	INT2MSKR1[15] INT2MSKCR1 [15]	INT2A01 [15] INT2A11 [15]	—		
MMCIF	FSTAT	H'D00	INT2PRI6 [12:8]	INT2MSKR[22] INT2MSKCR[22]	INT2A0[22] INT2A1[22]	INT2B5[0]	High	
	TRAN	H'D20				INT2B5[1]		
	ERR	H'D40				INT2B5[2]		
	FRDY	H'D60				INT2B5[3]	Low	
SIM	ERI	H'D80	INT2PRI10 [20:16]	INT2MSKR1[10] INT2MSKCR1 [10]	INT2A01 [10] INT2A11 [10]	INT2B11[0]		
	RXI	H'DA0				INT2B11[1]		
	TXI	H'DC0				INT2B11[2]		
	TEND	H'DE0				INT2B11[3]		
TMU3	TUNI3*[1]	H'E00	INT2PRI1 [28:24]	INT2MSKR[1] INT2MSKCR[1]	INT2A0[1] INT2A1[1]	INT2B0[4]		
TMU4	TUNI4*[1]	H'E20	INT2PRI1 [20:16]			INT2B0[5]		
TMU5	TUNI5*[1]	H'E40	INT2PRI1 [12:8]			INT2B0[6]		
ADC	ADI	H'E60	INT2PRI3 [12:8]	INT2MSKR1[12] INT2MSKCR1 [12]	INT2A01 [12] INT2A11 [12]	—		
SSI0	SSII0	H'E80	INT2PRI6 [4:0]	INT2MSKR[23] INT2MSKCR[23]	INT2A0[23] INT2A1[23]	—		
SSI1	SSII1	H'EA0	INT2PRI8 [12:8]	INT2MSKR1[1] INT2MSKCR1[1]	INT2A01[1] INT2A11[1]	—		Low

Interrupt Source		INTEVT Code	Interrupt Priority	MASK/CLEAR Register	Interrupt Source Register	Detail Source Register	Priority in the Source	Default Priority
SSI2	SSII2	H'EC0	INT2PRI8 [20:16]	INT2MSKR1[2] INT2MSKCR1[2]	INT2A01[2] INT2A11[2]	—		High
SSI3	SSII3	H'EE0	INT2PRI8 [28:24]	INT2MSKR1[3] INT2MSKCR1[3]	INT2A01[3] INT2A11[3]	—		
SCIF2	ERI2	H'F00	INT2PRI7 [28:24]	INT2MSKR1[25] INT2MSKCR1[25]	INT2A01 [25] INT2A11 [25]	INT2B6[0]	High	
	RXI2	H'F20				INT2B6[1]		
	BRI2	H'F40				INT2B6[2]		
	TXI2	H'F60				INT2B6[3]	Low	
GPIO	CH0	H'F80	INT2PRI7 [20:16]	INT2MSKR[25] INT2MSKCR[25] [2]	INT2A0[25] INT2A1[25]	INT2B7 [3:0]	High	
	CH1	H'FA0				INT2B7 [11:8]		
	CH2	H'FC0				INT2B7 [19:16]		
	CH3	H'FE0				INT2B7 [27:24]	Low	Low

Notes: 1. ITI: Interval timer interrupt
TUNI0 to TUNI5: TMU channel 0 to 5 under flow interrupt
TICPI2: TMU channel 2 input capture interrupt
DMINT0 to DMINT5: DMAC channel 0 to 5 transfer end interrupt
DMAE: DMAC address error interrupt (channel 0 to 5)
ERI0, ERI1: SCIF channel 0, 1 receive error interrupt
RXI0, RXI1: SCIF channel 0, 1 receive data full interrupt
BRI0, BRI1: SCIF channel 0, 1 break interrupt
TXI0, TXI1: SCIF channel 0, 1 transmission data empty interrupt
2. This bit is reserved in the R5S77631.

9.5 Operation

9.5.1 Interrupt Sequence

The sequence of interrupt operations is described below. Figure 9.4 is the flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the INTC.
2. The INTC selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in INTPRI and INT2PRI0 to INT2PRI14. Lower-priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected according to table 9.7.
3. The priority level of the interrupt selected by the INTC is compared with the interrupt mask level (IMASK) set in SR of the CPU. If the priority level is higher than the mask level, the INTC accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. SR and program counter (PC) are saved to SSR and SPC, respectively. R15 is saved to SGR at this time.
7. The BL, MD, and RB bits in SR are set to 1.
8. Execution jumps to the start address of the interrupt exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

In the exception handling routine, execution may branch with the INTEVT value used as its offset in order to identify the interrupt source. This enables execution to branch to the handling routine for the individual interrupt source.

Notes: 1. When the INTMU bit in the CPU operating mode register (CPUOPM) is set to 1, the interrupt mask level (IMASK) in SR is automatically set to the level of the accepted interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not affected by the accepted interrupt.

2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt source that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, wait for the time shown in table 9.8, and then clear the BL bit or execute an RTE instruction.
3. For some interrupt sources, the interrupt mask setting (INTMSK) for each interrupt source must be cleared by using INTMSKCLR.

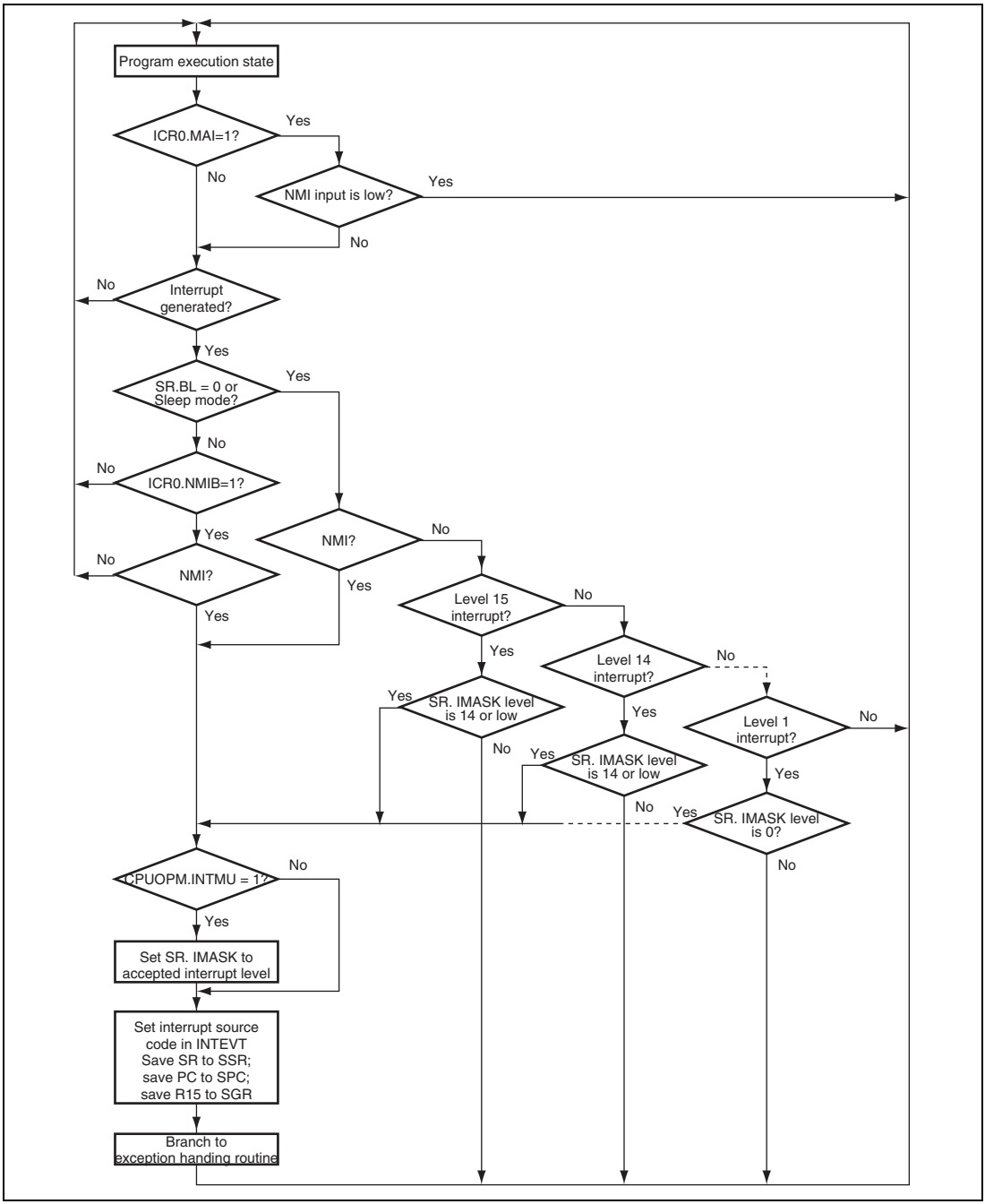


Figure 9.4 Interrupt Operation Flowchart

9.5.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handling routine should include the following procedures:

1. To identify the interrupt source, branch to a specific interrupt handling routine for the interrupt source by using the INTEVT code as an offset.
2. Clear the interrupt source in each specific interrupt handling routine.
3. Save SSR and SPC to the stack.
4. Clear the BL bit in SR. When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is automatically modified to the level of the accepted interrupt. When the INTMU bit in CPUOPM is cleared to 0, set the IMASK bit in SR by software to the accepted interrupt level.
5. Handle the interrupt as required.
6. Set the BL bit in SR to 1.
7. Restore SSR and SPC from memory.
8. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted if multiple interrupts occur after step 4. This reduces the interrupt response time for urgent processing.

9.5.3 Interrupt Masking by MAI Bit

Setting the MAI bit in ICR0 to 1 masks interrupts while the NMI signal is low regardless of the BL and IMASK bit settings in SR.

- Normal operation or sleep mode

All interrupts are masked while the NMI signal is low. Note that only NMI interrupts due to NMI signal input occur.

9.6 Interrupt Response Time

Table 9.8 shows the interrupt response time, which is the interval from when an interrupt request occurs until the interrupt exception handling is started and the start instruction of the exception handling routine is fetched.

Table 9.8 Interrupt Response Time

		Number of States					
					Peripheral Module		
Item		NMI	IRL	IRQ	Other than GPIO/PCIC/ RTC	GPIO/PCIC/ RTC	Remarks
Priority determination time		5Bcyc + 2Pcyc	8Bcyc + 2Pcyc	4Bcyc + 2Pcyc	5Pcyc	7Pcyc	
Wait time until the CPU finishes the current sequence				S-1 (≥ 0) × Icyc			
Interval from when interrupt exception handling begins (saving SR and PC) until a SHwy bus request is issued to fetch the start instruction of the exception handling routine				11Icyc + 1Scyc			
Response time	Total	(S + 10) Icyc + 1Scyc + 5Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 8Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 4Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 5Pcyc	(S + 10) Icyc + 1Scyc + 7Pcyc	
	Minimum	40Icyc + SxIcyc	52Icyc + SxIcyc	36Icyc + SxIcyc	32Icyc + SxIcyc*	40Icyc + SxIcyc*	When Icyc:Scyc: Bcyc:Pcyc = 4:2:1:1

[Legend]
Icyc: Period for one CPU clock cycle
Scyc: Period for one SHwy clock cycle
Bcyc: Period for one bus clock cycle
Pcyc: Period for one peripheral clock cycle (Pck0)
S: Number of instruction execution states
Note * In the case of Pcyc = Pck.

9.7 Usage Notes

9.7.1 Example of Interrupt Handling Routine for Level-Encoded IRL and Level-Sensed IRQ

If an interrupt request is accepted when level-sensed IRQ or level-encoded IRL interrupt request is selected, the held request must be cleared in the interrupt handling routine. Figure 9.5 shows an example of clearing the interrupt request held in the detection circuit.

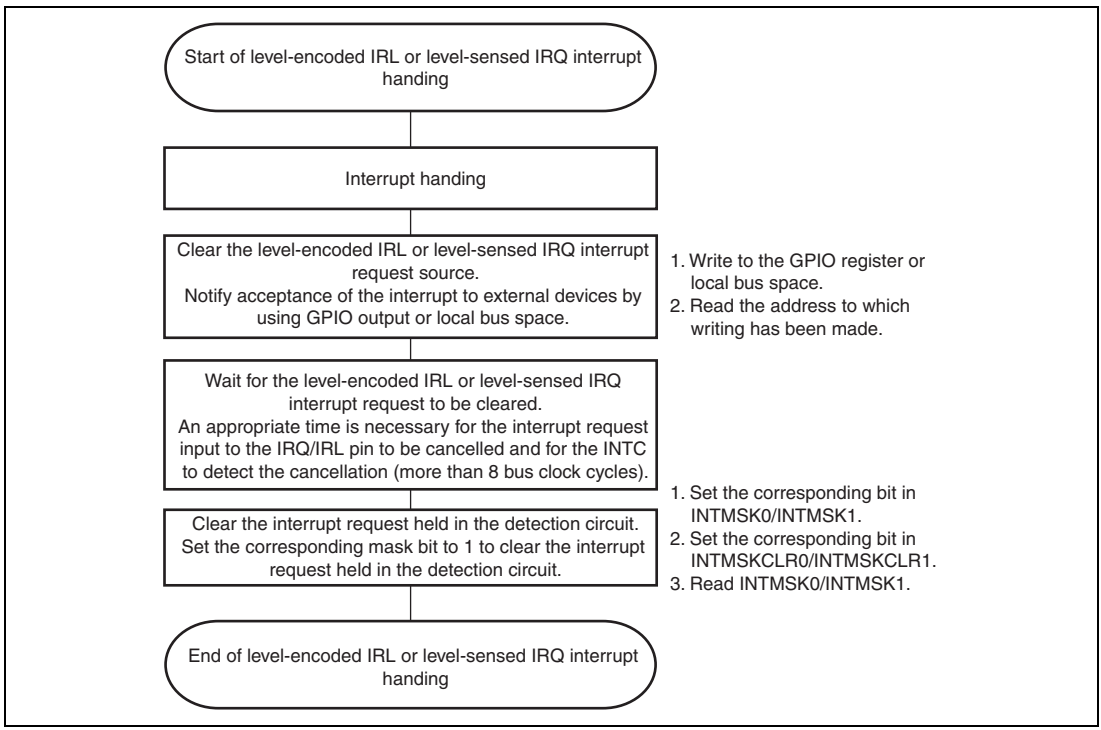


Figure 9.5 Example of Interrupt Handling Routine

After the CPU accepts an interrupt request, acceptance of the request should be notified to the external devices and the request should be cancelled. For example, acceptance can be notified by outputting the accepted level and pin-related information via the GPIO (general I/O port) and writing the acceptance information to a specific address in the local bus space. Here, writing to the GPIO register or local bus space and reading from the location should be consecutively executed.

When clearing the interrupt requests held in the detection circuit, adequate time is necessary for the CPU to detect the cancellation of the interrupt request. To secure the time, writing to

INTMSK0/INTMSK1 and INTMSKCLR0/INTMSKCLR1 and reading from INTMSK0/INTMSK1 should be consecutively executed.

9.7.2 Notes on Setting IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ Pin Function

When switching the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pin function, it is possible that the INTC may hold an interrupt by mistake. Therefore, to prevent detecting unintentional interrupts, mask both all IRQ and IRL interrupts and then switch the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pin function.

Table 9.9 Switching Sequence of IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ Pin Function

Sequence	ITEM	PROCEDURE
1	IRL interrupt request and IRQ interrupt request masking	Write 1 to all bits in INTMSK0 and INTMSK1
2	IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ pins setting to IRL7 to IRL4 interrupt request input	Write 010 to the PTSEL3[14:12] bits and the PTSEL3[2:0] bits in PSEL3 of GPIO Write 00 to the PL3MD[1:0], PL1MD[1:0], PL0MD[1:0] bits in PLCR
3	IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pins setting to IRL or IRQ interrupt request input	Set the IRLM[1:0] bit in ICR0
4	IRL and IRQ interrupt detection start	Write 1 to the corresponding bit in INTMSKCLR0 and INTMSKCLR1

9.7.3 To Clear IRQ and IRL Interrupt Requests

Clearing procedure of the interrupt held in the INTC is as follows

- **To clear IRL interrupt requests**

  To clear an IRL interrupt request from the IRQ3/$\overline{\text{IRL3}}$ to IRQ0/$\overline{\text{IRL0}}$ pins, write 1 to the IM10 bit in INTMSK1, and to clear an IRL interrupt request from the IRQ7/$\overline{\text{IRL7}}$ to IRQ4/$\overline{\text{IRL4}}$ pins, write 1 to the IM11 bit in INTMSK1. The IRL interrupt requests detected by the INTC is not cleared even if each of the corresponding interrupt level is masked by setting INTMSK2.
- **To clear IRQ level-sense interrupt requests**

  To clear an IRQ level-sense interrupt request from the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pins, write 1 to the corresponding mask bit (IM07 to IM00) in INTMSK0.

  The IRQ interrupt requests detected by the INTC is not cleared even if 0 is written to a corresponding bit in INTPRI. The IRQ interrupt sources detected by the INTC (be cleared)

- **To clear IRQ edge-detection interrupt requests**

  To clear an IRQ edge-detection interrupt request from the IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ pins, write 0 after reading 1 in the corresponding IRn (n = 0 to 7) bit in INTREQ.

  The IRQ interrupt requests detected by the INTC is not cleared even if 1 is written to a corresponding bit in INTMSK0.

Section 10 SuperHyway Bus Bridge (SBR)

The SuperHyway bus bridge (SBR) performs access protocol conversion between the SuperHyway (Shwy) bus and the SuperHyway bridge bus. At the same time, it also arbitrates between the accesses to the SuperHyway bus by the three peripheral modules (SECURITY, GETHER, and USBH) connected to the SuperHyway bridge bus.

10.1 Features

- SuperHyway bus interface

  Performs access protocol conversion between the SuperHyway bus and the SuperHyway bridge bus.
- Arbitration

  The arbiter arbitrates between the accesses to the SuperHyway bus by the SECURITY, GETHER, and USBH modules connected to the SuperHyway bridge bus. Priority can be set for the individual modules or ports.

Figure 10.1 shows a block diagram of the SBR.

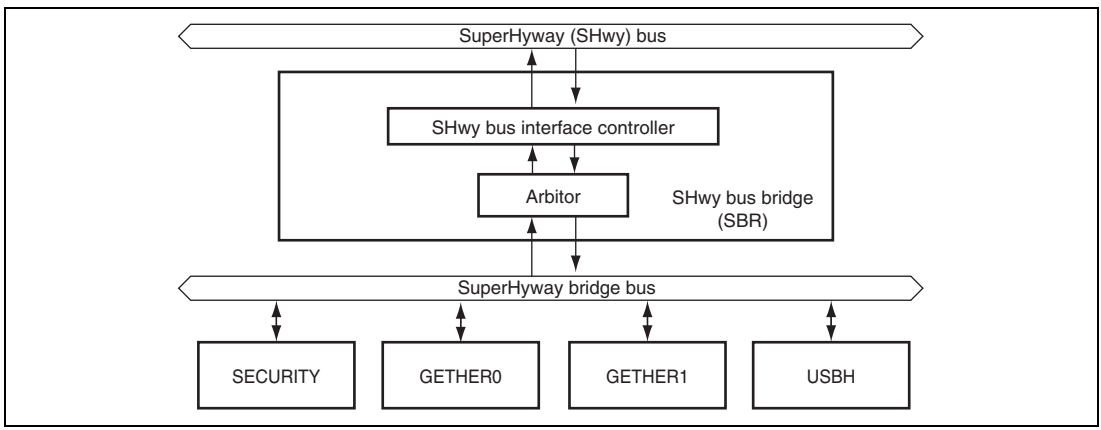


Figure 10.1 SBR Block Diagram

10.2 Register Descriptions

Table 10.1 shows the SBR register configuration. Table 10.2 shows the register state in each operating mode.

Table 10.1 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Bus arbitration priority level setting register	SBRIVCLV	R/W	H'FF40 0010	H'1F40 0010	32
SuperHyway bus priority control register	PRPRICR	R/W	H'FE60 0018	H'1F60 0018	32

Note: The area P4 address is the address when the P4 area of a virtual address space is used. The area 7 address is the address when the register is accessed through area 7 of a physical address space by using the TLB.

Table 10.2 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Bus arbitration priority level setting register	SBRIVCLV	H'0000 0000	H'0000 0000	Retained	Retained
SuperHyway bus priority control register	PRPRICR	H'0000 0001	H'0000 0001	Retained	Retained

10.2.1 Bus Arbitration Priority Level Setting Register (SBRIVCLV)

SBRIVCLV sets the priority levels used when SuperHyway bus access requests from the SECURITY, GETHER, and USBH coincide.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SEC LV	GEC0 LV	GEC1 LV	—	—	—	USBH LV
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W

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	SECLV	0	R/W	SECURITY Access Priority Level 0: Level 3 1: Level 2
5	GEC0LV	0	R/W	GETHER0 Access Priority Level 0: Level 3 1: Level 2
4	GEC1LV	0	R/W	GETHER1 Access Priority Level 0: Level 3 1: Level 2
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	USBHLV	0	R/W	USBH Access Priority Level 0: Level 3 1: Level 2

10.2.2 SuperHyway Bus Priority Control Resister (PRPRICR)

PRPRICR controls the SuperHyway bus access priority given to the CPU and the other function modules.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SBAPR	1	R/W	SuperHyway Bus Access Priority Sets whether to give the CPU priority over the other function modules when accessing the SuperHyway bus. 0: The CPU is given the same priority as the other function modules. 1: The CPU is given priority over the other function modules. Note: With this bit set to 1, when both of the data area accessed by the CPU and the data area accessed by the other function modules are placed in the areas for slow accesses, the CPU may be given priority, thus disabling accesses by the other function modules. To avoid this, place the data area accessed by the CPU or the other function modules in the DDR-SDRAM area, or set this bit to 0.

10.3 Operation

10.3.1 SuperHyway Bus Interface

The SuperHyway bus bridge (SBR) performs access protocol conversion between the SuperHyway bus and the SuperHyway bridge bus.

10.3.2 Bus Arbitration

The SBR performs arbitration for the access requests from the four ports of the three modules (SECURITY, GETHER, and USBH) connected to the SuperHyway bridge bus. Figure 10.2 shows the concept of arbitration by the SBR.

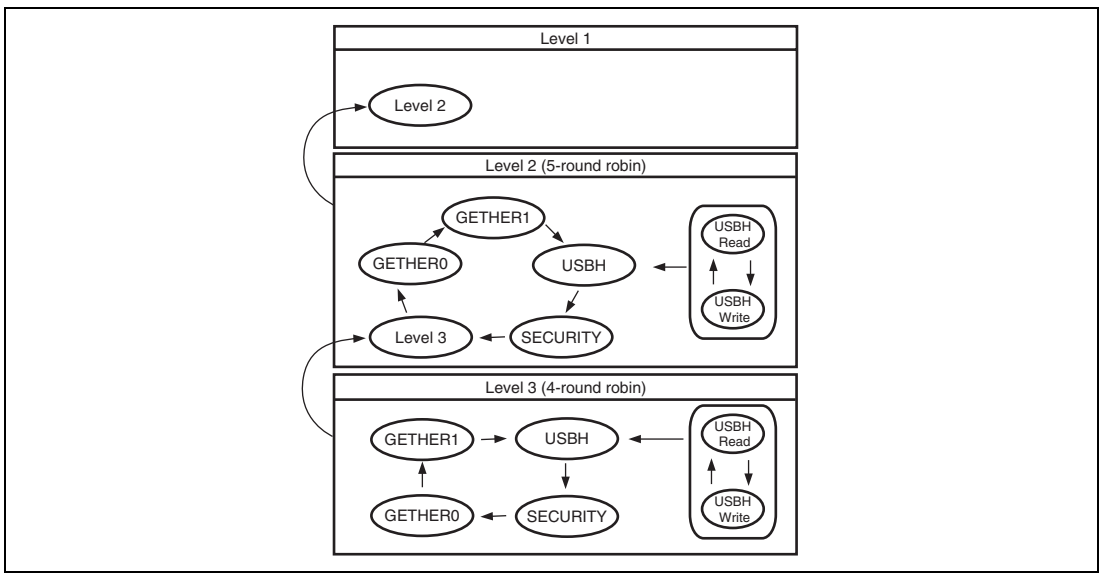


Figure 10.2 Bus Arbitration by the SBR

The SBR performs arbitration for three groups — level 1, level 2, and level 3. At level 3, round-robin arbitration is performed for a total of four ports of three modules; at level 2, round-robin arbitration is performed for four ports of three modules plus the result of arbitration at level 3. Since there are no modules to compete with at level 1, the access from the module that won the arbitration at level 2 is immediately executed.

The priority level of access requests from SECURITY, GETHER1, GETHER0, and USBH can be set to level 2 or level 3 through the SBRIVCLV register. Note that, in the USBH module, round-robin arbitration is first performed between read and write requests, then inter-module arbitration is performed using the result.

Section 11 Local Bus State Controller (LBSC)

The local bus state controller (LBSC) divides the external memory space and outputs control signals corresponding to the specifications of various types of memory and bus interfaces. The LBSC enables the connection of SRAM or ROM, etc., to this LSI. It also supports the PCMCIA interface protocol, which is used to implement simplified system design and high-speed data transfers in a compact system.

11.1 Features

The LBSC has the following features.

- Controls six areas, areas 0 to 2 and 4 to 6, of an external memory space divided into seven areas.
  - Maximum 64 Mbytes for each of areas 0 to 2 and 4 to 6
  - Bus width of each area can be controlled through register settings (except area 0, which is controlled by the external pin setting)
  - Wait-cycle insertion by the $\overline{\text{RDY}}$ pin
  - Wait-cycle insertion can be controlled by a program
  - Types of memory are specifiable for connection to each area
  - Output of the control signals of memory to each area
  - Automatic wait cycle insertion to prevent data bus collisions on consecutive memory accesses to different areas, or a read access followed by a write access to the same area
  - Insertion of cycles to ensure the setup time and hold time to the write strobe on a write cycle enables connection to low-speed memory
- SRAM interface
  - Wait-cycle insertion can be controlled by a program
  - Insertion of the wait cycle through the $\overline{\text{RDY}}$ pin

    Connectable areas : 0 to 2 and 4 to 6

    Settable bus widths: 32, 16, and 8 bits
- Burst ROM interface
  - Wait-cycle insertion can be controlled by a program
  - Burst length specified by the register

    Connectable areas: 0 to 2 and 4 to 6

    Settable bus widths: 32, 16, and 8 bits

- MPX interface
  - Address/data multiplexing
    Connectable areas: 0 to 2 and 4 to 6
    Settable bus width: 32 bits
- Byte control SRAM interface
  - SRAM interface with byte control
    Connectable areas: 1 and 4
    Settable bus widths: 32 and 16 bits
- PCMCIA interface
  - Wait-cycle insertion can be controlled by a program
  - Bus sizing function for I/O bus width
  - Little endian
    Connectable areas: 5 and 6
    Settable bus widths: 16 and 8 bits

Figure 11.1 shows a block diagram of the LBSC.

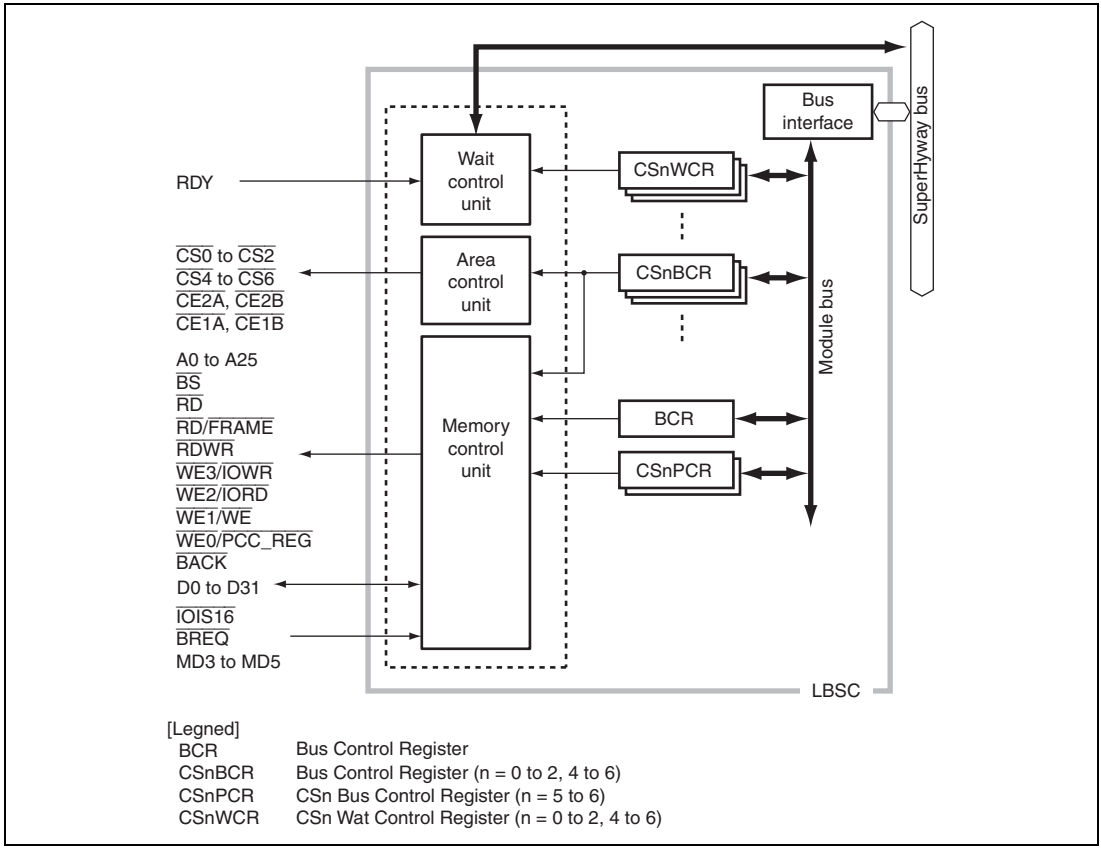


Figure 11.1 LBSC Block Diagram

11.2 Input/Output Pins

Table 11.1 shows the LBSC pin configuration.

Table 11.1 Pin Configuration

Pin Name	Function	I/O	Description
A25 to A0	Address Bus	Output	Address output
D31 to D0	Data Bus	I/O	Data input/output
$\overline{BS}$	Bus Cycle Start	Output	Signal that indicates the start of a bus cycle. Asserted once for a burst transfer when setting MPX interface. Asserted each data cycle for a burst transfer when setting other interfaces.
$\overline{CS6}$ to $\overline{CS4}$, $\overline{CS2}$ to $\overline{CS0}$	Chip Select 6 to 4 and 2 to 0	Output	Chip select signal that indicates the area being accessed. $\overline{CS5}$ and $\overline{CS6}$ can also be used as $\overline{CE1A}$ to $\overline{CE1B}$ of PCMCIA.
$\overline{RDWR}$	Read/Write	Output	Data bus input/output direction designation signal. Also used as PCMCIA interface write designation signal.
$\overline{RD}/\overline{FRAME}$	Read/Cycle Frame	Output	Strobe signal indicating a read cycle. $\overline{FRAME}$ signal when setting MPX interface.
$\overline{WE0}/\overline{PCC_REG}$	Data Enable 0	Output	When setting SRAM interface: write strobe signal for D7 to D0 When setting PCMCIA interface: $\overline{REG}$ signal
$\overline{WE1}/\overline{WE}$	Data Enable 1	Output	When setting SRAM interface: write strobe signal for D15 to D8 When setting PCMCIA interface: Write strobe signal
$\overline{WE2}/\overline{IORD}$	Data Enable 2	Output	When setting SRAM interface: write strobe signal for D23 to D16 When setting PCMCIA interface: $\overline{IORD}$ signal
$\overline{WE3}/\overline{IOWR}$	Data Enable 3	Output	When setting SRAM interface: write strobe signal for D31 to D24 When setting PCMCIA interface: $\overline{IOWR}$ signal
$\overline{RDY}$	Ready	Input	Wait cycle request signal
$\overline{IOIS16}$	16-Bit I/O	Input	16-bit I/O signal when setting PCMCIA interface. Valid only in little endian mode

Pin Name	Function	I/O	Description
$\overline{\text{BREQ}}$	Bus Release Request	Input	Bus release request signal
$\overline{\text{BACK}}$	Bus Request Acknowledge	Output	Bus release acknowledge signal
$\overline{\text{CE2A}}$*1, $\overline{\text{CE2B}}$*1	PCMCIA Card Select	Output	When setting PCMCIA, corresponds to PCMCIA card select signal D15 to D8. Valid only in little endian mode
$\overline{\text{CE1A}}$*1, $\overline{\text{CE1B}}$*1	PCMCIA Card Select	Output	When setting PCMCIA, corresponds to PCMCIA card select signal D7 to D0.
MD3, MD4	Area 0 Bus Width	Input	Signal setting area 0 bus width and MPX interface at power-on reset
MD5	Endian Switchover	Input	Endian setting at a power-on reset
$\overline{\text{DACK0}}$*2	DMAC0 Acknowledge Signal	Output	Data acknowledge of DMAC channel 0
$\overline{\text{DACK1}}$*2	DMAC1 Acknowledge Signal	Output	Data acknowledge of DMAC channel 1
$\overline{\text{DACK2}}$*2	DMAC2 Acknowledge Signal	Output	Data acknowledge of DMAC channel 2
$\overline{\text{DACK3}}$*2	DMAC3 Acknowledge Signal	Output	Data acknowledge of DMAC channel 3
$\overline{\text{TEND0}}$*2	DMAC0 Transfer End Signal	Output	Transfer end of DMAC channel 0
$\overline{\text{TEND1}}$*2	DMAC1 Transfer End Signal	Output	Transfer end of DMAC channel 1
$\overline{\text{TEND2}}$*2	DMAC2 Transfer End Signal	Output	Transfer end of DMAC channel 2
$\overline{\text{TEND3}}$*2	DMAC3 Transfer End Signal	Output	Transfer end of DMAC channel 3

Notes: 1. When bits TYPE [2:0] in the CS5 bus control register (CS5BCR) are set to B'100, $\overline{\text{CE2A}}$ and $\overline{\text{CE1A}}$ act as PCMCIA output pins, and bits TYPE [2:0] in the CS6 bus control register (CS6BCR) are set to b'100, $\overline{\text{CE2B}}$ and $\overline{\text{CE1B}}$ act as PCMCIA output pins.

2. Can be selectable the polarity (initial state is low active). For details, see section 14, Direct Memory Access Controller (DMAC).

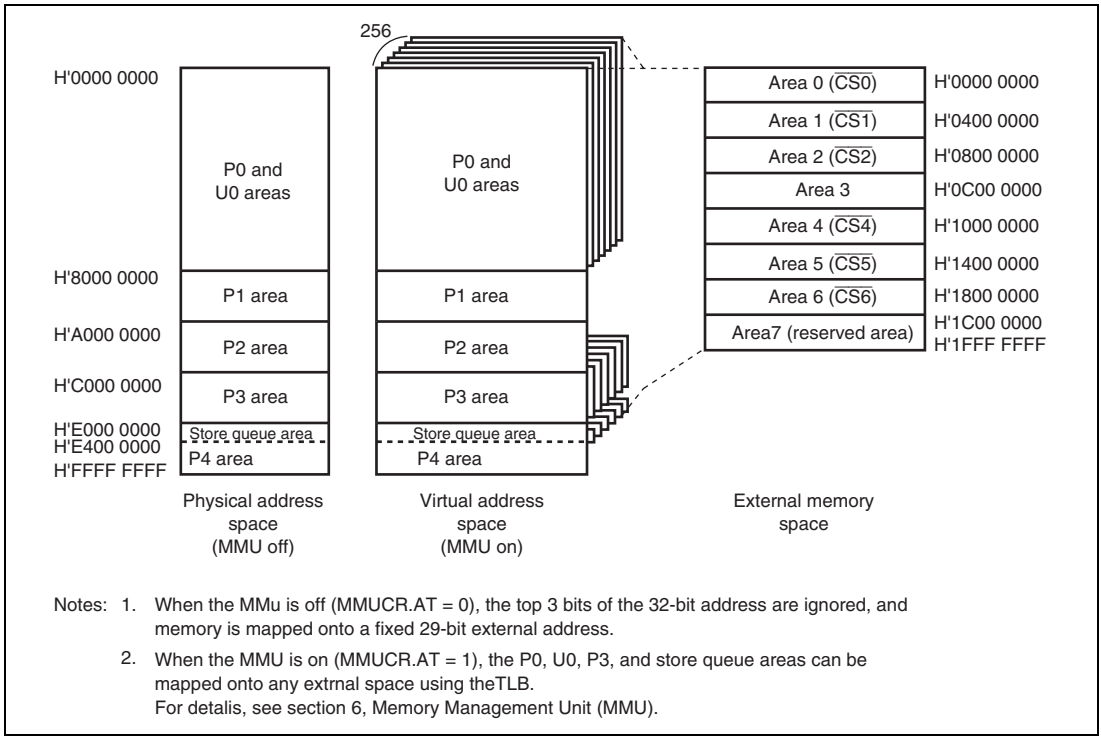
11.3 Area Overview

11.3.1 Space Divisions

The architecture of this LSI provides a 32-bit virtual address space. The virtual address space is divided into five areas according to the upper address value. The external memory space indicated by the remaining 29 address bits is divided into eight areas.

The virtual address can be allocated to any external address using the address translation function of the MMU. For details, see section 6, Memory Management Unit (MMU). This section describes the area division of the external address space.

With this LSI, various types of memory or PC cards can be connected to each of the seven areas in the external address space as shown in table 11.2, and accordingly output the chip select signals ($\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{CE2A}$, and $\overline{CE2B}$). Area 3 is used for DDR-SDRAM. $\overline{CS0}$ to $\overline{CS2}$ are asserted when accessing area 0 to 2, and $\overline{CS4}$ to $\overline{CS6}$ when accessing area 4 to 6. When the PCMCIA interface is selected for area 5 or 6, $\overline{CE2A}$ or $\overline{CE2B}$ is asserted along with $\overline{CS5}$/$\overline{CE1A}$ or $\overline{CS6}$/$\overline{CE1B}$ for the bytes to be accessed.



Notes: 1. When the MMu is off (MMUCR.AT = 0), the top 3 bits of the 32-bit address are ignored, and memory is mapped onto a fixed 29-bit external address.

2. When the MMU is on (MMUCR.AT = 1), the P0, U0, P3, and store queue areas can be mapped onto any extrnal space using theTLB.
For detalis, see section 6, Memory Management Unit (MMU).

Figure 11.2 Correspondence between Virtual Address Space and External Memory Space

Table 11.2 LBSC External Memory Space Map

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width	Access Size*[7]
0	H'0000 0000 to H'03FF FFFF	64 Mbytes	SRAM	8, 16, 32*[1]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[1]	
			MPX	32*[1]	
1	H'0400 0000 to H'07FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	

Area	External addresses	Size	Connectable Memory	Specifiable Bus Width	Access Size*[7]
2	H'0800 0000 to H'0BFF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			(DDR-SDRAM*[4])	32	
3*[3]	H'0C00 0000 to H'0FFF FFFF	64 Mbytes	(DDR-SDRAM)	32	8/16/32 bits and 32 bytes
4	H'1000 0000 to H'13FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	
			(DDR-SDRAM*[4])	32	
			(PCI*[4])	32	
5	H'1400 0000 to H'17FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			PCMCIA	8, 16*[2]*[5]	
			(DDR-SDRAM*[4])	32	
6	H'1800 0000 to H'1BFF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 bits and 32 bytes
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			PCMCIA	8, 16*[2]*[5]	
7*[6]	H'1C00 0000 to H'1FFF FFFF	64 Mbytes	—	—	—

Notes: 1. The memory bus width is specified by external pins.
2. The memory bus width is specified by the register.
3. Area 3 is used specifically for the DDR-SDRAM. For details, see section 12, DDR-SDRAM Interface (DDRIF).
4. These areas can be used for the DDR-SDRAM or PCI by setting MMSELR. For details, see section 12, DDR-SDRAM Interface (DDRIF) or see section 13, PCI Controller (PCIC).
5. With the PCMCIA interface, the bus width is either 8 bits or 16 bits.
6. If a reserved area is accessed, operation cannot be guaranteed.
7. If 8 or 16 bytes access transfer by another LSI internal bus master module is being executed, the LBSC is executing two or four times 32-bit access individually.

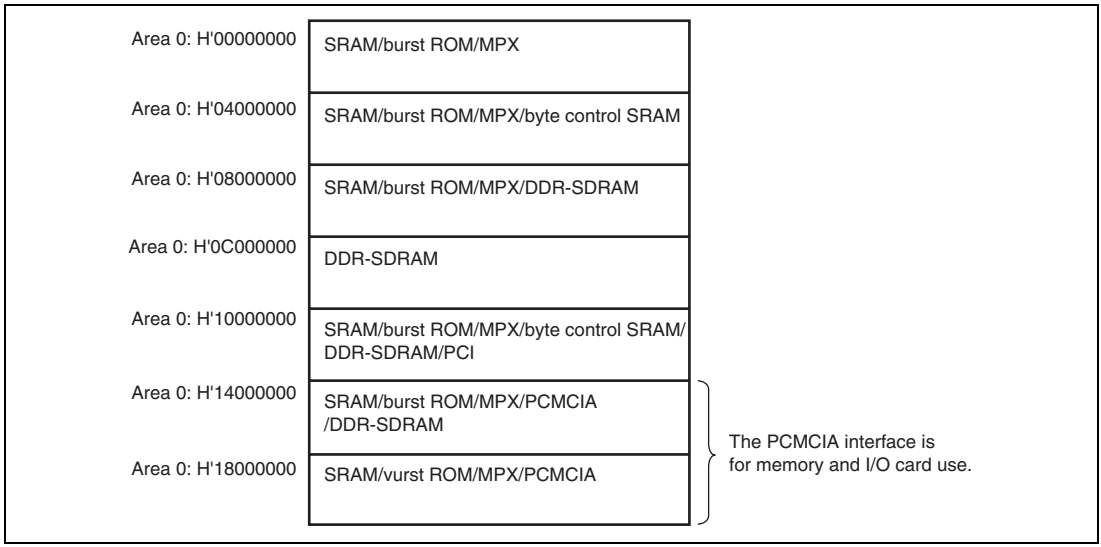


Figure 11.3 External Memory Space Allocation

11.3.2 Memory Bus Width

The memory bus width of the LBSC can be set independently for each area. For area 0, a bus width of 8, 16, or 32 bits is set according to the external pin settings at a power-on reset by the $\overline{\text{PRESET}}$ pin. The correspondence between the external pins (MD4 and MD3) and the bus width at a power-on reset is shown in table 11.3.

Table 11.3 Setting of bus width for area 0

MD4	MD3	Bus Width
0	0	32 bits (MPX interface)
0	1	8 bits
1	0	16 bits
1	1	32 bits

When either the SRAM or ROM interface is used in areas 1 to 2 and 4 to 6, a bus width of 8, 16, or 32 bits can be selected through the CSn bus control register (CSnBCR). When the burst ROM interface is used, a bus width of 8, 16, or 32 bits can be selected. When the byte-control SRAM interface is used, a bus width of 16 or 32 bits can be selected. When the MPX interface is used, a bus width of 32 bits should be selected.

When using the PCMCIA interface, a bus width of 8 or 16 bits should be selected. For details, see section 11.5.5, PCMCIA Interface.

For details, see section 11.4.3, CSn Bus Control Register (CSnBCR).

The bus width of the DDR-SDRAM and the PCI interfaces is 32 bits. For details, see section 12, DDR-SDRAM Interface (DDRIF), and section 13, PCI Controller (PCIC).

The addresses of area 7 (H'1C00 0000 to H'1FFF FFFF) are reserved and must not be used.

11.3.3 Data Alignment

This LSI supports big endian and little endian as data alignment. Data alignment is determined by the level of the external pin (MD5) at a power-on reset.

Table 11.4 Correspondence between External Pin (MD5) and Endian

MD5	Data Alignment
Low	Big endian
High	Little endian

11.3.4 PCMCIA Support

This LSI supports the PCMCIA interface specifications for areas 5 and 6 in the external memory space.

The IC memory card interface and I/O card interface prescribed in JEIDA specifications version 4.2 (PCMCIA2.1) are supported.

Both the IC memory card interface and the I/O card interface are supported in areas 5 and 6 in the external memory space.

The PCMCIA interface is only supported in little endian mode.

Table 11.5 PCMCIA Interface Features

Item	Features
Access	Random access
Data bus	8/16 bits
Memory type	Masked ROM, OTPROM, EPROM, flash memory, SRAM
Common memory capacity	Maximum 64 Mbytes
Attribute memory capacity	Maximum 64 Mbytes
Others	Dynamic bus sizing for I/O bus width

Table 11.6 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name*[1]	I/O*[1]	Function	Signal Name*[1]	I/O*[1]	Function	Corresponding Pin of this LSI
1	GND		Ground	GND		Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CS5}$ or $\overline{CS6}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	$\overline{RD}$
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$	I	Write enable	$\overline{WE}$	I	Write enable	$\overline{WE1}$
16	$\overline{READY}$	O	Ready	$\overline{IREQ}$	O	Interrupt request	Sensed on port
17	VCC		Operation power supply	VCC		Operation power supply	—
18	VPP1		Programming power supply	VPP1		Programming/ peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name*[1]	I/O*[1]	Function	Signal Name*[1]	I/O*[1]	Function	Corresponding Pin of this LSI
30	D0	I/O	Data	D0	I/O	Data	D0
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	$\overline{WP}$	O	Write protect	$\overline{IOIS16}$	O	16-bit I/O port	$\overline{IOIS16}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{CD1}$	O	Card detection	$\overline{CD1}$	O	Card detection	Sensed on port
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{CE2}$	I	Card enable	$\overline{CE2}$	I	Card enable	$\overline{CE2A}$ or $\overline{CE2B}$
43	$\overline{VS1}$	I	Refresh request	$\overline{VS1}$	I	Refresh request	Output from port
44	RSRVD		Reserved	$\overline{IORD}$	I	I/O read	$\overline{IORD}$
45	RSRVD		Reserved	$\overline{IOWR}$	I	I/O write	$\overline{IOWR}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—
52	VPP2		Programming power supply	VPP2		Programming/ peripheral power supply	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	RSRVD		Reserved	RSRVD		Reserved	—
58	RESET	I	Reset	RESET	I	Reset	Output from port
59	$\overline{WAIT}$	O	Wait request	$\overline{WAIT}$	O	Wait request	$\overline{RDY}$*[2]

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal Name*[1]**	**I/O*[1]**	**Function**	**Signal Name*[1]**	**I/O*[1]**	**Function**	**Corresponding Pin of this LSI**
60	RSRVD		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital voice signal	Sensed on port
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed on port
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	Sensed on port
68	GND		Ground	GND		Ground	—

Notes: 1. I/O means input/output on the side of the PCMCIA card.

The polarity of the PCMCIA card interface means that on the side of the card, while the polarity of the corresponding pin of this LSI means that on the side of this LSI.

2. Check the polarity.

11.4 Register Descriptions

The LBSC has 16 registers as shown in table 11.7 and 11.8. The following registers control memory interfaces, wait cycles, etc.

Table 11.7 Register Configuration

Register Name	Abbrev.	R/W	Initial Value	P4 Address	Area 7 Address	Access Size
Memory Address Map Select Register	MMSELR	R/W	H'0000 0000	H'FE60 0020	H'1E60 0020	32
Bus Control Register	BCR	R/W	H'x000 0000	H'FF80 1000	H'1F80 1000	32
CS0 Bus Control Register	CS0BCR	R/W	H'7777 7770	H'FF80 2000	H'1F80 2000	32
CS1 Bus Control Register	CS1BCR	R/W	H'7777 7770	H'FF80 2010	H'1F80 2010	32
CS2 Bus Control Register	CS2BCR	R/W	H'7777 7770	H'FF80 2020	H'1F80 2020	32
CS4 Bus Control Register	CS4BCR	R/W	H'7777 7770	H'FF80 2040	H'1F80 2040	32
CS5 Bus Control Register	CS5BCR	R/W	H'7777 7770	H'FF80 2050	H'1F80 2050	32
CS6 Bus Control Register	CS6BCR	R/W	H'7777 7770	H'FF80 2060	H'1F80 2060	32
CS0 Wait Control Register	CS0WCR	R/W	H'7777 770F	H'FF80 2008	H'1F80 2008	32
CS1 Wait Control Register	CS1WCR	R/W	H'7777 770F	H'FF80 2018	H'1F80 2018	32
CS2 Wait Control Register	CS2WCR	R/W	H'7777 770F	H'FF80 2028	H'1F80 2028	32
CS4 Wait Control Register	CS4WCR	R/W	H'7777 770F	H'FF80 2048	H'1F80 2048	32
CS5 Wait Control Register	CS5WCR	R/W	H'7777 770F	H'FF80 2058	H'1F80 2058	32
CS6 Wait Control Register	CS6WCR	R/W	H'7777 770F	H'FF80 2068	H'1F80 2068	32
CS5 PCMCIA Control Register	CS5PCR	R/W	H'7700 0000	H'FF80 2070	H'1F80 2070	32
CS6 PCMCIA Control Register	CS6PCR	R/W	H'7700 0000	H'FF80 2080	H'1F80 2080	32

Table 11.8 Register State in Each Operating Made.

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Memory Address Map Select Register	MMSELR	H'0000 0000	H'0000 0000	Retained	Retained
Bus Control Register	BCR	H'x000 0000	H'x000 0000	Retained	Retained
CS0 Bus Control Register	CS0BCR	H'7777 7770	H'7777 7770	Retained	Retained
CS1 Bus Control Register	CS1BCR	H'7777 7770	H'7777 7770	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
CS2 Bus Control Register	CS2BCR	H'7777 7770	H'7777 7770	Retained	Retained
CS4 Bus Control Register	CS4BCR	H'7777 7770	H'7777 7770	Retained	Retained
CS5 Bus Control Register	CS5BCR	H'7777 7770	H'7777 7770	Retained	Retained
CS6 Bus Control Register	CS6BCR	H'7777 7770	H'7777 7770	Retained	Retained
CS0 Wait Control Register	CS0WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS1 Wait Control Register	CS1WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS2 Wait Control Register	CS2WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS4 Wait Control Register	CS4WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS5 Wait Control Register	CS5WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS6 Wait Control Register	CS6WCR	H'7777 770F	H'7777 770F	Retained	Retained
CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	H'7700 0000	Retained	Retained
CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	H'7700 0000	Retained	Retained

11.4.1 Memory Address Map Select Register (MMSELR)

The memory address map select register (MMSELR) is a 32-bit register that selects memory address maps for areas 2 to 5. This register should be accessed at the address H'FE60 0020 in longwords. Writing is accepted only when the upper 16-bit data is H'A5A5 to prevent unintentional writing. The upper 29 bits are always read as 0. This register is initialized by a power-on reset or a manual reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R/W	Reserved Set these bits to H'A5A5 only when writing to AREASEL bits in this register. These bits are always read as 0.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	AREASEL	000	R/W	DDRIF/PCIC Memory Space Select 000: Sets area 3 (H'0C00 0000 to H'0FFF FFFF) as the DDRIF space and other areas as the LBSC space 001: Sets area 3 (H'0C00 0000 to H'0FFF FFFF) as the DDRIF space, area 4 (H'1000 0000 to H'13FF FFFF) as the PCI memory space, and other areas as the LBSC space 010: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDRIF space and other areas as the LBSC space 011: Sets areas 2 and 3 (H'0800 0000 to H'0FFF FFFF) as the DDRIF space, area 4 (H'1000 0000 to H'13FF FFFF) as PCI memory space, and other areas as the LBSC space 100: Sets areas 2 to 5 (H'0800 0000 to H'17FF FFFF) as the DDRIF space 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

The MMSELR must be written by the CPU. Writing to MMSELR, DMAC and PCIC module should be set not to access this register, and all processing in execution should be finished, for example execute SYNCO instruction preceding MOV instruction, and then modify MMSELR.

And execute twice MOV instruction of read out MMSELR (dummy read) and SYNCO instruction in succession immediately after MOV instruction of write to MMSELR.

Example:

```
------------------------------------------------------------------------
MOV.L   #H'FE600020, R0         ;
MOV.L   #MMSELR_DATA, R1        ; MMSELR_DATA=Writing value of MMSELR
SYNCO                           ;                   (upper word=H'A5A5)
MOV.L   R1, @R0                 ; Writing to MMSELR
MOV.L    @R0, R2
MOV.L    @R0, R2
SYNCO
------------------------------------------------------------------------
```

Modify executing instruction of MMSELR should allocate non-cacheable P2 area and the address that should not be affected by address map change.

Write to MMSELR before enable Instruction cache, Operand cache, and MMU address translation and after this never write again until execute power-on reset or manual reset.

11.4.2 Bus Control Register (BCR)

The bus control register (BCR) is a 32-bit readable/writable register that specifies the function and bus cycle status for each area. It is initialized to H'0000 0000 in big-endian mode or H'8000 0000 in little-endian mode by a power-on reset or a mammal reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ENDIAN	—	—	—	—	DPUP	—	OPUP	DACKBST[3:0]				—	—	BREQEN	DMABST
Initial value:	0/1*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31	ENDIAN	Undefined	R	Endian Flag The value of the external pin (MD5) designating the endian mode is sampled at a power-on reset by the $\overline{\text{PRESET}}$ pin. This bit determines the endian mode of all spaces. 0: Indicates that the external pin (MD5) designating the endian mode is low at a power-on reset and big-endian mode is specified for this LSI. 1: Indicates that the external pin (MD5) designating the endian mode is high at a power-on reset and little-endian mode is specified for this LSI.
30 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	DPUP	0	R/W	Data Pin Pull-Up Resistor Control Specifies the pull-up resistor state of the data pins (D31 to D0). This bit is initialized by a power-on reset. The pins are not pulled up when access is performed or when the bus is released, even if the pull-up resistor is on. Also, in standby mode, pins D31 to D16 are not pulled up regardless of the setting. 0: Cycles in which the pull-up resistors of the data pins (D31 to D0) are turned on are inserted before and after a memory access.* 1: Pull-up resistor is off for data pins (D31 to D0). Note: * We recommend that a pull-up resistor be externally connected to the data pins if it is required.
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
24	OPUP	0	R/W	Control Output Pin Pull-Up Resistor Control Specifies the pull-up resistor state (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{RD}$, $\overline{WEn}$, $\overline{RDWR}$, $\overline{CE2A}$, and $\overline{CE2B}$) when the control output pins are high-impedance. This bit is initialized by a power-on reset. 0: Pull-up resistors are on for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{RD}$, $\overline{WEn}$, $\overline{RDWR}$, $\overline{CE2A}$, and $\overline{CE2B}$) 1: Pull-up resistors are off for control output pins (A25 to A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{RD}$, $\overline{WEn}$, $\overline{RDWR}$, $\overline{CE2A}$, and $\overline{CE2B}$) Note: In standby mode, the control output pins are pulled up, regardless of the bit setting.
23 to 20	DACKBST [3:0]	All 0	R/W	DACK Burst Select assert period of $\overline{DACKn}$ signals of DMA burst transfer mode during DMA transfer start to end. 0: $\overline{DACKn}$ signals does not keep assert from burst start to end 1: $\overline{DACKn}$ signals keep assert from burst start to end DACKBST[3]: $\overline{DACK3}$ DACKBST[2]: $\overline{DACK2}$ DACKBST[1]: $\overline{DACK1}$ DACKBST[0]: $\overline{DACK0}$
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	BREQEN	0	R/W	$\overline{BREQ}$ Enable Indicates whether or not an external bus request can be accepted. This bit is initialized to the state where an external bus request is not accepted at a power-on reset. 0: An external bus request is not accepted 1: An external bus request is accepted

Bit	Bit Name	Initial Value	R/W	Description
16	DMABST	0	R/W	DMAC Burst Mode Transfer Priority Setting Specifies the priority of burst mode transfers by the DMAC. When this bit is cleared to 0, the priority is as follows: bus release, DMAC, CPU. When this bit is set to 1, the bus release is not performed until the completion of the DMAC burst transfer. This bit is initialized at a power-on reset. 0: DMAC burst mode transfer priority setting off 1: DMAC burst mode transfer priority setting on
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	HIZCNT	0	R/W	High Impedance (Hi-Z) Control Specifies the state of signals $\overline{\text{WEn}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ during the software standby mode and the bus-released state. 0: Signals of $\overline{\text{WEn}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ are high-impedance during the bus-released state 1: Signals of $\overline{\text{WEn}}$ and $\overline{\text{RD}}/\overline{\text{FRAME}}$ are output during the bus-released state
13 to 7	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	ASYNC[6:0]	All 0	R/W	Asynchronous Input Enable asynchronous input to the corresponding pins. 0: Input signals to the corresponding pins are synchronized with CLKOUT 1: Input signals to the corresponding pins are asynchronous to CLKOUT ASYNC[6]: $\overline{\text{DREQ3}}$ ASYNC[5]: $\overline{\text{DREQ2}}$ ASYNC[4]: $\overline{\text{DREQ1}}$ ASYNC[3]: $\overline{\text{DREQ0}}$ ASYNC[2]: $\overline{\text{IOIS16}}$ ASYNC[1]: $\overline{\text{BREQ}}$ ASYNC[0]: $\overline{\text{RDY}}$

11.4.3 CSn Bus Control Register (CSnBCR)

CSnBCR is a 32-bit readable/writable register that specifies the bus width for area n (n = 0 to 2 and 4 to 6), numbers of wait, setup, and hold cycles to be inserted, burst length, and memory types.

Some types of memory continue to drive the data bus immediately after the read signal is inactivated. Therefore, a data bus collision may occur when there is consecutive memory access to different areas or writing to a memory immediately after reading. This LSI automatically inserts the number of idle cycles set by CSnBCR to prevent data bus collision. During idle cycles, corresponding signals $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{RD}$, $\overline{WE}$, $\overline{CE2A}$, $\overline{CE2B}$, and $\overline{BS}$ are not asserted and $\overline{RDWR}$ is in the high state and the data is not driven.

CSnBCR is initialized to H'7777 7770 by a power-on reset or a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		IWW				IWRWD				IWRWS				IWRRD		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		IWRRS			BST		SZ		RDSPL	BW			MPX	TYPE		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W*	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW	111	R/W	Idle Cycles between Write-Read/Write-Write Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are write-read cycles and write-write cycles. For details, see section 11.5.8, Wait Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	IWRWD	111	R/W	Idle Cycles between Read-Write to Different Spaces Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-write cycles to different spaces. For details, see section 11.5.8, Wait Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	IWRWS	111	R/W	Idle Cycles between Read-Write to Same Space Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-write cycles to the same space. For details, see section 11.5.8, Wait Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	IWRRD	111	R/W	Idle Cycles between Read-Read to Different Spaces Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-read cycles to different spaces. For details, see section 11.5.8, Wait Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	IWRRS	111	R/W	Idle Cycles between Read-Read to Same Space Specify the number of idle cycles to be inserted after an access to a memory connected to the space is completed. The target cycles are read-read cycles to the same space. For details, see section 11.5.8, Wait Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
11, 10	BST	01	R/W	Burst Length When a burst ROM interface is used, these bits specify the number of accesses in a burst. The MPX interface is not affected. 00: 4 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 01: 8 consecutive accesses (Can be used with 8-, 16-, or 32-bit bus width) 10: 16 consecutive accesses (Can be used with 8-, or 16-bit bus width) 11: 32 consecutive accesses (Can be used with 8-bit bus width)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	SZ	11	R/W*	Bus Width Specify the bus width. In CS0BCR, the external pins (MD3 and MD4) are sampled at a power-on reset. Set to 11 for the MPX interface, and set to 10 or 11 for the byte control SRAM interface. 00: Reserved 01: 8 bits 10: 16 bits 11: 32 bits Note: * Bits SZ in CS0BCR are read-only. The SZ bits in CS0BCR are set to 11 when area 0 is set to MPX interface by the MD3 and MD4 pins.
7	RDSPL	0	R/W	$\overline{\text{RD}}$ Hold Cycle Specify the number of cycles to be inserted into the $\overline{\text{RD}}$ assertion period to ensure the data hold time to the read data sample timing. When set this bit to 1, specify the number of $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycle to be 1 or more by setting the RDH bit in CSnWCR. And $\overline{\text{RD}}$ negation-$\overline{\text{CS}}$n negation delay cycle is reduced 1 cycle to set this bit to 1 (Available only when the SRAM interface or byte control SRAM interface). 0: No hold cycle inserted 1: 1 hold cycle inserted
6 to 4	BW	111	R/W	Burst Pitch When the burst ROM interface is used, these bits specify the number of wait cycles to be inserted after the second data access in a burst transfer. 000: No idle cycle inserted, $\overline{\text{RDY}}$ signal disabled 001: 1 idle cycle inserted, $\overline{\text{RDY}}$ signal enabled 010: 2 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 011: 3 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 100: 4 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 101: 5 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 110: 6 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled 111: 7 idle cycles inserted, $\overline{\text{RDY}}$ signal enabled

Bit	Bit Name	Initial Value	R/W	Description
3	MPX	0	R/W*	MPX Interface Setting Selects the type of MPX interface 0: SRAM/byte-control SRAM interface selected 1: MPX interface selected Note: * The MPX bit in CS0BCR is read-only.
2 to 0	TYPE	000	R/W	Memory Type Setting Specify the type of memory connected to the space. 000: SRAM (Initial value) 001: SRAM with byte selection*1 010: Burst ROM (burst at read/SRAM at write) 011: Reserved (Setting prohibited) 100: PCMCIA *2 101: Reserved (Setting prohibited) 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited) Note: 1. Setting possible only in CS1BCR and CS4BCR 2. Setting possible only in CS5BCR and CS6BCR

11.4.4 CSn Wait Control Register (CSnWCR)

The CSn wait control register (CSnWCR) is a 32-bit readable/writable register that specifies the number of wait cycles to be inserted, the pitch of data access for burst memory accesses, and the number of cycles to be inserted for the address setup time to the read/write strobe assertion or for the data hold time from the write strobe negation.

This allows direct connection of even low-speed memories without an external circuit.

CSnBCR is initialized to H'7777 770F by a power-on reset or a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ADS			—	ADH			—	RDS			—	RDH		
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WTS			—	WTH			—	BSH			IW[3:0]			
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	1	1	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	ADS	111	R/W	Address Setup Cycle Specify the number of cycles to be inserted to ensure the address setup time to the $\overline{\text{CSn}}$ assertion. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	ADH	111	R/W	Address Hold Cycle Specify the number of cycles to be inserted to ensure the address hold time to the $\overline{CSn}$ negation. However, setting to over one cycle, one cycle decremented from the setting value when $\overline{RD}$ strobe cycle in read access or $\overline{WE}$ strobe cycle in write access is set to over 1 cycle. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) Note that, it will be no inserted cycle when setting to 0 for inserted wait cycle and setting to 0 for $\overline{RD}$ strobe hold wait in read access or $\overline{WE}$ strobe hold wait in write access. 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22 to 20	RDS	111	R/W	$\overline{RD}$ Setup Cycle ($\overline{CSn}$ Assert–$\overline{RD}$ Assert Delay Cycle) Specify the number of cycles to be inserted to ensure the $\overline{RD}$ setup time to the T1. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	RDH	111	R/W	$\overline{RD}$ Hold Cycle ($\overline{RD}$ Negate–$\overline{CSn}$ Negate Delay Cycle) Specify the number of cycles to be inserted to ensure the $\overline{RD}$ hold time to the T2. However, setting to over 1 cycle for insertion, one cycle incremented for the setting value when address hold cycle is set to over 1 cycle. (Available only when the SRAM interface, byte control an SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	WTS	111	R/W	$\overline{WEn}$ Setup Cycle ($\overline{CSn}$ Assert–$\overline{WEn}$ Assert Delay Cycle) Specify the number of cycles to be inserted to ensure the $\overline{WE}$ setup time to the T2. (Available only when the SRAM interface, byte control an SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	WTH	111	R/W	$\overline{\text{WE}}$n Hold Cycle ($\overline{\text{WE}}$n Negate–$\overline{\text{CS}}$n Negate Delay Cycle) Specify the number of cycles to be inserted to ensure the $\overline{\text{WE}}$n hold time to the T2. However, setting to over 1 cycle for insertion, one cycle incremented for the setting value when address hold cycle is set to over 1 cycle. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	BSH	000	R/W	$\overline{\text{BS}}$ Hold Cycle Specify the number of cycles to be inserted to elongate the $\overline{\text{BS}}$ assertion time. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 000: 1 cycle inserted 001: 2 cycles inserted 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	IW[3:0]	1111	R/W	Insert Wait Cycle Specify the number of wait cycles to be inserted. (Available only when the SRAM interface, byte control SRAM interface, or burst ROM interface is selected.) 0000: No cycle inserted 0001: 1 cycle inserted 0010: 2 cycles inserted 0011: 3 cycles inserted 0100: 4 cycles inserted 0101: 5 cycles inserted 0110: 6 cycles inserted 0111: 7 cycles inserted 1000: 8 cycles inserted 1001: 9 cycles inserted 1010: 11 cycles inserted 1011: 13 cycles inserted 1100: 15 cycles inserted 1101: 17 cycles inserted 1110: 21 cycles inserted 1111: 25 cycles inserted Note: IW[2:0] specify the number of wait cycles to be inserted into read and write cycles when MPX interface is selected. IW[1:0] specify the number of wait cycles to be inserted into first data. 00: 1 cycle inserted into read cycle and no cycle inserted into write cycle 01: 1 cycle inserted into read cycle and 1 cycle inserted into write cycle 10: 2 cycle inserted into read cycle and 2 cycle inserted into write cycle 11: 3 cycle inserted into read cycle and 3 cycle inserted into write cycle IW2 specifies the number of wait cycle to be inserted into second data or after. 0: No cycle inserted 1: 1 cycle inserted IW3: Reserved

11.4.5 CSn PCMCIA Control Register (CSnPCR)

CSnPCR is a 32-bit readable/writable register that specifies the timing for the PCMCIA interface connected to area n (n = 5 and 6), the space property, and the assert/negate timing for the $\overline{OE}$ and $\overline{WE}$ signals. The pulse widths of $\overline{OE}$ and $\overline{WE}$ are set using the wait control bits in CSnWCR. CSnPCR is initialized to H'7700 0000 by a power-on reset or a manual reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SAA			—	SAB			PCWA		PCWB		PCIW			
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	SAA	111	R/W	Space Property A Specify the space property of PCMCIA connected to first half of area 5 or 6. 000: ATA complement mode 001: Dynamic I/O bus sizing 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory
27	—	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
26 to 24	SAB	111	R/W	Space Property B Specify the space property of PCMCIA connected to second half of area 5 or 6. 000: ATA complement mode 001: Dynamic I/O bus sizing 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory
23, 22	PCWA	00	R/W	PCMCIA Wait A Wait cycle for low-speed PCMCIA. The number of wait cycles specified by these bits is added to the number designated by CSnWCR. These bits are valid, when the access area of PCMCIA interface is first half of area 5 or 6, 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 01: 50 wait cycles inserted
21, 20	PCWB	00	R/W	PCMCIA Wait B Wait cycle for low-speed PCMCIA. The number of wait cycles specified by these bits is added to the number designated by PCIW. These bits are valid, when the access area of PCMCIA interface is second half of area 5 or 6, 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 01: 50 wait cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
19 to 16	PCIW	0000	R/W	PCMCIA Insert Wait Cycle B Specify the number of wait cycles to be inserted. These bits are valid, when the access area of PCMCIA interface is second half of area 5 or 6, 0000: No cycle inserted 0001: 1 cycle inserted 0010: 2 cycles inserted 0011: 3 cycles inserted 0100: 4 cycles inserted 0101: 5 cycles inserted 0110: 6 cycles inserted 0111: 7 cycles inserted 1000: 8 cycles inserted 1001: 9 cycles inserted 1010: 11 cycles inserted 1011: 13 cycles inserted 1100: 15 cycles inserted 1101: 17 cycles inserted 1110: 21 cycles inserted 1111: 25 cycles inserted Note: Specify the number of wait cycle designated by CSnWCR when the access area of PCMCIA interface is first half of area 5 or 6.
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	TEDA	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay A These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion for the access of first half area of PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	TEDB	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay B These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion for the access of second half area of PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	TEHA	000	R/W	$\overline{OE}$/$\overline{WE}$ Negate-Address Delay A These bits set the delay time from $\overline{OE}$/$\overline{WE}$ negation to address hold for the access of first half area of PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TEHB	000	R/W	$\overline{OE}$/$\overline{WE}$ Negate-Address Delay B These bits set the delay time from $\overline{OE}$/$\overline{WE}$ negation to address hold for the access of second half area of PCMCIA interface. 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted

11.5 Operation

11.5.1 Endian/Access Size and Data Alignment

This LSI supports both big-endian mode, in which the upper byte (MSByte) in a string of byte data is at address 0, and little-endian mode, in which the lower byte (LSByte) in a string of byte data is at address 0. The mode is specified by the external pin (MD5 pin) at a power-on reset through the RESET pin. At a power-on reset by $\overline{\text{PRESET}}$, big-endian mode is specified when the MD5 pin is low, and little-endian mode is specified when the MD5 pin is high.

A data bus width of 8, 16, or 32 bits can be selected for the normal memory interface, and one of 8 or 16 bits can be selected for the PCMCIA interface. Data alignment is carried out according to the data bus width and endian mode of each device. Accordingly, when the data bus width is smaller than the access size, multiple bus cycles are automatically generated to reach the access size. In this case, access is performed by incrementing the addresses corresponding to the bus width. For example, when a longword access is performed at the area with an 8-bit width in the SRAM interface, each address is incremented one by one, and then access is performed four times. In the 32-byte transfer, a total of 32-byte data is continuously transferred according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during these transfers. In this LSI, data alignment and data length conversion between different interfaces is performed automatically.

The relationship between the endian mode, device data length, and access unit are shown in tables 11.9 to 11.14.

Data Configuration

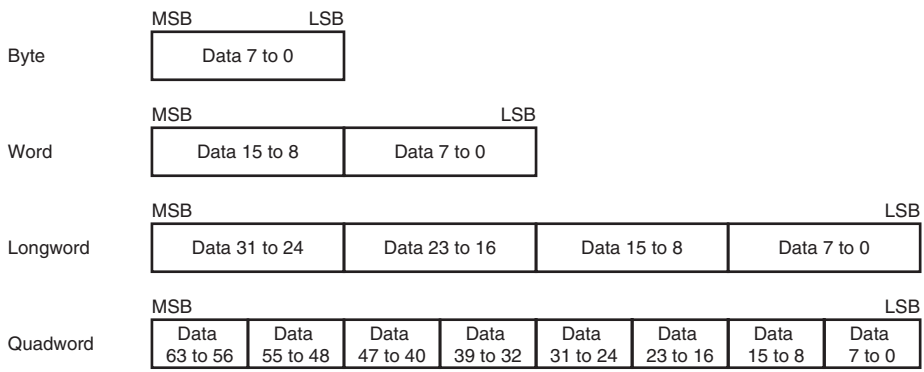


Table 11.9 32-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	Data 7 to 0	—	—	—	Assert			
	4n + 1	1	—	Data 7 to 0	—	—		Assert		
	4n + 2	1	—	—	Data 7 to 0	—			Assert	
	4n + 3	1	—	—	—	Data 7 to 0				Assert
Word	4n	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
	4n + 2	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.10 16-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	Data 7 to 0	—			Assert	
	2n + 1	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert
	4n + 2	2	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert

Table 11.11 8-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	—	Data 15 to 8				Assert
	2n + 1	2	—	—	—	Data 7 to 0				Assert
Longword	4n	1	—	—		Data 31 to 24				Assert
	4n + 1	2	—	—	—	Data 23 to 16				Assert
	4n + 2	3	—	—	—	Data 15 to 8				Assert
	4n + 3	4	—	—	—	Data 7 to 0				Assert

Table 11.12 32-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	4n	1	—	—	—	Data 7 to 0				Assert
	4n + 1	1	—	—	Data 7 to 0	—			Assert	
	4n + 2	1	—	Data 7 to 0	—	—		Assert		
	4n + 3	1	Data 7 to 0	—	—	—	Assert			
Word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	4n + 2	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 11.13 16-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	2n	1	—	—	—	Data 7 to 0				Assert
	2n + 1	1	—	—	Data 7 to 0	—			Assert	
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	4n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert

Table 11.14 8-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
Byte	n	1	—	—	—	Data 7 to 0				Assert
Word	2n	1	—	—	—	Data 7 to 0				Assert
	2n + 1	2	—	—	—	Data 15 to 8				Assert
Longword	4n	1	—	—	—	Data 7 to 0				Assert
	4n + 1	2	—	—	—	Data 15 to 8				Assert
	4n + 2	3	—	—	—	Data 23 to 16				Assert
	4n + 3	4	—	—	—	Data 31 to 24				Assert

11.5.2 Areas

(1) Area 0

For area 0, external address bits A28 to A26 are 000.

The interfaces that can be set for this area are the SRAM, MPX, and burst ROM interfaces.

A bus width of 8, 16, or 32 bits is selectable with external pins MD4 and MD3 at a power-on reset. For details, see section 11.3.2, Memory Bus Width.

When area 0 is accessed, the $\overline{CS0}$ signal is asserted. In addition, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS0WCR can be selected.

When the burst ROM interface is used, a burst pitch number in the range of 0 to 7 is selectable with bits BW[2:0] in CS0BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{RDY}$). (When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

When the burst ROM interface is used, the number of transfer cycles for a burst cycle is selected from a range of 2 to 9 according to the number of wait cycles.

The setup time and hold time (cycle number) of the address and $\overline{CS0}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS0WCR. The $\overline{BS}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

(2) Area 1

For area 1, external address bits A28 to A26 are 001.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM and byte-control SRAM interfaces.

A bus width of 8, 16, or 32 bits is selectable with bits SZ[1:0] in CS1BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ[1:0] in CS1BCR. When using the byte-control SRAM interface, select a bus width of 16 or 32 bits.

When area 1 is accessed, the $\overline{CS1}$ signal is asserted. In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and write control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS1WCR can be selected.

When the burst ROM interface is used, a burst pitch number in the range of 0 to 7 is selectable with bits BW[2:0] in CS1BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS1}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS1WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

(3) Area 2

For area 2, external address bits A28 to A26 are 010.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM and DDR-SDRAM interfaces.

When the SRAM interface is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ[1:0] in CS2BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ[1:0] in CS2BCR.

When area 2 is accessed, the $\overline{\text{CS2}}$ signal is asserted (except for DDR-SDRAM area). In the case where the SRAM interface is set, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS2WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS2}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS2WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

When the DDR-SDRAM is used, see section 12, DDR-SDRAM Interface (DDRIF).

(4) Area 3

For area 3, external address bits A28 to A26 are 011.

This area is used only for the DDR-SDRAM interface.

For details, see section 12, DDR-SDRAM Interface (DDRIF).

(5) Area 4

For area 4, physical address bits A28 to A26 are 100.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM, byte control SRAM, DDR-SDRAM and PCI interfaces.

A bus width of 8, 16, or 32 bits is selectable with bits SZ [1:0] in CS4BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ[1:0] in CS4BCR. When the byte control SRAM interface is used, select a bus width of 16 or 32 bits. For details, see section 11.3.2, Memory Bus Width.

When area 4 is accessed, the $\overline{\text{CS4}}$ signal is asserted (except for DDR-SDRAM and PCI areas). In the case where the SRAM interface is set, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS4WCR can be selected. Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS4}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS4WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

When the DDR-SDRAM or PCI is used, see section 12, DDR-SDRAM Interface (DDRIF) or section 13, PCI Controller (PCIC), respectively.

(6) Area 5

For area 5, external address bits A28 to A26 are 101.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM, PCMCIA, and DDR-SDRAM interfaces.

When the SRAM or burst ROM interface is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ[1:0] in CS5BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ[1:0] in CS5BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ[1:0] in CS5BCR. For details, see section 11.3.2, Memory Bus Width.

While the SRAM interface is used, the $\overline{\text{CS5}}$ signal is asserted when area 5 is accessed. In addition, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

While the PCMCIA interface is used, the $\overline{\text{CE1A}}$/$\overline{\text{CS5}}$ and $\overline{\text{CE2A}}$ signals, the $\overline{\text{RD}}$ signal, (which can be used as $\overline{\text{OE}}$), the $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, and $\overline{\text{WE3}}$ signals, (which can be used as, $\overline{\text{PCC_REG}}$, $\overline{\text{WE}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$, respectively) are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS5WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS5}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS5WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{\text{CE1A}}$/$\overline{\text{CS5}}$ and $\overline{\text{CE2A}}$) can be specified within a range from 0 to 15 cycles through bits TEDA[2:0], TEDB[2:0], TEHA[2:0], and TEHB[2:0] in CS5PCR. In addition, the number of wait cycles can be specified within a range from 0 to 50 cycles through bits PCWA[1:0] and PCWB[1:0] The number of wait cycles specified by CS5PCR is added to the value specified by IW[3:0] in CS5WCR or PCIW[3:0] in CS5PCR.

When the DDR-SDRAM is used, see section 12, DDR-SDRAM Interface (DDRIF).

(7) Area 6

For area 6, external address bits A28 to A26 are 110.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM, and PCMCIA interfaces.

When the SRAM or burst ROM is used, a bus width of 8, 16, or 32 bits is selectable with bits SZ[1:0] in CS6BCR. When the MPX interface is used, a bus width of 32 bits should be selected through bits SZ[1:0] in CS6BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 bits with SZ[1:0] in CS6BCR. For details, see section 11.3.2, Memory Bus Width.

While the SRAM interface is used, the $\overline{\text{CS6}}$ signal is asserted when area 6 is accessed. In addition, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted. While the PCMCIA interface is used, the $\overline{\text{CE1B}}$/$\overline{\text{CS6}}$ and $\overline{\text{CE2B}}$ signals, the $\overline{\text{RD}}$ signal (which can be used as $\overline{\text{OE}}$), and the $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, and $\overline{\text{WE3}}$ signals (which can be used as $\overline{\text{PCC_REG}}$, $\overline{\text{WE}}$, $\overline{\text{IORD}}$, and $\overline{\text{IOWR}}$, respectively) are asserted.

As regards the number of bus cycles, 0 to 25 wait cycles inserted by CS6WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{\text{RDY}}$). (When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS6}}$ signals to the read and write strobe signals can be set within a range of 0 to 7 cycles by CS6WCR. The $\overline{\text{BS}}$ hold cycles can be set within a range of 0 to 1 when the number of read and write strobe setup wait is 1 or more.

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{\text{CE1B}}$/$\overline{\text{CS6}}$ and $\overline{\text{CE2B}}$) can be specified within a range from 0 to 15 cycles by bits TEDA[2:0], TEDB[2:0], TEHA[2:0], and TEHB[2:0] in CS6PCR. In addition, the number of wait cycles can be specified within a range from 0 to 50 cycles by bits PCWA[1:0] and PCWB[1:0]. The number of wait cycles specified by CS6PCR is added to the value specified by IW[3:0] in CS6WCR or PCIW[3:0] in CS6PCR.

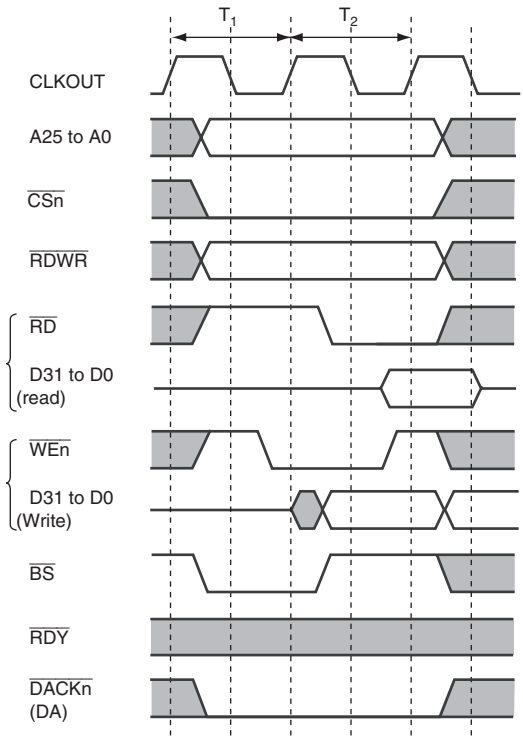
11.5.3 SRAM interface

(1) Basic Timing

The strobe signals for the SRAM interface of this LSI are output primarily based on the SRAM connection. Figure 11.4 shows the basic timing of the SRAM interface. A no-wait normal access is completed in two cycles. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{\text{CSn}}$ signal is asserted at the rising edge of the clock in the T1 state, and negated at the next rising edge of the clock in the T2 state. Therefore, there is no negation period in the case of access at minimum pitch.

During reading, specification of an access size is not needed. The output of an access address on the address pins (A25 to A0) is correct, however, since the access size is not specified, 32-bit data is always output when a 32-bit device is in use, and 16-bit data is output when a 16-bit device is in use. During writing, only the $\overline{\text{WE}}$ signal corresponding to the byte to be written is asserted. For details, see section 11.5.1, Endian/Access Size and Data Alignment.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the bus width set. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during this transfer.



DA: Dual address DMA

Figure 11.4 Basic Timing of SRAM Interface

Figures 11.5 to 11.7 show examples of the connection to SRAM with data width of 32, 16, and 8 bits.

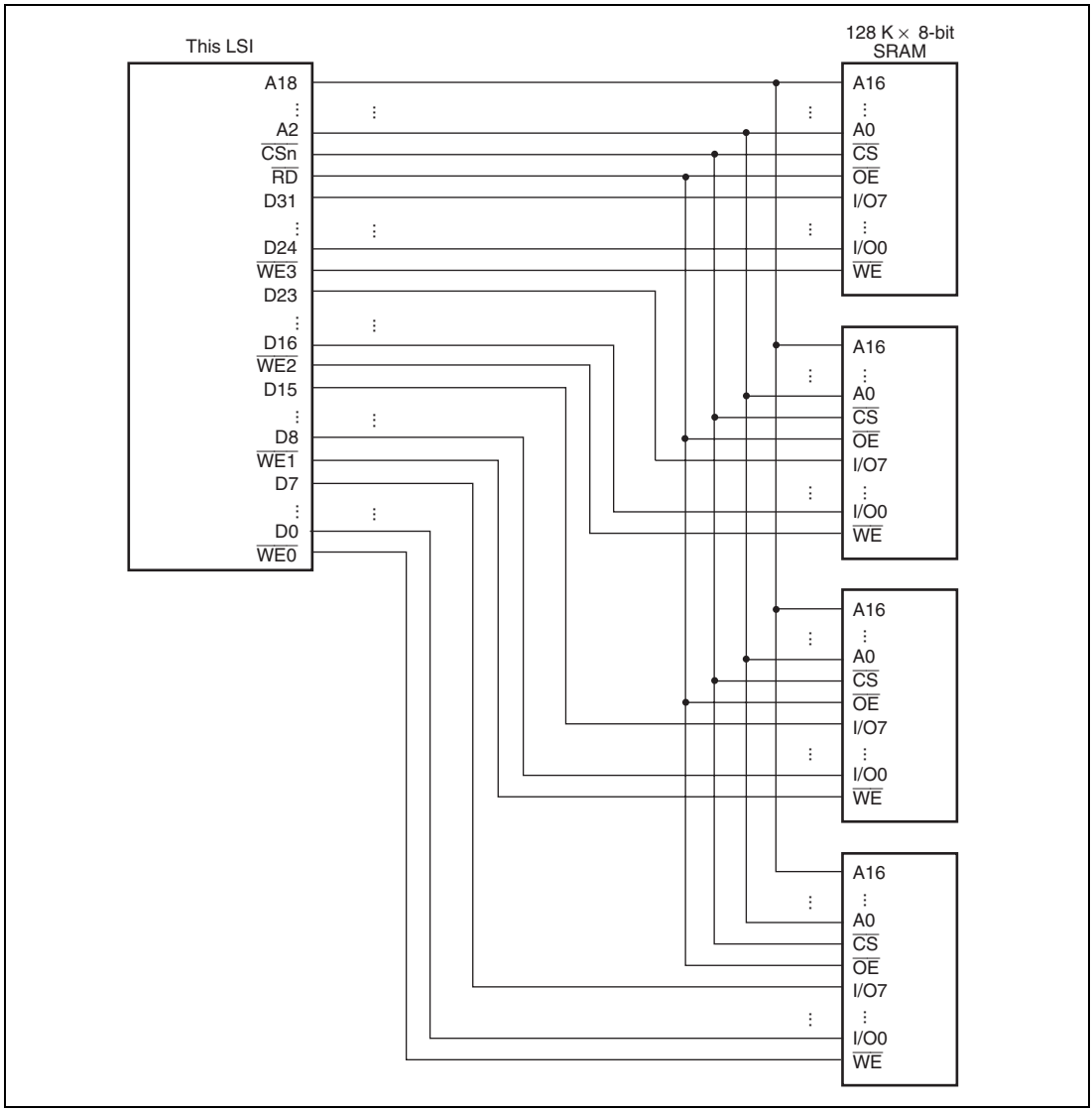


Figure 11.5 Example of 32-Bit Data-Width SRAM Connection

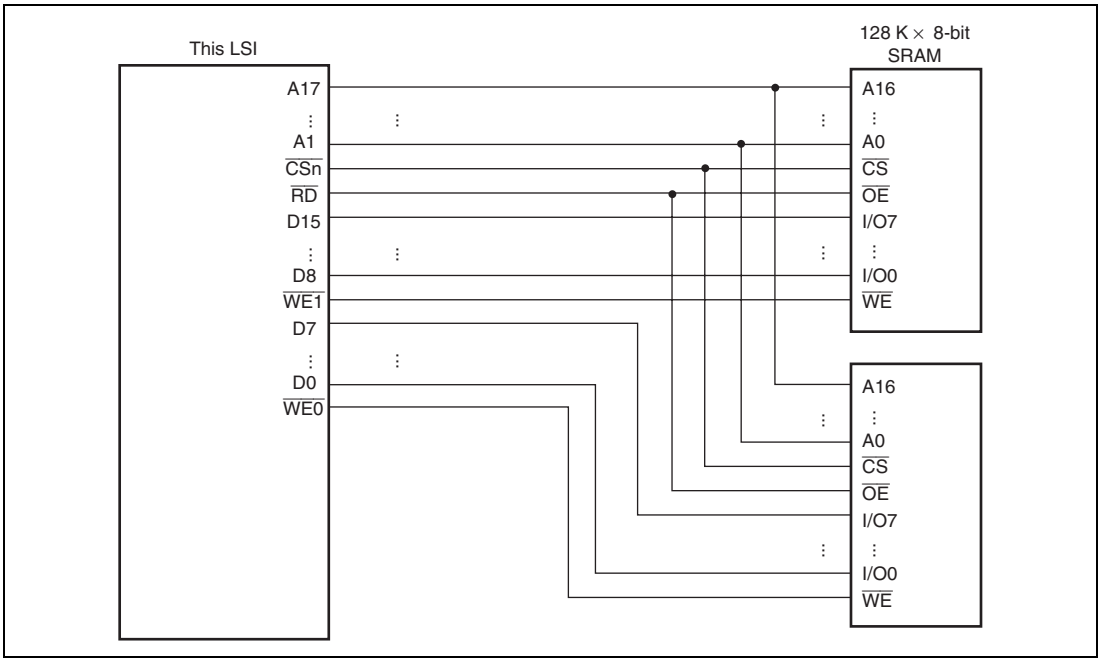


Figure 11.6 Example of 16-Bit Data-Width SRAM Connection

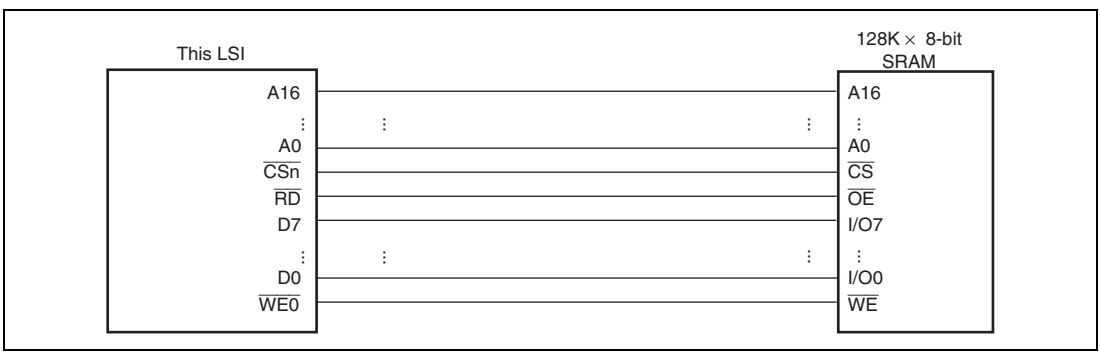


Figure 11.7 Example of 8-Bit Data-Width SRAM Connection

(2) Wait Cycle Control

Wait cycle insertion for the SRAM interface can be controlled by CSnWCR. If the IW bits for each area in CSnWCR is not 0, a software wait is inserted in accordance with the wait-control bits. For details, see section 11.4.4, CSn Wait Control Register (CSnWCR).

A specified number of Tw cycles is inserted as wait cycles in accordance with the CSnWCR setting. The insertion timing of the wait cycle is shown in figure 11.8.

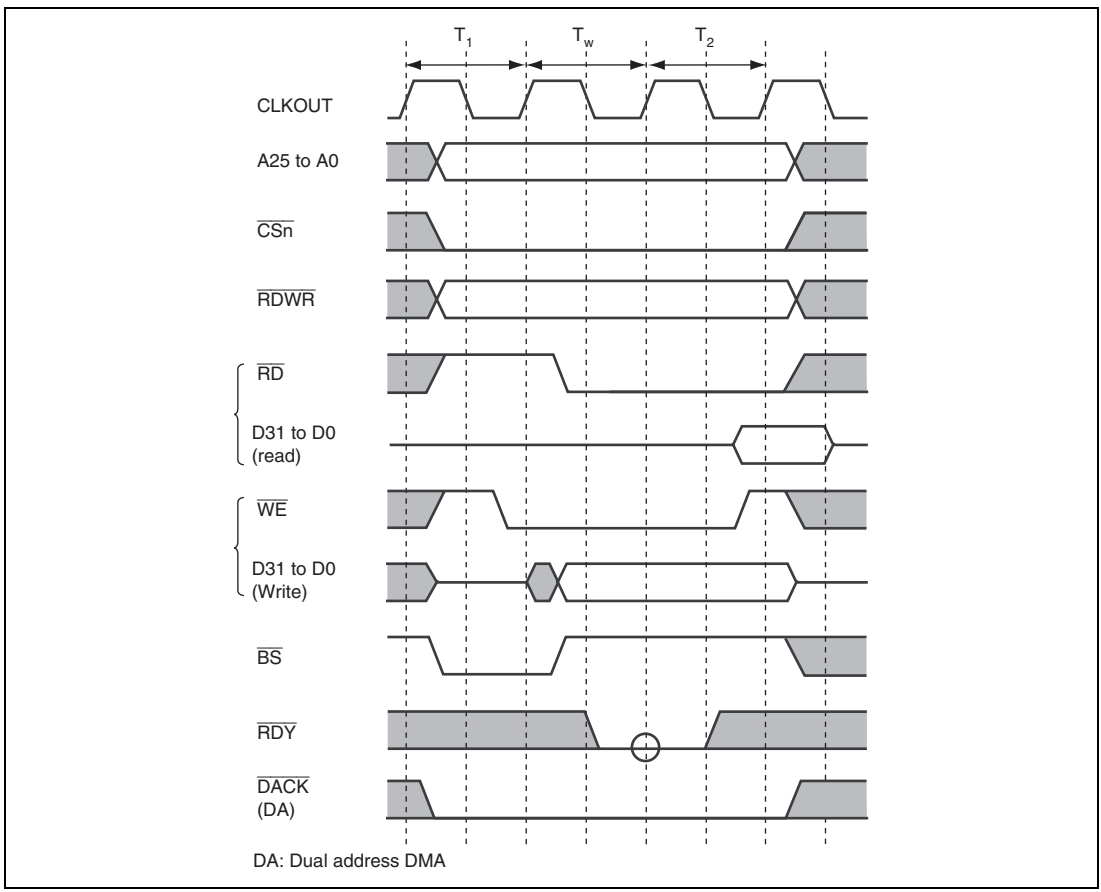


Figure 11.8 SRAM Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by CSnWCR, the external wait input signal ($\overline{RDY}$) is also sampled. The $\overline{RDY}$ signal sampling timing is shown in figure 11.9, where a single wait cycle is specified as a software wait. The $\overline{RDY}$ signal is sampled at the transition from the Tw state to the T2 state. Therefore, the assertion of the $\overline{RDY}$ signal has no effect in the T1 cycle or in the first Tw cycle. The $\overline{RDY}$ signal is sampled on the rising edge of the clock.

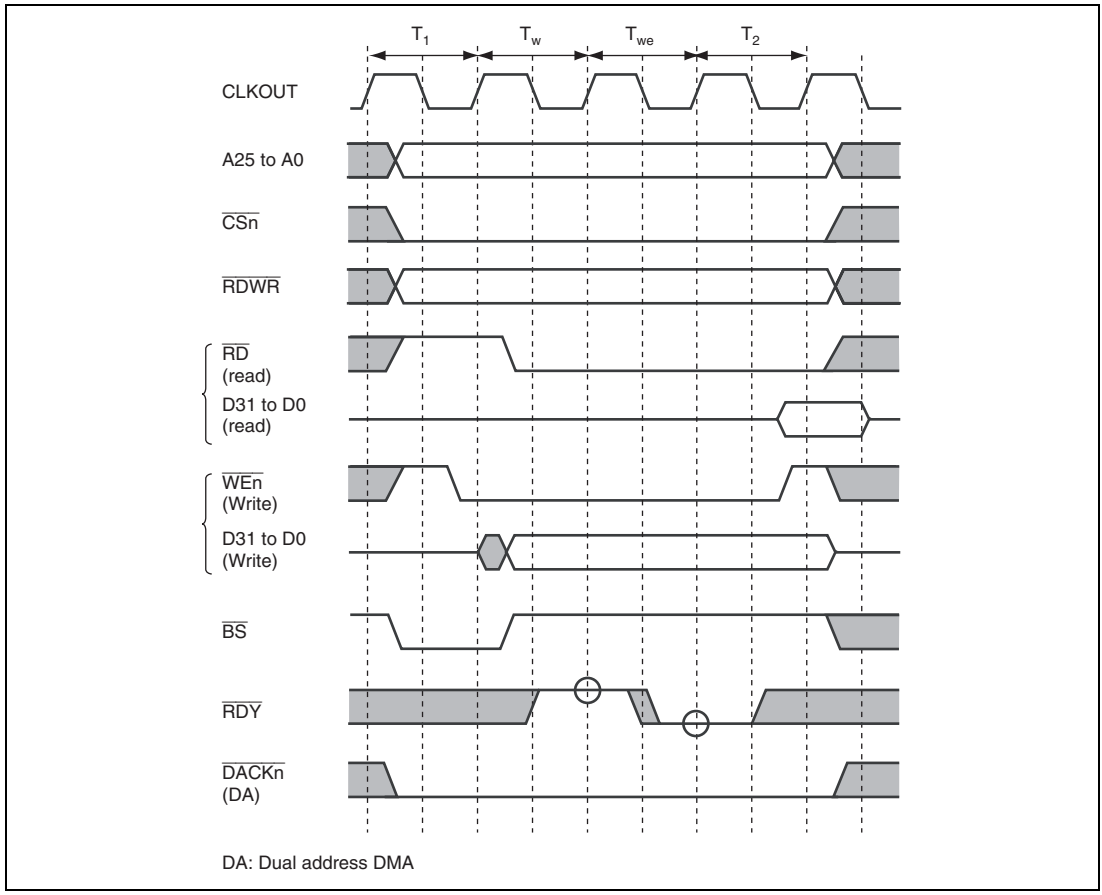
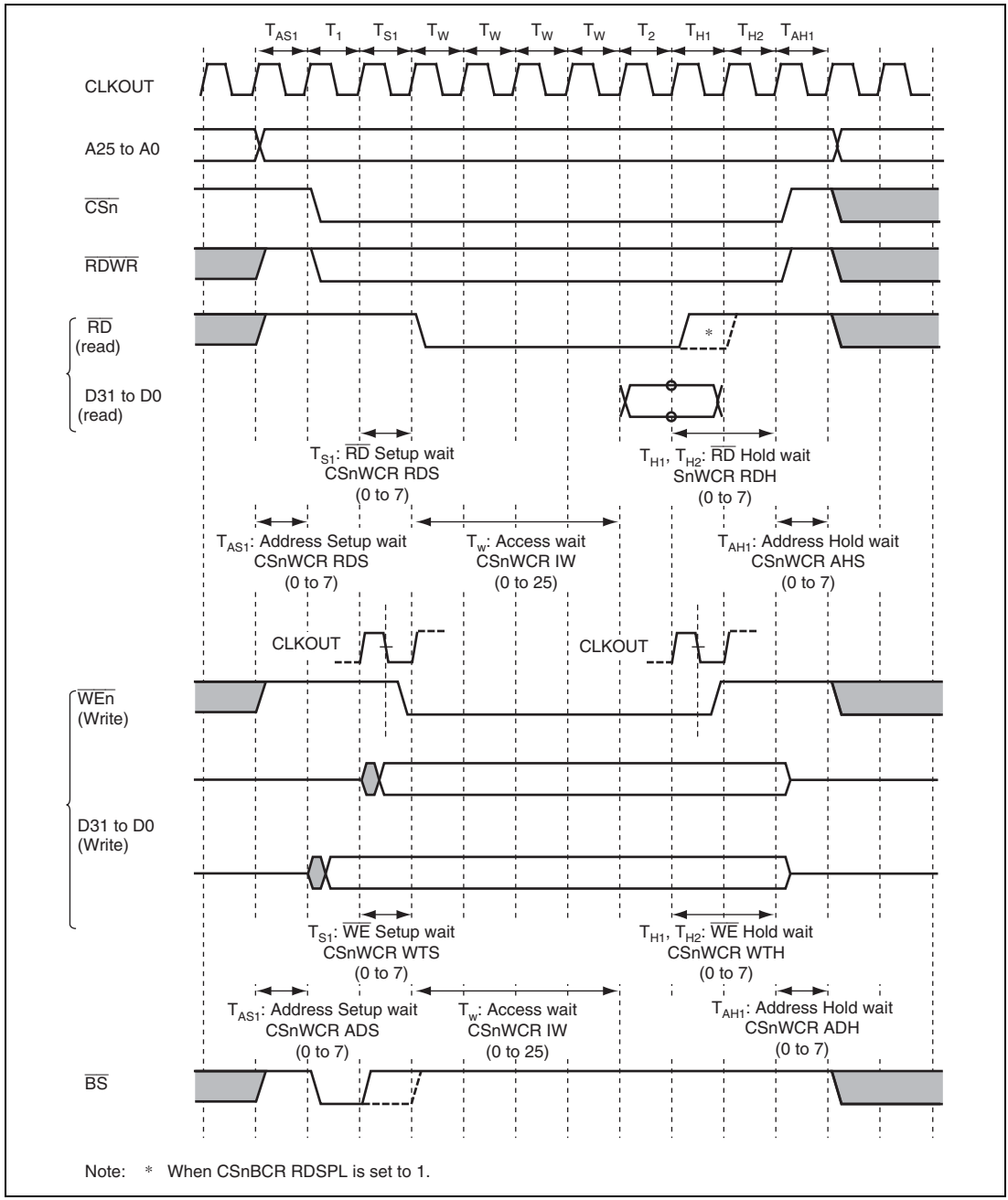


Figure 11.9 SRAM Interface Wait Cycle Timing (Wait Cycle Insertion by $\overline{RDY}$ Signal)

(3) Read-Strobe Negate Timing

When the SRAM interface is used, the negation timing of the strobe signal during a read operation can be specified through the RDH bit in CSnWCR.



Note: * When CSnBCR RDSPL is set to 1.

Figure 11.10 SRAM Interface Wait State Timing (Read-Strobe Negate Timing Setting)

11.5.4 Burst ROM Interface

Setting the TYPE bit in CSnBCR(n=0 to 2 and 4 to 6) to 010 allows a burst ROM to be connected to areas 0 to 2 and 4 to 6. The burst ROM interface provides high-speed access to ROM that has a burst access function. The burst access timing of burst ROM is shown in figure 11.11. The wait cycle is set to 0 cycle. Although the access is similar to that of the SRAM interface, only the address is changed when the first cycle ends and then the next access is started. When 8-bit ROM is used, the number of consecutive accesses can be set as 4, 8, 16, or 32 through bits BST[2:0] in CSnBCR(n=0 to 2 and 4 to 6). Similarly, when 16-bit ROM is used, 4, 8 or 16 accesses can be set; when 32-bit ROM is used, 4 or 8 accesses can be set.

The $\overline{\text{RDY}}$ signal is always sampled when one or more wait cycles are set. Even when no wait is specified in the burst ROM settings, two access cycles are inserted in the second and subsequent accesses as shown in figure 11.12.

A writing operation for the burst ROM interface is performed in the same way as for the SRAM interface.

In a 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during this transfer.

Figure 11.13 shows the timing chart when the burst ROM is used and setup/hold is specified by CSnWCR.

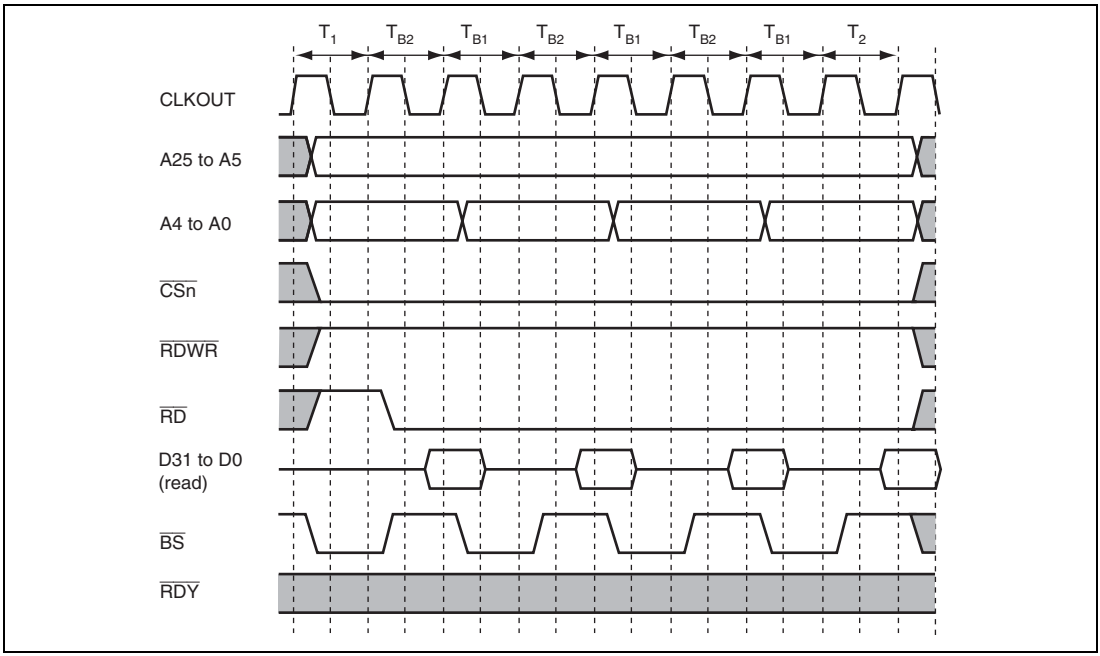


Figure 11.11 Burst ROM Basic Access Timing

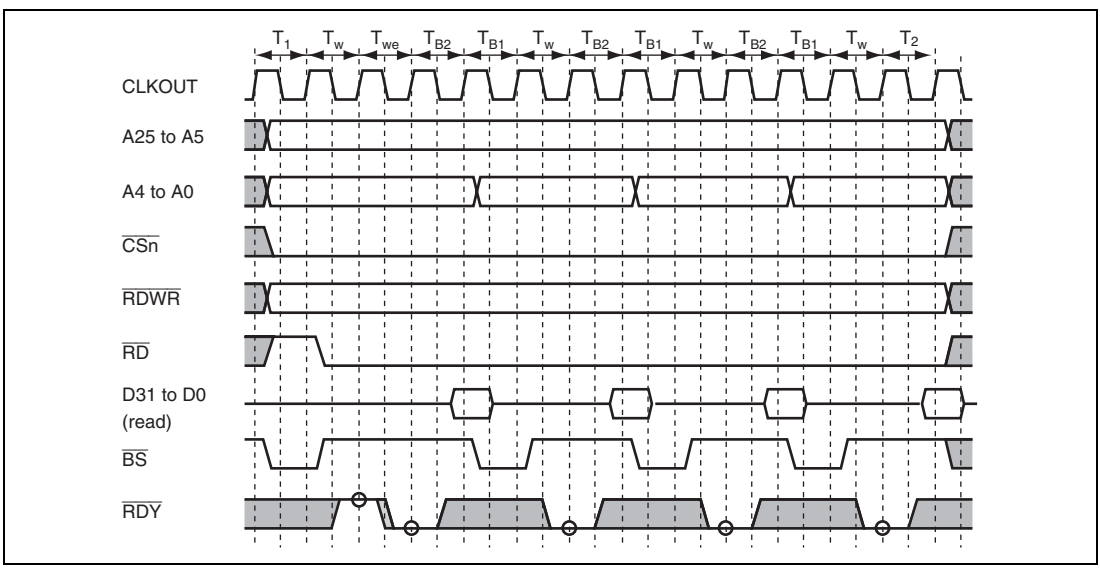


Figure 11.12 Burst ROM Wait Access Timing

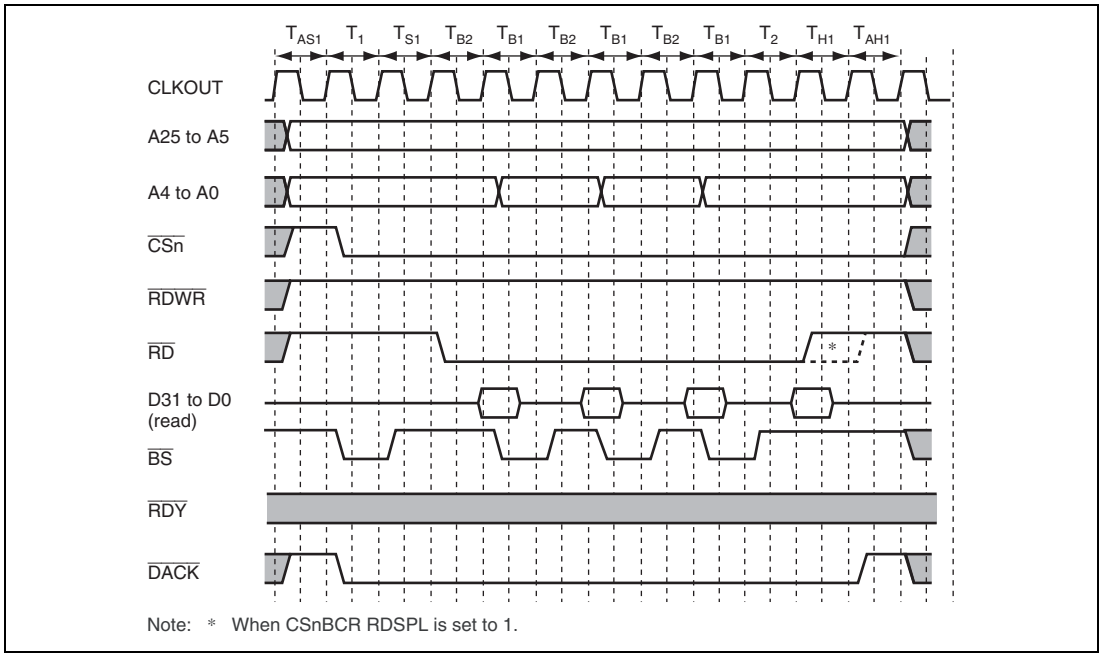


Figure 11.13 Burst ROM Wait Access Timing

11.5.5 PCMCIA Interface

Areas 5 and 6 can be set to the IC memory card interface or I/O card interface, which is stipulated in JEIDA specification version 4.2 (PCMCIA 2.1), by setting bits TYPE[2:0] in CS5BCR and CS6BCR.

Figure 11.14 shows an example of PCMCIA card connection to this LSI. To enable hot insertion of PCMCIA cards (i.e., insertion or removal while system power is being supplied), a three-state buffer must be connected between this LSI bus interface and the PCMCIA cards.

Since operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA standard, this LSI supports the PCMCIA interface only in little-endian mode through little-endian mode setting.

PCMCIA interface can select space property from among 8-bit common memory, 16-bit common memory, 8-bit attribute memory, 16-bit attribute memory, 8-bit I/O space, 16-bit I/O space, dynamic I/O bus sizing, and ATA complement mode by setting bits SAA[2:0] and SAB[2:0] in CSnPCR.

When the first half area is accessed, bits IW[3:0] in CSnWCR (n=5 or 6) and bits PCWA[1:0], TEDA[2:0], and TEHA[2:0] in CSnPCR (n=5 or 6) are selected. When the second half area is accessed, bits IW[3:0] in CSnWCR (n=5 or 6) and bits PCWB[1:0], TEDB[2:0], and TEHB[2:0] in CSnPCR (n=5 or 6) are selected.

Bits PCWA[1:0] and PCWB[1:0] can be used to set the number of wait cycles to be inserted in a low-speed bus cycle as 0, 15, 30, or 50. This value is added to the number of inserted wait cycles specified by IW bit in CSnWCR or PCIW bit in CSnPCR. Bits PEDA[2:0] and PEDB[2:0] (with a setting range from 0 to 15) can be used to ensure the setup times of the address, $\overline{\text{CSn}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, and $\overline{\text{PCC_REG}}$ to the $\overline{\text{RD}}$ and $\overline{\text{WE1}}$ signals. Bits TEHA[2:0] and TEHB[2:0] (with a setting range from 0 to15) can be used to ensure the hold times of the address, $\overline{\text{CSn}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, and $\overline{\text{PCC_REG}}$ $\overline{\text{REG}}$ to the $\overline{\text{RD}}$ and $\overline{\text{WE1}}$ signals.

Bits IW[3:0] in the CS5 bus control register (CS5BCR) or CS6 bus control register (CS6BCR) are used to set the number of idle cycles between cycles. The selected number of wait cycles between cycles depends only on the area to be accessed (area 5 or 6). When area 5 is accessed, bits IW[3:0] in CS5WCR are selected, and when area 6 is accessed, bits IW[3:0] in CS6WCR are selected.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during this transfer.

ATA complement mode is to access the ATA device register that connected to this LSI. Device Control Register, Alternate Status Register, Data Register, and Data Port can be accessed in ATA complement mode.

To access Device Control Register and Alternate Status Register, PIO byte access is used, and to access Data Register, PIO word access is used. When PIO byte access is executed, $\overline{\text{CE1x}}$ is negated and $\overline{\text{CE2x}}$ is asserted. When PIO word access is executed, $\overline{\text{CE1x}}$ is asserted and $\overline{\text{CE2x}}$ is negated.

To access Data Port is used DMA transfer. Then DMAC must be set burst mode, level detection, overrun 0, DACK output to the correspondent PCMCIA connected area, and set to 1 DACKBST[2:0] bit in BCR of correspondent DMA transfer channels. When DMA transfer of ATA complement mode area is executed, both $\overline{\text{CE1x}}$ and $\overline{\text{CE2x}}$ are not asserted.

And set to 1 the DACKBST bit in BCR of correspondent DMA transfer channel, then the correspondent $\overline{\text{DACK}}$ signal is being asserted from the first to the end of the DMA transfer cycle.

Specify the number of wait cycles between accesses to be 0 for the $\overline{\text{DACK}}$ assertion area, when setting the size of DMA transfer is 16-byte. After the DMA burst transfer has finished, that DACKBST was enabled, set the DACKBST bit to 1 again before starting the next DMA transfer.

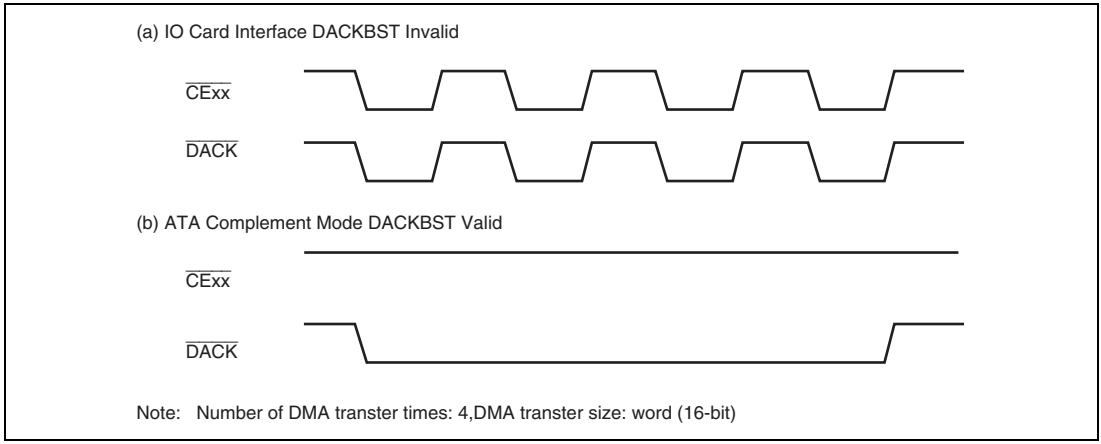


Figure 11.14 $\overline{\text{CExx}}$ and $\overline{\text{DACK}}$ Output of ATA Complete Mode in DMA Transfer

Table 11.15 Relationship between Address and CE When Using PCMCIA Interface

Bus (Bits)	Read/ Write	Access Size (bits)*1	Odd/ Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
8	Read	8	Even	×	—	1	0	0	Invalid	Read data
			Odd	×	—	1	0	1	Invalid	Read data
		16	Even	×	First	1	0	0	Invalid	Lower read data
			Even	×	Second	1	0	1	Invalid	Upper read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	1	0	0	Invalid	Write data
			Odd	×	—	1	0	1	Invalid	Write data
		16	Even	×	First	1	0	0	Invalid	Lower write data
			Even	×	Second	1	0	1	Invalid	Upper write data
			Odd	×	—	—	—	—	—	—
16	Read	8	Even	×	—	1	0	0	Invalid	Read data
			Odd	×	—	0	1	1	Read data	Invalid
		16	Even	×	—	0	0	0	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	1	0	0	Invalid	Write data
			Odd	×	—	0	1	1	Write data	Invalid
		16	Even	×	—	0	0	0	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
Dynamic Bus Sizing*2	Read	8	Even	0	—	1	0	0	Invalid	Read data
			Odd	0	—	0	1	1	Read data	Invalid
		16	Even	0	—	0	0	0	Upper read data	Lower read data
			Odd	0	—	—	—	—	—	—
	Write	8	Even	0	—	1	0	0	Invalid	Write data
			Odd	0	—	0	1	1	Write data	Invalid
		16	Even	0	—	0	0	0	Upper write data	Lower write data
			Odd	0	—	—	—	—	—	—
	Read	8	Even	1	—	1	0	0	Invalid	Read data
			Odd	1	First	0	1	1	Invalid	Invalid
			Odd	1	Second	1	0	1	Invalid	Read data
		16	Even	1	First	0	0	0	Invalid	Lower read data
			Even	1	Second	1	0	1	Invalid	Upper read data
			Odd	1	—	—	—	—	—	—

Bus (Bits)	Read/ Write	Access Size (bits)*1	Odd/ Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
Dynamic Bus Sizing*2	Write	8	Even	1	—	1	0	0	Invalid	Write data
			Odd	1	First	0	1	1	Invalid	Write data
			Odd	1	Second	1	0	1	Invalid	Write data
		16	Even	1	First	0	0	0	Upper write data	Lower write data
			Even	1	Second	1	0	1	Invalid	Upper write data
			Odd	1	—	—	—	—	—	—
ATA comple-ment mode	PIO read	8	Even	×	—	0	1	0	Invalid	Read data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	1	0	0	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	PIO write	8	Even	×	—	0	1	0	Invalid	Write data
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	1	0	0	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—
	DMA read	8	Even	×	—	0	1	0	Invalid	Read data
			Odd	×	—	1	1	0	Read data	Invalid
		16	Even	×	—	1	1	1	Upper read data	Lower read data
			Odd	×	—	—	—	—	—	—
	DMA write	8	Even	×	—	1	1	0	Invalid	Write data
			Odd	×	—	1	1	0	Write data	Invalid
		16	Even	×	—	1	1	1	Upper write data	Lower write data
			Odd	×	—	—	—	—	—	—

[Legend]

×: Don't care

Notes: 1. In 32-bit/64-bit/32-byte transfer, the addresses are automatically incremented by the bus width, and then above accesses are repeated until the transfer data size is reached.

2. PCMCIA I/O card interface only.

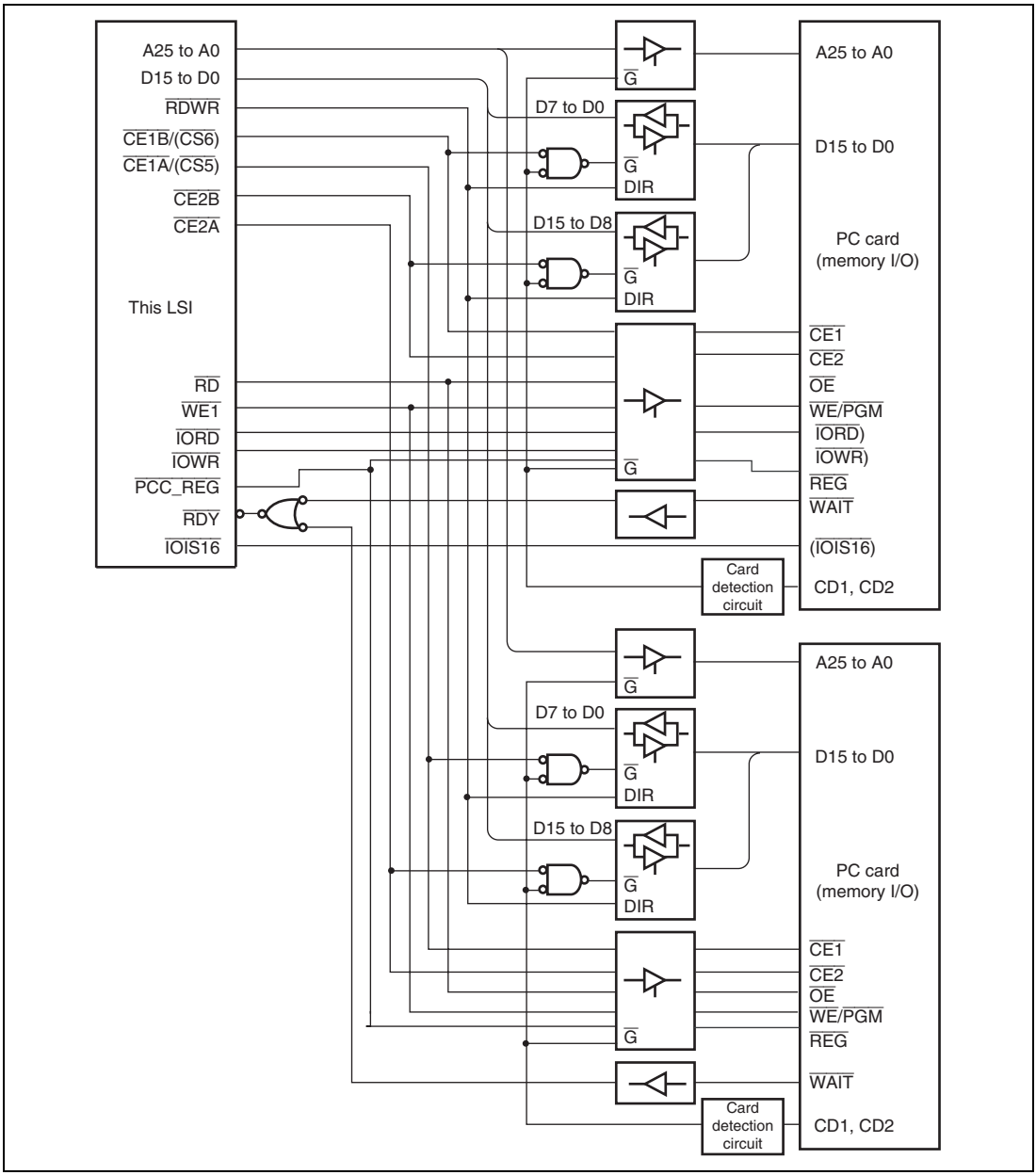


Figure 11.15 Example of PCMCIA Interface

(1) Memory Card Interface Basic Timing

Figure 11.16 shows the basic timing for the PCMCIA memory card interface, and figure 11.17 shows the wait timing for the PCMCIA memory card interface.

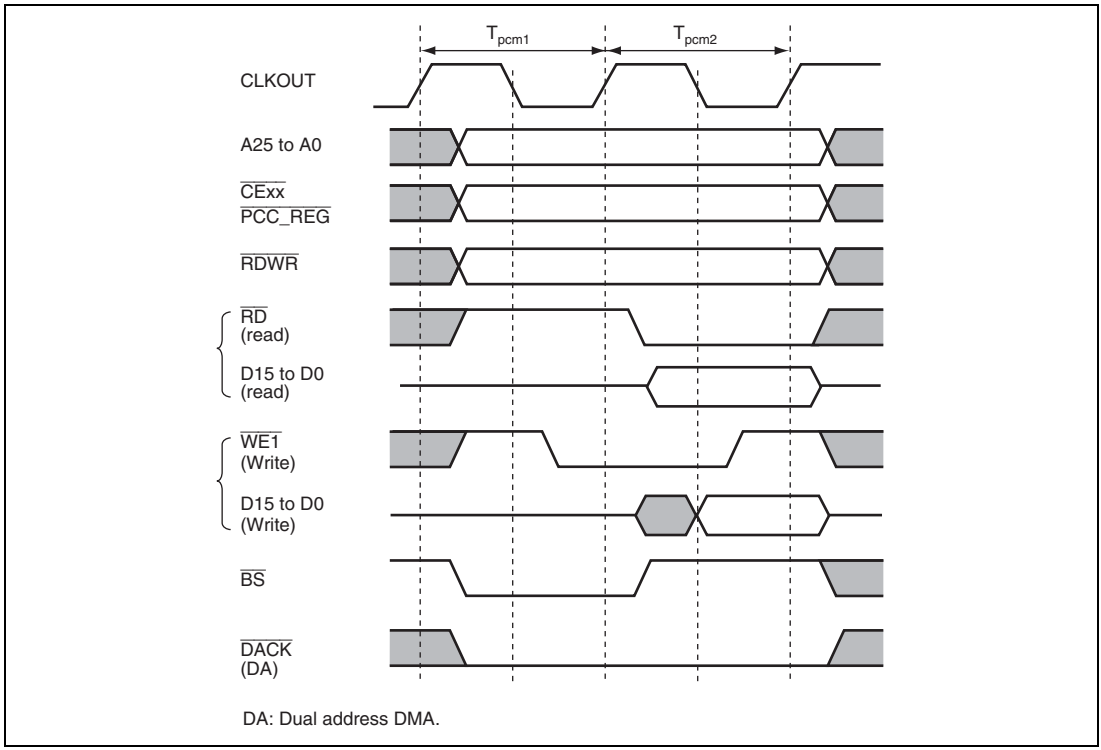


Figure 11.16 Basic Timing for PCMCIA Memory Card Interface

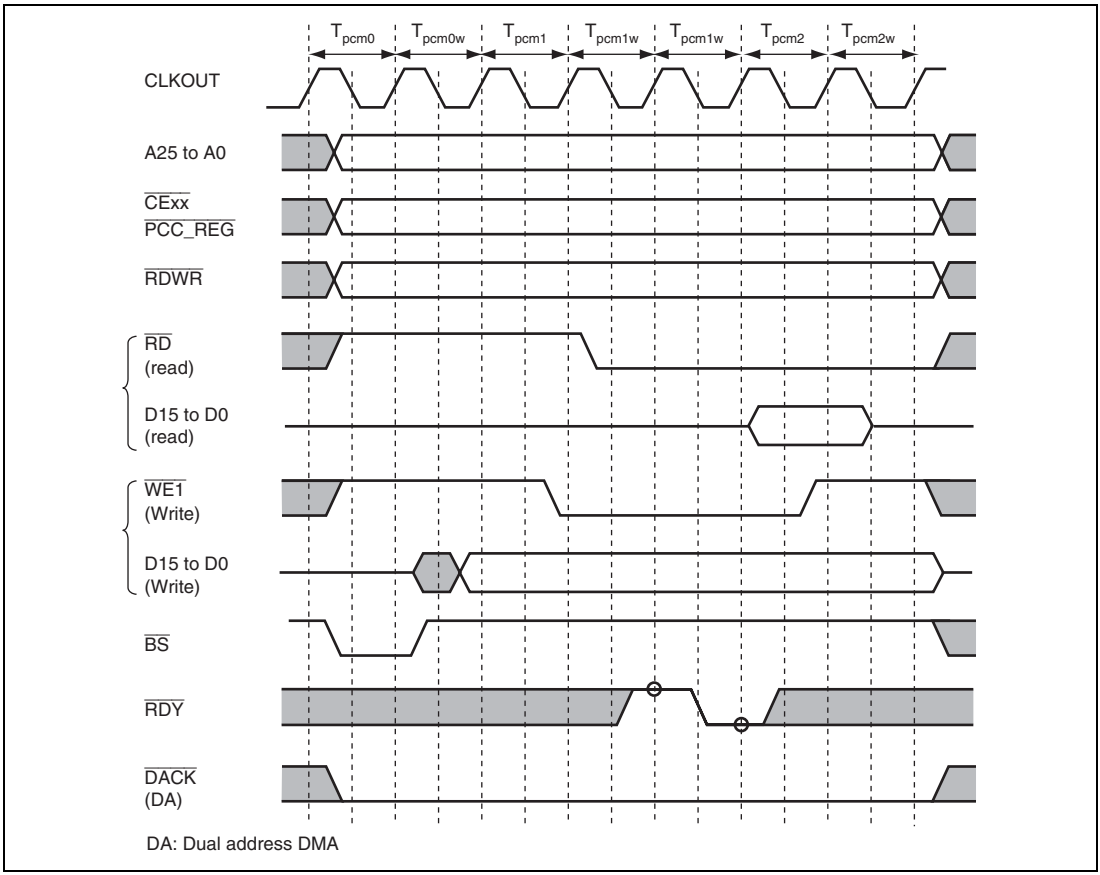


Figure 11.17 Wait Timing for PCMCIA Memory Card Interface

(2) I/O Card Interface Timing

Figures 11.18 and 11.19 show the timing for the PCMCIA I/O card interface.

When accessing a PCMCIA card via the I/O card interface, it is possible to perform dynamic sizing of the I/O bus width using the $\overline{\text{IOIS16}}$ pin. With the 16-bit bus width selected, if the $\overline{\text{IOIS16}}$ signal is high during the word-size I/O bus cycle, the I/O port is recognized as eight bits in bus width. In this case, a data access for only eight bits is performed in the I/O bus cycle being executed, and this is automatically followed by a data access for the remaining eight bits. Dynamic bus sizing is also performed for byte-size access to address 2n + 1.

Figure 11.20 shows the basic timing for dynamic bus sizing.

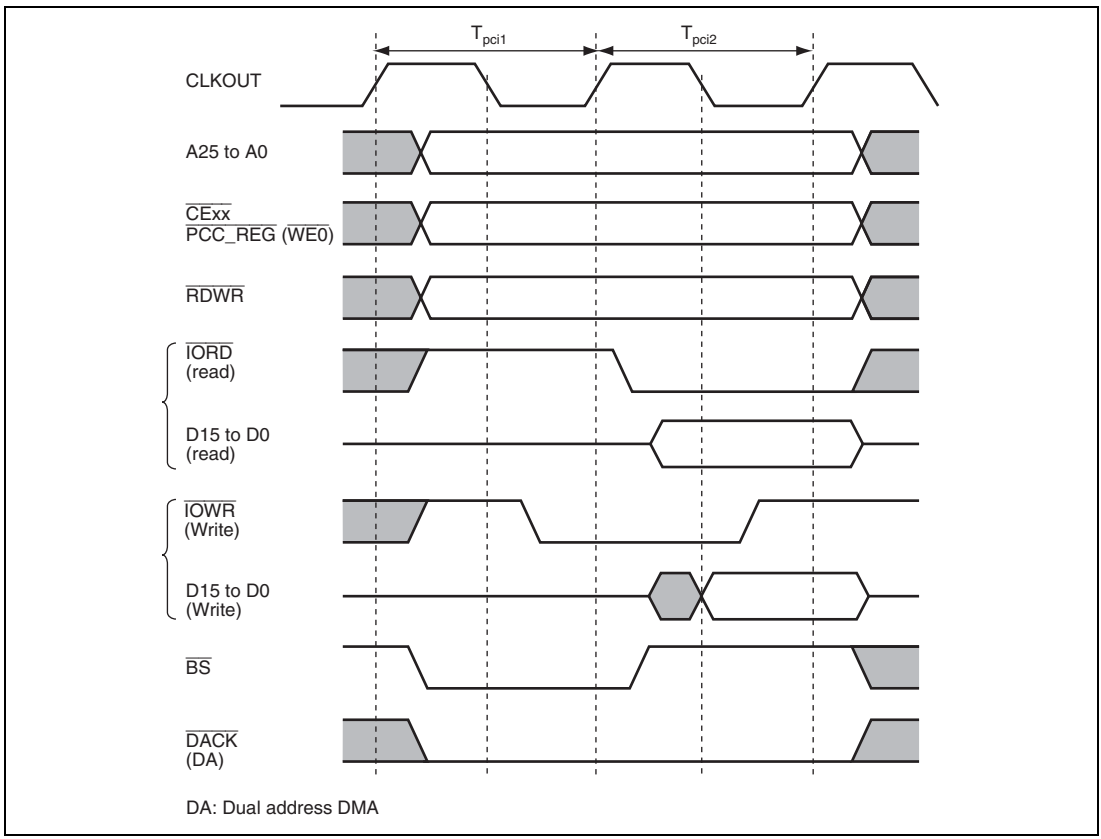


Figure 11.18 Basic Timing for PCMCIA I/O Card Interface

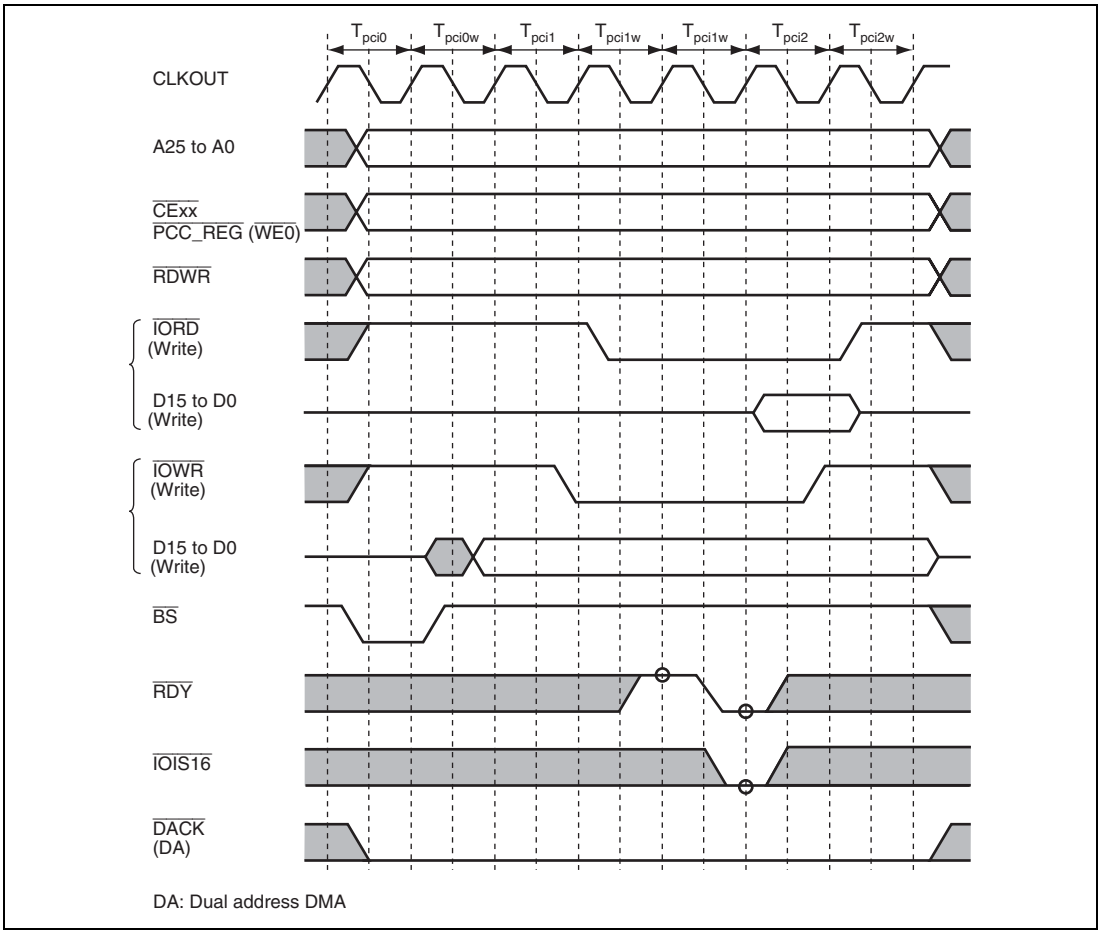


Figure 11.19 Wait Timing for PCMCIA I/O Card Interface

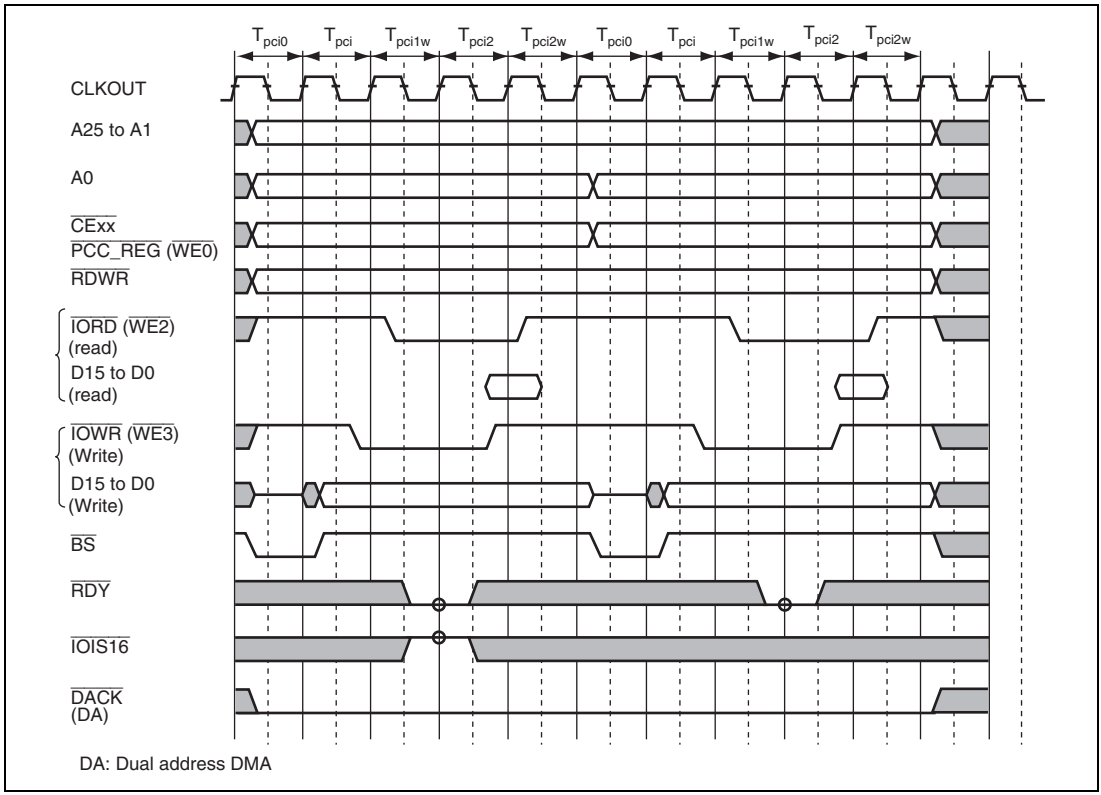


Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

11.5.6 MPX Interface

When both the MODE4 and MODE3 pins are set to 0 at a power-on reset by the $\overline{\text{PRESET}}$ pin, the MPX interface is selected for area 0. The MPX interface is selected for areas 1, 2, and 4 to 6 by the MPX bit in CS1BCR, CS2BCR, and CS4BCR to CS6BCR. The MPX interface provides an address/data multiplex-type bus protocol and facilitates connection with external memory controller chips using an address/data multiplex-type 32-bit single bus. A bus cycle consists of an address phase and a data phase. Address information is output on D25 to D0 and the access size is output on D31 to D29 in the address phase. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the address phase. The $\overline{\text{CSn}}$ signal is asserted at the rising edge in Tm1 and is negated after the end of the last data transfer in the data phase. Therefore, a negation cycle does not occur in the case of minimum pitch access. The $\overline{\text{FRAME}}$ signal is asserted at the rising edge in Tm1 and negated at the start of the last data transfer cycle in the data phase. Therefore, an external device for the MPX interface must internally store the address information and access size output in the address phase and perform data input/output for the data phase. For details, see section 11.5.1, Endian/Access Size and Data Alignment.

Values output on address pins A25 to A20 are not guaranteed.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed according to the set bus width. If the access size is larger than the bus width in this case, a burst access with continuing multiple data cycle occurs after one address output. The bus is not released during this transfer.

D31	D30	D29	Access Size
0	0	0	Byte
		1	Word
	1	0	Longword
		1	Quadword
1	X	X	32-byte burst

[Legend]
X: Don't care

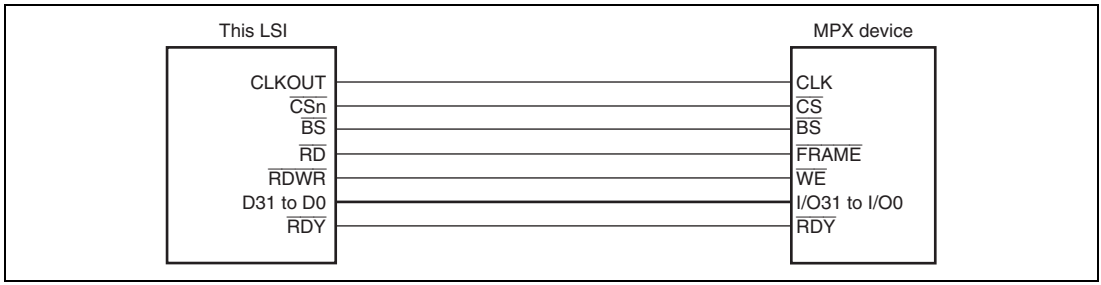


Figure 11.21 Example of 32-Bit Data Width MPX Connection

The MPX interface timing is shown below.

When the MPX interface is used for areas 1, 2, and 4 to 6, a bus size of 32 bits should be specified by CSnBCR.

In wait control, either waits by CSnWCR or waits by the RDY pin can be inserted.

In a read, one wait cycle is automatically inserted after address output, even if CSnWCR is cleared to 0.

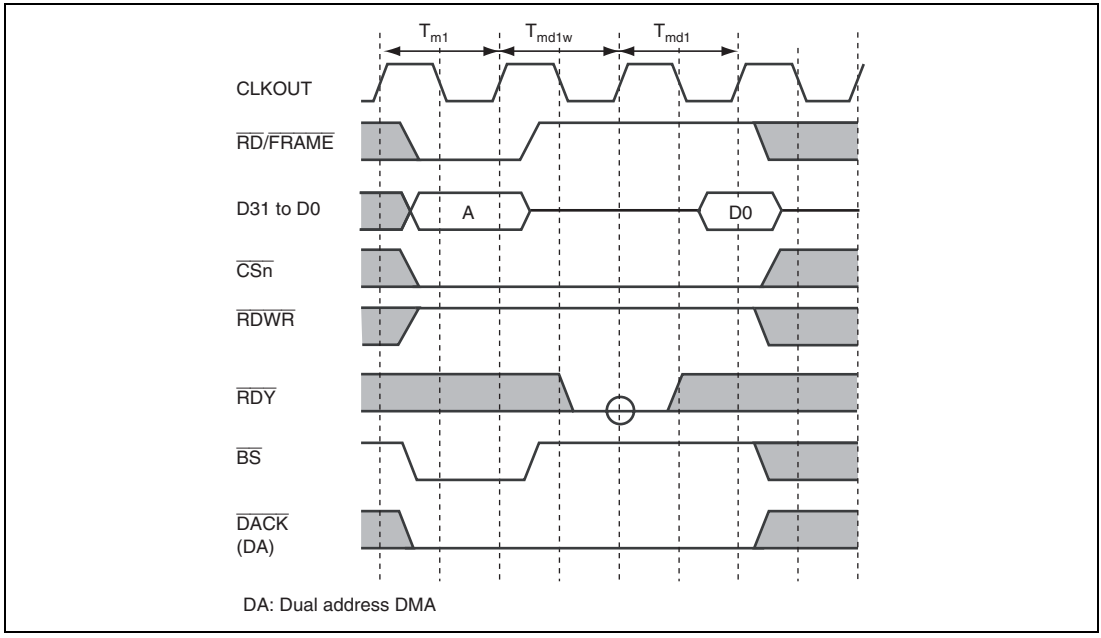


Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External Wait)

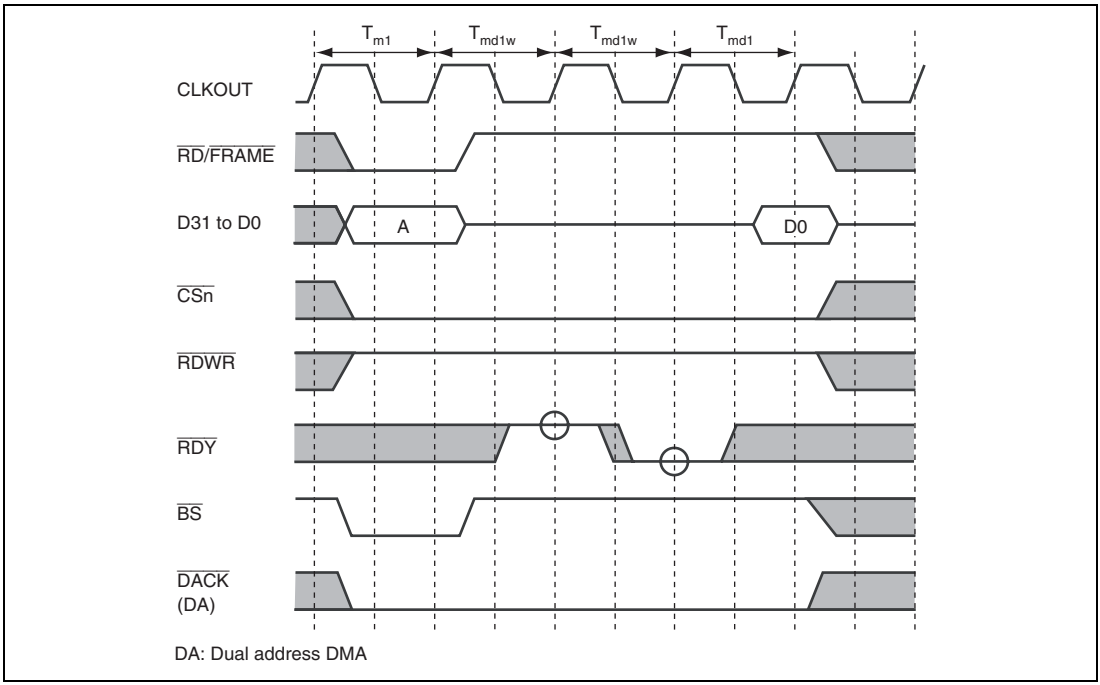


Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait Inserted)

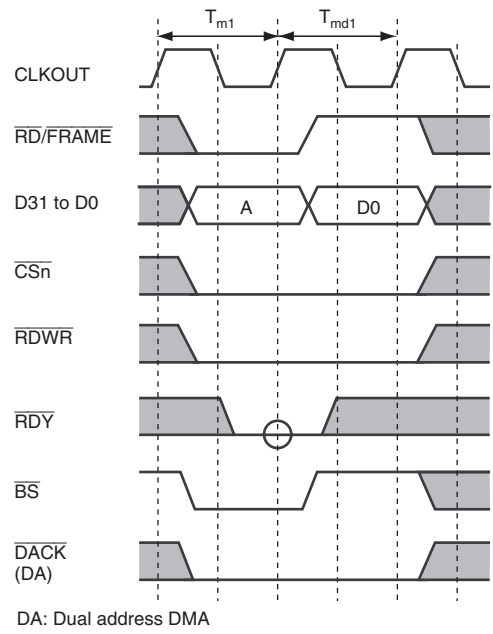


Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No External Wait)

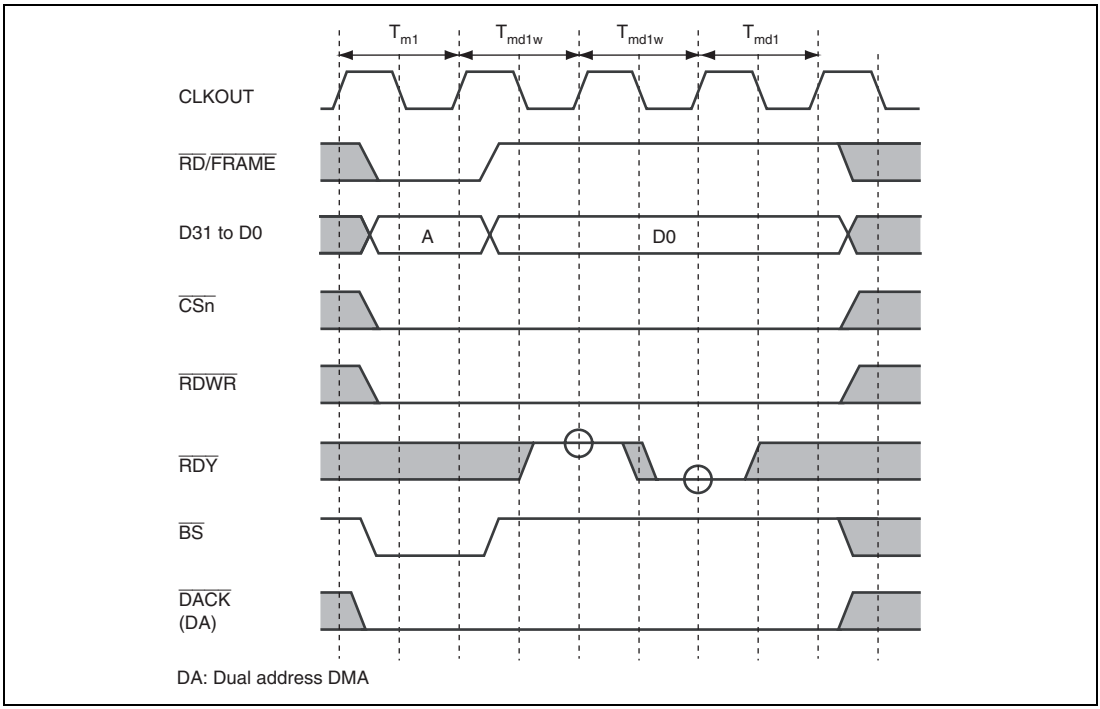


Figure 11.25 MPX Interface Timing 4 (Single Write Cycle, IW = 1, One External Wait Inserted)

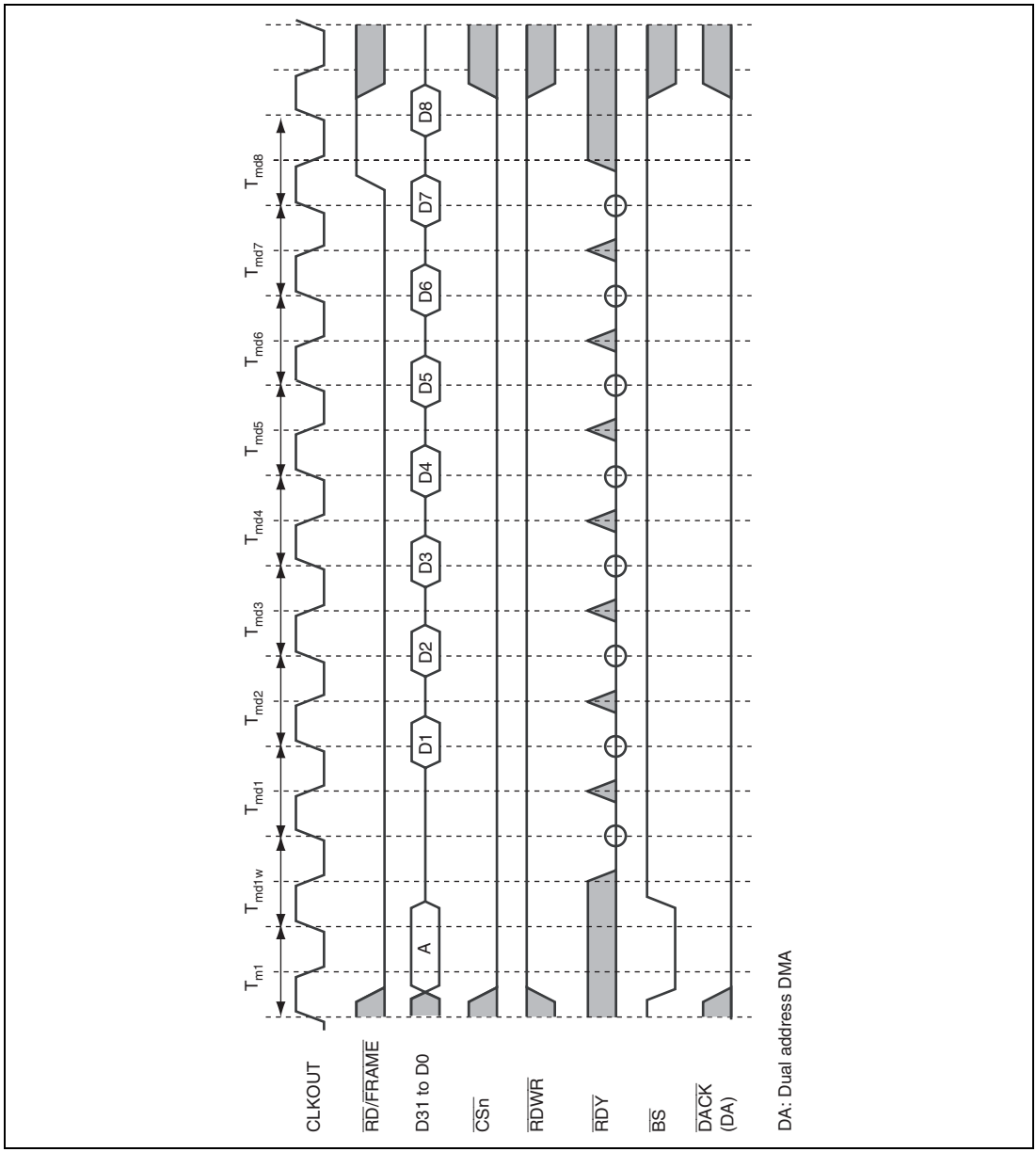


Figure 11.26 MPX Interface Timing 5 (Burst Read Cycle, IW = 0, No External Wait)

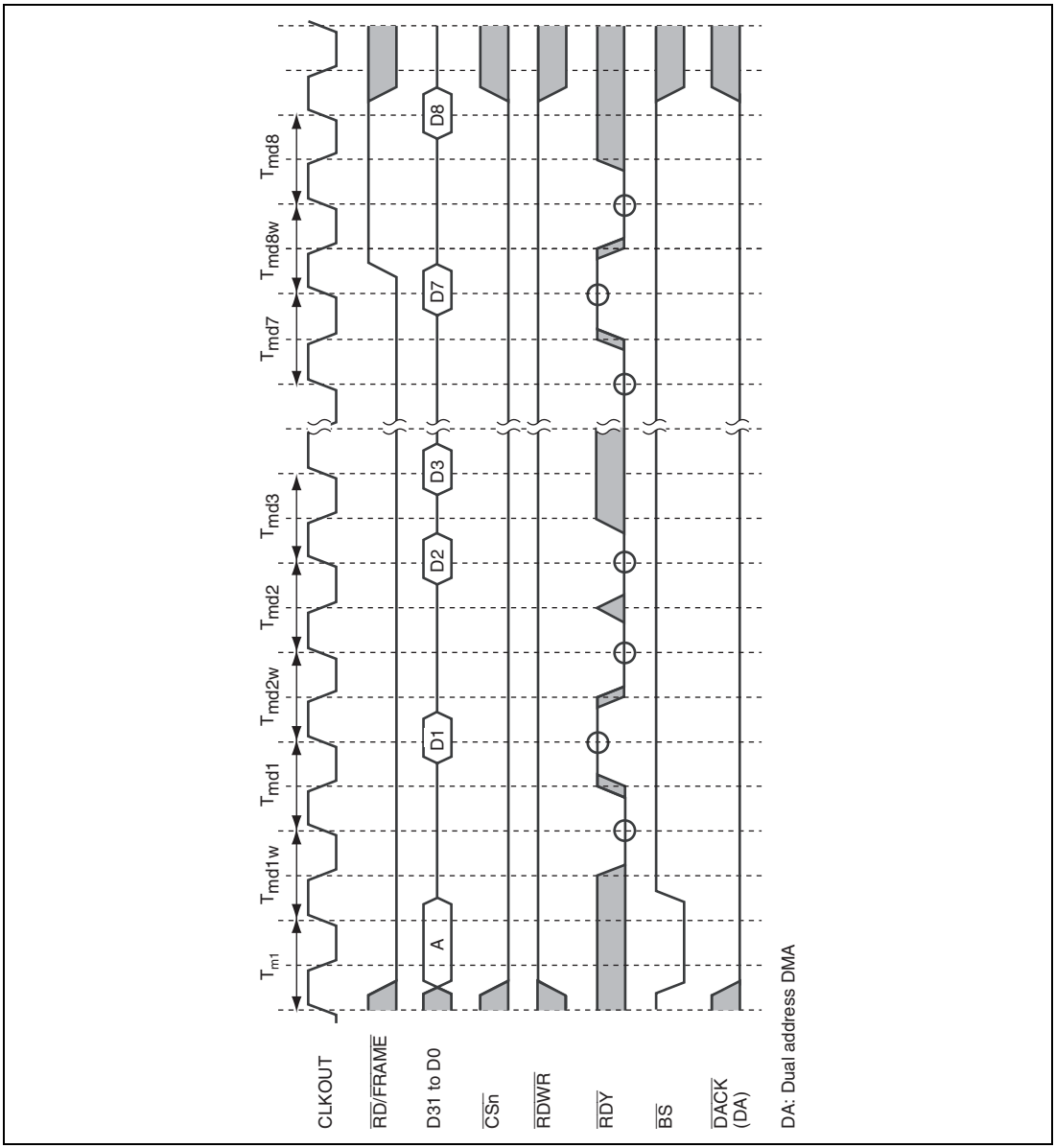


Figure 11.27 MPX Interface Timing 6 (Burst Read Cycle, IW = 0, External Wait Control)

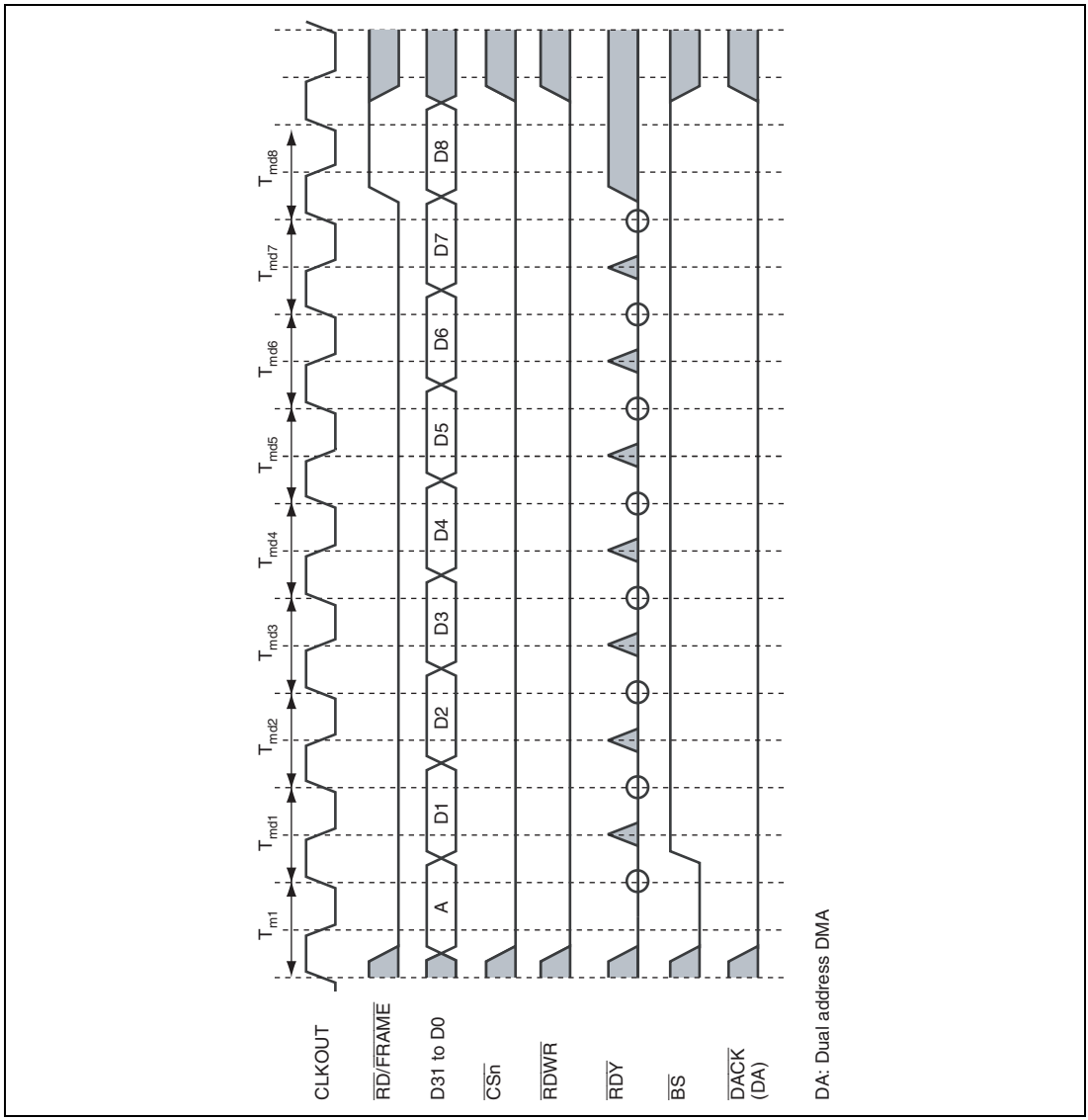


Figure 11.28 MPX Interface Timing 7 (Burst Write Cycle, IW = 0, No External Wait)

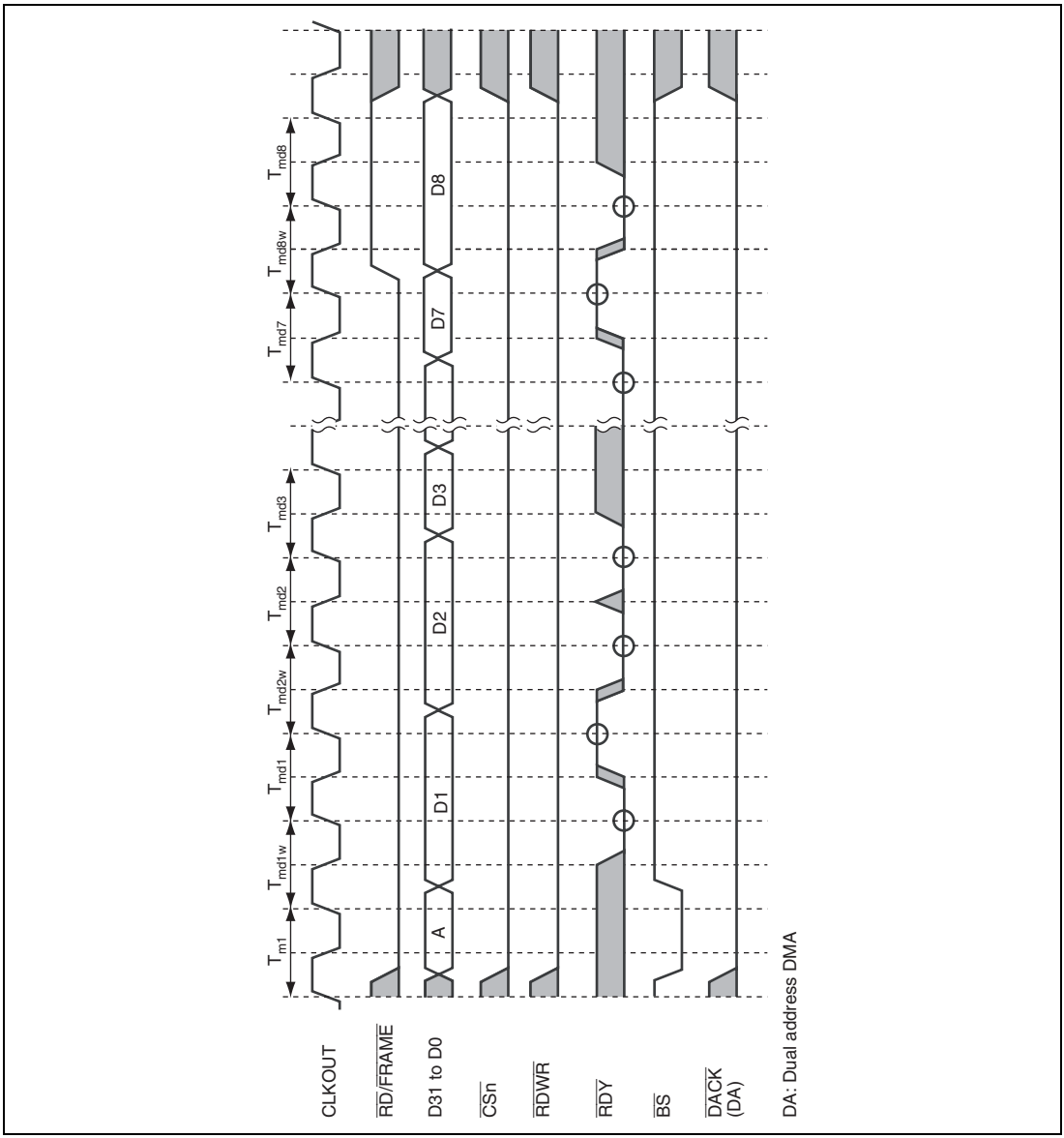


Figure 11.29 MPX Interface Timing 8 (Burst Write Cycle, IW = 1, External Wait Control)

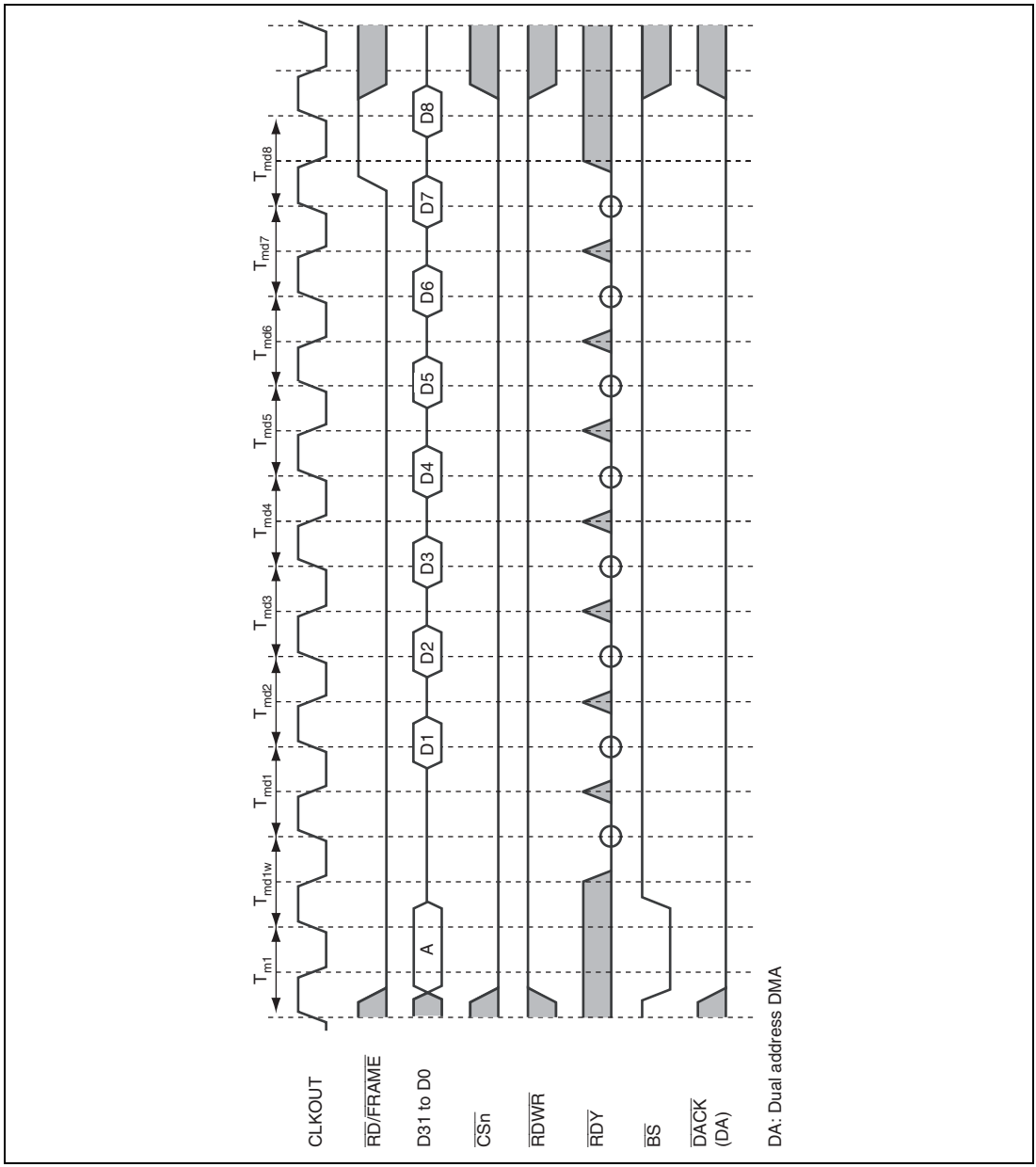


Figure 11.30 MPX Interface Timing 9 (Burst Read Cycle, IW = 0, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)

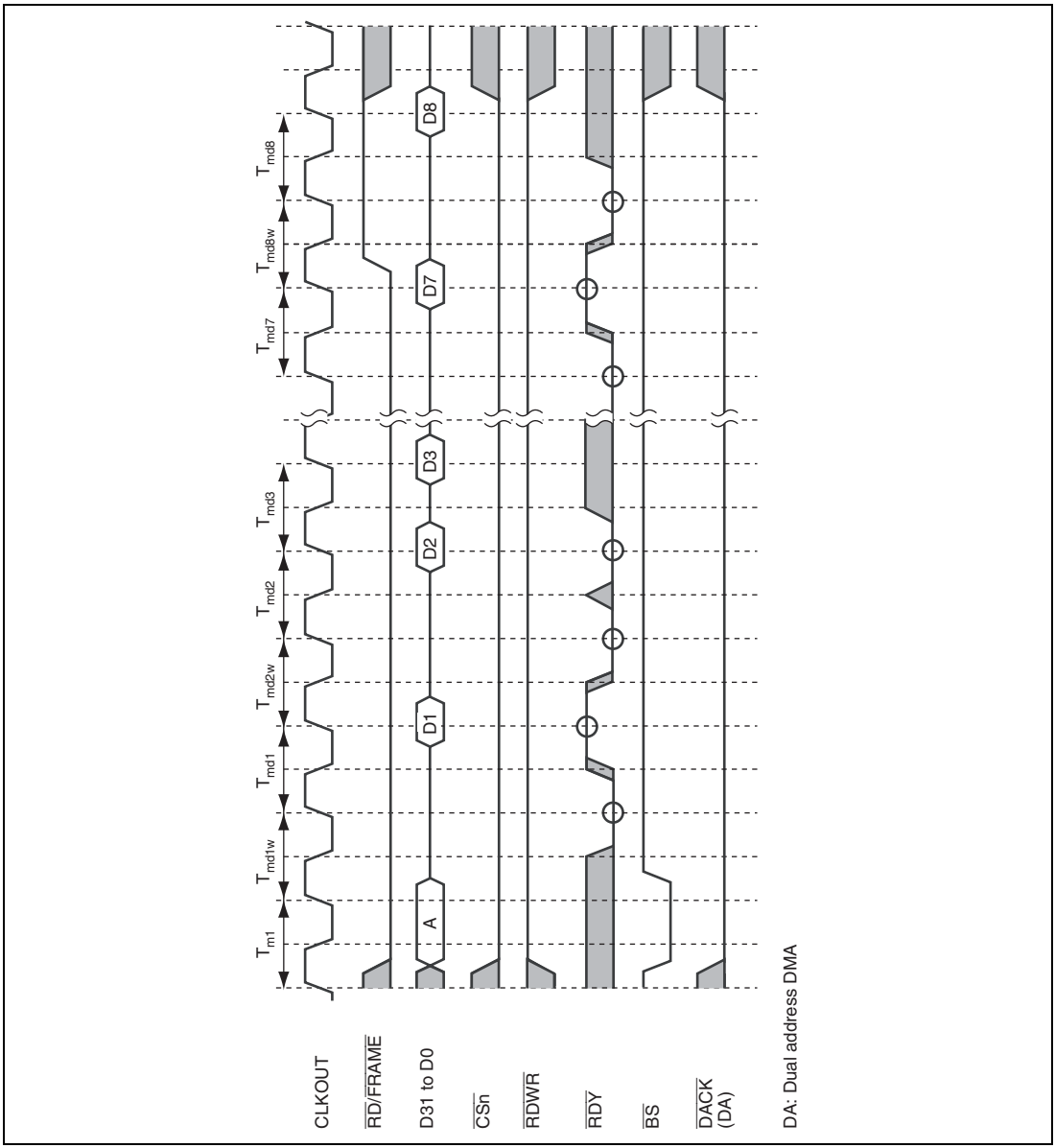


Figure 11.31 MPX Interface Timing 10 (Burst Read Cycle, IW = 0, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)

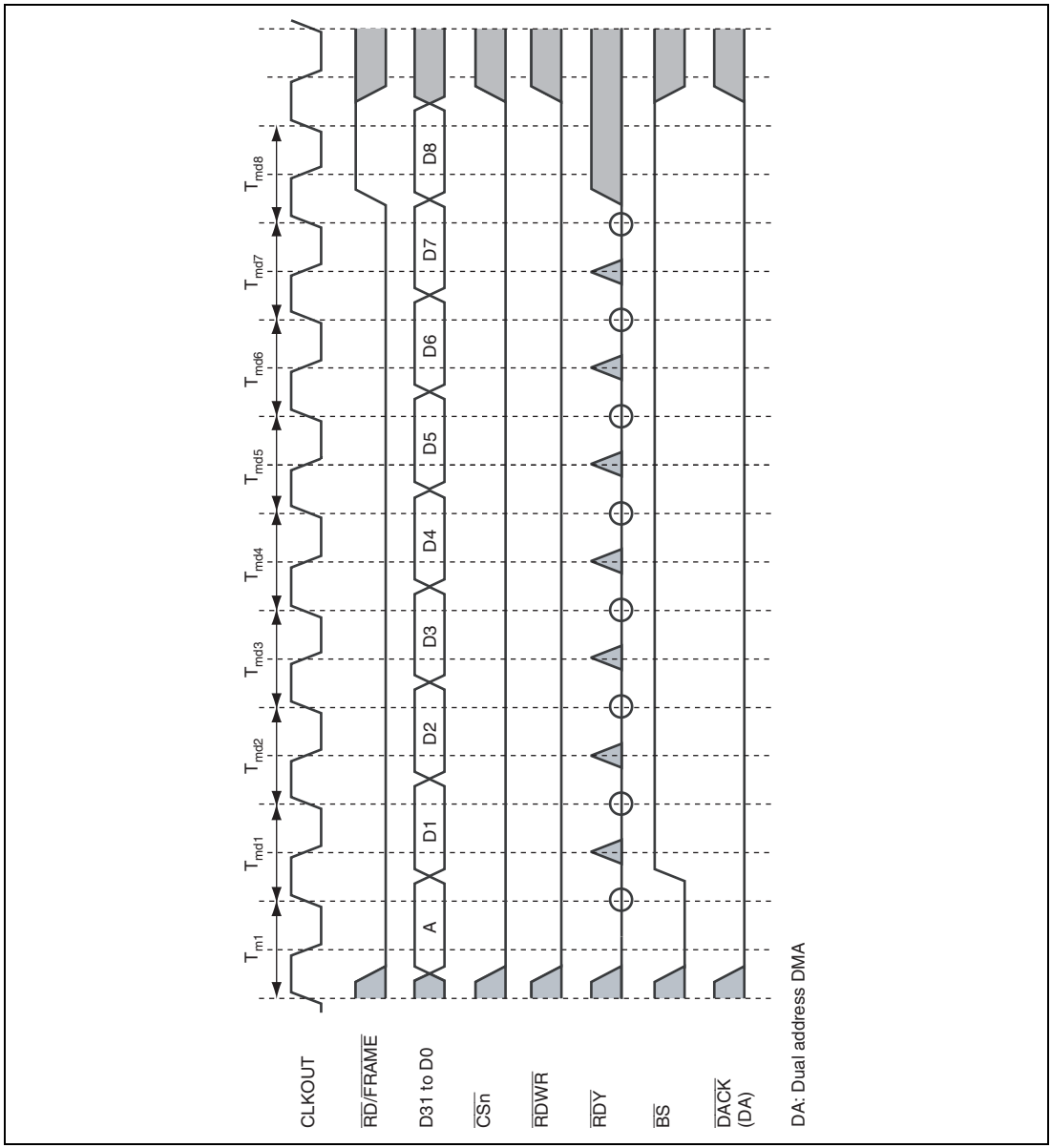


Figure 11.32 MPX Interface Timing 11 (Burst Write Cycle, IW = 0, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)

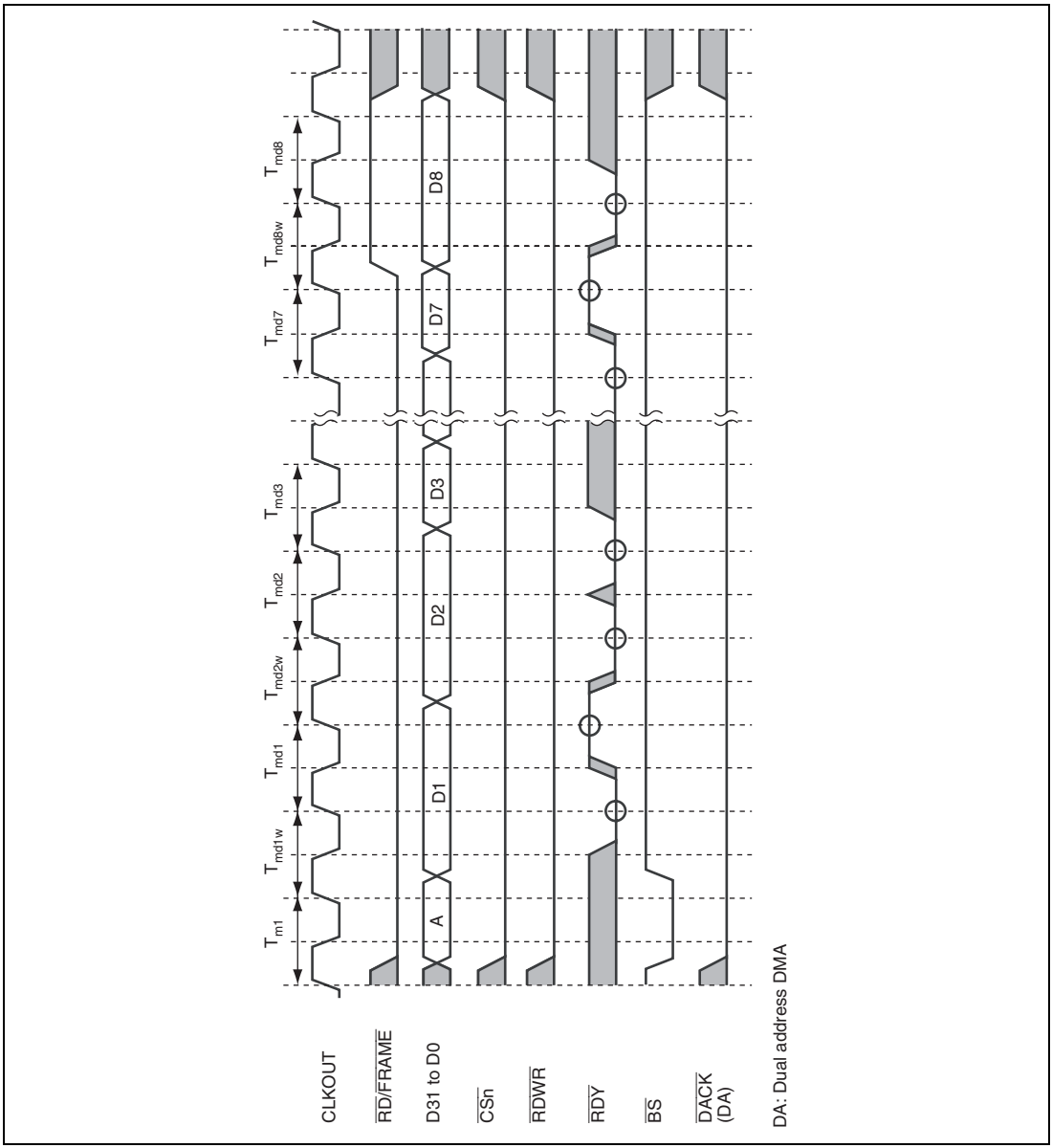


Figure 11.33 MPX Interface Timing 12 (Burst Write Cycle, IW = 1, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)

11.5.7 Byte Control SRAM Interface

The byte control SRAM interface is a memory interface that outputs a byte-select strobe ($\overline{\text{WEn}}$) in both read and write bus cycles. This interface has 16-bit data pins and can be connected to SRAM having an upper byte select strobe and lower select strobe functions, such as UB and LB.

Areas 1 and 4 can be specified as a byte control SRAM interface. However, when these areas are set to the MPX interface, the MPX interface has priority.

The write timing for the byte control SRAM interface is identical to that of a normal SRAM interface.

In read operations, on the other hand, the $\overline{\text{WEn}}$ pin timing is different. In a read access, only the $\overline{\text{WE}}$ signal for the byte being read is asserted. Assertion is synchronized with the falling edge of the CLKOUT clock in the same way as for the $\overline{\text{WE}}$ signal, while negation is synchronized with the rising edge of the CLKOUT clock in the same way as for the $\overline{\text{RD}}$ signal.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wrap around method according to the set bus width. The bus is not released during this transfer.

Figure 11.37 shows an example of a byte control SRAM connection, and figures 11.38 to 11.40 show examples of byte-control SRAM read cycles.

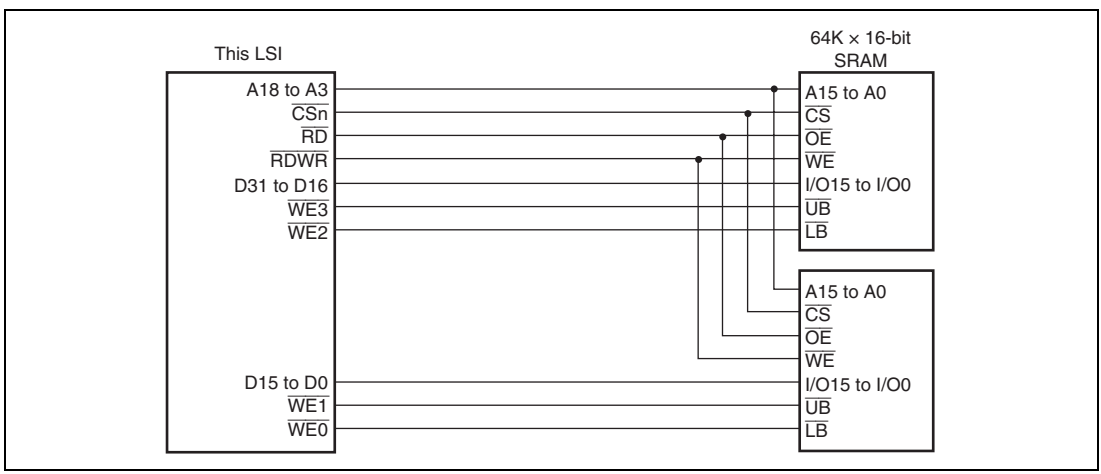
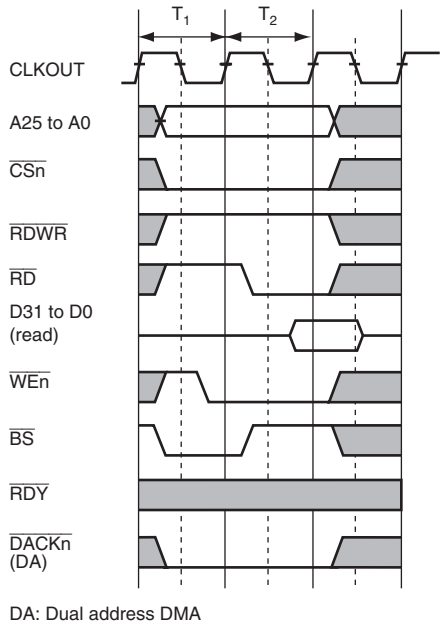


Figure 11.34 Example of 32-Bit Data-Width Byte-Control SRAM



DA: Dual address DMA

Figure 11.35 Byte-Control SRAM Basic Read Cycle (No Wait)

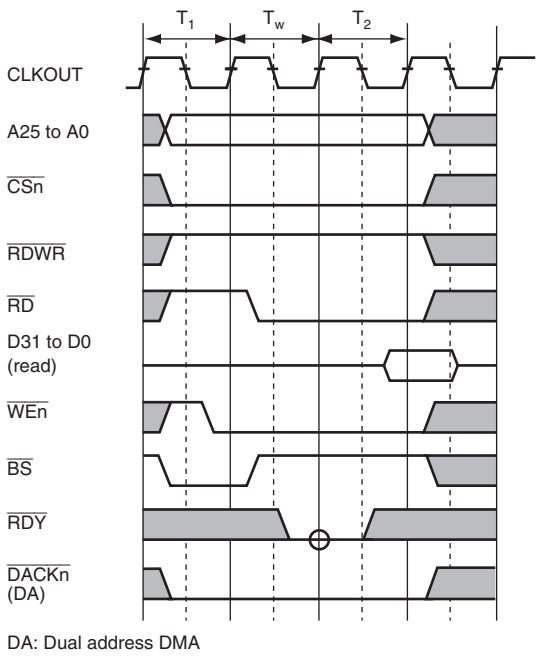
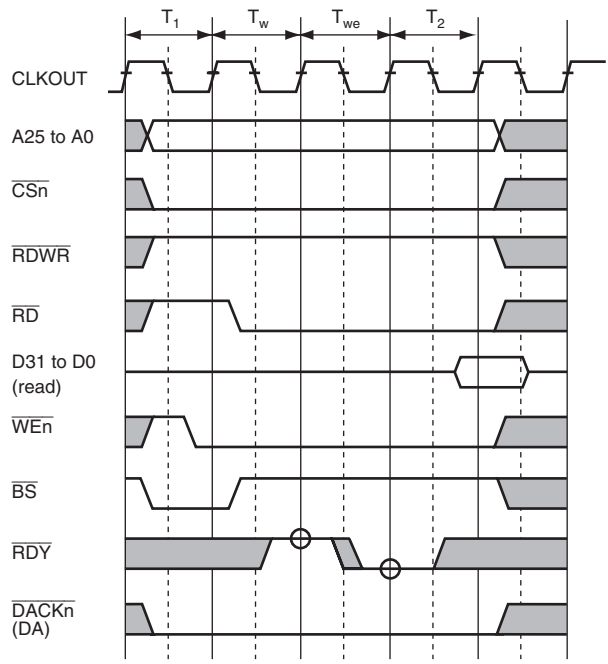


Figure 11.36 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cycle)



DA: Dual address DMA

Figure 11.37 Byte-Control SRAM Basic Read Cycle (One Internal Wait + One External Wait)

11.5.8 Wait Cycles between Accesses

A problem associated with higher operating frequencies for external memory buses is that the data buffer turn-off after completion of a read from a low-speed device may be too slow, causing a collision with the data in the next access, and resulting in lower reliability or malfunctions. To prevent this problem, this module provides a data collision prevention function. It stores the preceding access area and the type of read/write and inserts a wait cycle before the access cycle if there is a possibility of a bus collision when the next access is started. The process for wait cycle insertion consists of inserting idle cycles between the access cycles as shown in section 11.4.3, CSn Bus Control Register (CSnBCR). If bits IWW, IWRWD, IWRWS, IWRRD and IWRRS in CSnBCR (n = 0 to 2 and 4 to 6) are used to set the number of idle cycles between accesses, the number of inserted idle cycles is only the specified number of idle cycles minus the number of idle cycles specified by the bits.

When bus arbitration is performed, the bus is released after wait cycles are inserted between the cycles.

When a DMA transfer (dual address mode) is performed, wait cycles are inserted as set in CSnBCR idle cycle bits.

When access the MPX interface area continuously after read access, 1 wait cycle is inserted even if set the wait cycle to 0.

When the access size is 8-byte or 16-byte, wait cycles are inserted every 4-byte access.

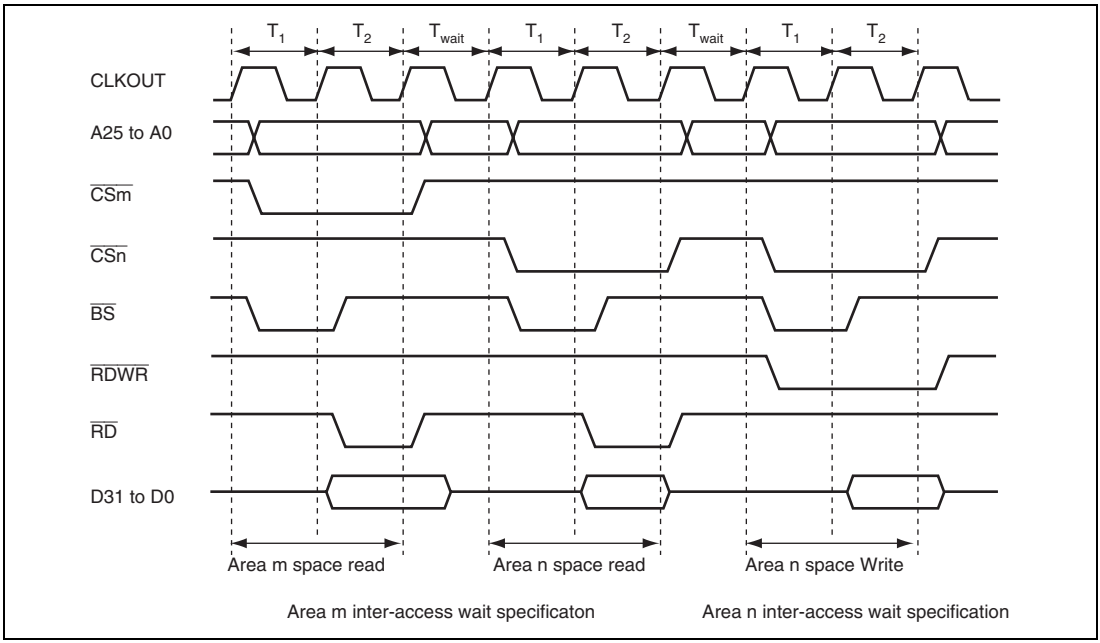


Figure 11.38 Wait Cycles between Access Cycles

11.5.9 Bus Arbitration

This LSI is provided with a bus arbitration function that grants the bus to an external device when it makes a bus request.

This bus arbitration supports master mode. In master mode the bus is held on a constant basis, and is released to another device in response to a bus request. The bus goes to the high-impedance state when not being held and it is possible to connect an external device that issues bus requests. In the following description, an external device that issues bus requests is also referred to as a slave.

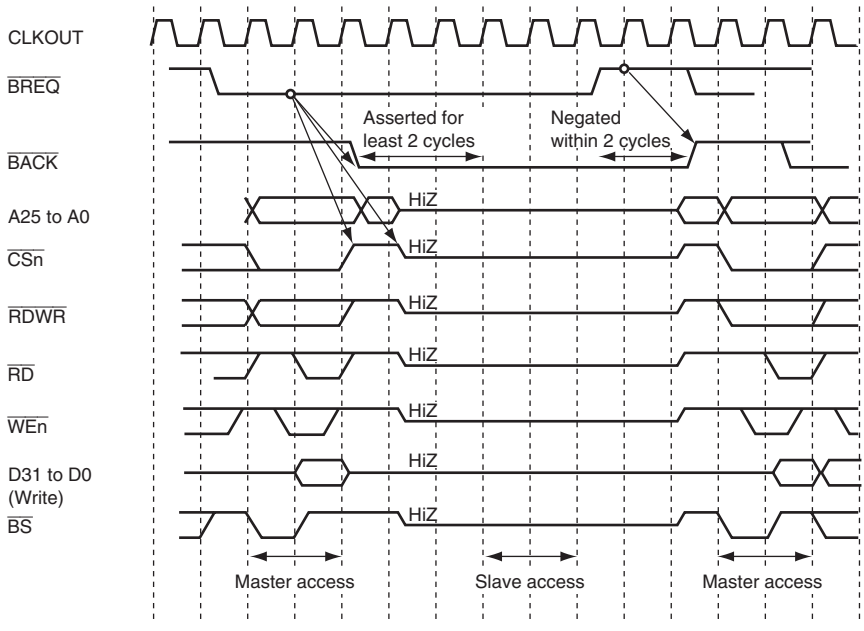
This LSI has three internal bus masters: the CPU, DMAC, and PCIC. In addition to these are bus requests from external devices. If requests occur simultaneously, priority is given, in high-to-low order, to a bus request from an external device, the PCIC, the DMAC, and the CPU.

To prevent incorrect operation of connected devices when the bus is transferred between master and slave, all bus control signals are negated before the bus is released. When mastership of the bus is received, also, bus control signals begin driving the bus from the negated state. Since signals are driven to the same value by the master and slave exchanging the bus, output buffer collisions can be avoided. By turning off the output buffer on the side releasing the bus, and turning on the output buffer on the side receiving the bus, simultaneously with respect to the bus control signals, it is possible to eliminate the signal high-impedance period. It is not necessary to provide the pull-up resistors usually inserted in these control signal lines to prevent incorrect operation due to external noise in the high-impedance state.

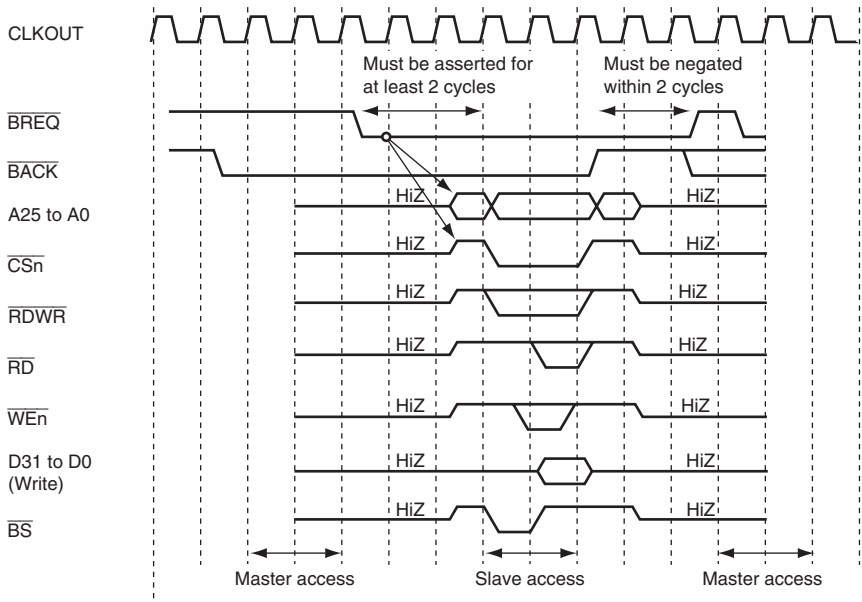
Bus transfer is executed between bus cycles.

When the bus release request signal ($\overline{\text{BREQ}}$) is asserted, this LSI releases the bus as soon as the currently executing bus cycle ends, and outputs the bus use permission signal ($\overline{\text{BACK}}$). However, bus release is not performed during multiple bus cycles generated because the data bus width is smaller than the access size (for example, when performing longword access to 8-bit bus width memory) or during a 32-byte transfer such as a cache fill or write-back. In addition, bus release is not performed between read and write cycles during execution of a TAS instruction, or between read and write cycles when DMAC dual address transfer is executed. When $\overline{\text{BREQ}}$ is negated, $\overline{\text{BACK}}$ is negated and use of the bus is resumed.

As the CPU in this LSI is connected to cache memory by a dedicated internal bus, reading from cache memory can still be carried out when the bus is being used by another bus master inside or outside this LSI. When writing from the CPU, an external write cycle is generated when write-through has been set for the cache in this LSI, or when an access is made to a cache-off area. There is consequently a delay until the bus is returned.



(a) Master mode device access



(b) Slave mode device access

Figure 11.39 Arbitration Sequence

11.5.10 Master Mode

The master mode processor holds the bus itself unless it receives a bus request.

On receiving an assertion (low level) of the bus request signal ($\overline{\text{BREQ}}$) from off-chip, the master mode processor releases the bus and asserts (drives low) the bus use permission signal ($\overline{\text{BACK}}$) as soon as the currently executing bus cycle ends. On receiving the $\overline{\text{BREQ}}$ negation (high level) indicating that the slave has released the bus, the processor negates (drives high) the $\overline{\text{BACK}}$ signal and resumes use of the bus.

When the bus is released, all bus interface related output signals and input/output signals go to the high-impedance state, except for the synchronous DRAM interface M_CKE signal and bus arbitration $\overline{\text{BACK}}$ signal, $\overline{\text{DACK0}}$ to $\overline{\text{DACK3}}$, and $\overline{\text{TEND0}}$ to $\overline{\text{TEND3}}$ which control DMA transfers.

The actual bus release sequence is as follows.

First, the bus use permission signal is asserted in synchronization with the rising edge of the clock. The address bus and data bus go to the high-impedance state in synchronization from next rising edge of the clock after this $\overline{\text{BACK}}$ assertion. At the same time, the bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, $\overline{\text{RDWR}}$, $\overline{\text{CE2A}}$, and $\overline{\text{CE2B}}$) go to the high-impedance state. These bus control signals are negated no later than one cycle before going to high-impedance. Bus request signal sampling is performed on the rising edge of the clock.

The sequence for re-acquiring the bus from the slave is as follows.

As soon as $\overline{\text{BREQ}}$ negation is detected on the rising edge of the clock, $\overline{\text{BACK}}$ is negated and bus control signal driving is started. Driving of the address bus and data bus starts at the next rising edge of an in-phase clock. The bus control signals are asserted and the bus cycle is actually started, at the earliest, at the clock rising edge at which the address and data signals are driven.

In order to reacquire the bus and start execution of bus access, the $\overline{\text{BREQ}}$ signal must be negated for at least two cycles.

11.5.11 Cooperation between Master and Slave

To enable system resources to be controlled in a harmonious fashion by master and slave, their respective roles must be clearly defined.

When designing an application system that includes this LSI, all control, including initialization, and low power consumption control, are supposed to be carried out by this LSI.

In a power-on reset, this LSI will not accept bus requests from the slave until the $\overline{\text{BREQ}}$ enable bit (BCR.BREQEN) is set to 1.

To ensure that the slave processor does not access memory requiring initialization before use, write 1 to the $\overline{\text{BREQ}}$ enable bit only after this LSI has performed the initialization.

Section 12 DDR-SDRAM Interface (DDRIF)

The memory controller is a module that arbitrates accesses from the CPU and modules and outputs control signals for the DDR-SDRAM. This module allows direct connection with the DDR-SDRAM. This module is provided with two interface modules (SHIF: SuperHyway bus interface and LCDIF: LCD interface) and one DDR-SDRAM controller (DDRC), and an arbiter (ARBT) that arbitrates accesses from interface modules to the DDRC.

12.1 Features

- The data bus width of the DDRIF is 32 bits
- Supports DDR-SDRAM self-refreshing
- Supports the DDR266 (133MHz); DDR200 (100MHz)
- Efficient data transfer is possible using the SuperHyway bus (Internal bus)
- Supports a 4-bank DDR-SDRAM
- Supports a burst length of 2
- Connectable memory size: 256 Mbits, 512 Mbits, 1 Gbit, and 2 Gbits
  Address × bit width (bit) for supported memory is as follows.
  DDR-SDRAM data bus width is 32 bits:
  - Parallel connection of two 128-Mbit DDRs (× 16) (Total size 256-Mbits)
  - Parallel connection of two 256-Mbit DDRs (× 16) (Total size 512-Mbits)
  - Parallel connection of two 512-Mbit DDRs (× 16) (Total size 1-Gbit)
  - Parallel connection of two 1-Gbit DDRs (× 16) (Total size 2-Gbits)
- Big or little endian for external memory access can be switched at a power-on reset

Figure 12.1 shows a block diagram of the DDRIF.

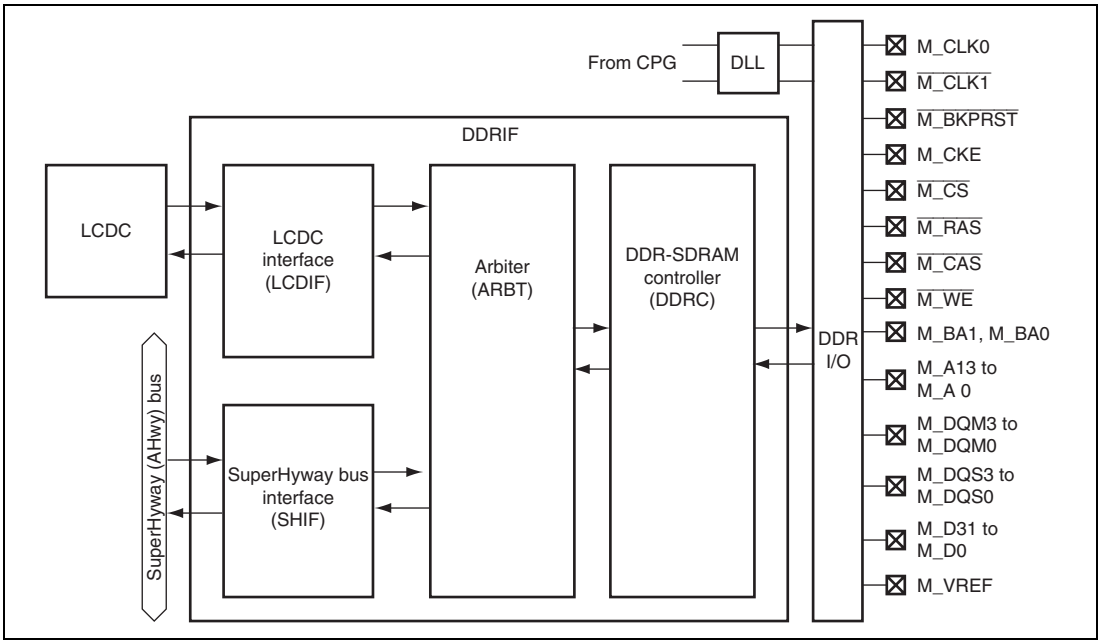


Figure 12.1 DDRIF Block Diagram

- The SuperHyway bus interface (SHIF) is an interface between the CPU and SDRAM. The SuperHyway bus protocol is used for interface. The bus width is 64 bits and the maximum operating frequency is 133 MHz.
- The LCDC interface (LCDIF) is an interface with the LCDC. The bus width is 32 bits and the maximum operating frequency is 66 MHz.
- The arbiter (ARBT) arbitrates SHIF that accesses the DDR-SDRAM and LCDIF requests among the requests from the abovementioned interfaces.
- The DDR-SDRAM controller (DDRC) controls read/write accesses to the DDR-SDRAM. Commands are issued and read data is received according to the specified timing of the DDR-SDRAM. Arbitration between the interfaces described above is performed with the priority order determination defined separately. The I/O block that drives the DDR-SDRAM interface signal and the register block related to the control of the DDR-SDRAM are included.

12.2 Input/Output Pins

Table 12.1 shows the DDRIF pin configuration. For details on connection with the DDR-SDRAM, see the LSI pin information section. Note that clock-related signals will be determined later.

Table 12.1 Pin Configuration

Pin Name	Function	I/O	Description
M_CLK0	DDR-SDRAM clock	Output	Clock output for DDR-SDRAM
$\overline{\text{M_CLK1}}$	DDR-SDRAM clock	Output	Clock output for DDR-SDRAM Inverted clock output of M_CLK0
M_CKE	Clock enable	Output	When this pin goes high, the clock signal is active. When this pin goes low, the clock signal is inactive.
$\overline{\text{M_CS}}$	Chip select	Output	Chip select output
$\overline{\text{M_WE}}$	Write enable	Output	Write enable output
M_A13 to M_A0	Address	Output	Row/column address
M_BA1, M_BA0	Bank active	Output	Bank address output
M_D31 to M_D0	Data	I/O	Data I/O
M_DQS3 to M_DQS0	I/O data strobe	I/O	I/O data strobe
M_DQM3 to M-DQM0	Data mask	Output	I/O data mask signal
$\overline{\text{M_RAS}}$	Row address strobe	Output	Row address strobe signal
$\overline{\text{M_CAS}}$	Column address strobe	Output	Column address strobe signal
$\overline{\text{M_BKPRST}}$	Power back-up reset	Input	When this pin goes low, the M_CKE pin shall go low
M_VREF	Reference voltage input	Input	Input reference voltage

12.3 Data Conversion

12.3.1 Data Alignment

Data Alignment in DDR-SDRAM: The DDRIF supports both big endian mode where an address of the upper byte is 0 and little endian mode where an address of the lower byte is 0. These modes can be switched by using external pins at a power-on reset.

The data alignment shown in the following tables is used when access is made from the peripheral modules. The data alignment in little endian mode differs from that in the physical memory map of the DDR-SDRAM.

Table 12.2 Access and Data Alignment in Little Endian Mode (External Bus Width is 32 Bits)

	M_D31 to M_D24	M_D23 to M_D16	M_D15 to M_D8	M_D7 to M_D0
Byte access at address 0				Data 7 to 0
Byte access at address 1			Data 7 to 0	
Byte access at address 2		Data 7 to 0		
Byte access at address 3	Data 7 to 0			
Byte access at address 4				Data 7 to 0
Byte access at address 5			Data 7 to 0	
Byte access at address 6		Data 7 to 0		
Byte access at address 7	Data 7 to 0			
Word access at address 0			Data 15 to 8	Data 7 to 0
Word access at address 2	Data 15 to 8	Data 7 to 0		
Word access at address 4			Data 15 to 8	Data 7 to 0
Word access at address 6	Data 15 to 8	Data 7 to 0		
Longword access at address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Longword access at address 4	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0 (first time: address 0)	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0 (second time: address 4)	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32

Table 12.3 Access and Data Alignment in Big Endian Mode (External Bus Width is 32 Bits)

	M_D31 to M_D24	M_D23 to M_D16	M_D15 to M_D8	M_D7 to M_D0
Byte access at address 0	Data 7 to 0			
Byte access at address 1		Data 7 to 0		
Byte access at address 2			Data 7 to 0	
Byte access at address 3				Data 7 to 0
Byte access at address 4	Data 7 to 0			
Byte access at address 5		Data 7 to 0		
Byte access at address 6			Data 7 to 0	
Byte access at address 7				Data 7 to 0
Word access at address 0	Data 15 to 8	Data 7 to 0		
Word access at address 2			Data 15 to 8	Data 7 to 0
Word access at address 4	Data 15 to 8	Data 7 to 0		
Word access at address 6			Data 15 to 8	Data 7 to 0
Longword access at address 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Longword access at address 4	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0
Quadword access at address 0 (first time: address 0)	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32
Quadword access at address 0 (second time: address 4)	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0

12.3.2 Data Alignment in Peripheral Modules

The endian mode in the DDRIF matches that in the CPU, and both big endian and little endian are available.

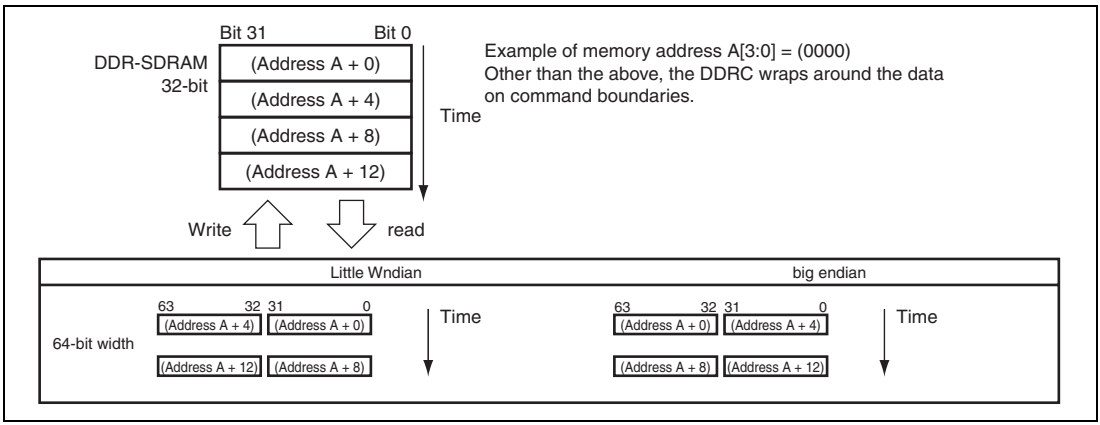


Figure 12.2 Data Alignment in DDR-SDRAM and DDRIF

12.4 Register Descriptions

Table 12.4 shows the DDRIF registers.

These registers should be set when access is not made to the DDR-SDRAM from peripheral modules. When the access is not made and the DCE bit (DDR-SDRAM control enable) in the memory interface mode register is cleared to 0 or the SELFS bit (self-refresh status) in that register is set to 1, set other registers.

Although the bit width for registers is 64 bits, access the registers in longwords (32 bits). Writing to this register is reflected in longwords. Reading this register returns a current longword value. In big endian mode, when accessing bits 63 to 32, specify address 8n + 0. When accessing bits 31 to 0, specify address 8n + 4.

Table 12.4 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*[1]	Area 7 Address*[1]	Access Size
Memory interface mode register	MIM	R/W	H'FE80 0008	H'1E80 0008	32
DDR-SDRAM control register	SCR	R/W	H'FE80 0010	H'1E80 0010	32
DDR-SDRAM timing register	STR	R/W	H'FE80 0018	H'1E80 0018	32
DDR-SDRAM row attribute register	SDR	R/W	H'FE80 0030	H'1E80 0030	32
DDR-SDRAM mode register	SDMR	(W)	H'FE9x xxxx*[2]	H'1E9x xxxx*[2]	32
DDR-SDRAM back-up register	DBK	R	H'FE80 0400	H'1E80 0400	32

Notes: 1. P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.
2. For details, see section 12.4.5, DDR-SDRAM Mode Register (SDMR).

Table 12.5 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Memory interface mode register	MIM	H'0000 0000 0C34 xx00*[1]	H'0000 0000 0C34 xx00*[1]	Retained	Retained
DDR-SDRAM control register	SCR	H'0000 0000 0000 0000	H'0000 0000 0000 0000	Retained	Retained
DDR-SDRAM timing register	STR	H'0000 0000 0000 0000	H'0000 0000 0000 0000	Retained	Retained
DDR-SDRAM row attribute register	SDR	H'0000 0000 0000 0100	H'0000 0000 0000 0100	Retained	Retained
DDR-SDRAM mode register	SDMR	Only writing	Only writing	Only writing	Only writing
DDR-SDRAM back-up register	DBK	H'0000 0000 0000 000x*[2]	H'0000 0000 0000 000x*[2]	Retained	Retained

Notes: 1. The initial value of bit 8 (ENDIAN bit) depends on the setting of external pins (MD5).
2. The initial value of bit 0 (SDBUP bit) depends on the setting of external pin ($\overline{\text{M_BKPRST}}$).

All bits are active-high signals and are initialized by a reset unless otherwise specified.

All access is made in longwords using the SuperHyway bus.

12.4.1 Memory Interface Mode Register (MIM)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	BOMODE			PCKE										SELFS	RMODE	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				DRI[12:0]												
Initial value:	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK						DRE	ENDIAN					DLLEN			DCE
Initial value:					0	0	0		0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R	R	R/W

Note: * Depends on the setting of external pins (MD5).

Bit	Bit Name	Initial Value	R/W	Description
63 to 49	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
48	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
47, 46	BOMODE1, BOMODE0	All 0	R/W	Access Mode Switch Switch access modes for the DDR-SDRAM. The DDRIF supports two DDR-SDRAM access modes. For details on the operation in each mode, see section 12.5.4, DDR-SDRAM Access Mode. 00: Bank open mode 01: Bank close mode Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
45	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
44	PCKE	0	R/W	Power Down When the DDR-SDRAM is not accessed (in the idle state or bank active state), this bit sets the CKE pin low and the power-down mode is entered. When this bit is set to 1, the power-down mode is entered to reduce the DDR-DSRAM power consumption. For details, see section 12.5.5, Power-Down Modes. Note that the setting for enabling the CKE pin by the SMS bit in SCR is used for DDR-SDRAM initialization.
43 to 35	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
34	SELFS	0	R	Self-Refresh Decision Decides whether the DDR-SDRAM is in the self-refresh state. When this bit is set to 1, the DDR-SDRAM is in the self-refresh state. When this bit is cleared to 0, the DDR-SDRAM is not in the self-refresh state.
33	RMODE	0	R/W	Refresh Mode Select Specifies whether auto-refresh mode or self-refresh mode is set to the DDR-SDRAM. This bit is valid only when the DRE bit in MIM is set to 1. 0: Auto-refresh mode 1: Self-refresh mode
32 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
28 to 16	DRI	H'0C34	R/W	DRAM Refresh Interval When refreshing is valid (the DRE bit in MIM is set to 1), these bits specify the maximum refresh interval (auto refresh). One count is the same as the cycle of the memory clock. At 133-MHz operation, one count corresponds to 7.5 ns. The minimum settable value is H'020. When a value less than H'020 is set, H'020 is added to the count value. The DDRIF has a 13-bit internal counter. When the DCE or DRE bit is cleared to 0, or the RMODE bit is set to 1, this counter is cleared to 0. Otherwise, this counter will increment by the external memory clock. This counter is compared with the DRI bit. If a match occurs, an auto-refresh request is generated in the controller and auto-refreshing is performed. Note that the counter is cleared to 0 at the match and then begins incrementing again. The single auto-refresh request that has been generated is recorded (max.). When the DCE and DRE bits are set to 1 and the RMODE bit is cleared to 0, an auto-refresh request is not cleared until auto refreshing is performed. To set this bit, the DRE bit should be cleared to 0 and should be written to, and then 1 should be written to the DRE bit. In this case, the previous written value should be set to the DRI bits.
15 to 12	LOCK	Undefined	R	DLL Lock Status These bits indicate the lock status of the DLL for generating the read timing for the DDR-SDRAM. When these bits are all set to 1, access to memory is possible.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	DRE	0	R/W	DRAM Refresh Enable Sets whether the refresh mode is valid or invalid. 1: Valid 0: Invalid

Bit	Bit Name	Initial Value	R/W	Description
8	ENDIAN	Undefined	R	Endian Identification Indicates whether the big endian mode or little endian mode is set to the external data bus. 1: Big endian mode 0: Little endian mode
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	DLLEN	0	R/W	DLL Enable Sets whether the DLL for generating the read timing for the DDR-SDRAM is valid or invalid. When this bit is set to 1, the DLL is enabled and read access to memory is possible.
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DCE	0	R/W	DDR Controller Enable Enables SDRAM control by the DDRIF. 1: Enable 0: Disable

12.4.2 DDR-SDRAM Control Register (SCR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	SMS	000	R/W	SDRAM Mode Select These bits initialize the DDR-SDRAM at a power-on or after release of a reset. If this bit is set by software, the following commands are issued. For details on the initialization procedure, see section 12.5.2, DDR-SDRAM Initialization Sequence. After the DDR-SDRAM has been initialized, normal operation (000) is specified. 000: Normal operation 001: The NOP command is issued (valid only when the DCE bit in MIM is set to 1). 010: The PREALL command is issued (valid only when the DCE bit in MIM is set to 1). 011: The M_CKE pin is enabled. At that time, the DESELECT command is issued (valid only when the DCE bit in MIM is set to 1). 100: The REFA (auto) refresh command is issued (valid only when the DCE bit in MIM is set to 1). Settings other than above are prohibited. If such settings are made, correct operation is not guaranteed. Note that setting the M_CKE pin low by the PCKE bit in MIM is used to reduce the DDR-SDRAM power consumption.

12.4.3 DDR-SDRAM Timing Register (STR)

STR specifies the DDR-SDRAM timing. (Details on the number range depend on the parameters used by each memory manufacturer.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRFC			SWR	SRRD	SRAS			SRC			SCL			SRCD	SRP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
63 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19, 18	WR	00	R/W	Minimum Number of Cycles from Write command to Read Commands These bits specify the minimum number of cycles for the READ command issuance after the WRITE command is issued for the DDR-SDRAM. 00: 3 cycles 01: 4 cycles 10: 5 cycles 11: 6 cycles

Bit	Bit Name	Initial Value	R/W	Description
17, 16	RW	00	R/W	Minimum Number of Cycles from Read command to Write Commands These bits specify the minimum number of cycles for the WRITE command issuance after the READ command is issued for the DDR-SDRAM. 00: 3 cycles 01: 4 cycles 10: 5 cycles 11: 6 cycles
15 to 13	SRFC	000	R/W	Number of Cycles in Same Bank These bits specify the number of cycles in the same bank for the following access times (Trfc). (1) From auto refresh to ACT command issuance (2) From auto refresh to auto refresh 000: 11 cycles 001: 12 cycles 010: 13 cycles 011: 14 cycles 100: 15 cycles Other than above: Setting prohibited
12	SWR	0	R/W	PRE/PREALL Command Issuance Cycle Specifies the number of cycles from the last postamble to PRE/PREALL command issuance in a write cycle (Twr). 0: 2 cycles 1: 3 cycles
11	SRRD	0	R/W	ACT Command Issuance Cycle between Banks Specifies the minimum number of cycles from ACT command issuance to ACT command issuance between different banks (Trrd). 0: 2 cycles 1: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	SRAS	000	R/W	Minimum Number of Cycles between ACT and PRE Commands These bits specify the minimum number of cycles from ACT command issuance to PRE command issuance in the same bank (Tras). 000: 6 cycles 001: 7 cycles 010: 8 cycles 011: 9 cycles Other than above: Setting prohibited
7 to 5	SRC	000	R/W	Auto-Refresh/ACT Command Issuance Cycle These bits specify the number of cycles in the same bank for the following access times (Trc). (1) From ACT command issuance to auto refresh (2) From ACT command issuance to ACT command issuance 000: 6 cycles 001: 7 cycles 010: 8 cycles 011: 9 cycles 100: 10 cycles 101: 11 cycles 110: 12 cycles 111: 13 cycles Other than above: Setting prohibited
4 to 2	SCL	000	R/W	CAS Latency These bits specify the CAS latency (CL) in data read. 000: 2.5 cycles The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	SRCD	0	R/W	Number of Cycles between RAS and CAS Commands Specifies the number of cycles from RAS (ACT) command issuance to CAS (READ/READA, WRITE/WRITEA) command issuance (Trcd). 0: 3 cycles 1: 4 cycles
0	SRP	0	R/W	Number of Cycles between PRE and ACT Commands Specifies the number of cycles from PRE command issuance to ACT command issuance (Trp). 0: 3 cycles 1: 4 cycles

12.4.4 DDR-SDRAM Row Attribute Register (SDR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	SPLIT	0001	R/W	DDR-SDRAM Memory Configuration These bits specify the DDR-SDRAM row/column configuration. 0001: 12 × 9 (= 8 M × 16 bits product) 0011: 13 × 9 (= 16 M × 16 bits product) 0100: 13 × 10 (= 32 M × 16 bits product) 0110: 14 × 10 (= 64 M × 16 bits product) Other than above: Setting prohibited The relationship between the SPLIT bits and row/column is shown in section 12.5.12, Address Multiplexing.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

12.4.5 DDR-SDRAM Mode Register (SDMR)

SDMR is used to set the DDR-SDRAM mode register and extended mode register. Since SDMR is not physically contained in the DDRIF, reading this register is invalid. Only write addresses have a meaning for the DDR-SDRAM and the write data is ignored.

When SDMR is written to, signals are output to pins connected to the DDR-SDRAM according to the table shown below.

Address bits 12 to 3 correspond to external pins M_A9 to M_A0, address bits 14 and 13 to external pins M_BA1 and M_BA0, and address bits 18 to 15 to external pins M_A13 to M_A10.

M_CKE						Address Bit Correspondence		
n-1	n	$\overline{\text{M_CS}}$	$\overline{\text{M_RAS}}$	$\overline{\text{M_CAS}}$	$\overline{\text{M_WE}}$	M_BA1 and M_BA0	M_A13 to M_A10	M_A9 to M_A0
H	H	L	L	L	L	Bits 14 and 13	Bits 18 to 15	Bits 12 to 3

Figure 12.3 shows the relationship between write values in SDMR and output signals to the memory pins.

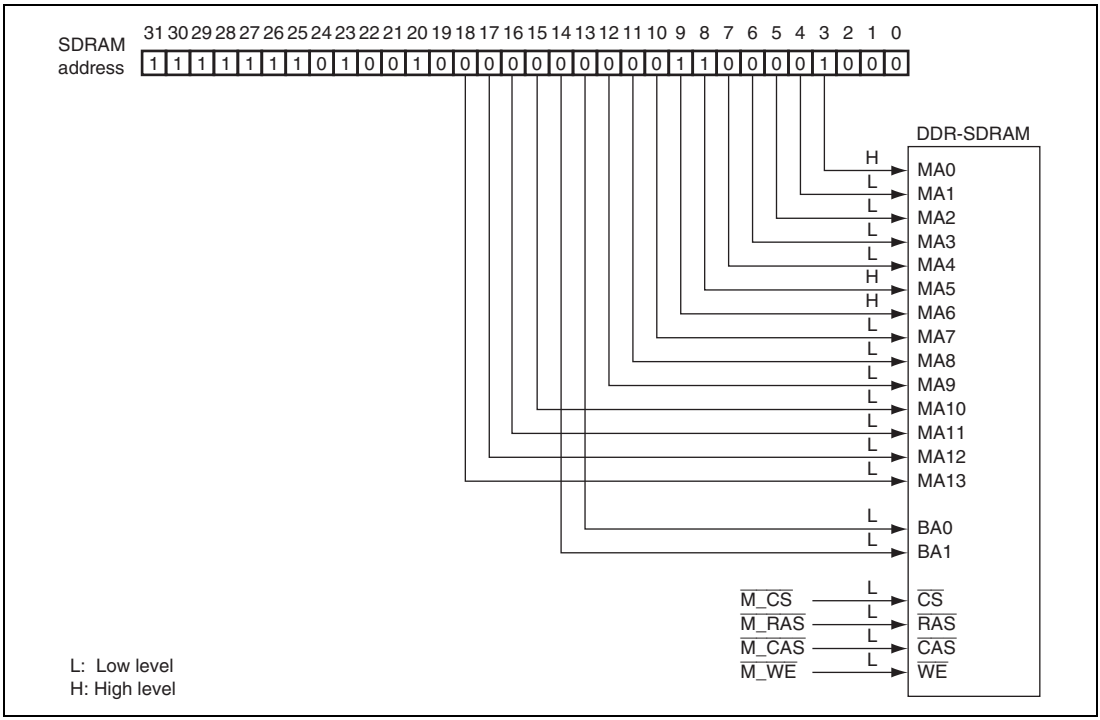


Figure 12.3 Relationship between Write Values in SDMR and Output Signals to Memory Pins

For example, when the DLL reset release, CAS latency of 2.5 cycles, sequential burst sequence, and burst length of 2 are set to the mode register in the DDR-SDRAM, the following signals must be output to the DDR-SDRAM pins.

$\overline{CS}$ = low, $\overline{RAS}$ = low, $\overline{CAS}$ = low, $\overline{WE}$ = low, BA0 = low, BA1 = low, MA13/MA12/MA11/MA10/MA9 = low, MA8 = low, MA7 = low, MA6 = high, MA5 = high, MA4 = low, MA3 = low, MA2 = low, MA1 = low, and MA0 = high

To output the above control signals, write access to address H'FE90 0308 in SDMR is made in longwords. Then the above control signals are output to the DDR-SDRAM pins. Write data to SDMR is Don't care.

12.4.6 DDR-SDRAM Back-up Register (DBK)

This register indicates the DDR-SDRAM back-up status. For details, see section 18, Power-Down Mode.

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	47
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

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

Bit	Bit Name	Initial Value	R/W	Description
63 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SDBUP	Undefined*	R	Determine whether DDR-SDRAM is battery back-up or not 0: Battery back-up 1: Not back-up

Note: * Depends on the setting of external pin $\overline{\text{BKPRST}}$.

12.5 Operation

12.5.1 DDR-SDRAM Access

The DDR-SDRAM is accessed with a burst length of 2. Read or write commands that hit the page are issued continuously and read data continuously.

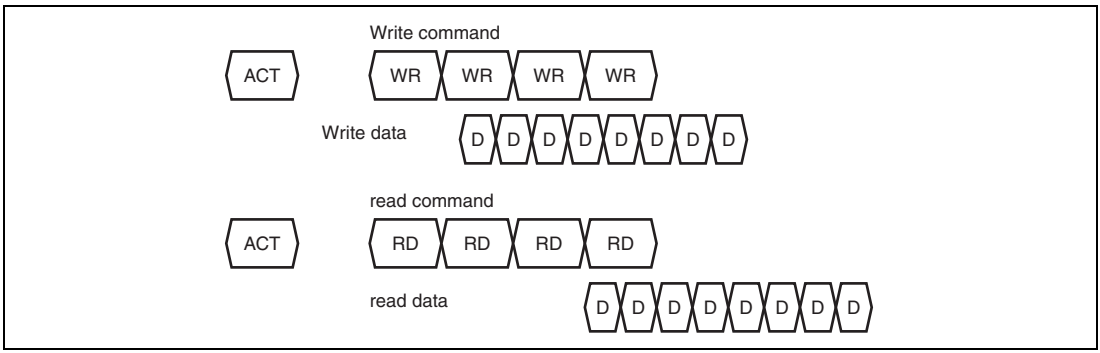


Figure 12.4 DDR-SDRAM Access

12.5.2 DDR-SDRAM Initialization Sequence

Since the internal state of the DDR-SDRAM is undefined immediately after a power-on, initialize the SDRAM according to the following sequence. Otherwise the device may be damaged.

An example of the initialization sequence for the DDR-SDRAM is shown below. For details, see each memory manufacturer's datasheet.

1. Turn on the four power supplies to the DDR-SDRAM in the following order: VDD, VDDQ, VREF, and VTT.
2. After the power supply, reference voltage, and clock are stabilized, maintain the current state for at least 200 μs.
3. Perform a dummy read to any DDR-SDRAM address.
4. Set MIM to enable the DDR-SDRAM controller, set the endian mode, and so on.
5. Set SDR and STR.
6. Use the SMS field in SCR to enable the CKE pin.
7. Use the SMS field in SCR to issue the all-bank precharge (PREALL) command.
8. Use SDMR to issue the EMRS command and enable the DLL.

9. Use SDMR to issue the MRS command and reset the DLL. Also set the burst length, CAS latency, and so on.
10. After the PREALL command is issued, use the SMS field in SCR to issue the REF command twice.
11. Use SDMR to issue the MRS command, release the DLL reset (MA8 = low), and determine the operating mode. In this case, use the setting for the burst length, etc. that was specified in step 8.
12. After the DLL is reset, wait for 200 cycles of the memory clock: normal memory access will then be possible.

Match the SDMR setting, etc. of the DDR-SDRAM with the register settings of the DDRIF.

12.5.3 Supported DDR-SDRAM Commands

Table 12.6 shows the DDR-SDRAM commands supported by the DDRIF.

Table 12.6 DDR-SDRAM Commands Issued by DDRIF

Function	Symbol	CKEn – 1	CKEn	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	MA13 to MA11	AP (MA10)	BA1 and BA0	MA9 to MA0
Device deselect	DESL	H	X	H	X	X	X	X	X	X	X
No operation	NOP	H	X	L	H	H	H	X	X	X	X
Read	READ	H	X	L	H	L	H	V	L	V	V
Read with auto precharge	READA	H	X	L	H	L	H	V	H	V	V
Write	WRITE	H	X	L	H	L	L	V	L	V	V
Write with auto precharge	WRITEA	H	X	L	H	L	L	V	H	V	V
Bank activate	ACT	H	X	L	L	H	H	V	V	V	V
Precharge select bank	PRE	H	X	L	L	H	L	X	L	V	X
Precharge all banks	PREALL	H	X	L	L	H	L	X	H	X	X
Auto refresh	REFA	H	H	L	L	L	H	X	X	X	X
Self-refresh entry from IDLE	REFS	H	L	L	L	L	H	X	X	X	X
Self-refresh exit	REFSX	L	H	H	X	X	X	X	X	X	X
Power-down entry	PWRDN	H	L	H	X	X	X	X	X	X	X
Power-down exit	PWRDNX	L	H	H	X	X	X	X	X	X	X
Mode register set	MRS/ EMRS	H	X	L	L	L	L	V	V	V	V

[Legend]
H: High level
L: Low level
X: High level or low level (either is acceptable)
V: Valid data

The DESL command in table 12.6 is issued when the DDR-SDRAM is not accessed by modules. The DESL command therefore cannot be issued by the user.

12.5.4 DDR-SDRAM Access Mode

The DDRIF supports the following two DDR-SDRAM access modes. Each mode can be set using the BOMODE bits in MIM.

Bank Open Mode: The DDR-SDRAM is accessed without the PRE command immediately after memory read or memory write, meaning that the bank is always open. This mode is useful for applications in which bank hits continue to occur during consecutive memory accesses. When a bank miss occurs, the PRE command is automatically issued.

Bank Close Mode: Immediately after memory read or memory write, the PRE command is output and the bank is closed. This mode is useful for applications in which a bank hit does not continue to occur in consecutive memory accesses.

12.5.5 Power-Down Modes

(1) Self-Refresh Mode

The self-refresh mode is a standby state where the refresh timing and refresh addresses are generated in the DDR-SDRAM. The self-refresh state is retained even if the CPU enters the sleep mode after the self-refresh mode is set by setting the DRE and RMODE bits in MIM to 1. If the sleep mode of CPU is canceled due to an interrupt, the self-refresh state is retained.

Although the self-refresh state is entered through the register setting of the DDRIF, the following sequence should be used.

[Transition to self-refresh state]

1. Confirm that the transaction to the memory controller is completed.
2. Through software control, set the SMS bits in SCR to issue the PREALL command. This closes any DDR-SDRAM bank that was open. After that, use the SMS bits in SCR to issue the REFA (auto-refresh) command to perform concentrated refresh on all memory rows ((REFA) (REFA) (REFA)). The STR settings do not establish a relationship between the timing of the PREALL and REFA commands that are issued by using SCR. A period of waiting that is suitable for the memory unit must be inserted by software. At that time, if concentrated refresh is necessary, execute the REFA command.
3. To make the DDR-SDRAM enter the self-refresh state, set the DRE and RMODE bits in MIM. (In this case, the DCE bit should remain at 1.)
4. The memory controller automatically issues the self-refresh command and set the CKE pin low. The DDR-SDRAM then automatically enters the power-down mode.

5. Whether the DDR-SDRAM enters the self-refresh mode can be checked by reading the SELFS bit in MIM.

[Recovery from self-refresh state]

1. Clear the RMODE and DRE bits in MIM to 0 to clear the self-refresh state.
2. Whether the DDR-SDRAM recovers from the self-refresh mode can be checked by reading the SELFS bit in MIM.
3. After the self-refresh state is cleared, wait for the time required by the DDR-SDRAM before accessing the DDR-SDRAM (130 ns before issuing a command other than read and 200 clock cycles before issuing a read command).
4. When access becomes possible, use the SMS bits in SCR to issue the REFA command so that the concentrated refresh (REFA) is performed on all memory rows.
5. Dummy read a byte from any address.
6. Use the SMS bits in SCR to issue the PREALL command.
7. Use the SMS bits in SCR to issue the REFA command. This operation is required to make the delay adjustment unit in the memory controller operate.
8. Set MIM so that the counter for the auto-refresh function starts counting and thus drives auto-refreshing at a regular interval. After this, normal memory access is possible.

(2) Power-Down Mode (when CKE Goes Low)

When the PCKE bit in MIM is set, the CKE pin level is automatically changed and the DDR-SDRAM then enters or leaves the power-down mode. This mode reduces DDR-SDRAM power consumption.

Since the DDR-SDRAM enters the power-down mode after a memory access and leaves the power-down mode before a memory access, an overhead of one cycle in the external memory clock occurs in each case.

12.5.6 Registers that Set DDR-SDRAM Timing Restrictions

Registers that support connection to memory other than the DDR-SDRAM in a conventional microcomputer and registers that set timing restrictions for the DDR-SDRAM of the DDRIF differ with regard to the setting of the memory timing restrictions. The DDRIF registers are specialized with respect to the DDR-SDRAM timing restrictions. For details, see section 12.4, Register Descriptions.

12.5.7 Operating Frequency

The DDRIF is supported only when the clock ratio between the SuperHyway bus clock and the external memory clock is 1:1 (DDR266 or DDR200). The maximum operating frequency for the SuperHyway bus is 133 MHz. The minimum operating frequency depends on the DDR-SDRAM clock frequency. Therefore see the DDR-SDRAM datasheet.

12.5.8 Note on Clock Stop

The clock supplied to the DDRIF is stopped in the following three modes:

- DDR-SDRAM power supply backup mode
- Software standby mode
- RTC power supply backup mode

Since the clock is not supplied in the above cases, auto-refreshing is not performed. As a result, the refresh cycle is not held and then the DDR-SDRAM data will be damaged. To prevent this, the DDR-SDRAM should enter the self-refresh state through software before the clock supply is stopped. For details on entering/canceling the self-refresh mode, see section 12.5.5 (1), Self-Refresh Mode.

12.5.9 Using SCR to Issue REFA Commands (Outside the Initialization Sequence)

This memory controller automatically opens the DDR-SDRAM bank by memory access (read/write). When issuing the REFA command with the SMS bits in SCR, be sure to close the bank by issuing the PREALL command with the SMS bits in SCR. This operation is also necessary when SCR is used to perform concentrated refresh (REFA) on all rows in the memory before self-refresh operations.

12.5.10 Note on Timing of Connected DDR-SDRAM

This memory controller only supports memory in which the number of cycles (tRAP) required from issuing an ACT command to issuing a read or write with auto-precharge command and the number of cycles (tRDC) required from issuing an ACT command to issuing a read or write command are the same. If the two numbers differ, the DDR-SDRAM should be accessed in bank-open mode.

12.5.11 Note on Setting Auto-Refresh Interval

The auto-refresh interval is specified by the DRI bits in MIM. If the DRE bit is set to 1 at the same time as the DRI bits are set, the time until the first auto-refresh is that selected by the value of the DRI bits before the new setting was made. However, the second and subsequent auto-refresh intervals take on the value corresponding to the new setting for the DRI bits. To avoid this situation, clear the DRE bit to 0 when setting the DRI bits. When the DRE bit is subsequently set to 1, auto-refreshing proceeds with the specified interval from the first round. When writing 1 to the DRE bit, the previously written cycle number should be set to the DRI bits.

12.5.12 Address Multiplexing

Address multiplexing is performed so that the DDR-SDRAM is connected without the external address multiplexing circuit according to the setting of the BW bits in MIM and the SPLIT bits in SDR. Table 12.7 shows the relationship between the DDR-SDRAM bus width and the addresses that are output to the address pins according to the setting of the SPLIT bits. If a setting not specified in table 12.7 is used, correct operation is not guaranteed.

Table 12.7 DDR-SDRAM Address Multiplexing (32-Bit Data Bus)

	SPLIT[3:0]	ROW × COL		M_ BA1	M_ BA0	M_ A13	M_ A12	M_ A11	M_ A10	M_ A9	M_ A8	M_ A7	M_ A6	M_ A5	M_ A4	M_ A3	M_ A2	M_ A1	M_ A0
128 M bits × 2	0001	12 × 9	ROW	13	12	—	—	11	24	23	22	21	20	19	18	17	16	15	14
(8 M × 16-bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	4	3	2
256 M bits × 2	0011	13 × 9	ROW	13	12	—	11	25	24	23	22	21	20	19	18	17	16	15	14
(16 M × 16-bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	4	3	2
512 M bits × 2	0100	13 × 10	ROW	13	12	—	26	25	24	23	22	21	20	19	18	17	16	15	14
(32 M × 16-bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	4	3	2
1 G bits × 2	0110	14 × 10	ROW	13	12	27	26	25	24	23	22	21	20	19	18	17	16	15	14
(64 M × 16-bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	4	3	2

Note: * Auto-precharge

12.5.13 DDR-SDRAM Access Arbitration

(1) Priority Order of Access Arbitration

The DDRIF has the access arbitration function that arbitrates accesses to the DDR-SDRAM between the CPU and the LCDC. The priority order of the arbitration is divided in the following two levels.

At level 0, DDR-SDRAM controls such as DDR-SDRAM refresh and page management have the highest priority. The memory is refreshed according to the memory refresh intervals specified separately.

At level 1, access is rotated between access from the SHway bus and access from the LCDC (in round-robin method). However, immediately after a reset, access from the SHwy bus has priority over access from the LCDC.

Access is not arbitrated based on the order of requests but by a request signal that is asserted between transactions. When read and write requests are made for the same device simultaneously, a read request has priority. Access arbitration is performed between transactions.

(2) Access Arbitration When Burst and Non-Burst Transfers Coexist

The arbiter block receives inputs from the SHwy bus with the 133*[1] MHz interface and the LCDC with the 66*[2] MHz interface. Therefore, arbitration operation differs depending on a burst and non-burst transfers

- In burst transfers, an arbitrated module can perform continuous transfers. Therefore, the SHwy bus and the LCDC have the same possibility of being arbitrated.
- In non-burst transfers, arbitration is performed in 133*[1] MHz units. Requests are continuously output to the DDRC.

However, in actuality, burst and non-burst transfers coexist. Signals output from the interfaces are used to determine whether a burst transfer is expected or not. When making arbitration from non-burst transfers to burst transfers, if the arbitrated transaction is a burst transfer, the non-burst transfer is continued after the burst transfer.

Notes: 1. This indicates the clock frequency when DDR266-SDRAM is used. The clock frequency is 100 MHz when DDR200-SDRAM is used.
2. This indicates the clock frequency when DDR266-SDRAM is used. The clock frequency is 50 MHz when DDR200-SDRAM is used.

12.5.14 Coherency When Accessing DDR-SDRAM

In some cases, writing the DDR-SDRAM via the SHwy bus by software may be held for some reason and reading the DDR-SDRAM by the subsequently activated LCDC may be executed first. That is, incorrect operation may occur if coherency for accessing the DDR-SDRAM is not guaranteed.

In this case, execute the SYNCO instruction between the write instruction for the DDR-SDRAM by software and the LCDC activation instruction. When the SYNCO instruction is executed, the next instruction is not activated until the data access being performed is completed.

12.6 DDRIF Basic Timing

Figures 12.5 to 12.14 show examples of basic DDRIF timing.

In every figure, the DDR-SDRAM is idle at T0.

The various timings should be set in the STR register within the range specified by the DDR-SDRAM used.

Note that the DDRIF only supports 2.5-cycle CAS latency (CL).

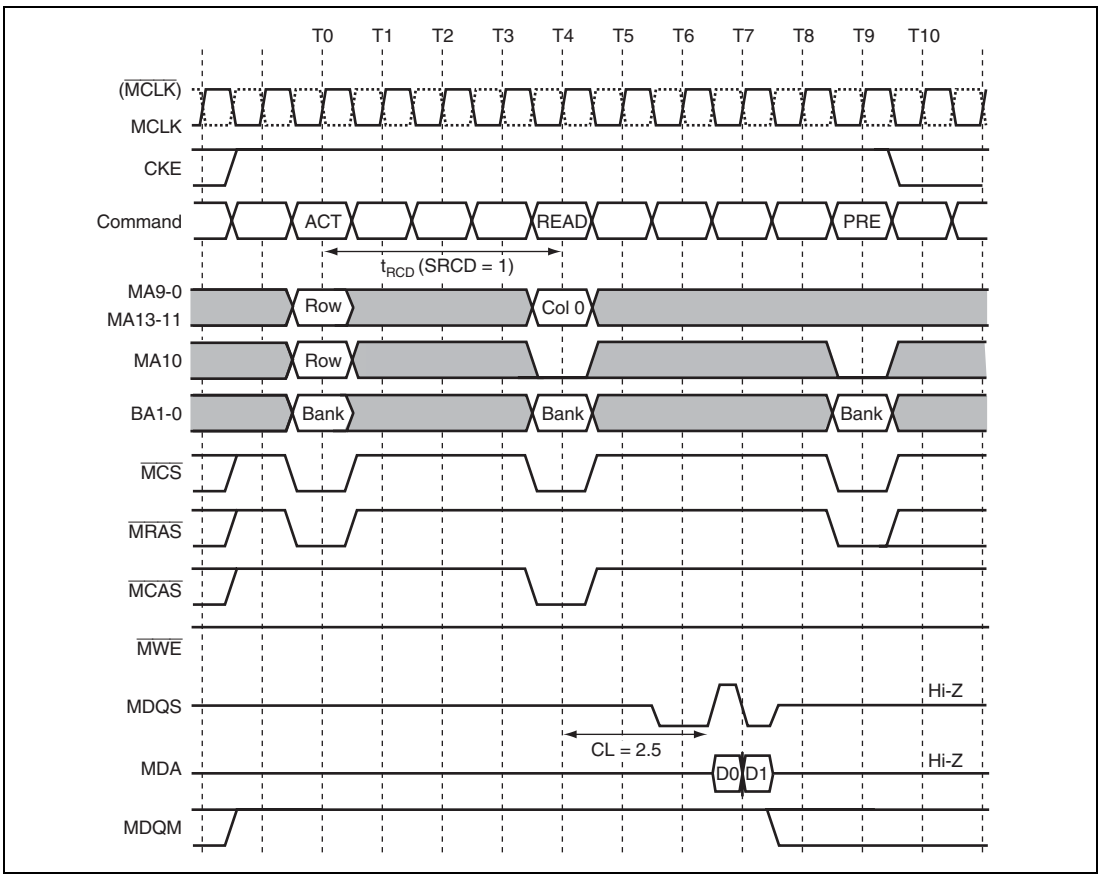


Figure 12.5 Basic DDRIF Timing (1 Burst Read: 1, 2, 4, or 8 Bytes; Without Auto-Precharge)

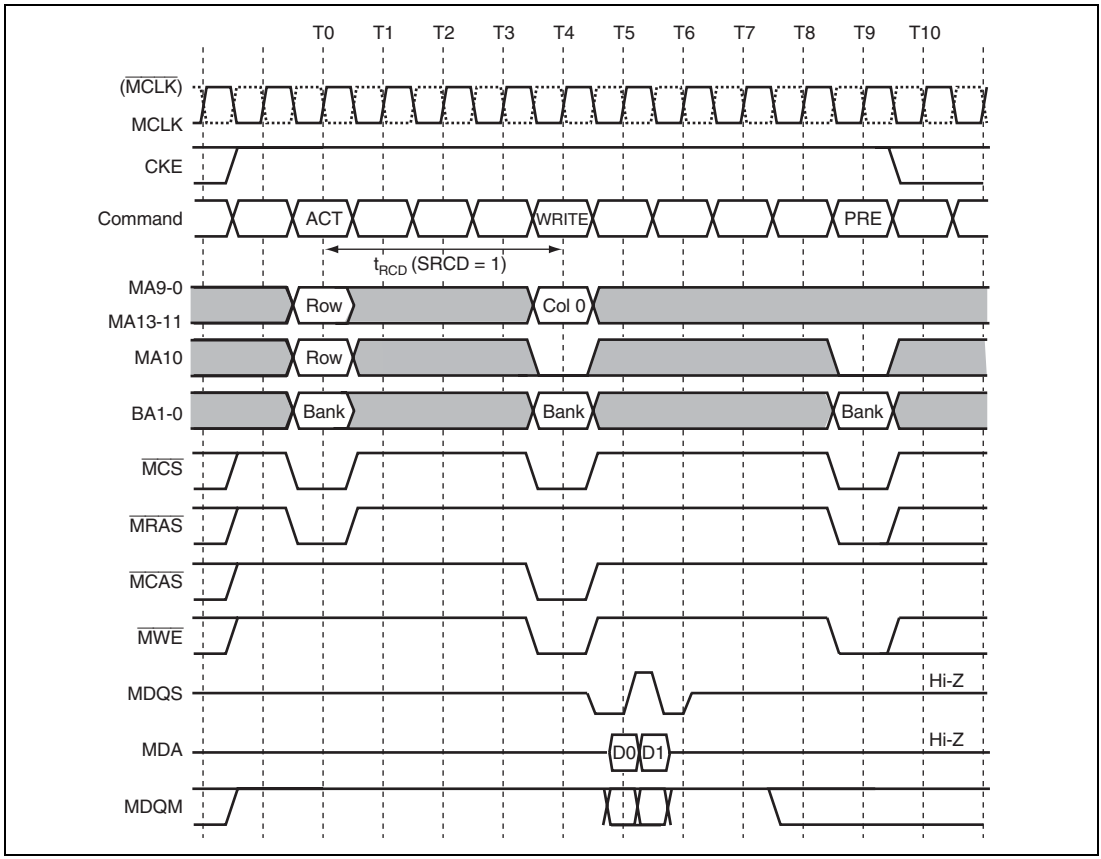


Figure 12.6 Basic DDRIF Timing (1 Burst Write: 1, 2, 4, or 8 Bytes; Without Auto-Precharge)

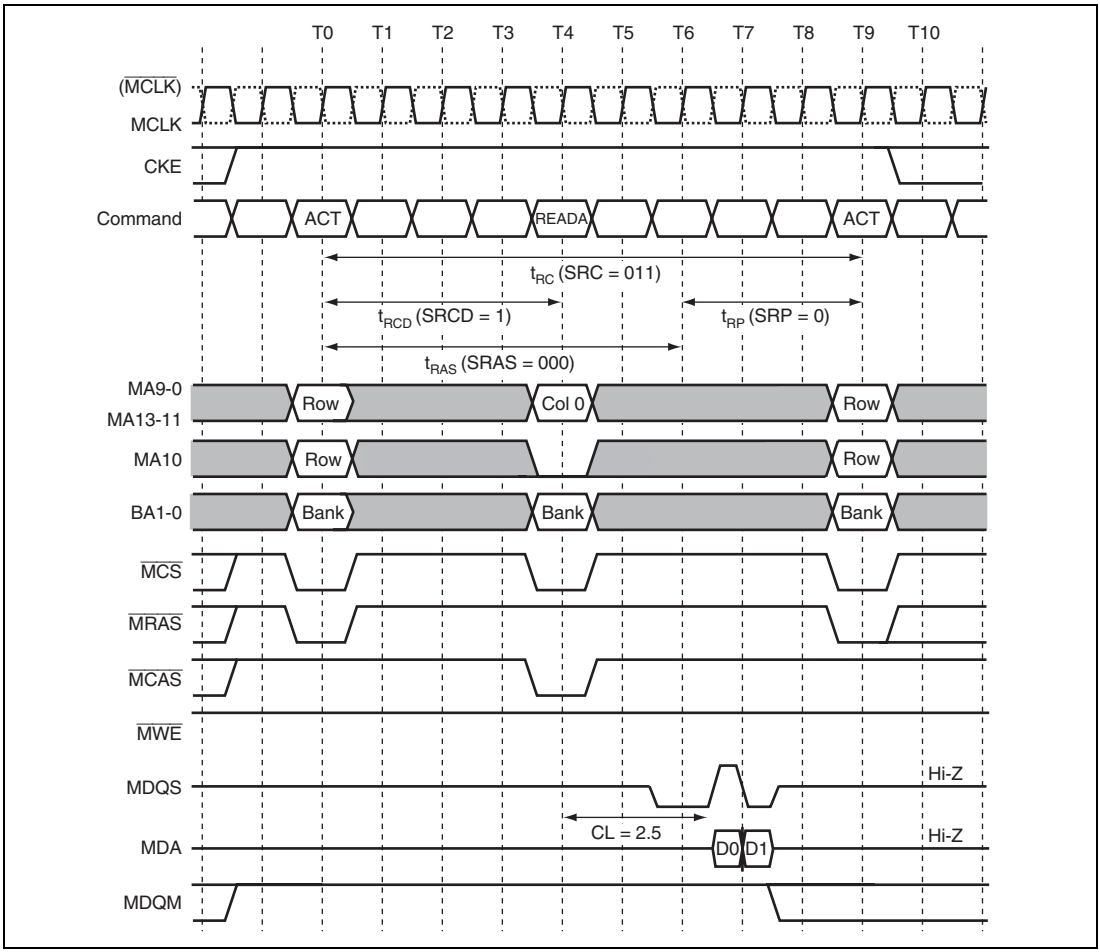


Figure 12.7 Basic DDRIF Timing (1 Burst Read: 1, 2, 4, or 8 Bytes; With Auto-Precharge)

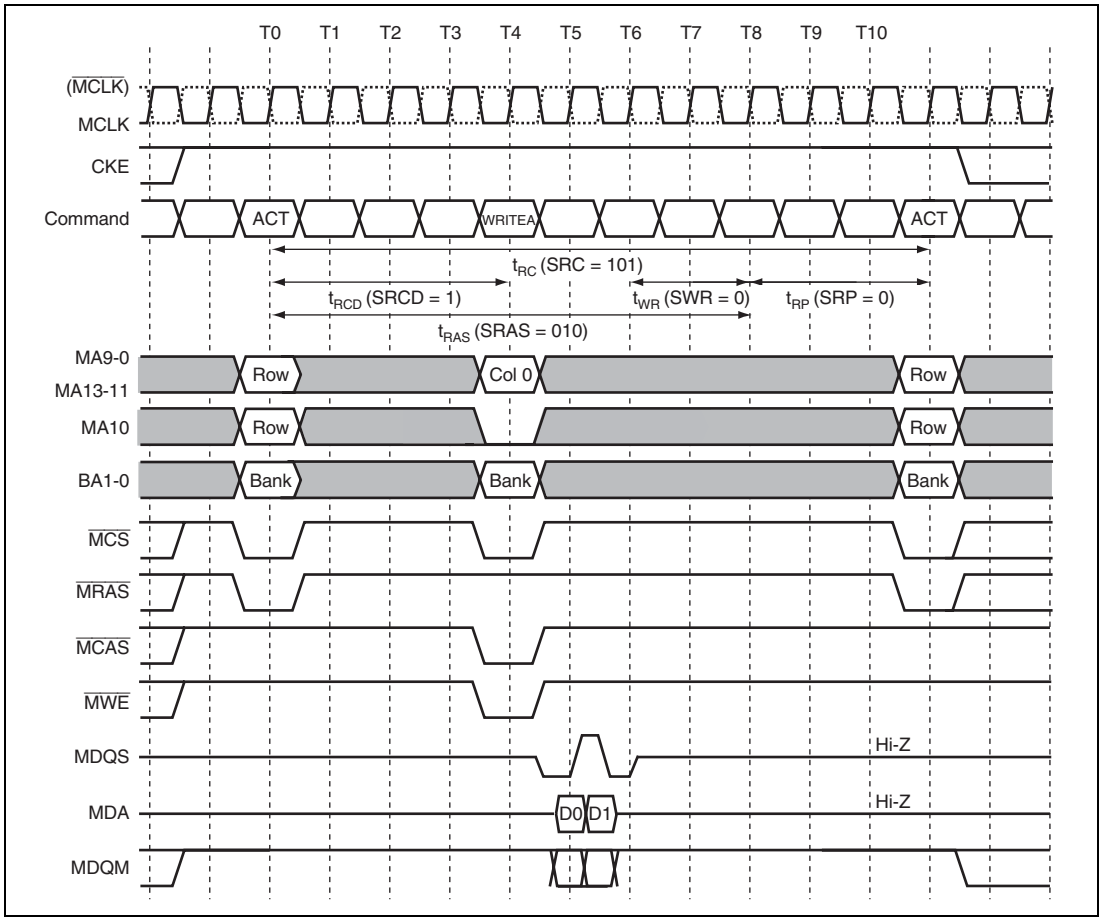


Figure 12.8 Basic DDRIF Timing (1 Burst Write: 1, 2, 4, or 8 Bytes; With Auto-Precharge)

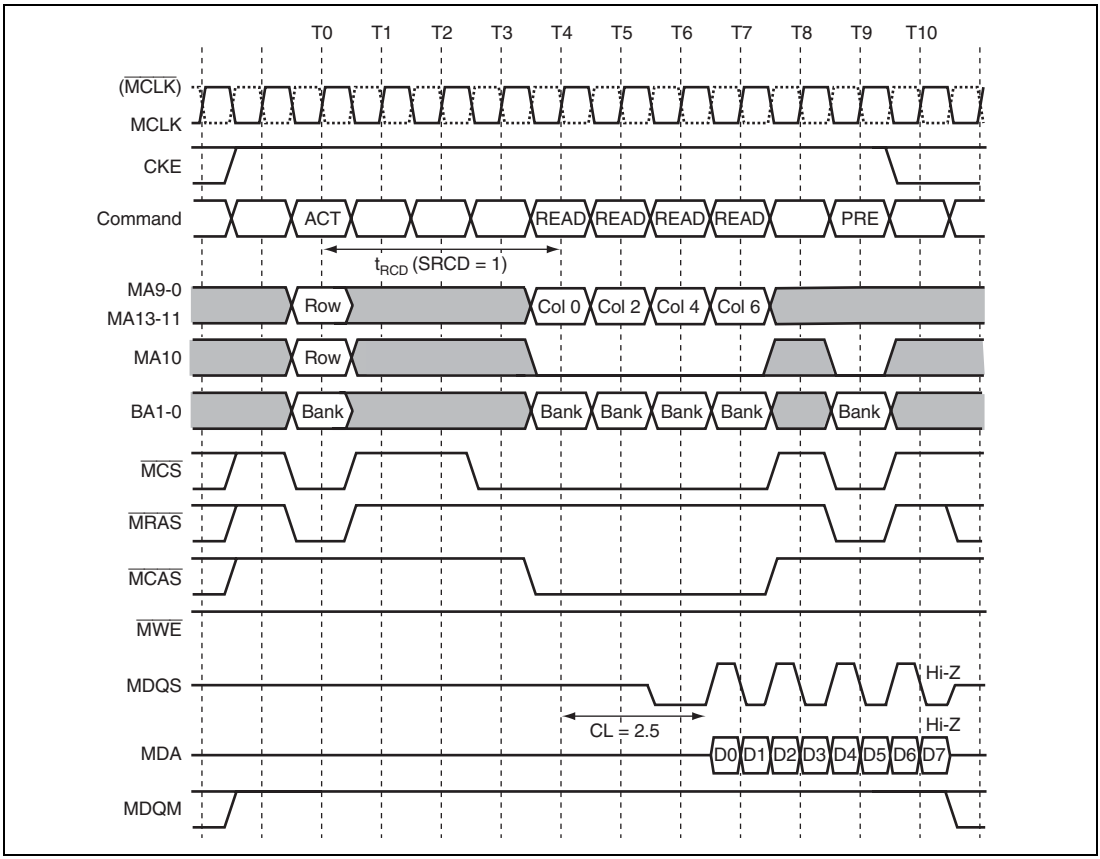


Figure 12.9 Basic DDRIF Timing (4 Burst Read: 32 Bytes; Without Auto-Precharge)

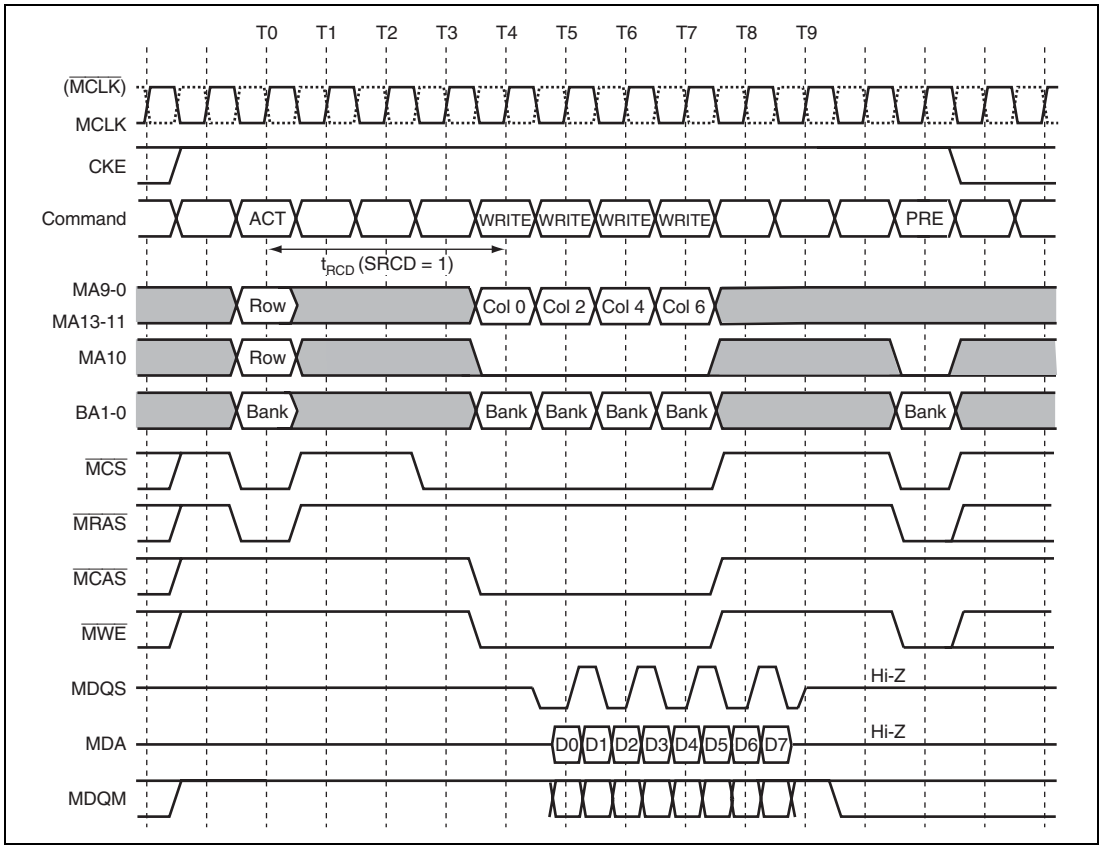


Figure 12.10 Basic DDRIF Timing (4 Burst Write: 32 Bytes; Without Auto-Precharge)

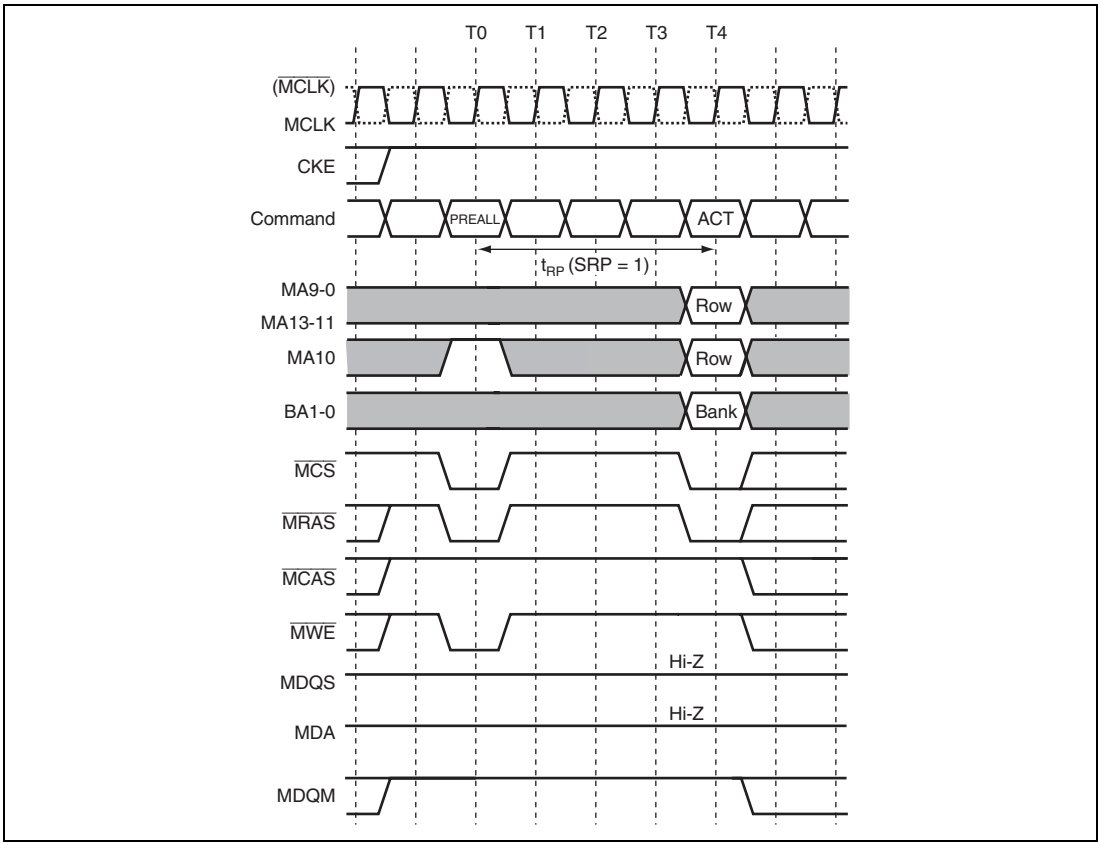
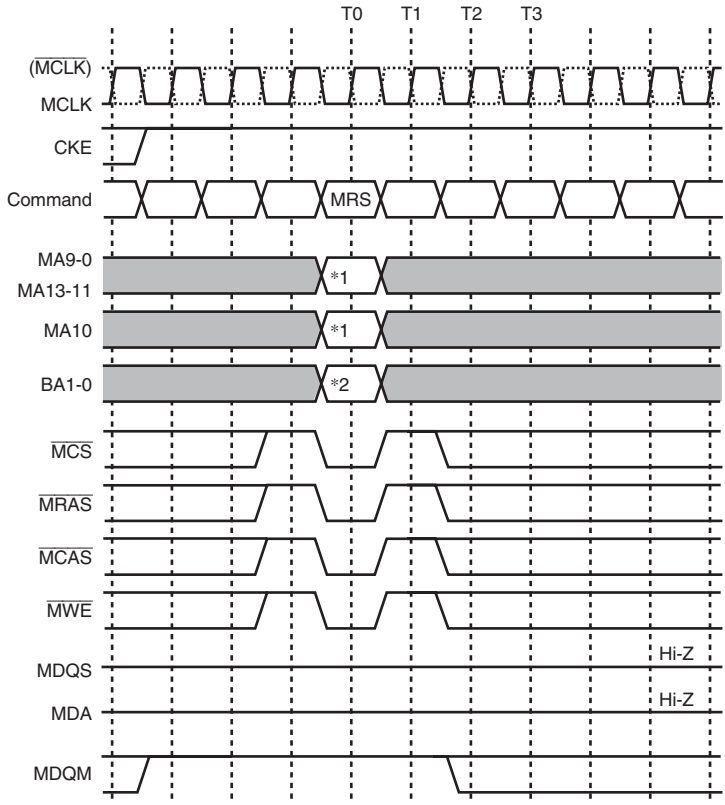


Figure 12.11 Basic DDRIF Timing (Precharge all Banks (PREALL) to Bank Activate (ACT))



Notes: 1. Sets the operating mode and other necessary parameters.
2. For mode register setting: BA1 = low, BA0 = low
For extended mode register setting: BA1 = low, BA0 = high

Figure 12.12 Basic DDRIF Timing (Mode Register Set (MRS))

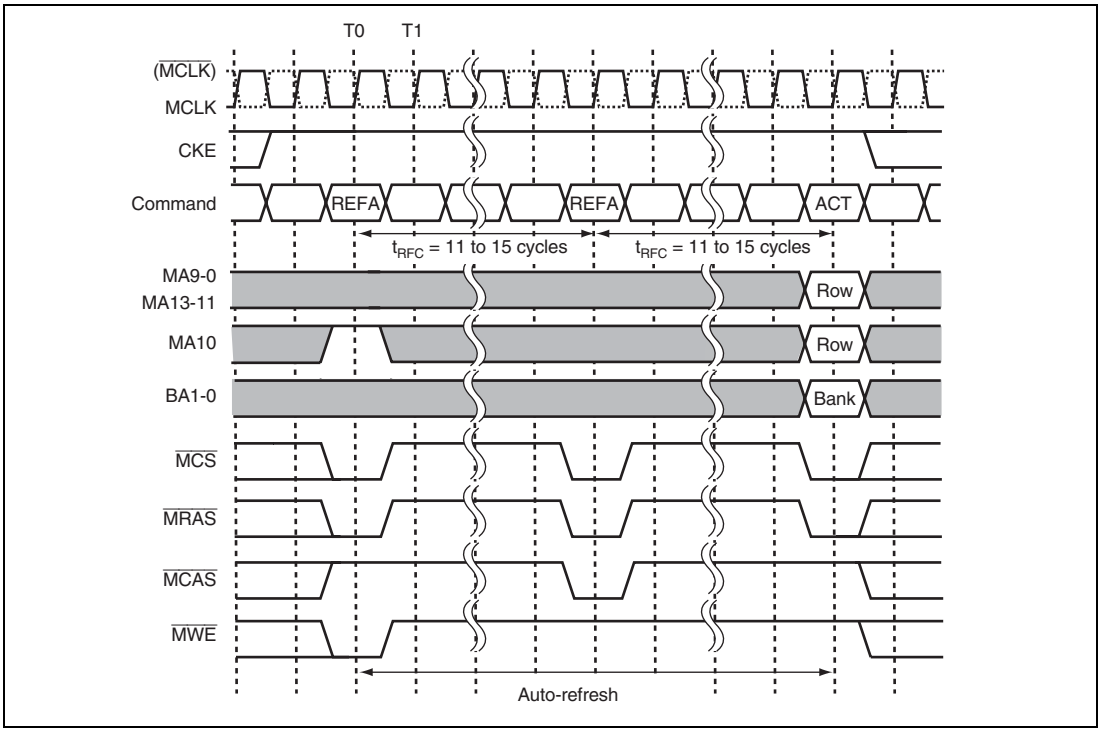
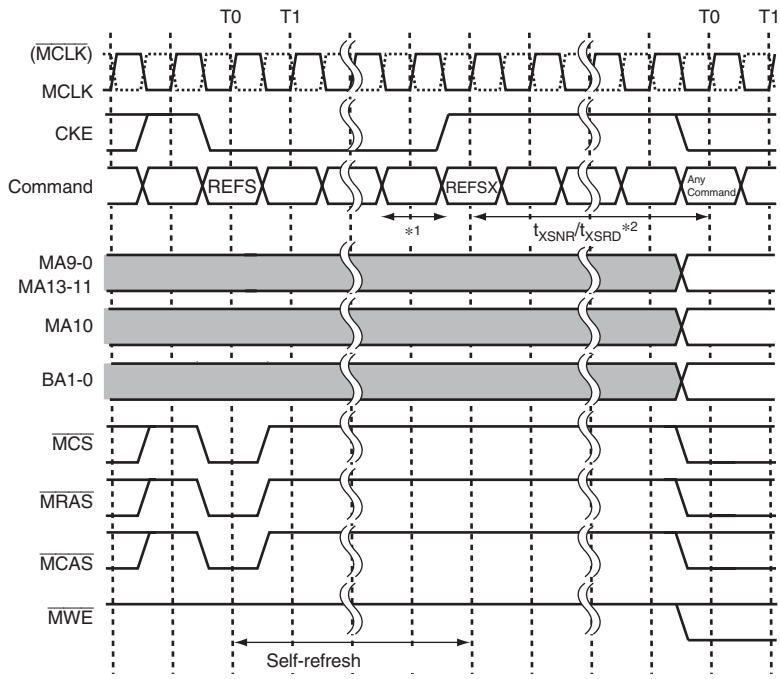


Figure 12.13 Basic DDRIF Timing (Auto-Refresh (REFA) Enter/Exit to Bank Activate (ACT))



Notes: 1. This timing should satisfy the conditions specified by the DDR-SDRAM used when driving CKE high.
2. This timing should satisfy the conditions specified by the DDR-SDRAM used.
(txSNR is for a non-READ command and txSRD is for a READ command;
txSRD should usually be 200 clock cycles or longer.)

Figure 12.14 Basic DDRIF Timing (Self-Refresh Entry from IDLE (REFS)/Self-Refresh Exit (REFSX) to Any Command Input)

Section 13 PCI Controller (PCIC)

The PCI controller (PCIC) controls the PCI bus for data transfers between memory connected to an external bus and a PCI device connected to the PCI bus. The ability to connect PCI devices facilitates the design of systems using the PCI bus and enables more compact systems capable of faster data transfer.

The PCIC functions as a bus bridge which connects an external PCI bus to the internal SuperHyway bus. It provides a device connected to the external PCI bus with a channel for access to the on-chip modules connected to the SuperHyway bus. The PCIC supports both the host bus bridge mode and normal mode (non-host mode). In host busbridge mode, PCI bus arbitration control is available and in normal mode, arbitration is executed by the external PCI bus arbiter.

13.1 Features

The PCIC has the following features:

- Supports subset of PCI Local Bus Specification Revision 2.2
- PCI bus operating speeds of 33 MHz/66 MHz
- 32-bit data bus
- PCI master and target functions
- Supports subset of PCI power management Revision 1.1
- Supports the host bus bridge mode and normal mode (selectable by MD6 pin settings)
- Supports the PCI bus arbiter (in host bus bridge mode)
  - Supports four external masters
  - Pseudo-round-robin or fixed priority arbitration
  - Supports external bus arbiter mode
- Supports configuration mechanism #1 (in host bus bridge mode)
- Supports burst transfer
- Parity check and error report

- Exclusive access (target only)
  - Once locked, only accessible from the device that accessed the $\overline{\text{LOCK}}$ signal
  - The SuperHyway bus in not locked during lock transfer
- Can support cache coherency between a device connected to the PCI bus and system memory (PCI target) although device performance may become suboptimal
- Supports four external interrupt inputs ($\overline{\text{INTD}}$ to $\overline{\text{INTA}}$) in host bus bridge mode
- Supports one external interrupt output ($\overline{\text{INTA}}$) in normal mode
- Supports both big endian and little endian formats for the SuperHyway bus (the PCI bus operates in the little endian format)
- Number of devices which can be connected
  - 33 MHz: 4 or less
  - 66 MHz: 1

The PCIC does not support the following PCI functions.

- Cache support (no $\overline{\text{SBO}}$ or SDONE pin)
- Address wrap-around mechanism
- PCI JTAG (other modules in this LSI can support the JTAG feature)
- Dual address cycles
- Interrupt acknowledge cycles
- Fast back-to-back transfer initiation (supported when performed as a target device)
- Extended ROM for initialization and system boot

  etc.

Note: When the ratio of the clocks (SHwy clock : PCICLK clock) is in the ranges of (2.1 : 1) to (3.3 : 1), the PCIC cannot be used.

Figure 13.1 is a block diagram of the PCIC.

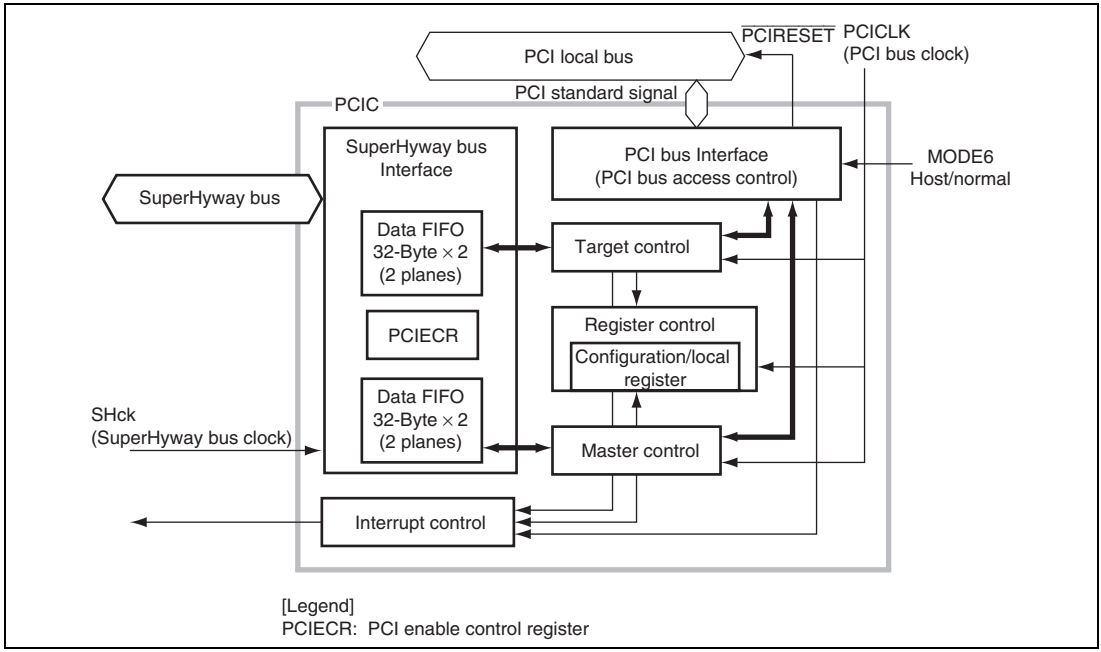


Figure 13.1 PCIC Block Diagram

The PCIC comprises two blocks: the PCI bus interface and SuperHyway bus interface block.

The PCI bus interface block comprises the PCI configuration register, local register, PCI master, and PCI target controller.

The functions of the PCI bus interface are transaction control on the PCI local bus.

The SuperHyway bus interface block comprises the control register (PCIECR) and the data FIFO.

The functions of the SuperHyway bus interface are access translation between the PCI bus interface and the CPU or DMAC via SuperHyway bus.

The interrupt controller requests interrupt request to the INTC of this LSI.

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the PCIC.

Table 13.1 Input/Output Pins

Pin Name	PCI standard signal name	I/O	Description
AD31 to AD0	AD[31:0]	I/O	PCI Address/Data Bus
		(TRI)	Address and data buses are multiplexed. Each bus transaction consists of an address phase followed by one or more data phases.
CBE3 to CBE0	C/$\overline{\text{BE}}$[3:0]	I/O	PCI Command/Byte Enable
		(TRI)	Bus command and byte enables are multiplexed. These signals indicate the type of transaction during the address phase and the byte enables during the data phases.
PAR	PAR	I/O	PCI Parity
		(TRI)	Generates/checks even parity across AD[31:0] and CBE[3:0].
PCICLK	CLK	Input	PCI Clock
			Provides timing for all transactions on the PCI bus.
$\overline{\text{PCIFRAME}}$	$\overline{\text{FRAME}}$	I/O	PCI Frame
		(STRI)	Current initiator drives this signal, which indicates the start and duration or end of a transaction.
$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	I/O	PCI Target Ready
		(STRI)	Selected target drives this signal, which indicates the target is ready to execute a transaction. During a write, this signal indicates that the target is ready to accept data. During a read, this signal indicates that valid data is present on the AD [31:0] lines.
$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	I/O	PCI Initiator Ready
		(STRI)	The current bus master drives this signal. During a write, this signal indicates that valid data is present on the AD [31:0] lines. During a read, this signal indicates that the master is ready to accept data.
$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	I/O	PCI Stop
		(STRI)	Selected target drives this signal to stop the current transaction.
$\overline{\text{LOCK}}$	$\overline{\text{LOCK}}$	I/O	PCI Lock
		(STRI)	

Pin Name	PCI standard signal name	I/O	Description
IDSEL	IDSEL	Input	PCI Configuration Device Select This signal is input to the PCI device to select configuration cycles (only for normal mode).
$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	I/O (STRI)	PCI Device Select Indicates the device driving this signal has decoded its address as the target. As an input, this signal indicates that a device has been selected.
$\overline{\text{INTD}}$ $\overline{\text{INTC}}$ $\overline{\text{INTB}}$	$\overline{\text{INTD}}$ $\overline{\text{INTC}}$ $\overline{\text{INTB}}$	Input	Interrupts D, C, and B Indicate that a PCI device is requesting an interrupt. Only these signals are available in host bus bridge mode.
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	I/O (output: O/D)	Interrupt A Indicates that a PCI device is requesting an interrupt (input) in host bus bridge mode. This signal is used to request an interrupt (output: O/D) in normal mode.
$\overline{\text{REQ3}}$ to $\overline{\text{REQ1}}$*[4]	$\overline{\text{REQ[3:1]}}$	Input	PCI Bus Request Available only in host bus bridge mode.
$\overline{\text{GNT3}}$ to $\overline{\text{GNT1}}$	$\overline{\text{GNT[3:1]}}$	Output (TRI)	PCI Bus Grant Available only in host bus bridge mode.
$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$	$\overline{\text{REQ0}}$	I/O (TRI)	PCI Bus Request Functions as an input or an output in host bus bridge mode and as an output in normal mode.
$\overline{\text{GNT0}}$/ $\overline{\text{GNTIN}}$	$\overline{\text{GNT0}}$	I/O (TRI)	PCI Bus Grant Functions as an input or an output in host bus bridge mode and as an input in normal mode.
$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	I/O (output: O/D)	PCI System Error
$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	I/O (TRI)	PCI Parity Error
$\overline{\text{PCIRESET}}$	—	Output	PCI reset output (only for host bus bridge mode)
MD6*	—	Input	PCI Operating Mode Select Low: PCI normal mode in which the PCIC operates as a PCI bridge on the PCICLK High: PCI host bus bridge mode in which the PCIC operates as a PCI bridge on the PCICLK

[Legend]

TRI: Tri-state

STRI: Sustained tri-state

O/D: Open Drain

Note: * Clear the PCIC-related interrupt masks only after the PCIC-related pins are selected as the PCIC.

13.3 Register Descriptions

Table 13.2 shows the PCIC register configuration. Table 13.3 shows the register states in each operating mode. The PCI configuration register address and its offset are used for little endian operation.

Table 13.2 List of PCIC Registers

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
Control register space						
PCIC enable control register	PCIECR	R/W	—	H'FE00 0008	H'1E00 00008	32
PCI configuration register space						
PCI vendor ID register	PCIVID	R	R	H'FE04 0000	H'1E04 0000	16
PCI device ID register	PCIDID	R	R	H'FE04 0002	H'1E04 0002	16
PCI command register	PCICMD	R/W	R/W	H'FE04 0004	H'1E04 0004	16
PCI status register	PCISTATUS	R/WC	R/WC	H'FE04 0006	H'1E04 0006	16
PCI revision ID register	PCIRID	R	R	H'FE04 0008	H'1E04 0008	8
PCI program interface register	PCIPIF	R/W	R	H'FE04 0009	H'1E04 0009	8
PCI sub class code register	PCISUB	R/W	R	H'FE04 000A	H'1E04 000A	8
PCI base class code register	PCIBCC	R/W	R	H'FE04 000B	H'1E04 000B	8
PCI cacheline size register	PCICLS	R	R	H'FE04 000C	H'1E04 000C	8
PCI latency timer register	PCILTM	R/W	R/W	H'FE04 000D	H'1E04 000D	8
PCI header type register	PCIHDR	R	R	H'FE04 000E	H'1E04 000E	8
PCI BIST register	PCIBIST	R	R	H'FE04 000F	H'1E04 000F	8
PCI I/O base address register	PCIIBAR	R/W	R/W	H'FE04 0010	H'1E04 0010	32
PCI Memory base address register 0	PCIMBAR0	R/W	R/W	H'FE04 0014	H'1E04 0014	32
PCI Memory base address register 1	PCIMBAR1	R/W	R/W	H'FE04 0018	H'1E04 0018	32
PCI subsystem vendor ID register	PCISVID	R/W	R	H'FE04 002C	H'1E04 002C	16
PCI subsystem ID register	PCISID	R/W	R	H'FE04 002E	H'1E04 002E	16
PCI capabilities pointer register	PCICP	R	R	H'FE04 0034	H'1E04 0034	8
PCI interrupt line register	PCIINTLINE	R/W	R/W	H'FE04 003C	H'1E04 003C	8
PCI interrupt pin register	PCIINTPIN	R/W	R	H'FE04 003D	H'1E04 003D	8

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
PCI minimum grant register	PCIMINGNT	R	R	H'FE04 003E	H'1E04 003E	8
PCI maximum latency register	PCIMAXLAT	R	R	H'FE04 003F	H'1E04 003F	8
PCI capability ID register	PCICID	R	R	H'FE04 0040	H'1E04 0040	8
PCI next item pointer register	PCINIP	R	R	H'FE04 0041	H'1E04 0041	8
PCI power management capability register	PCIPMC	R/W	R/W	H'FE04 0042	H'1E04 0042	16
PCI power management control/status register	PCIPMCSR	R/W	R/W	H'FE04 0044	H'1E04 0044	16
PCI PMCSR bridge support extension register	PCIPMCSR BSE	R	R	H'FE04 0046	H'1E04 0046	8
PCI power consumption/dissipation data register	PCIPCDD	R/W	R	H'FE04 0047	H'1E04 0047	8
PCI local register space						
PCI control register	PCICR	R/W	R	H'FE04 0100	H'1E04 0100	32
PCI local space register 0	PCILSR0	R/W	R	H'FE04 0104	H'1E04 0104	32
PCI local space register 1	PCILSR1	R/W	R	H'FE04 0108	H'1E04 0108	32
PCI local address register 0	PCILAR0	R/W	R	H'FE04 010C	H'1E04 010C	32
PCI local address register 1	PCILAR1	R/W	R	H'FE04 0110	H'1E04 0110	32
PCI interrupt register	PCIIR	R/WC	R	H'FE04 0114	H'1E04 0114	32
PCI interrupt mask register	PCIIMR	R/W	R	H'FE04 0118	H'1E04 0118	32
PCI error address information register	PCIAIR	R	R	H'FE04 011C	H'1E04 011C	32
PCI error command information register	PCICIR	R	R	H'FE04 0120	H'1E04 0120	32
PCI arbiter interrupt register	PCIAINT	R/WC	R	H'FE04 0130	H'1E04 0130	32
PCI arbiter interrupt mask register	PCIAINTM	R/WC	R	H'FE04 0134	H'1E04 0134	32
PCI arbiter bus master error information register	PCIBMIR	R	R	H'FE04 0138	H'1E04 0138	32
PCI PIO*[3] address register	PCIPAR	R/W	—	H'FE04 01C0	H'1E04 01C0	32
PCI power management interrupt register	PCIPINT	R/WC	—	H'FE04 01CC	H'1E04 01CC	32
PCI power management interrupt mask register	PCIPINTM	R/W	—	H'FE04 01D0	H'1E04 01D0	32

Name	Abbreviation	SH*[1] R/W	PCI*[1] R/W	P4 address	Area 7 address	Access Size*[2]
PCI memory bank register 0	PCIMBR0	R/W	—	H'FE04 01E0	H'1E04 01E0	32
PCI memory bank mask register 0	PCIMBMR0	R/W	—	H'FE04 01E4	H'1E04 01E4	32
PCI memory bank register 1	PCIMBR1	R/W	—	H'FE04 01E8	H'1E04 01E8	32
PCI memory bank mask register 1	PCIMBMR1	R/W	—	H'FE04 01EC	H'1E04 01EC	32
PCI memory bank register 2	PCIMBR2	R/W	—	H'FE04 01F0	H'1E04 01F0	32
PCI memory bank mask register 2	PCIMBMR2	R/W	—	H'FE04 01F4	H'1E04 01F4	32
PCI I/O bank register	PCIIOBR	R/W	—	H'FE04 01F8	H'1E04 01F8	32
PCI I/O bank master register	PCIIOBMR	R/W	—	H'FE04 01FC	H'1E04 01FC	32
PCI cache snoop control register 0	PCICSCR0	R/W	—	H'FE04 0210	H'1E04 0210	32
PCI cache snoop control register 1	PCICSCR1	R/W	—	H'FE04 0214	H'1E04 0214	32
PCI cache snoop address register 0	PCICSAR0	R/W	—	H'FE04 0218	H'1E04 0218	32
PCI cache snoop address register 1	PCICSAR1	R/W	—	H'FE04 021C	H'1E04 021C	32
PCI PIO*[3] data register	PCIPDR	R/W	—	H'FE04 0220	H'1E04 0220	32

Notes: 1. SH: SuperHyway bus (internal bus). PCI: PCI local bus. WC: Cleared by writing 1 (Writing of 0 is no effect). —: Accessing is prohibited.
2. When accessing a register, do not use a size smaller than the register's access size.
3. PIO: Programmed I/O.

Table 13.3 Register States in Each Operating Mode

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode	Standby
Control register space					
PCIC enable control register	PCIECR	H'0000 0000	H'0000 0000	Retained	Retained
PCI configuration register space					
PCI vendor ID register	PCIVID	H'1912	H'1912	Retained	Retained
PCI device ID register	PCIDID	H'0004	H'0004	Retained	Retained
PCI command register	PCICMD	H'0080	H'0080	Retained	Retained
PCI status register	PCISTATUS	H'0290	H'0290	Retained	Retained
PCI revision ID register	PCIRID	H'00	H'00	Retained	Retained
PCI program interface register	PCIPIF	H'00	H'00	Retained	Retained
PCI sub class code register	PCISUB	H'00	H'00	Retained	Retained
PCI base class code register	PCIBCC	H'00	H'00	Retained	Retained
PCI cache line size register	PCICLS	H'20	H'20	Retained	Retained
PCI latency timer register	PCILTM	H'00	H'00	Retained	Retained
PCI header type register	PCIHDR	H'00	H'00	Retained	Retained
PCI BIST register	PCIBIST	H'00	H'00	Retained	Retained
PCI I/O base address register	PCIIBAR	H'0000 0001	H'0000 0001	Retained	Retained
PCI Memory base address register 0	PCIMBAR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI Memory base address register 1	PCIMBAR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI subsystem vendor ID register	PCISVID	H'0000	H'0000	Retained	Retained
PCI subsystem ID register	PCISID	H'0000	H'0000	Retained	Retained
PCI capabilities pointer register	PCICP	H'40	H'40	Retained	Retained
PCI interrupt line register	PCIINTLINE	H'00	H'00	Retained	Retained
PCI interrupt pin register	PCIINTPIN	H'01	H'01	Retained	Retained
PCI minimum grant register	PCIMINGNT	H'00	H'00	Retained	Retained
PCI maximum latency register	PCIMAXLAT	H'00	H'00	Retained	Retained
PCI capability ID register	PCICID	H'01	H'01	Retained	Retained
PCI next item pointer register	PCINIP	H'00	H'00	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode	Standby
PCI power management capability register	PCIPMC	H'000A	H'000A	Retained	Retained
PCI power management control/status register	PCIPMCSR	H'0000	H'0000	Retained	Retained
PCI PMCSR bridge support extension register	PCIPMCSR BSE	H'00	H'00	Retained	Retained
PCI power consumption/dissipation data register	PCIPCDD	H'00	H'00	Retained	Retained
PCI local register space					
PCI control register	PCICR	H'0000 00xx	H'0000 00xx	Retained	Retained
PCI local space register 0	PCILSR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI local space register 1	PCILSR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI local address register 0	PCILAR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI local address register 1	PCILAR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI interrupt register	PCIIR	H'0000 0000	H'0000 0000	Retained	Retained
PCI interrupt mask register	PCIIMR	H'0000 0000	H'0000 0000	Retained	Retained
PCI error address information register	PCIAIR	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
PCI error command information register	PCICIR	H'xx00 000x	H'xx00 000x	Retained	Retained
PCI arbiter interrupt register	PCIAINT	H'0000 0000	H'0000 0000	Retained	Retained
PCI arbiter interrupt mask register	PCIAINTM	H'0000 0000	H'0000 0000	Retained	Retained
PCI arbiter bus master error information register	PCIBMIR	H'0000 00xx	H'0000 00xx	Retained	Retained
PCI PIO address register	PCIPAR	H'80xx xxxx	H'80xx xxxx	Retained	Retained
PCI power management interrupt register	PCIPINT	H'0000 0000	H'0000 0000	Retained	Retained
PCI power management interrupt mask register	PCIPINTM	H'0000 0000	H'0000 0000	Retained	Retained
PCI memory bank register 0	PCIMBR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI memory bank mask register 0	PCIMBMR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI memory bank register 1	PCIMBR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI memory bank mask register 1	PCIMBMR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI memory bank register 2	PCIMBR2	H'0000 0000	H'0000 0000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep Mode	Standby
PCI memory bank mask register 2	PCIMBMR2	H'0000 0000	H'0000 0000	Retained	Retained
PCI I/O bank register	PCIIOBR	H'0000 0000	H'0000 0000	Retained	Retained
PCI I/O bank master register	PCIIOBMR	H'0000 0000	H'0000 0000	Retained	Retained
PCI cache snoop control register 0	PCICSCR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI cache snoop control register 1	PCICSCR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI cache snoop address register 0	PCICSAR0	H'0000 0000	H'0000 0000	Retained	Retained
PCI cache snoop address register 1	PCICSAR1	H'0000 0000	H'0000 0000	Retained	Retained
PCI PIO data register	PCIPDR	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained

[Legend] x: Undefined

13.3.1 PCIC Enable Control Register (PCIECR)

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ENBL	0	R/W	PCI Enable Bit. Enable the PCIC 0: PCIC disable The access from both the CPU and external PCI devices to the PCIC is invalid (including the configuration and local register), except PCIECR. 1: PCIC enable

13.3.2 Configuration Registers

The configuration registers defines the programming model and usages for the configuration register space in a PCI compliant device. For details, refer to “PCI Local Bus Specification Revision 2.2 Chapter 6 Configuration Space ”.

(1) PCI Vender ID Register (PCIVID)

This register identifies the manufacturer of device.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	VID	H'1912	SH: R	PCI Vender ID
			PCI: R	Indicates the PCI device manufacture identifier (vender ID) that is allocated by PCI-SIG. Renesas Technology’s vendor ID is H'1912.

(2) PCI Device ID Register (PCIDID)

This register uniquely identifies this LSI amongst PCI devices manufactured by the vendor.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DID	H'0004	SH: R	PCI Device ID
			PCI: R	These bits uniquely identify this LSI amongst PCI devices manufactured by the vendor indicated by the PCI vender field. This LSI’s device ID is H'0004.

(3) PCI Command Register (PCICMD)

The PCI command register provides coarse control over a device's ability to generate and respond to PCI cycles. When 0 is written to this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FBBE	SERRE	WCC	PER	VGAPS	MWIE	SC	BM	MS	IOS
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
9	FBBE	0	SH: R	PCI Fast Back-to-Back Enable
			PCI: R	Controls whether or not a master can do fast back-to-back transactions to different device.
				0: Fast back-to-back transactions are only allowed to the same target
				1: Master is allowed to generate fast back-to-back transactions to different targets (not supported)
8	SERRE	0	SH: R/W	PCI $\overline{\text{SERR}}$ Output Control
			PCI: R/W	Controls the $\overline{\text{SERR}}$ output.
				0: $\overline{\text{SERR}}$ output disabled
				1: $\overline{\text{SERR}}$ output enabled
7	WCC	1	SH: R/W	Wait Cycle Control
			PCI: R/W	Controls the address/data stepping.
				When WCC = 1, both an address and data for a master write, only an address for a master read, and only data for a target read are output for at least two clock cycles.
				0: Address/data stepping control disabled
				1: Address/data stepping control enabled

Bit	Bit Name	Initial Value	R/W	Description
6	PER	0	SH: R/W	Parity Error
			PCI: R/W	Controls the device's response when the PCIC detects a parity error or receives a parity error. When this bit is set to 1, the $\overline{\text{PERR}}$ signal is asserted.
				0: No response parity error
				1: Response parity error
5	VGAPS	0	SH: R	VGA Palette Snoop Control
			PCI: R	0: VGA compatible device
				1: Palette register write is not supported (not supported)
4	MWIE	0	SH: R	PCI Memory Write and Invalidate Control
			PCI: R	Controls issuance of a memory write and invalidate command in a master access.
				0: Memory write is used
				1: Memory write and invalidate command is executable (not supported)
3	SC	0	SH: R	PCI Special Cycles
			PCI: R	Indicates whether or not to support the special cycle operations in a target access.
				0: Special cycles ignored
				1: Special cycles monitored (not supported)
2	BM	0	SH: R/W	PCI Bus Master Control
			PCI: R/W	Controls a bus master.
				0: Bus master function disabled
				1: Bus master function enabled
1	MS	0	SH: R/W	PCI Memory Space Control
			PCI: R/W	Controls accesses to memory space of this LSI. When this bit is cleared to 0, a memory transfer to the PCIC is terminated with a master abort.
				0: Does not respond to memory space accesses
				1: Respond to memory space accesses
0	IOS	0	SH: R/W	PCI I/O Space
			PCI: R/W	Controls accesses to I/O space of this LSI. When this bit is cleared to 0, a I/O transfer to the PCIC is terminated with a master abort.
				0: Does not respond to I/O space accesses
				1: Respond to I/O space accesses

(4) PCI Status Register (PCISTATUS)

This status register is used to record status information for PCI bus related events. The definition of each of the bits is given in the table below. A device may not need to implement all the bits, depending on device functionality. For instance, since a device that acts as a target does not inform a target abort, bit 11 does not need to be implemented. Reserved bits should be read-only and return zero when the bits are read.

Reads from this register operates normally. Writes are slightly different in that bits can be cleared, but not set. A one bit is cleared whenever the register is written to, and the write data in the corresponding bit location is a 1. For instance, to clear bit 14 and not affect any other bits, write the value of B'0100 0000 0000 0000 to the register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPE	SSE	RMA	RTA	STA	DEVSEL		MDPE	FBBC	—	66C	CL	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
SH R/W:	R/WC	R/WC	R/WC	R/WC	R/WC	R	R	R/WC	R	R/W	R/W	R	R	R	R	R
PCI R/W:	R/WC	R/WC	R/WC	R/WC	R/WC	R	R	R/WC	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	DPE	0	SH: R/WC	Parity Error Detect Status
			PCI: R/WC	Indicates that a parity error has been detected in read data when the PCIC is a master or in write data when the PCIC is a target. This bit must be set by the device whenever it detects a parity error, even if parity error handling is disabled. 0: Device is not detecting parity error. 1: Device is detecting parity error.
14	SSE	0	SH: R/WC	System Error Output Status
			PCI: R/WC	Indicates that the PCIC has asserted the $\overline{\text{SERR}}$ signal. 0: $\overline{\text{SERR}}$ has not been asserted 1: $\overline{\text{SERR}}$ has been asserted (the value retained until cleared)

Bit	Bit Name	Initial Value	R/W	Description
13	RMA	0	SH: R/WC	Master Abort Receive Status
			PCI: R/WC	Indicates that the PCIC has terminated a transaction with a master abort when the PCIC is a master.
				0: PCIC has not terminated a transaction with a master abort
				1: PCIC has terminated a transaction with a master abort
12	RTA	0	SH: R/WC	Target Abort Receive Status
			PCI: R/WC	Indicates that a transaction is terminated by a target device with a target abort when the PCIC functions as a master.
				0: Transaction has not been terminated with a target abort
				1: Transaction has been terminated with a target abort
11	STA	0	SH: R/WC	Target Abort Execution Status
			PCI: R/WC	Indicates that the PCIC has terminated a transaction with a target-abort when the PCIC functions as a target.
				0: PCIC has not terminated a transaction with a target-abort
				1: PCIC has terminated a transaction with target-abort
10, 9	DEVSEL	01	SH: R	DEVSEL Timing Status
			PCI: R	Indicate the response timing status of the $\overline{\text{DEVSEL}}$ signal when the PCIC functions as a target.
				00: Fast (not support)
				01: Medium
				10: Slow (not support)
				11: Reserved
8	MDPE	0	SH: R/WC	Data parity error
			PCI: R/WC	Indicates that the PCIC has asserted the $\overline{\text{PERR}}$ signal or detected the assertion of the $\overline{\text{PERR}}$ signal if the PCIC functions as a master. Only when the parity response bit has been set to 1, this bit is set to 1.
				0: Data parity error has not been generated
				1: Data parity error has been generated

Bit	Bit Name	Initial Value	R/W	Description
7	FBBC	1	SH: R	Fast Back-to-Back Status
			PCI: R	Indicates whether or not the PCIC is capable of accepting fast back-to-back transactions when the transactions are not to the same agent if the PCIC functions as a target.
				0: Fast back-to-back transactions to different agents not supported
				1: Fast back-to-back transactions to different agents supported
6	—	0	SH: R/W	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
5	66C	0	SH: R/W	66MHz-Operation Capable Status
			PCI: R	Indicates whether or not the PCIC is capable of running at 66MHz.
				0: PCIC runs at 33 MHz
				1: PCIC runs at 66 MHz
4	CL	1	SH: R	PCI Power Management (Optional Function)
			PCI: R	Indicates whether or not the PCI power management function is supported.
				0: Power management not supported
				1: Power management supported
3 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

(5) PCI Revision ID Register (PCIRID)

This register specifies a device specific revision identifier.

Bit:	7	6	5	4	3	2	1	0
	RID							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RID	H'00	SH: R	Revision ID
			PCI: R	Indicates the PCIC revision.The initial value is H'00.RID value varies according to the logic version of the PCIC and it may be changed in the future.

(6) PCI Program Interface Register (PCIPIF)

This register is the programming interface for the IDE controller class code. For details of the class code, refer to “PCI Local Bus Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	MIDED	—	—	—	PIS	OMS	PIP	OMP
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MIDED	0	SH: R/W	PCI Master IDE Device
			PCI: R	Specifies the PCI master IDE device.
				1: PCI master IDE device
				0: PCI slave IDE device
				When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3	PIS	0	SH: R/W	PCI Programmable Indicator (Secondary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.
2	OMS	0	SH: R/W	PCI Operating Mode (Secondary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.
1	PIP	0	SH: R/W	PCI Programmable Indicator (Primary)
			PCI: R	When the CFINIT bit in CR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.
0	OMP	0	SH: R/W	PCI Operating Mode (Primary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is writable. When the CFINIT bit in PCICR is 1, writing is ignored. This bit is readable.

(7) PCI Sub Class Code Register (PCISUB)

This register identifies the sub class code. For details of the class code, refer to “PCI Local Bus Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	SUB							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SUB	H'00	SH: R/W	Sub Class Code
			PCI: R	Indicate the sub class code. The initial value is H'00.

(8) PCI Base Class Code Register (PCIBCC)

This register identifies the base class code. For details of the class code, refer to “PCI Local Bus Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	BCC							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BCC	H'00	SH: R/W	Base Class Code
			PCI: R	Indicates the base class code. The initial value is H'00.

(9) PCI Cacheline Size Register (PCICLS)

Bit:	7	6	5	4	3	2	1	0
	CLS							
Initial value:	0	0	1	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CLS	H'20	SH: R	Cache Line Size: Not supported
			PCI: R	A memory target does not support a cache. SDON and $\overline{\text{SBO}}$ are ignored.

(10) PCI Latency Timer Register (PCILTM)

This register specifies, in units of PCI bus clocks, the value of latency timer for this PCI bus master.

Bit:	7	6	5	4	3	2	1	0
	LTM							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	LTM	H'00	SH: R/W	PCI Latency Timer
			PCI: R/W	Specifies the maximum number of acquisition clocks of PCI bus when the PCIC is operating as the master.

(11) PCI Header Type Register (PCIHDR)

Bit:	7	6	5	4	3	2	1	0
	MFE	HDR						
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MFE	0	SH: R	Multiple Function Enable
			PCI: R	0: Single function
				1: Multiple (from two to eight) functions (not supported)
6 to 0	HDR	H'00	SH: R	Configuration Layout
			PCI: R	Indicates the layout type of configuration registers.
				H'00: Type "00h" layout supported
				H'01: Type "01h" layout supported (not supported)

(12) PCI BIST Register (PCIBIST)

Bit:	7	6	5	4	3	2	1	0
	BISTC	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BISTC	0	SH: R	This bit is used to control the BIST function and status.
			PCI: R	
				0: Function not available
				1: Function available (not supported)
6 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

(13) PCI I/O Base Address Register (PCIIBAR)

This register packages the I/O space base address register of the PCI configuration register that is prescribed with PCI local bus specification.

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IOB (upper)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOB (upper)								IOB (lower)						—	ASI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	IOB (upper)	H'000000	SH: R/W	I/O Space Base Address (upper 24 bits)
			PCI: R/W	Specifies the upper 24 bits of I/O base address that corresponds the PCIC local register space (PCIC control register space).
7 to 2	IOB (lower)	000000	SH: R	I/O Space Base Address (lower 6 bits)
			PCI: R	These bits are fixed 000000 by hardware.
1	—	0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
0	ASI	1	SH: R	Address Space Indicator
			PCI: R	Indicates whether the base address in this register indicates the I/O or memory space.
				0: Memory space
				1: I/O space

(14) PCI Memory Base Address Register 0 (PCIMBAR0)

This register packages the memory space base address register of the PCI configuration register that is prescribed with PCI local bus specification.

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBA (upper)												MBA (lower)			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBA (lower)												LAP	LAT		ASI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	MBA (upper)	H'000	SH: R/W	Memory Space 0 Base Address (upper 12 bits)
			PCI: R/W	Specifies the upper 12 bits of memory base address that corresponds the local address space 0 (SuperHyway bus address space of this LSI).
				Update value
				PCILSR [28:20] — Address space — Effective bit of MBA (upper)
				0 0000 0000 — 1 Mbyte — [31:20]
				0 0000 0001 — 2 Mbytes — [31:21]
				0 0000 0011 — 4 Mbytes — [31:22]
				⋮ — ⋮ — ⋮
				0 1111 1111 — 256 Mbytes — [31:28]
				1 1111 1111 — 512 Mbytes — [31:29]
19 to 4	MBA (lower)	H'0000	SH: R	Memory Space 0 Base Address (lower 16 bits)
			PCI: R	These bits are fixed H'0000 by hardware.
3	LAP	0	SH: R	Prefetch Control
			PCI: R	Indicates whether or not local address space 0 is prefetchable.
				0: Not prefetchable
				1: Prefetchable (not supported)

Bit	Bit Name	Initial Value	R/W	Description
2, 1	LAT	00	SH: R	Memory Type
			PCI: R	Indicates the memory type of local address space 0.
				00: 32-bit base address and 32-bit space
				01: 32-bit base address and 1-Mbyte space (Not supported)
				10: 64-bit base address (Not supported)
				11: Reserved
0	ASI	0	SH: R	Address Space Indicator
			PCI: R	Indicates whether the base address in this register indicates the I/O or memory space.
				0: Memory space
				1: I/O space

(15) PCI Memory Base Address Register 1 (PCIMBAR1)

This register packages the memory space base address register of the PCI configuration register that is prescribed with PCI local bus specification.

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBA (upper)												MBA (lower)			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBA (lower)												LAP	LAT		ASI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	MBA (upper)	H'000	SH: R/W PCI: R/W	PCI Memory Space 1 Base Address (upper 12 bits) Specifies the upper 12 bits of PCI memory base address that corresponds the base address of local address space 1 (SuperHyway bus address space of this LSI). PCILSE0 [28:20] — Address space — Effective bit of MBA (upper) 0 0000 0000 — 1 Mbyte — [31:20] 0 0000 0001 — 2 Mbytes — [31:21] 0 0000 0011 — 4 Mbytes — [31:22] ⋮ — ⋮ — ⋮ 0 1111 1111 — 256 Mbytes — [31:28] 1 1111 1111 — 512 Mbytes — [31:29]
19 to 4	MBA (lower)	H'0000	SH: R PCI: R	Memory Space 1 Base Address (lower 16 bits) These bits are fixed H'0000 by hardware.
3	LAP	0	SH: R PCI: R	Prefetch Control Indicates whether or not local address space 1 is prefetchable. 0: Not prefetchable 1: Prefetchable (Not supported)
2, 1	LAT	00	SH: R PCI: R	Memory Type Indicates the memory type of local address space 1. 00: 32-bit base address and 32-bit space 01: 32-bit base address and 1-Mbyte space (Not supported) 10: 64-bit base address (Not supported) 11: Reserved
0	ASI	0	SH: R PCI: R	Address Space Indicator Indicates whether the base address in this register indicates the I/O or memory space. 0: Memory space 1: I/O space

(16) PCI Subsystem vender ID Register (PCISVID)

Refer to miscellaneous registers section of PCI local bus specification Revision 2.2.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	SVID	H'0000	SH: R/W	Subsystem Vendor ID
			PCI: R	Specifies the subsystem vendor ID of the PCIC. The initial value is H'0000.
				This field can be modified during initializing PCIC registers (PCICR.CFINIT = 0), but cannot be modified after initialized PCIC register (PCICR.CFINIT = 1) even if writing this field.

(17) PCI Subsystem ID Register (PCISID)

Refer to section about miscellaneous registers of PCI local bus specification Revision 2.2.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	SSID	H'0000	SH: R/W	Subsystem ID
			PCI: R	Specifies the subsystem ID of the PCIC. The initial value is H'0000.
				This field can be modified during initializing PCIC registers (PCICR.CFINIT = 0), but cannot be modified after initialized PCIC register (PCICR.CFINIT = 1) even if writing this field.

(18) PCI Capability Pointer Register (PCICP)

This register is the expansion function pointer register of the PCI configuration register that is prescribed in the PCI power management specification.

Bit:	7	6	5	4	3	2	1	0
	CP							
Initial value:	0	1	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CP	H'40	SH: R	Capabilities pointer
			PCI: R	The offset address of the expansion function register.

(19) PCI Interrupt Line Register (PCIINTLINE)

Bit:	7	6	5	4	3	2	1	0
	INTLINE							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	INTLINE	H'00	SH: R/W	PCI Interrupt Line
			PCI: R/W	PCI interrupt connected to the external interrupt of this LSI. Specify these bits by system software during initialization. The initial value is H'00. The setting value of this field does not affect the operation of this LSI.

(20) PCI Interrupt Pin Register (PCIINTPIN)

Bit:	7	6	5	4	3	2	1	0
	INTPIN							
Initial value:	0	0	0	0	0	0	0	1
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	INTPIN	H'01	SH: R/W	Interrupt Pin Select
			PCI: R	Specifies which interrupt pin is used for connection when the PCIC outputs interrupt request.
				H'00: Does not connect $\overline{\text{INTD}}$ to $\overline{\text{INTA}}$
				H'01: $\overline{\text{INTA}}$ is used to request an interrupt
				H'02: $\overline{\text{INTB}}$ is used to request an interrupt
				H'03: $\overline{\text{INTC}}$ is used to request an interrupt
				H'04: $\overline{\text{INTD}}$ is used to request an interrupt
				H'05 to H'FF: Reserved

(21) PCI Minimum Grant Register (PCIMINGNT)

This register is not programmable.

Bit:	7	6	5	4	3	2	1	0
	MINGNT							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	MINGNT	H'00	SH: R	Minimum Grant
			PCI: R	Specify the burst time to be required by the master device (not supported).

(22) PCI Maximum Latency Register (PCIMAXLAT)

This register is not programmable.

Bit:	7	6	5	4	3	2	1	0
	MAXLAT							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	MAXLAT	H'00	SH: R	Maximum Latency
			PCI: R	Specify the worst time from the bus request by the PCI master device to the bus acquisition (not supported).

(23) PCI Capability Identifier Register (PCICID)

When H'01 is read by system software, it indicates that the data structure currently being pointed to is the PCI power management data structure. Each function of a PCI device may have only one item in its capability list with PCICID set to H'01.

Bit:	7	6	5	4	3	2	1	0
	CID							
Initial value:	0	0	0	0	0	0	0	1
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CID	H'01	SH: R	Expansion Function ID
			PCI: R	Specifies the expansion function ID.
				H'01: The expansion function is power management.

(24) PCI Next Item Pointer Register (PCINIP)

PCINIP gives the location of the next item in the function's capability list.

Bit:	7	6	5	4	3	2	1	0
	NIP							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NIP	H'00	SH: R	Next Item Pointer
			PCI: R	Specifies the offset to the next expansion function.
				H'00: Power management function is listed as the last item.

(25) PCI Power Management Capability Register (PCIPMC)

PCIPMCS is a 16-bit register that provides information on the capabilities of the power management related functions. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”. This register must be set during initializing the PCIC registers (PCICR.CFINIT = 0).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCS					D2S	D1S	—	—	—	DSI	—	PMEC	PMV		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
SH R/W:	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	PMCS	00000	SH: R PCI: R	PME_SUPPORT This 5-bit field indicates the power states in which the function may assert $\overline{\text{PME}}$. A value of 0b for any bit indicates that the function is not capable of asserting the $\overline{\text{PME}}$ signal while in that power state. Bit11: xxxx1 - $\overline{\text{PME}}$ can be asserted from D0 Bit12: xxx1x - $\overline{\text{PME}}$ can be asserted from D1 Bit13: xx1xx - $\overline{\text{PME}}$ can be asserted from D2 Bit14: x1xxx - $\overline{\text{PME}}$ can be asserted from D3 hot Bit15: 1xxxx - $\overline{\text{PME}}$ can be asserted from D3 cold Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
10	D2S	0	SH: R/W PCI: R	When this bit is 1, This function supports the D2 power management state. When the D2 power management state is not supported, this bit is read as 0.
9	D1S	0	SH: R/W PCI: R	When this bit is 1, This function supports the D1 power management state. When the D1 power management state is not supported, this bit is read as 0.

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
5	DSI	0	SH: R	DSI
			PCI: R	Specifies whether or not the device requires the specific initialization. 0: Does not require the specific initialization
4	—	0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3	PMEC	1	SH: R/W	PCI PME clock
			PCI: R	Specifies whether or not the device requires the clock to support $\overline{\text{PME}}$ generation. 1: Requires the clock to support $\overline{\text{PME}}$ generation Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
2 to 0	PMV	010	SH: R/W	Version
			PCI: R	Specifies the version of the power management specifications. 010: This LSI's power management specification is conformed to revision 1.1

(26) PCI Power Management Control/Status Register (PCIPMCSR)

This 16-bit register is used to manage the PCI function's power management status as well as to enable/monitor PMEs. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMES	DSC		DSL				PME EN	—	—	—	—	—	—	PS	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PMES	0	SH: R	PME Status
			PCI: R	Indicates the state of the $\overline{\text{PME}}$ signal. (Not supported)
				Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
14, 13	DSC	00	SH: R	Data Scale
			PCI: R	Specify the scaling of data field. (Not supported)
12 to 9	DSL	0000	SH: R	Data Select
			PCI: R	Specify the data output in the data filed.
8	PMEEN	0	SH: R	PME Enable
			PCI: R	Controls the $\overline{\text{PME}}$ output. (Not supported)
				Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
7 to 2	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
1, 0	PS	00	SH: R/W	Power State
			PCI: R/W	Specifies the power state.
				If software attempts to write an unsupported, optional state to these bits, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.
				00: D0 state
				01: D1 state
				10: D2 state
				11: D3 hot state (power-down mode)

(27) PCIPMCSR Bridge Support Extension Register (PCIPMCSRBSE)

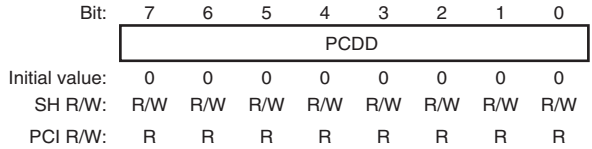
This register supports PCI bridge specific functionality and is required for all PCI-to-PCI bridges.

Bit:	7	6	5	4	3	2	1	0
	BPCCEN	B2B3N	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BPCCEN	0	SH: R PCI: R	When the bus power/clock control mechanism is disabled, the power state bits in bridge's PCIPMCSR cannot be used by the system software to control the power or clock of the bridge's secondary bus.
6	B2B3N	0	SH: R PCI: R	The state of this bit determines the action that is to occur as a direct result of programming the function to the D3 hot state. 0: Indicates that when the bridge function is set to the D3 hot state, its secondary bus will have its power removed (B3). 1: Indicates that when the bridge function is set to the D3 hot state, its secondary bus's PCI clock will be stopped (B2). This bit is only valid if bit 7 (BPCCEN) is set to 1.
5 to 0	—	All 0	SH: R PCI: R	Reserved These bits are always read as 0. The write value should always be 0.

(28) PCI Power Consumption/Radiation Register (PCIPCDD)

The data register is an 8-bit register that provides a mechanism for the function to report state dependent operating data such as power consumed or heat dissipation. For details, refer to “PCI Bus Power Management Interface Specification Revision 1.1 Chapter 3 PCI Power Management Interface”.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PCDD	H'00	SH: R/W PCI: R	This register is used to report the state dependent data requested by the PCIPMCSR.DSL bits. The value of this register is scaled by the value reported by the PCIPMCSR.DSC bits.

13.3.3 Local Register

(1) PCI Control Register (PCICR)

PCICR is a 32-bit register which specifies the operation of the PCIC.

The register is write protected; only writes in which the upper eight bits (that is, bits 31 to 24) have the value H'A5 are performed. All other writes are ignored.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PFCS	FTO	PFE	TBS	—	BMAM	—	—	—	IOCS	RST CTL	CFI NIT
Initial value:	0	0	0	0	0	0	0	0	0	0	—	—	0	0	0	0
SH R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	H'00	SH: R/W	Reserved
			PCI: R	Set these bits to H'A5 only when writing to bits 11 to 8, 6, and 3 to 0.
				These bits are always read as 0.
23 to 12	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
11	PFCS	0	SH: R/W	PCI Pre-Fetch Command Setting
			PCI: R	This bit is valid only when the PFE bit is 1.
				0: Always 8-byte pre-fetching
				1: Always 32-byte pre-fetching
10	FTO	0	SH: R/W	PCI $\overline{\text{TRDY}}$ Control Enable
			PCI: R	In a target access, negate the $\overline{\text{TRDY}}$, within 5 cycles before disconnection.
				0: Disabled
				1: Enabled
9	PFE	0	SH: R/W	PCI Pre-Fetch Enable
			PCI: R	0: Disabled
				1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
8	TBS	0	SH: R/W	Byte Swap
			PCI: R	Specifies whether or not byte data is swapped when accessing to the PCI local bus. 0: No swap 1: Byte data is swapped For details, see section 13.4.3 (5), Endian or section 13.4.4 (6), Endian.
7	—	0	SH: R	Reserved
			PCI: R	This bit is always read as 0. The write value should always be 0.
6	BMAM	0	SH: R/W	Bus Master Arbitration
			PCI: R	Controls the PCI bus arbitration mode when the PCIC operates in host bus bridge mode. This bit is ignored when the PCIC operates in normal mode. 0: Fixed mode (PCIC > device0 > device1 > device2 > device3) 1: Pseudo round robin (the most recently granted device is assigned the lowest priority)
5, 4	—	Undefined	SH: R	Reserved
			PCI: R	These bits are always read as an undefined value. The write value should always be 0.
3	—	0	SH: R	Reserved
			PCI: R	This bit is always read as 0. The write value should always be 0.
2	IOCS	0	SH: R/W	$\overline{\text{INTA}}$ Output
			PCI: R	Controls the $\overline{\text{INTA}}$ output by software. This bit is valid only in normal mode. 0: Makes $\overline{\text{INTA}}$ output high-impedance state (driven high by pull-up register) 1: Asserts $\overline{\text{INTA}}$ output (low level output)

Bit	Bit Name	Initial Value	R/W	Description
1	RSTCTL	0	SH: R/W	$\overline{\text{PCIRESET}}$ Output
			PCI: R	Controls the $\overline{\text{PCIRESET}}$ output by software. This bit is valid when the PCIC operates in host bus bridge mode. 0: Negates $\overline{\text{PCIRESET}}$ output (high level output) 1: Asserts $\overline{\text{PCIRESET}}$ output (low level output) Note: The $\overline{\text{PCIRESET}}$ is also asserted during power-on reset.
0	CFINIT	0	SH: R/W	PCI Internal Register Initialize Control
			PCI: R	Set this bit to 1 after the initialization of the PCIC internal registers are completed. Setting this bit enables accesses from the PCI bus. During initialization in host bus bridge mode, the bus is not given to the device on the PCI bus. In normal mode, the PCIC returns RETRY when it is accessed from the PCI bus. 0: During initialization 1: Initialization completed

(2) PCI Local Space Register 0 (PCILSR0)

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
28 to 20	LSR	0 0000 0000	SH: R/W	Size of Local Address Space 0 (9 bits)
			PCI: R	Specify the size of local address space 0 (SuperHyway bus address space of this LSI) in units of Mbyte. The value set in these bits must be the size minus 1 Mbytes. Setting all the bits to 0 ensures 1-Mbyte space. 0 0000 0000: 1 Mbyte 0 0000 0001: 2 Mbytes 0 0000 0011: 4 Mbytes 0 0000 0111: 8 Mbytes 0 0000 1111: 16 Mbytes 0 0001 1111: 32 Mbytes 0 0011 1111: 64 Mbytes 0 0111 1111: 128 Mbytes 0 1111 1111: 256 Mbytes 1 1111 1111: 512 Mbytes Other than above: Setting prohibited
19 to 1	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
0	MBARE	0	SH: R/W	PCI Memory Base Address Register 0 Enable
			PCI: R	The local address space 0 can be accessed by setting this bit to 1. 0: PCIMBAR0 disabled 1: PCIMBAR0 enabled

(3) PCI Local Space Register 1 (PCILSR1)

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
28 to 20	LSR	0 0000 0000	SH: R/W	Size of Local Address Space 1 (9 bits)
			PCI: R	Specify the size of local address space 1 (SuperHyway bus address space of this LSI) in units of Mbyte. The value set in these bits must be the size minus 1 Mbytes. Setting all the bits to 0 ensures 1-Mbyte space. 0 0000 0000: 1 Mbyte 0 0000 0001: 2 Mbytes 0 0000 0011: 4 Mbytes 0 0000 0111: 8 Mbytes 0 0000 1111: 16 Mbytes 0 0001 1111: 32 Mbytes 0 0011 1111: 64 Mbytes 0 0111 1111: 128 Mbytes 0 1111 1111: 256 Mbytes 1 1111 1111: 512 Mbytes Other than above: Setting prohibited
19 to 1	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	MBARE	0	SH: R/W	PCI Memory Base Address Register 1 Enable
			PCI: R	The local address space 1 can be accessed by setting this bit to 1.
				0: PCIMBAR1 disabled
				1: PCIMBAR1 enabled

(4) PCI Local Address Register 0 (PCILAR0)

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	LAR	H'000	SH: R/W	Local Address (12 bits)
			PCI: R	Specify bits 31 to 20 of the start address in local address space 0.
				The effective bits of LAR depend on the capacity of local address space 0 as specified in PCILSR0.
				The effective bits are as follows:
				PCILSR0.LS0([28:20]) = 0 0000 0000: Effective bits are [31:20]
				PCILSR0.LS0([28:20]) = 0 0000 0001: Effective bits are [31:21]
				PCILSR0.LS0([28:20]) = 0 0000 0011: Effective bits are [31:22]
				| |
				PCILSR0.LS0([28:20]) = 0 1111 1111: Effective bits are [31:28]
				PCILSR0.LS0([28:20]) = 1 1111 1111: Effective bits are [31:29]
19 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

(5) PCI Local Address Register 1 (PCILAR1)

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	LAR	H'000	SH: R/W	Local Address (12 bits)
			PCI: R	Specify bits 31 to 20 of the start address in local address space 1. The effective bits of LAR depend on the capacity of local address space 1 as specified in PCILSR1. The effective bits are as follows: PCILSR1.LS1([28:20]) = 0 0000 0000: Effective bits are [31:20] PCILSR1.LS1([28:20]) = 0 0000 0001: Effective bits are [31:21] PCILSR1.LS1([28:20]) = 0 0000 0011: Effective bits are [31:22] | | PCILSR0.LS1([28:20]) = 0 1111 1111: Effective bits are [31:28] PCILSR1.LS1([28:20]) = 1 1111 1111: Effective bits are [31:29]
19 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

(6) PCI Interrupt Register (PCIIR)

PCIIR records the source of an interrupt.

When multiple interrupts occur, only the first source is registered.

When an interrupt is disabled, the source is registered in corresponding bit (set to 1) in this register, however, no interrupt occurs.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TTADI	—	—	—	—	TMTOI	MDEI	APEDI	SEDI	DPEITW	DPEITR	TADIM	MADIM	MWPDI	MRDPEI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R/WC	R	R	R	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
14	TTADI	0	SH: R/WC	Target Target-Abort Interrupt
			PCI: R	Indicates that the PCIC has terminated a transaction with a target-abort when the PCIC functions as a target. A target-abort is detected as an illegal byte enable when the lower two bits (bits 1 and 0) of the address and the byte enable do not match during an I/O transfer (target). 0: Target-abort interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target-abort interrupt occurs [Set condition] When a target-abort interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
13 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
9	TMTOI	0	SH: R/WC	Target Memory Read Retry Timeout Interrupt
			PCI: R	When the PCIC functions as a target, the master did not attempt a retry within the prescribed number of PCICLK clocks (2^{15}) (detected only in the case of memory read operations). 0: Target memory read retry timeout interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target memory read retry timeout interrupt occurs [Set condition] When a target memory read retry timeout interrupt occurs.
8	MDEI	0	SH: R/WC	Master Function Disable Error Interrupt
			PCI: R	The PCIC attempted a master access when such accesses are disabled, that is, when PCICMD.BM is cleared to 0. 0: Master function disable error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master function disable error interrupt occurs [Set condition] When a master function disable error interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
7	APEDI	0	SH: R/WC	Address Parity Error Detection Interrupt
			PCI: R	Indicates an address parity error has been detected.
				When both the PER and SERRE bits in the PCI command register are set to 1, an address parity error is detected.
				0: Address parity error detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Address parity error detection interrupt occurs
				[Set condition]
				When an address parity error detection interrupt occurs.
6	SEDI	0	SH: R/WC	$\overline{\text{SERR}}$ Detection Interrupt
			PCI: R	Indicates that the assertion of the $\overline{\text{SERR}}$ signal has been detected when the PCIC operates in host bus bridge mode.
				0: $\overline{\text{SERR}}$ detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: $\overline{\text{SERR}}$ detection interrupt occurs
				[Set condition]
				When a $\overline{\text{SERR}}$ detection interrupt occurs.
5	DPEITW	0	SH: R/WC	Data Parity Error Interrupt for Target Write
			PCI: R	Indicates that a data parity error has been detected during a target write access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a target.
				0: Data parity error detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Data parity error detection interrupt occurs
				[Set condition]
				When a data parity error detection interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
4	PEDITR	0	SH: R/WC	Data Parity Error Interrupt for Target $\overline{\text{PERR}}$
			PCI: R	Indicates that the $\overline{\text{PERR}}$ signal has been received during a target read access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a target.
				0: $\overline{\text{PERR}}$ detection interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: $\overline{\text{PERR}}$ detection interrupt occurs
				[Set condition]
				When a $\overline{\text{PERR}}$ detection interrupt occurs.
3	TADIM	0	SH: R/WC	Target-Abort Detection Interrupt for Master
			PCI: R	When the PCIC functions as a master, it has detected a target-abort, that is, the transaction is terminated.
				0: Target-abort interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Target-abort interrupt occurs
				[Set condition]
				When a target-abort interrupt occurs.
2	MADIM	0	SH: R/WC	Master-Abort Interrupt for Master
			PCI: R	Indicates that the PCIC has terminated a transaction with a master-abort when the PCIC functions as a master.
				0: Master-abort interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master-abort interrupt occurs
				[Set condition]
				When a master-abort interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
1	MWPDI	0	SH: R/WC	Master Write $\overline{\text{PERR}}$ Detection Interrupt
			PCI: R	Indicates that the $\overline{\text{PERR}}$ signal has been received during a master write access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a master. 0: Master write $\overline{\text{PERR}}$ interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master write $\overline{\text{PERR}}$ interrupt occurs [Set condition] When a master write $\overline{\text{PERR}}$ interrupt occurs.
0	MRDPEI	0	SH: R/WC	Master Read Data Parity Error Interrupt
			PCI: R	Indicates that a data parity error has been detected during a master read access (only detected when PCICMD.PER is set to 1) when the PCIC functions as a master. 0: Master read data perity error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master read data perity error interrupt occurs [Set condition] When a master read data perity error interrupt occurs.

(7) PCI Interrupt Mask Register (PCIIMR)

This register is the mask register for PCIIR.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TTA DIM	—	—	—	—	TMT OIM	MDE IM	APE DIM	SE DIM	DPEI TWM	DPEI TRM	TAD IMM	MAD IMM	MW PDIM	MRD PEIM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
14	TTADIM	0	SH: R/W	Target Target-Abort Interrupt Mask
			PCI: R	0: PCIIR.TTADI disabled (masked)
				1: PCIIR.TTADI enabled (not masked)
13 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
9	TMTOIM	0	SH: R/W	Target Retry Time Out Interrupt Mask
			PCI: R	0: PCIIR.TMTOI disabled (masked)
				1: PCIIR. TMTOI enabled (not masked)
8	MDEIM	0	SH: R/W	Master Function Disable Error Interrupt Mask
			PCI: R	0: PCIIR.MDEI disabled (masked)
				1: PCIIR.MDEI enabled (not masked)
7	APEDIM	0	SH: R/W	Address Parity Error Detection Interrupt Mask
			PCI: R	0: PCIIR.APEDI disabled (masked)
				1: PCIIR.APEDI enabled (not masked)

Bit	Bit Name	Initial Value	R/W	Description
6	SEDIM	0	SH: R/W	$\overline{\text{SERR}}$ Detection Interrupt Mask
			PCI: R	0: PCIIR.SEDI disabled (masked)
				1: PCIIR.SEDI enabled (not masked)
5	DPEITWM	0	SH: R/W	Data Parity Error Interrupt Mask for Target Write
			PCI: R	0: PCIIR.DPEITW disabled (masked)
				1: PCIIR.DPEITW enabled (not masked)
4	PEDITRM	0	SH: R/W	$\overline{\text{PERR}}$ Detection Interrupt Mask for Target Read
			PCI: R	0: PCIIR.PEDITR disabled (masked)
				1: PCIIR.PEDITR enabled (not masked)
3	TADIMM	0	SH: R/W	Target-Abort Interrupt Mask for Master
			PCI: R	0: PCIIR.TADIM disabled (masked)
				1: PCIIR.TADIM enabled (not masked)
2	MADIMM	0	SH: R/W	Master-Abort Interrupt Mask for Master
			PCI: R	0: PCIIR.MADIM disabled (masked)
				1: PCIIR.MADIM enabled (not masked)
1	MWPDIM	0	SH: R/W	Master Write Data Parity Error Interrupt Mask
			PCI: R	0: PCIIR.MWPDI disabled (masked)
				1: PCIIR.MWPDI enabled (not masked)
0	MRDPEIM	0	SH: R/W	Master Read Data Parity Error Interrupt Mask
			PCI: R	0: PCIIR.MRDPEI disabled (masked)
				1: PCIIR.MRDPEI enabled (not masked)

(8) PCI Error Address Information Register (PCIAIR)

This register records PCI address information when an error is detected.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	AIL	Undefined	SH: R	Address Information Log
			PCI: R	This register holds address information (the states of the AD signals) when an error occurs.

(9) PCI Error Command Information Register (PCICIR)

This register records the PCI command information when an error is detected.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31	MTEM	Undefined	SH: R	Master Error
			PCI: R	Indicates that an error has occurred during a master access.
				0: Master error does not occur
				1: Master error occurs
30 to 27	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
26	RWTET	Undefined	SH: R	Target Error
			PCI: R	Indicates that an error has occurred during a target read or a target write access.
				0: Target error does not occur
				1: Target error occurs
25 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3 to 0	ECL	Undefined	SH: R	Command Log
			PCI: R	Hold PCI command information (the state of the CBE[3:0] signal) when an error occurs.

(10) PCI Arbiter Interrupt Register (PCIAINT)

In host bus bridge mode, this register records source of an interrupt. When multiple interrupts occur, only the first source is registered. When an interrupt is disabled, source is registered in corresponding bit (set to 1) in this register, however, no interrupt occurs.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MBI	TB TOI	MB TOI	—	—	—	—	—	—	—	TAI	MAI	RD PEI	WD PEI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R/WC	R/WC	R/WC	R	R	R	R	R	R	R	R/WC	R/WC	R/WC	R/WC
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
13	MBI	0	SH: R/WC	Master-Broken Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{PCIFRAME}}$ signal is not asserted within 16 clock cycles, although the PCIC gave a master the bus.
				0: Master-broken interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master-broken interrupt occurs
				[Set condition]
				When a master-broken interrupt occurs.

Bit	Bit Name	Initial Value	R/W	Description
12	TBTOI	0	SH: R/WC	Target Bus Time-Out Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{TRDY}}$ or $\overline{\text{STOP}}$ signal is not asserted within 16 clock cycles on the first data transfer. An interrupt is detected when the $\overline{\text{TRDY}}$ or $\overline{\text{STOP}}$ signal is not asserted within eight clock cycles during the data transfer subsequent to the 2nd. 0: Target bus time-out interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target bus time-out interrupt occurs [Set condition] When a target bus time-out interrupt occurs.
11	MBTOI	0	SH: R/WC	Master Bus Time-Out Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{IRDY}}$ signal is not asserted within 8 clock cycles. 0: Master bus time-out interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master bus time-out interrupt occurs [Set condition] When a master bus time-out interrupt occurs.
10 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	TAI	0	SH: R/WC	Target-Abort Interrupt
			PCI: R	Indicates that a transaction is terminated with a target-abort when a device other than the PCIC functions as a bus master. 0: Target-abort interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target-abort interrupt occurs [Set condition] When a target-abort interrupt occurs.
2	MAI	0	SH: R/WC	Master-Abort Interrupt
			PCI: R	Indicates that a transaction is terminated with a master-abort when a device other than the PCIC functions as a bus master. 0: Master-abort interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Master-abort interrupt occurs [Set condition] When a master-abort interrupt occurs.
1	RDPEI	0	SH: R/WC	Read Parity Error Interrupt
			PCI: R	The $\overline{\text{PERR}}$ assertion is detected during a data read when a device other than the PCIC functions as a bus master. 0: Read parity error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Read parity error interrupt occurs [Set condition] When a read parity error interrupt is detected by the $\overline{\text{PERR}}$ assertion.

Bit	Bit Name	Initial Value	R/W	Description
0	WDPEI	0	SH: R/WC	Write Parity Error Interrupt
			PCI: R	The $\overline{\text{PERR}}$ assertion is detected during a data write when a device other than the PCIC functions as a bus master. 0: Write parity error interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Write parity error interrupt occurs [Set condition] When a write parity error interrupt is detected by the $\overline{\text{PERR}}$ assertion.

(11) PCI Arbiter Interrupt Mask Register (PCIAINTM)

This register is the mask register for PCIAINT.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MBIM	TBT OIM	MBT OIM	—	—	—	—	—	—	—	TAIM	MAIM	RDP EIM	WDP EIM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13	MBIM	0	SH: R/WC	Master-Broken Interrupt Mask
			PCI: R	0: PCIAINT.MBI disabled (masked)
				1: PCIAINT.MBI enabled (not masked)
12	TBTOIM	0	SH: R/WC	Target Bus Time-Out Interrupt Mask
			PCI: R	0: PCIAINT.TBTOI disabled (masked)
				1: PCIAINT.TBTOI enabled (not masked)
11	MBTOIM	0	SH: R/WC	Master Bus Time-Out Interrupt Mask
			PCI: R	0: PCIAINT.MBTOI disabled (masked)
				1: PCIAINT.MBTOI enabled (not masked)
10 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
3	TAIM	0	SH: R/WC	Target-Abort Interrupt Mask
			PCI: R	0: PCIAINT.TAI disabled (masked)
				1: PCIAINT.TAI enabled (not masked)
2	MAIM	0	SH: R/WC	Master-Abort Interrupt Mask
			PCI: R	0: PCIAINT.MAI disabled (masked)
				1: PCIAINT.MAI enabled (not masked)
1	RDPEIM	0	SH: R/WC	Read Data Parity Error Interrupt Mask
			PCI: R	0: PCIAINT.RDPEI disabled (masked)
				1: PCIAINT.RDPEI enabled (not masked)
0	WDPEIM	0	SH: R/WC	Write Data Parity Error Interrupt Mask
			PCI: R	0: PCIAINT.WDPEI disabled (masked)
				1: PCIAINT.WDPEI enabled (not masked)

(12) PCI Arbiter Bus Master Information Register (PCIBMIR)

In host bridge mode, this register records when the interrupt is invoked by PCIAINT.

When multiple interrupts occur, only the first source is registered.

When an interrupt is masked, the source is registered in corresponding bit (set to 1), however, an interrupt occurs.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	REQ4 BME	REQ3 BME	REQ2 BME	REQ1 BME	REQ0 BME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value should always be 0.
4	REQ4BME	Undefined	SH: R	REQ4 Error
			PCI: R	An error occurs when the PCIC functions as a bus master.
3	REQ3BME	Undefined	SH: R	REQ3 Error
			PCI: R	An error occurs when device 3 ($\overline{\text{REQ3}}$) functions as a bus master
2	REQ2BME	Undefined	SH: R	REQ2 Error
			PCI: R	An error occurs when device 2 ($\overline{\text{REQ2}}$) functions as a bus master
1	REQ1BME	Undefined	SH: R	REQ1 Error
			PCI: R	An error occurs when device 1 ($\overline{\text{REQ1}}$) functions as a bus master
0	REQ0BME	Undefined	SH: R	REQ0 Error
			PCI: R	An error occurs when device 0 ($\overline{\text{REQ0}}$) functions as a bus master

(13) PCI PIO Address Register (PCIPAR)

This register is configuration address register.

Refer to Section 13.4.5 (2), Configuration Space Access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCIE	—	—	—	—	—	—	—	BN							
Initial value:	1	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DN					FN			CRA						—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31	CCIE	1	SH: R	Configuration Cycle Issue Enable
			PCI: —	Enables a configuration cycle to be issued.
				1: Indicates the configuration cycle generation enable
30 to 24	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
23 to 16	BN	Undefined	SH: R/W	PCI Bus Number
			PCI: —	Specify the PCI bus number for a configuration access. The PCIC is connected to bus number 0. Bus numbers ranging from 0 to 255 are represented in 8 bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	DN	Undefined	SH: R/W	Device Number
			PCI: —	Specify the device number for a configuration access. Device numbers ranging from 0 to 31 are represented in five bits. A single bit of bits 31 to 16 of the AD signals is driven to high level instead of the IDSEL assertion. The bit driven to high level corresponds to the device number set in these bits. The correspondence between the device number and IDSEL (AD[31:16]) is shown below. If a device number is equal to H'10 or greater, all bits 31 to 16 of the AD signals are driven to low level. Device No. IDSEL — Device No. IDSEL H'0: AD[16] = high level — H'8: AD[24] = high level H'1: AD[17] = high level — H'9: AD[25] = high level H'2: AD[18] = high level — H'A: AD[26] = high level H'3: AD[19] = high level — H'B: AD[27] = high level H'4: AD[20] = high level — H'C: AD[28] = high level H'5: AD[21] = high level — H'D: AD[29] = high level H'6: AD[22] = high level — H'E: AD[30] = high level H'7: AD[23] = high level — H'F: AD[31] = high level Other than above AD[31:16] lines are driven to high level.
10 to 8	FN	Undefined	SH: R/W	Function Number
			PCI: —	Specify the function number for a configuration access. The function numbers ranging from 0 to 7 are represented in three bits.
7 to 2	CRA	Undefined	SH: R/W	Configuration Register Address
			PCI: —	Specify the register for a configuration access at a longword boundary.
1, 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(14) PCI Power Management Interrupt Register (PCIPINT)

This register controls the power management interrupt.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PMD 3H	PMD 2	PMD 1	PMD 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/WC	R/WC	R/WC	R/WC
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
3	PMD3H	0	SH: R/WC	PCI Power Management D3 Hot Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D3 is not detected
				1: Interrupt request for a transition to D3 is detected
2	PMD2	0	SH: R/WC	PCI Power Management D2 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D2 is not detected
				1: Interrupt request for a transition to D2 is detected
1	PMD1	0	SH: R/WC	PCI Power Management D1 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D1 is not detected
				1: Interrupt request for a transition to D1 is detected
0	PMD0	0	SH: R/WC	PCI Power Management D0 Status Transition Interrupt
			PCI: —	0: Interrupt request for a transition to D0 is not detected
				1: Interrupt request for a transition to D0 is detected

(15) PCI Power Management Interrupt Mask Register (PCIPINTM)

This is the mask register for PCIPINT.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PMD 3HM	PMD 2M	PMD 1M	PMD 0M
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	SH: R PCI: —	Reserved These bits are always read as 0. The write value should always be 0.
3	PMD3HM	0	SH: R/W PCI: —	PCI Power Management D3 Hot Status Transition Interrupt Mask 0: PCIPINT.PM D3H disabled (masked) 1: PCIPINT.PM D3H enabled (not masked)
2	PMD2M	0	SH: R/W PCI: —	PCI Power Management D2 Status Transition Interrupt Mask 0: PCIPINT.PMD2 disabled (masked) 1: PCIPINT.PMD2 enabled (not masked)
1	PMD1M	0	SH: R/W PCI: —	PCI Power Management D1 Status Transition Interrupt Mask 0: PCIPINT.PMD1 disabled (masked) 1: PCIPINT.PMD1 enabled (not masked)
0	PMD0M	0	SH: R/W PCI: —	PCI Power Management D0 Status Transition Interrupt Mask 0: PCIPINT.PMD0 disabled (masked) 1: PCIPINT.PMD0 enabled (not masked)

(16) PCI Memory Bank Register 0 (PCIMBR0)

This register specifies the upper 14-bit address of the PCI memory space 0 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA0														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA0	H'0000	SH: R/W	PCI Memory Space 0 Bank Address
			PCI: —	Specify the bank address in PCI memory space 0 for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(17) PCI Memory Bank Mask Register 0 (PCIMBMR0)

This register specifies the size of the PCI memory space 0.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MSBAM0						—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
23 to 18	MSBAM0	000000	SH: R/W	PCI Memory Space 0 Bank Address Mask
			PCI: —	0000 00 : 256 Kbytes
				0000 01 : 512 Kbytes
				0000 11 : 1 Mbyte
				0001 11 : 2 Mbytes
				0011 11 : 4 Mbytes
				0111 11 : 8 Mbytes
				1111 11 : 16 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(18) PCI Memory Bank Register 1 (PCIMBR1)

This register specifies the upper 14-bit address of the PCI memory space 1 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA1														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA1	All 0	SH: R/W	PCI Memory Space 1 Bank Address
			PCI: —	Specify the bank address in PCI memory space 1 for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(19) PCI Memory Bank Mask Register 1 (PCIMBMR1)

This register specifies the size of the PCI memory space 1.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MSBAM1								—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
25 to 18	MSBAM1	All 0	SH: R/W	PCI Memory Space 1 Bank Address Mask (8 bits)
			PCI: —	00 0000 00: 256 Kbytes
				00 0000 01: 512 Kbytes
				00 0000 11: 1 Mbyte
				00 0001 11: 2 Mbytes
				00 0011 11: 4 Mbytes
				00 0111 11: 8 Mbytes
				00 1111 11: 16 Mbytes
				01 1111 11: 32 Mbytes
				11 1111 11: 64 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(20) PCI Memory Bank Register 2 (PCIMBR2)

This register specifies the upper 14-bit address of the PCI memory space 2 (address bits 31 to 18).

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSBA2														—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA2	All 0	SH: R/W	PCI Memory Space 2 Bank Address
			PCI: —	Specify the bank address in PCI memory space 2 for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(21) PCI Memory Bank Mask Register 2 (PCIMBMR2)

This register specifies the size of the PCI memory space 2.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	MSBAM2											—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
28 to 18	MSBAM2	All 0	SH: R/W	PCI Memory Space 2 Bank Address Mask
			PCI: —	0 0000 0000 00: 256 Kbytes
				0 0000 0000 01: 512 Kbytes
				0 0000 0000 11: 1 Mbyte
				0 0000 0001 11: 2 Mbytes
				0 0000 0011 11: 4 Mbytes
				0 0000 0111 11: 8 Mbytes
				0 0000 1111 11: 16 Mbytes
				0 0001 1111 11: 32 Mbytes
				0 0011 1111 11: 64 Mbytes
				0 0111 1111 11: 128 Mbytes
				0 1111 1111 11: 256 Mbytes
				1 1111 1111 11: 512 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(22) PCI I/O Bank Register (PCIIOBR)

This register specifies the upper 14-bit address of the PCI I/O space (address bits 31 to 18).

Refer to Section 13.4.3 (3), Accessing PCI I/O Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PIOSBA	All 0	SH: R/W	PCI I/O Space Bank Address (14 bits)
			PCI: —	Specify the bank address in PCI I/O space for a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(23) PCI I/O Bank Mask Register (PCIIOBMR)

This register specifies the size of the PCI I/O space.

Refer to Section 13.4.3 (2), Accessing PCI Memory Space.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
20 to 18	IOBAMR	All 0	SH: R/W	PCI I/O Space Bank Address Mask (3 bits)
			PCI: —	000: 256 Kbytes
				001: 512 Kbytes
				011: 1 Mbyte
				111: 2 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.

(24) PCI Cache Snoop Control Register 0 (PCICSCR0)

An external device can access local memory of this LSI via the PCIC. When an external PCI device accesses cacheable areas of this LSI, the PCIC can support cache snoop function to the on-chip caches. The PCICSCR0 specifies this function that uses cache snoop address registers 0.

Refer to section 13.4.4 (7), Cache Coherency.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
4 to 2	RANGE	All 0	SH: R/W	Address Range to be Compared
			PCI: —	Specify the address range of PCICSAR0 to be compared. 000: PCICSAR[n].CADR[31:12] compared (4 Kbytes) 001: PCICSAR[n].CADR[31:16] compared (64 Kbytes) 010: PCICSAR[n].CADR[31:20] compared (1 Mbyte) 011: PCICSAR[n].CADR[31:24] compared (16 Mbytes) 100: PCICSAR[n].CADR[31:25] compared (32 Mbytes) 101: PCICSAR[n].CADR[31:26] compared (64 Mbytes) 110: PCICSAR[n].CADR[31:27] compared (128 Mbytes) 111: PCICSAR[n].CADR[31:28] compared (256 Mbytes) Valid only when PCICSCR0.SNPMD = 10 or 11.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SNPMD	All 0	SH: R/W	Snoop Mode for PCICSAR0
			PCI: —	Specify if PCICSAR0 is compared with address requested by an external device. Also, specify how snoop function is executed when PCICSAR0 is compared.
				00: PCICSAR0 not compared
				01: Reserved (setting prohibited)
				10: PCICSAR0 compared. If hit, snoop function is not executed, otherwise executed.
				11: PCICSAR0 compared. If hit, snoop function is executed, otherwise not executed.

(25) PCI Cache Snoop Control Register 1 (PCICSCR1)

An external device can access local memory of this LSI via the PCIC. When an external PCI device accesses cacheable areas of this LSI, the PCIC can support cache snoop function to the on-chip caches. The PCICSCR1 specifies this function that uses cache snoop address registers 1.

Refer to section 13.4.4 (7), Cache Coherency.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write value should always be 0.
4 to 2	RANGE	All 0	SH: R/W	Address Range to be Compared
			PCI: —	Specify the address range of PCICSAR1 to be compared. 000: PCICSAR[n].CADR[31:12] compared (4 Kbytes) 001: PCICSAR[n].CADR[31:16] compared (64 Kbytes) 010: PCICSAR[n].CADR[31:20] compared (1 Mbyte) 011: PCICSAR[n].CADR[31:24] compared (16 Mbytes) 100: PCICSAR[n].CADR[31:25] compared (32 Mbytes) 101: PCICSAR[n].CADR[31:26] compared (64 Mbytes) 110: PCICSAR[n].CADR[31:27] compared (128 Mbytes) 111: PCICSAR[n].CADR[31:28] compared (256 Mbytes) Valid only when PCICSCR1.SNPMD = 10 or 11.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SNPMD	All 0	SH: R/W	Snoop Mode for PCICSAR1
			PCI: —	Specify if PCICSAR1 is compared with address requested by an external device. Also, specify how snoop function is executed when PCICSAR1 is compared.
				00: PCICSAR1 not compared
				01: Reserved (setting prohibited)
				10: PCICSAR1 compared. If hit, snoop function is not executed, otherwise executed.
				11: PCICSAR1 compared. If hit, snoop function is executed, otherwise not executed.

(26) PCI Cache Snoop Address Register 0 (PCICSAR0)

PCICSAR0 specifies the address to be compared with the PCI address requested by an external device.

Refer to section 13.4.4 (7), Cache Coherency.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CADR	All 0	SH: R/W	Address to be compared
			PCI: —	Specify address to be compared with the PCI address requested by external PCI devices

(27) PCI Cache Snoop Address Register 1 (PCICSAR1)

PCICSAR1 specifies the address to be compared with the PCI address requested by an external device.

Refer to section 13.4.4 (7), Cache Coherency.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CADR	All 0	SH: R/W	Address to be compared
			PCI: —	Specify address to be compared with the PCI address requested by external PCI devices

(28) PCI PIO Data Register (PCIPDR)

When accessed, this register will cause the generation of a configuration cycle on the PCI bus.

Refer to section 13.4.5 (2), Configuration Space Access.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	PDR	Undefined	SH: R/W	PCI PIO Data Register
			PCI: —	A read from or write to this register will cause a PCI configuration cycle on the PCI bus.

13.4 Operation

13.4.1 Supported PCI Commands

Table 13.4 Supported Bus Commands

CBE[3:0]	Command Type	PCI Master	PCI Target
0000	Interrupt acknowledge cycle	No	—
0001	Special cycle	Yes*[1]	—
0010	I/O read	Yes	Yes*[2]
0011	I/O write	Yes	Yes*[2]
0100	Reserved	—	—
0101	Reserved	—	—
0110	Memory read	Yes	Yes
0111	Memory write	Yes	Yes
1000	Reserved	—	—
1001	Reserved	—	—
1010	Configuration read	Yes*[1]	Yes*[2]
1011	Configuration write	Yes*[1]	Yes*[2]
1100	Memory read multiple	No	Partially yes*[3]
1101	Dual address cycle	No	No
1110	Memory read line	No	Partially yes*[3]
1111	Memory write and invalidate	No	Partially yes*[4]

[Legend]

0: Low level

1: High level

Notes: 1. Only the host bus bridge mode is supported.

2. Single transfer only is performed.

3. Operation is the same as that for the memory read command.

4. Operation is the same as that for the memory write command.

13.4.2 PCIC Initialization

After a power-on reset, the PCIC enable bit (ENBL) of the PCIC enable control register (PCIECR) and the internal register initialization bit (CFINIT) of the PCI control register (PCICR) is cleared. At this point, if the PCIC is operating as the PCI bus host (host bus bridge mode), the bus privileges are permanently granted to the PCIC, and no device arbitration is performed on the PCI bus. When the PCIC is not operating as host (normal mode), retries are returned without accepting access from PCI external devices connected to the PCI bus. In addition, all accesses to the PCIC from the CPU are invalid except the access to the PCIECR if the PCIECR.ENBL is cleared to 0. A write access is invalid and a read access will read 0, none of the registers can be modified, and any access to the PCI bus will not be executed.

To initialize the PCIC, first setting the enable bit in the PCIECR to 1. The PCIC's internal configuration registers and local registers must be initialized before setting the CFINIT bit in the PCICR to 1 (while the CFINIT bit is cleared to 0). On completion of initialization, set the CFINIT bit to 1. When operating as host, arbitration is enabled; when operating as non-host, the PCIC can be accessed from the PCI bus.

Regardless of whether the PCIC is operating as the host or normal, external PCI devices cannot be accessed from the PCIC while the CFINIT bit is being cleared. Set the CFINIT bit to 1 before accessing an external PCIC device.

Be sure to initialize the following registers while the CFINIT bit is being cleared (before setting to 1): PCI command (PCICMD), PCI status (PCISTATUS), PCI sub system vender ID (PCISVID), PCI subsystem ID (PCISID), PCI local space register 0/1 (PCILSR 0/1) and PCI local address register 0/1.

13.4.3 Master Access

This section describes how the PCIC is accessed by software in this LSI and the restrictions on usage, such as buffering and synchronization with other devices, when the PCIC is used in both the host bus bridge and normal modes.

(1) Address Space of PCIC

Table 13.5 shows the PCIC address map.

Table 13.5 PCIC Address Map

	Physical Address		
Memory Area	**29-Bit Address Mode**	**32-Bit Address Extended Mode***	**Space Size**
PCI memory space1 (Area 4)	H'1000 0000 to H'13FF FFFF	H'1000 0000 to H'13FF FFFF	64 Mbytes
PCI memory space 2 (Only 32-bit address extended mode)	—	H'C000 0000 to H'DFFF FFFF	512 Mbytes
PCI memory space 0	H'FD00 0000 to H'FDFF FFFF	H'FD00 0000 to H'FDFF FFFF	16 Mbytes
Control register	H'FE00 0000 to H'FE03 FFFF	H'FE00 0000 to H'FE03 FFFF	256 Kbytes
PCIC internal register (configuration and local registers)	H'FE04 0000 to H'FE07 FFFF	H'FE04 0000 to H'FE07 FFFF	256 Kbytes
Reserved	H'FE08 0000 to H'FE1F FFFF	H'FE08 0000 to H'FE1F FFFF	1.5 Mbytes
PCI I/O space	H'FE20 0000 to H'FE3F FFFF	H'FE20 0000 to H'FE3F FFFF	2 Mbytes

Note: * For details, see section 7.8, Notes on Using 32-Bit Address Extended Mode.

The address space of the PCIC is divided into four main spaces (six spaces, altogether): the control register space (PCIECR), PCI internal control register (PCI configuration and PCI local registers) space, I/O space, and PCI memory (PCI memory space 0, PCI memory space 1, and PCI memory space 2).

(2) Accessing PCI Memory Space

Figure 13.2 shows the method for accessing the PCI bus allocated to the PCI memory space from the SuperHyway bus.

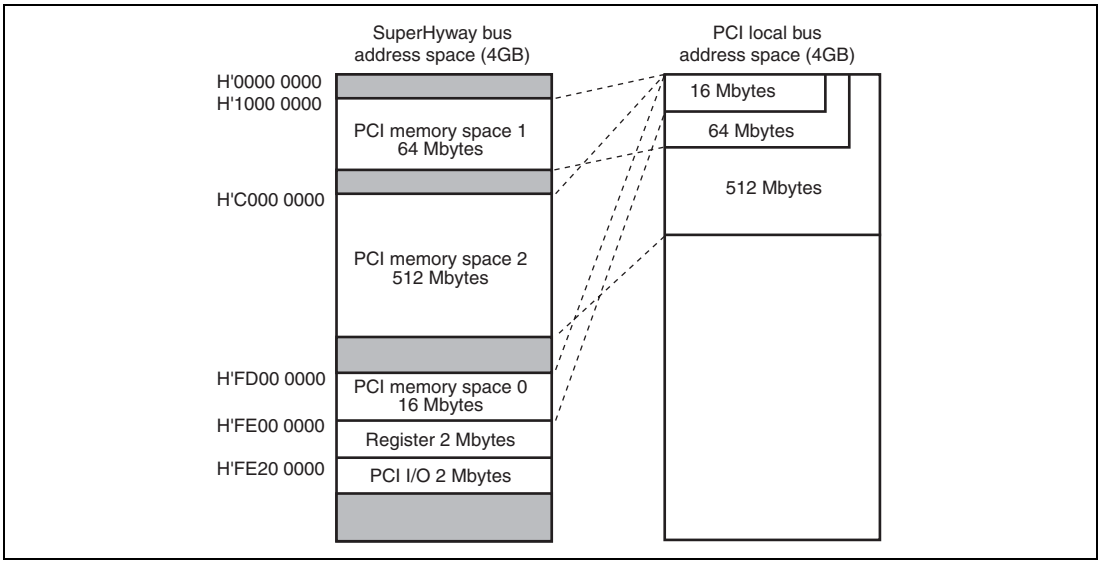


Figure 13.2 SuperHyway Bus to PCI Local Bus Access

To access to the PCI memory address space, use the PCI memory bank register (PCIMBR) and PCI memory bank mask register (PCIMBMR). These registers should have an address space ranging from 16 Mbytes to 512 Mbytes. PCI addresses can be allocated to by software.

The PCIC supports burst transfers to memory transfer.

Consecutive accesses with the SuperHyway load 32-byte or SuperHyway store 32-byte command result in a burst transfer of 32-byte or more (64-byte, 96-byte, etc.).

The PCI memory spaces are allocated from H'FD00 0000 to H'FDFF FFFF for PCI memory space 0 (16 Mbytes), H'1000 0000 to H'13FF FFFF for PCI memory space 1 (Area 4, 64 Mbytes, selection of the PCIC, DDRIF and LBSC spaces), and H'C000 0000 to H'DFFF FFFF for PCI memory space 2 (512 Mbytes, available only in 32-bit address extended mode).

Address translation from SuperHyway bus to PCI local bus

The lower 15 bits ([17:3]) of a SuperHyway bus address are sent without translation.

For PCI memory space 0 accesses, bits 23 to 18 of a SuperHyway bus address are controlled by PCI memory bank mask register 0 (PCIMBMR0).

Note: In the following items and figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus.

- PCIMBMR0 [23:18] B'1111 11: PCI address [23:18] = SH address [23:18]
- PCIMBMR0 [23:18] B'0111 11: PCI address [23:18] = PCIMBR0 [23], SH address [22:18]

≀

- PCIMBMR0 [23:18] B'0000 01: PCI address [23:18] = PCIMBR0 [23:19], SH address [18]
- PCIMBMR0 [23:18] B'0000 00: PCI address [23:18] = PCIMBR0 [23:18]

The upper eight bits ([31:24]) of a SuperHyway bus address are replaced with bits 31 to 24 in PCI memory bank register 0 (PCIMBR0).

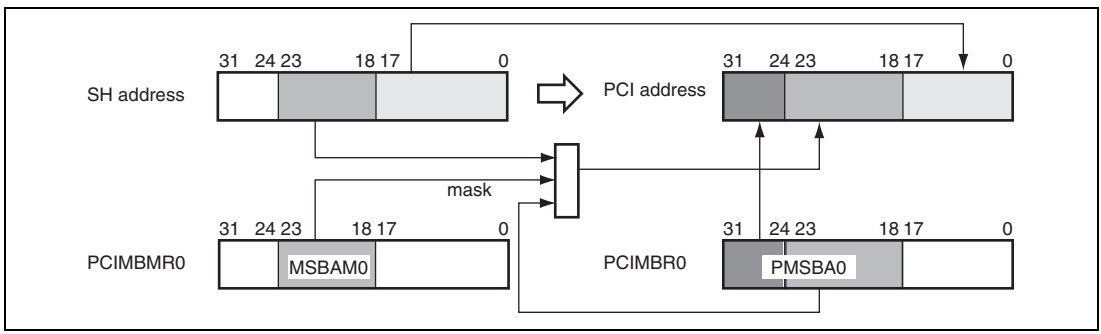


Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 0)

For PCI memory space 1 accesses, bits 25 to 18 of a SuperHyway address are controlled by PCI memory bank mask register 1 (PCIMBMR1).

- PCIMBMR1 [25:18] B'11 1111 11: PCI address [25:18] = SH address [25:18]
- PCIMBMR1 [25:18] B'01 1111 11: PCI address [25:18] = PCIMBR1 [25], SH address [24:18]

≀

- PCIMBMR1 [25:18] B'00 0000 01: PCI address [25:18] = PCIMBR1 [25:19], SH address [18]
- PCIMBMR1 [25:18] B'00 0000 00: PCI address [25:18] = PCIMBR1 [25:18]

The upper six bits ([31:26]) of a SuperHyway bus address are replaced with bits 31 to 26 in PCI memory bank register 1 (PCIMBR1).

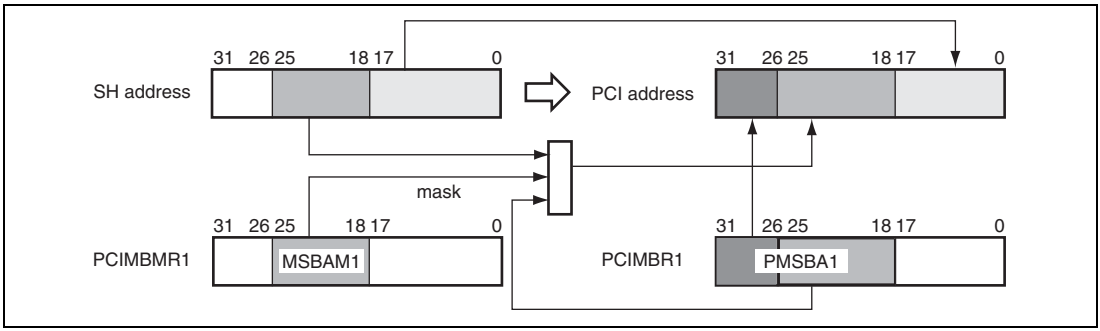


Figure 13.4 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 1)

For PCI memory space 2 accesses, bits 28 to 18 of a SuperHyway address are controlled by the PCI memory bank mask register 2 (PCIMBMR2).

- PCIMBMR2 [28:18] B'1 1111 1111 11: PCI address [28:18] = SH address [28:18]
- PCIMBMR2 [28:18] B'0 1111 1111 11: PCI address [28:18] = PCIMBR2 [28], SH address [27:18]

≀

- PCIMBMR2 [28:18] B'0 0000 0000 01: PCI address [28:18] = PCIMBR2 [28:19], SH address [18]
- PCIMBMR2 [28:18] B'0 0000 0000 00: PCI address [28:18] = PCIMBR2[28:18]

The upper three bits ([31:29]) of a SuperHyway bus address are replaced with bits 31 to 29 in PCI memory bank register 2 (PCIMBR2).

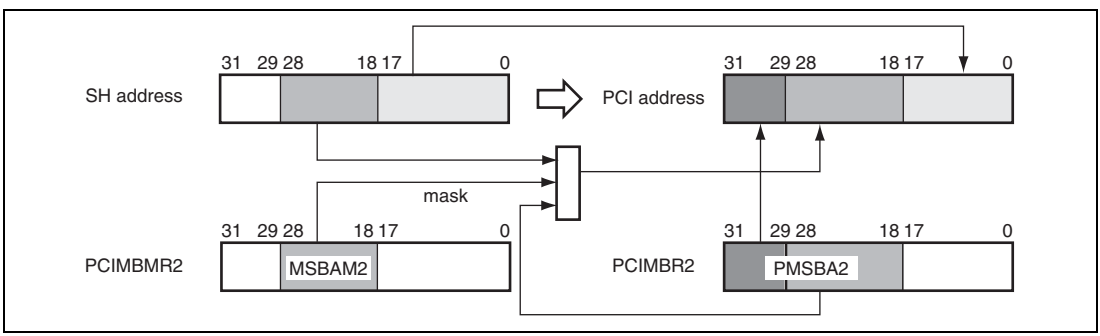


Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 2)

(3) Accessing PCI I/O Space

Access within the size of 4-byte.

Burst I/O transfers are not supported.

The PCI I/O address space is allocated from H'FD20 0000 to H'FE3F FFFF (2 Mbytes).

Address translation from SuperHyway bus to PCI local bus

The lower 15 bits ([17:3]) of a SuperHyway bus address are sent without translation.

Bits 20 to 18 of a SuperHyway bus address are controlled by the PCI I/O bank mask register (PCIIOBMR).

Note: In the following item and figure, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus.

- PCIIOMR0 [20:18] B'111: PCI address [20:18] = SH address [20:18]
- PCIIOMR0 [20:18] B'011: PCI address [20:18] = PCIIOBR [20], SH address [19:18]
- PCIIOMR0 [20:18] B'001: PCI address [20:18] = PCIIOBR [20:19], SH address [18]
- PCIIOMR0 [20:18] B'000: PCI address [20:18] = PCIIOBR [20:18]

The upper 11 bits ([31:21]) of a SuperHyway bus address are replaced with bits 31 to 21 in the PCI I/O bank register (PCIIOBR).

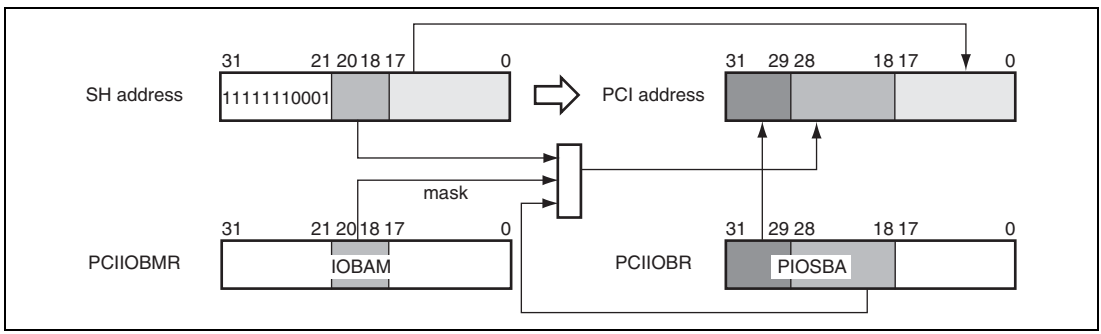


Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/O)

(4) Accessing Internal Registers of this LSI

All internal registers, that is, PCIECR, PCI configuration registers, and PCI local registers are accessible from the CPU.

4-byte, 2-byte, and byte transmission are supported.

(5) Endian

The PCIC of this LSI supports both the big endian and little endian formats. Since PCI local bus is inherently little endian, the PCIC supports both byte swapping and non-byte swapping.

The endian format is specified by the setting of the TBS bit in the PCI control register (PCICR) at a reset.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus. “MSByte” means the most significant byte and “LSByte” means the least significant byte.

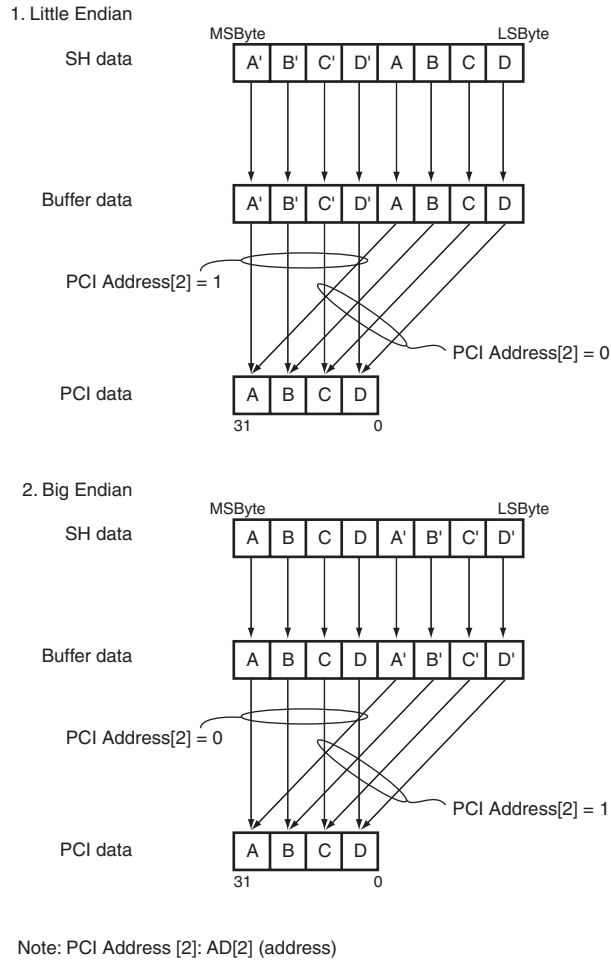


Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus (Non-Byte Swapping: TBS = 0)

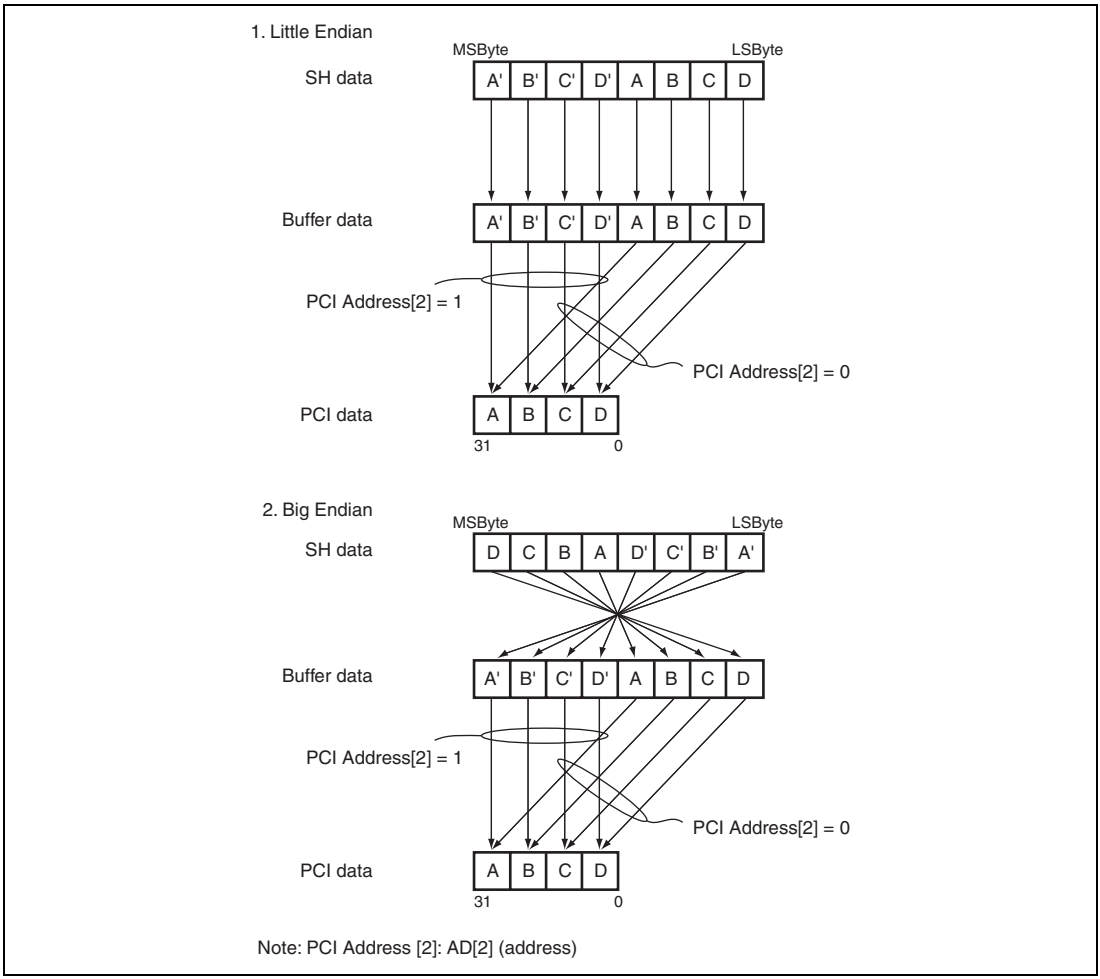


Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus (Byte Swapping: TBS = 1)

13.4.4 Target Access

This section describes how the PCIC of this LSI is accessed by an external PCI local bus master when the PCIC is used in both the host bus bridge and normal modes.

(1) Accessing This LSI Address Space

Accesses to the address space of this LSI by an external PCI bus master are described here.

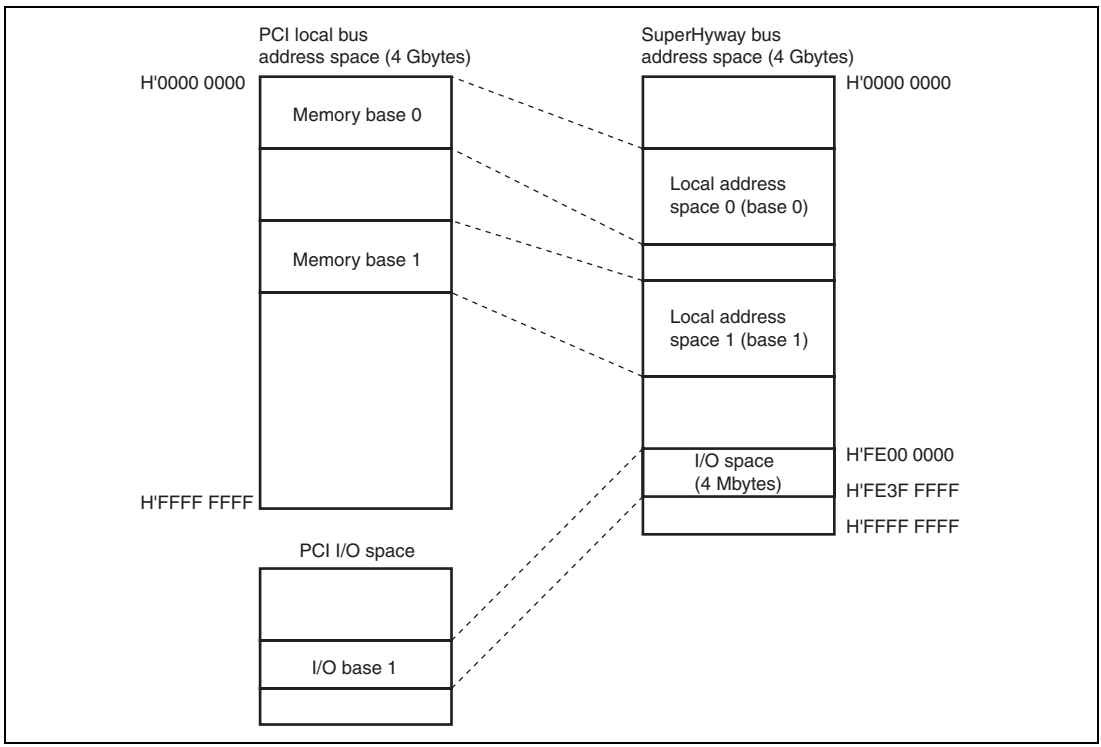


Figure 13.9 PCI local bus to SuperHyway bus Memory Map

To access the address space of this LSI, use the PCI memory base address register (PCIMBAR0/1), PCI local space register (PCILSR0/1), and PCI local address register (PCILAR0/1). The address spaces are mapped by software. The PCIC includes two memory mapping registers.

Setting these two registers enables the use of two spaces.

The size of these address spaces are selectable from 1 Mbyte to 512 Mbytes by setting the PCI local space register (PCILSR0/1).

Single longword and burst transfers are supported for the memory data transfer to a PCI target.

A certain range of the address space on the PCI local bus corresponds to the local address space on the SuperHyway bus. The local address space 0 is controlled by the PCIMBAR0, PCILSR0 and PCILAR0. And the local address space 1 is controlled by the PCIMBAR1, PCILSR1 and PCILAR1. Figure 13.10 shows the method of accessing the local address space.

The PCIMBAR0/1 indicates the starting address of the memory space used by the PCI device. The PCILAR0/1 specifies the starting address of the local address space 0/1. The PCILSR0/1 expresses the size of the memory used by the PCI device.

Address translation from PCI local bus to SuperHyway bus

For the PCIMBAR0/1 and PCILAR0/1, the more significant address bits that are higher than the memory size set in the PCILSR0/1 becomes valid. The more significant address bits of the PCIMBAR0/1 and the same field line bits of the PCI local bus address output from an external PCI device are compared for the purpose of determining whether the access is made to the PCIC. When the addresses correspond, the access to the PCIC is recognized, and a local address is generated from the more significant address bits of the PCILAR0/1 and the less significant bits of the PCI local bus address output from the external PCI device. The PCI command is executed for this local address.

If the more significant address bits of the PCI local bus address output from the external PCI device does not correspond with the more significant address bits of the PCIMBAR0/1, the PCIC does not respond to the PCI command.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI” means the PCI local bus.

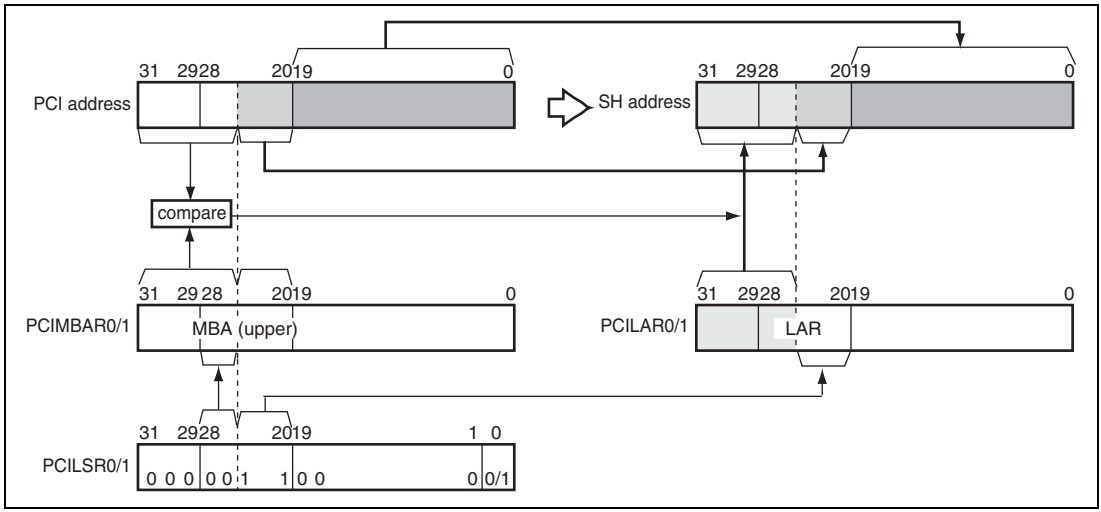


Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation (Local Address Space 0/1)

When all the MBARE bits in PCILSR0/1 are 0, the PCI local bus address is sent to the SuperHyway bus without translation.

Data prefetching for memory read commands is supported. When a PCI burst read is performed, 8 bytes, or 32 bytes of data block is prefetched. (this depends on the settings of the PFE and PFCS bits in PCICR).

(2) Accessing PCIC I/O Space

Allocate a 256-byte area to the I/O address space.

Address translation from PCI local bus to SuperHyway bus

The lower 8 bits ([7:0]) are sent to the SuperHyway bus without translation.

When bits 31 to 8 of a PCI local bus address match bits 31 to 8 in a PCI I/O base address register (PCIIBAR), the upper 24 bits of a PCI local bus address are replaced with H'FE04 01.

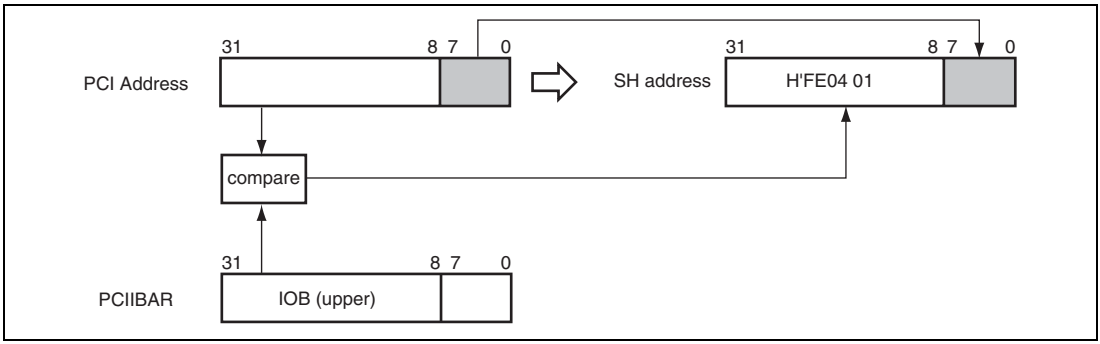


Figure 13.11 PCI Local Bus to SuperHyway Bus Address Translation (PCIC I/O Space)

(3) Accessing PCIC Registers

Configuration Registers: Access the configuration registers using an offset from the PCI configuration register space base address with the configuration read or write command. Only a single access which size should be under longword is performed. If a burst transfer is attempted, it is terminated to end the transaction.

Local Registers: Access the local registers using an offset from a PCI local register space base address with the I/O read or I/O write command. Only a single longword access is performed. If a burst transfer is attempted, it is terminated to end the transaction.

Control Register (PCIECR): Do not read or write access to the PCIECR from the PCI local bus.

(4) Access to this LSI Address Space

Memory Space: Refer to Section 13.4.4 (1), Accessing This LSI Address Space. Area 0 to area 2 and area 4 to area 6 and DDR-SDRAM space on this LSI address space can be accessed.

On-chip IO Space: Do not read or write access to the on-chip IO space using memory read or memory write command via PCI local bus. The operation of this read/write is not guaranteed.

(5) Exclusive Access

The lock access on the PCI bus is supported.

When the PCI local bus is locked, the PCIC is accessible from the device that activates the LOCK signal.

SuperHyway bus resource lock does not occur. (Another on-chip module can access the PCIC during a lock transfer.)

(6) Endian

This LSI supports both the big and little endian formats. Since the PCI local bus is inherently little endian, the PCIC supports both byte swapping and non-byte swapping.

The endian format is specified by the setting of the TBS bit in the PCI control register (PCICR).

Note: In the following figures, “MSByte” means the most significant byte and “LSByte” means the least significant byte.

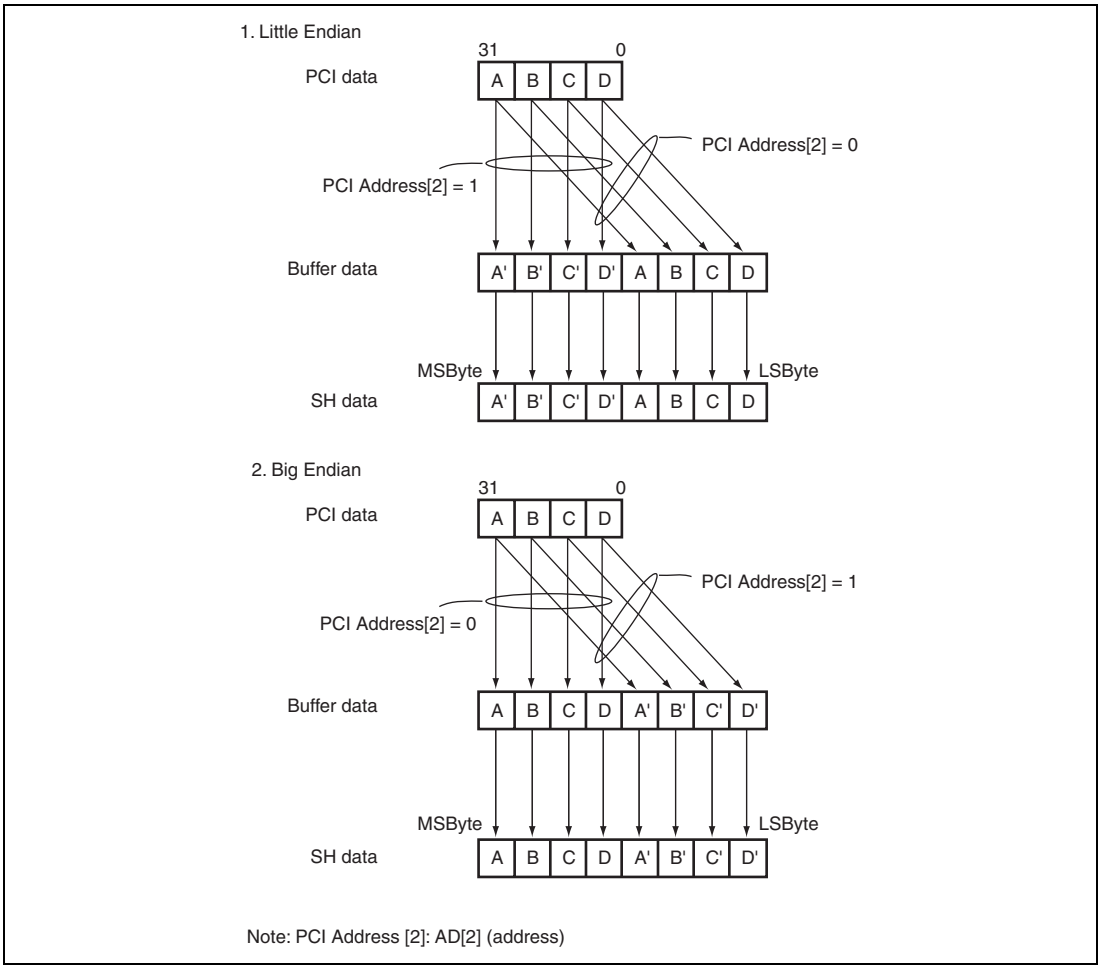


Figure 13.12 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 0)

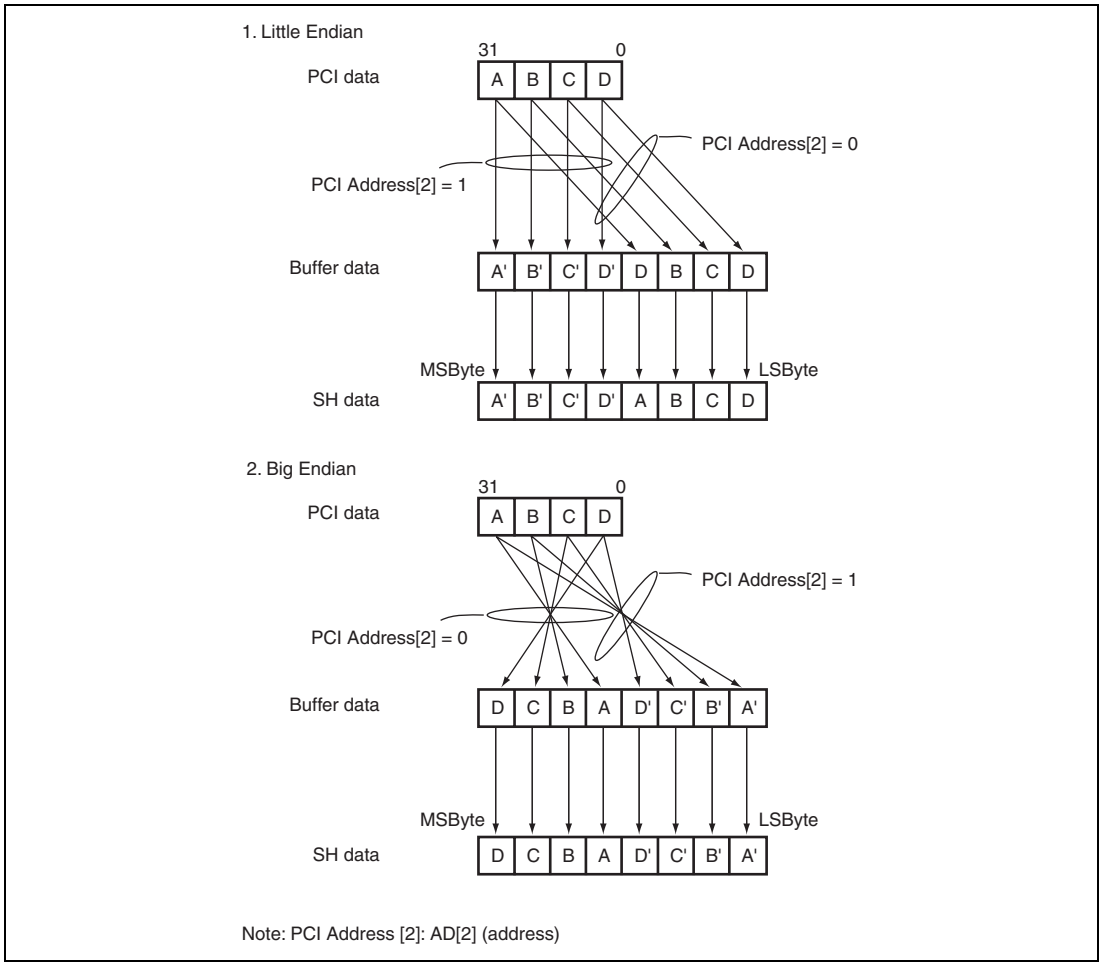


Figure 13.13 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 1)

(7) Cache Coherency

The PCIC supports cache snoop function.

When the PCIC functions as a target device, cache coherency is guaranteed for accesses from a master device connected to a PCI bus in both the host bus bridge mode and normal mode.

When accessing this LSI cacheable area, set the cache snoop registers: the PCI cache snoop control registers (PCICSCR0 and PCICSCR1) and PCI cache snoop address register (PCICSAR0 and PCICSAR1).

Usage Notes

- Up to 2 conditions can be set as snoop address. Address comparison is logical OR of setting 2 conditions.
- When using this function, execute memory read or write after flush/purge request issued to the CPU cache in the access of cache hit. It reduces PCI bus transfer speed and CPU performance.
- When using this function, do not use the prefetch function.
  (Do not set PFE bit in the PCICR to 1.)
- Do not use this function when the CPU is sleep state. If cache hit occurs in sleep state, it becomes an error access on the SuperHyway bus, and memory read or memory write does not execute. Specify the SNPMD bit in the PCICSCR to 00 before the CPU enters sleep mode. To keep the coherency before and after the CPU sleep, cache purge should be executed before sleep instruction executed.
- Do not use ether of the following functions and the cache shoop function simultaneously.
  — Debug function using an emulator (Disable this function when using an emulator).
  — L memory or memory mapped cache access from the DMAC.

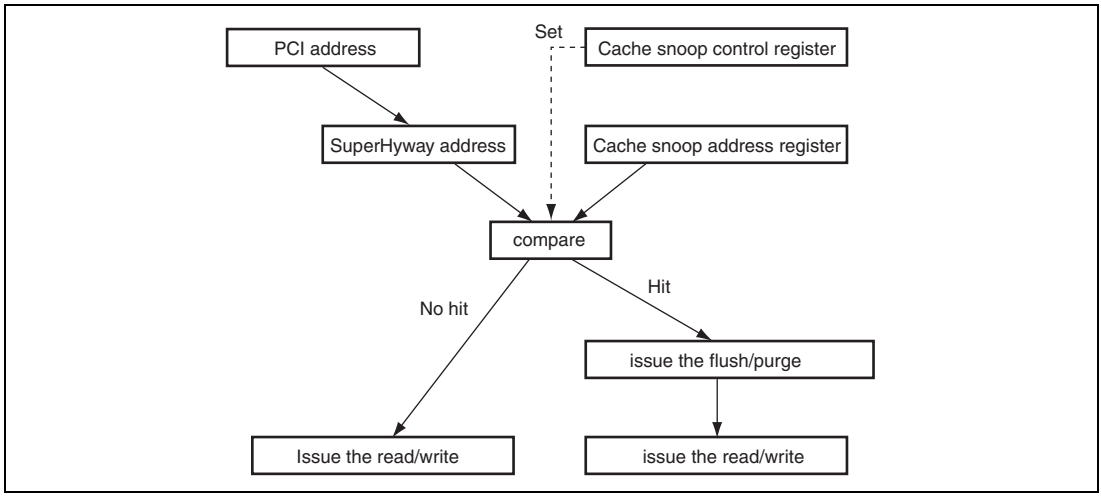


Figure 13.14 Cache Flush/Purge Execution Flow for PCI local Bus to SuperHyway Bus

13.4.5 Host Bus Bridge Mode

(1) PCI Host bus bridge Mode Operation

The PCIC supports a subset of the PCI Local Bus Specification Revision 2.2 and can be connected to a device with a PCI bus interface.

While the PCIC is set in host bus bridge mode, or while set in normal mode, operation differs according to whether or not bus parking is performed, and whether or not the PCI bus arbiter function is enabled or not.

In host bus bridge mode, the AD, CBE, PAR signal lines are driven by the PCIC when transfers are not being performed on the PCI bus. When the PCIC subsequently starts transfers as master, these signal lines continue to be driven until the end of the address phase.

The arbiter in the PCIC and the REQ and GNT between PCIC are connected internally. Here, pins $\overline{\text{REQ0}}$/REQOUT, $\overline{\text{REQ1}}$, $\overline{\text{REQ2}}$, and $\overline{\text{REQ3}}$ function as the REQ inputs from the external masters 0 to 3. Similarly, $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$, $\overline{\text{GNT1}}$, $\overline{\text{GNT2}}$, and $\overline{\text{GNT3}}$ function as the GNT outputs to external masters 0 to 3. Including the PCIC, arbitration of up to five masters is possible.

(2) Configuration Space Access

The PCIC supports configuration mechanism #1. The PCI PIO address register (PCIPAR) and PCI PIO data register (PCIPDR) correspond to the configuration address register and configuration data register, respectively.

When PCIPDR is read from or written to after PCIPAR has been set, a configuration cycle is issued on a PCI bus.

For a type 0 transfer, bits 10 to 2 of the configuration address register are sent without translation and bits 31 to 11 are translated so that these bits can be used as the IDSEL signal.

Bit 16 of the AD signal is driven to 1 and the other bits are made 0 by setting the device number to 0.

Bit 17 of the AD signal is driven to 1 and the other bits are made 0 by setting the device number to 1. Similarly, setting the device number to 2 drives bit 18 of the AD signal to 1 and setting the device number to 3 drives bit 19 of the AD signal to 0.

Bit 31 of the AD signal is driven to 1 and the other bits are made 0 by setting the device number to 16.

For details, refer to "PCI Local Bus Specification Revision 2.2, section 3.2.2.3 Configuration Space Decoding".

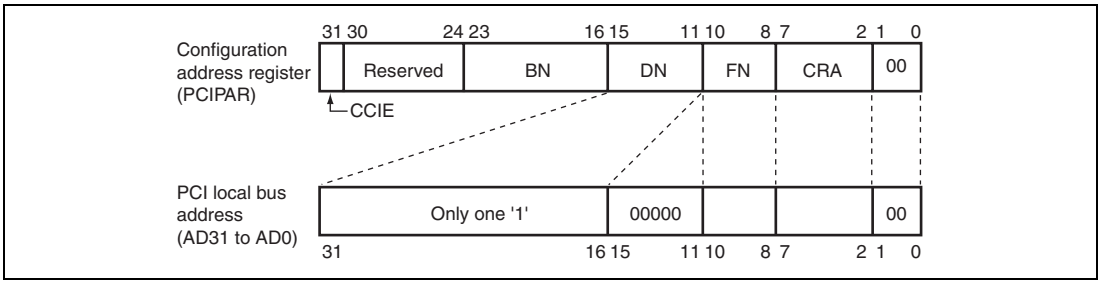


Figure 13.15 Address Generation for Type 0 Configuration Access

In configuration accesses, a PCI master abort (no device connected) will not cause an interrupt.

Configuration writes will end normally. Configuration reads will return a value of 0.

(3) Special Cycle Generation

When the PCIC operates as the host device, a special cycle is generated by setting H'8000 FF00 in the PCIPAR and writing to the PCIPDR.

(4) Arbitration

In host bus bridge mode, the PCI bus arbiter in the PCIC is activated.

The PCIC supports four external masters (i.e., four REQ and GNT pairs).

If use of the bus is simultaneously requested by more than one device, the bus is granted to the device with the highest priority.

The PCI bus arbiter supports two modes to determine the priority of devices: fixed priority and pseudo-round-robin. The mode is selected by the BMAM bit in PCICR.

Fixed Priority: When the BMAM bit in PCICR is cleared to 0, the priorities of devices are fixed the following default values.

PCIC > device 0 > device 1 > device 2 > device 3

The PCIC always gains use of the bus over other devices.

Pseudo-Round-Robin: When the BMAM bit in PCICR is set to 1, the most recently granted device is assigned the lowest priority.

The initial priority is the same as the fixed priority mode.

After device 1 has claimed and granted the bus, and transferred data, the priority is as follows:

PCIC > device 0 > device 2 > device 3 > device 1

Then, after the PCIC has claimed and granted the bus, and transferred data, the priority is changed to:

Device 0 > device 2 > device 3 > device 1 > PCIC

After device 3 has claimed and granted the bus, and transferred data, the priority is changed to:

Device 0 > device 2 > device 1 > PCIC > device 3

In host bus bridge mode, bus parking is always controlled by the PCIC.

(5) Interrupts

- 10 interrupts are available (these signals are connected to the INTC of this LSI)
- Interrupts are enabled/disabled and their priority levels are specified by the INTC of this LSI
- When the PCIC operates normal mode, $\overline{\text{INTA}}$ output is available to the host device on the PCI bus. The $\overline{\text{INTA}}$ pin is specified assert or negate by the IOCS bit in the PCICR.

Table 13.6 Interrupt Priority

Signal	Interrupt Source	Priority
PCISERR	SERR assertion detected in host bus bridge mode	High
PCIINTA	PCI interrupt A ($\overline{\text{INTA}}$) detected in host bus bridge mode	
PCIINTB	PCI interrupt B ($\overline{\text{INTB}}$) detected in host bus bridge mode	
PCIINTC	PCI interrupt C ($\overline{\text{INTC}}$) detected in host bus bridge mode	
PCIINTD	PCI interrupt D ($\overline{\text{INTD}}$) detected in host bus bridge mode	
PCIEER	Error on PCI bus occurs and reflected in PCIIR and PCIAINT. The interrupt can be masked.	
PCIPWD3	Power state transition to D3 caused by PCIPINT. The interrupt can be masked.	
PCIPWD2	Power state transition to D2 caused by PCIPINT. The interrupt can be masked.	
PCIPWD1	Power state transition to D1 caused by PCIPINT. The interrupt can be masked.	
PCIPWD0	Power state transition to D0 caused by PCIPINT. The interrupt can be masked.	Low

The PCIC can store the error information on the PCI bus. If an error occurs, the error address is stored in the PCI error address information register (PCIAIR), the types of transfer and command information are stored in the PCI error command information register. And then if the PCIC operates host bus bridge mode, the bus master information is stored in the PCI error bus master information register.

Error information is stored only one information. This causes only to store the first occurred error information, and not to store after second error information. The error information is initialized by a power-on reset.

13.4.6 Normal mode

When operating in normal mode, the PCI bus arbitration function in the PCIC is disabled and PCI bus arbitration is performed according to the specifications of the externally connected PCI bus arbiter.

In normal mode, the master performs bus parking is decided by the grant signal that asserted from the external bus arbiter. If the master that performing bus parking is different from the next transaction master, the bus will be high-impedance state for minimum one clock cycle before the address phase.

In normal mode, the $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$ pin is used for the grant input signal to the PCIC, and the $\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$ pin is used for the request output signal from the PCIC.

13.4.7 Power Management

The PCIC supports PCI power management revision 1.1. Supported features are shown below.

- Support for the PCI power management control configuration register.
- Support for the power-down/restore request interrupts from hosts on the PCI bus.

There are seven configuration registers for PCI power management control. PCI capabilities pointer register shows the address offset of the configuration registers for power management. In the PCIC, this offset is fixed at CP = H'40. PCI capability ID (PCICID), next item pointer (PCINIP), power management capability (PCIPMC), power management control/status (PCIPMCSR), PMCSR bridge support extension (PCIPMCSRBSE) and power consumption/dissipation (PCIPCDD) are power management registers. They support four states: power state D0 (normal) power state D1 (bus idle) power state D2 (clock stop) and power state D3 (power down mode).

Figure 13.16 shows the PCI local bus power down state transition.

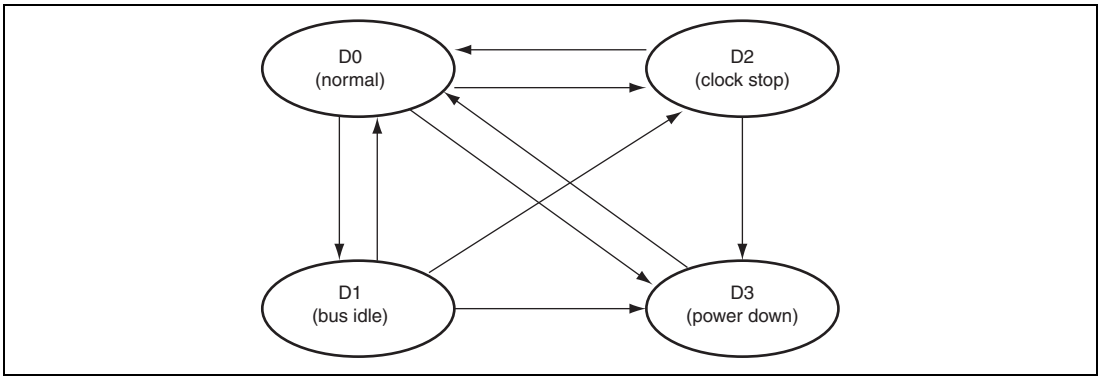


Figure 13.16 PCI Local Bus Power Down State Transition

The PCIC detects when the power state (PS) bit of the PCI power management control/status register changes (when it is written to from an external PCI device), and issues a power management interrupt. To control the power management interrupts, there are the PCI power management interrupt register (PCIPINT) and PCI power management interrupt mask register (PCIPINTM). Of the power management interrupts, the power state D0 interrupt (PCIPWD0) detects a transition from the power state D1/D2/D3 to D0, while power state D1 interrupt (PCIPWD1) detects a transition from the power state D0 to D1, while power state D2 interrupt (PCIPWD2) detects a transition from the power state D0/D1 to D2, while power state D3 interrupt (PCIPWD3) detects a transition from the power state D0/D1/D2 to D3. Interrupt masks can be set for each interrupt.

No power state D0 interrupt is generated at a power-on reset.

The following cautions should be noted when the PCIC is operating in normal mode and a power down interrupt is received from the host: In PCI power management, the PCI local bus clock stops within a minimum of 16 clocks after the host device has instructed a transition to power state D3. After detecting a power state D3 interrupt, do not, therefore, attempt to read or write to local registers and configuration registers that can be accessed from the SuperHyway bus and PCI local bus access (I/O and memory spaces). Because these accesses operate using the PCI local bus clock, the cycle for these accesses will not be completed if the clock stops and may be hung-up on the SuperHyway bus.

13.4.8 PCI Local Bus Basic Interface

The PCIC of this LSI conforms to the PCI local bus specification revision 2.2 stipulations and can be connected to a device with a PCI local bus interface. The following figures show the timing for each operation mode.

(1) Master Read/Write Cycle Timing

Figures13.17 is an example of a single-write cycle in host bus bridge mode. Figure 13.18 is an example of a single read cycle in host bus bridge mode. Figure 13.19 is an example of a burst write cycle in normal mode. And Figure 13.20 is an example of a burst read cycle in normal mode. Note that the response speed of $\overline{\text{DEVSEL}}$ and $\overline{\text{TRDY}}$ differs according to the connected target device. In host bus bridge mode, master accesses always use single read/write cycles. The issuing of configuration transfers is only possible in host bus bridge mode.

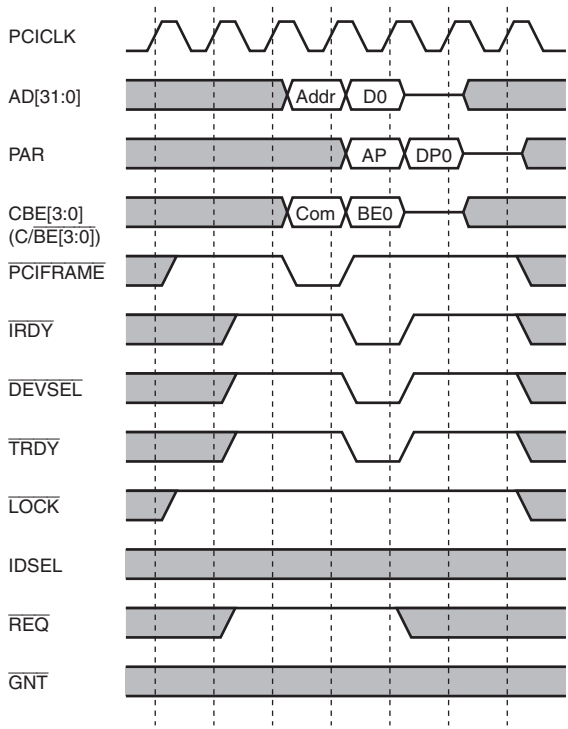


Figure 13.17 Master Write Cycle in Host Bus Bridge Mode (Single)

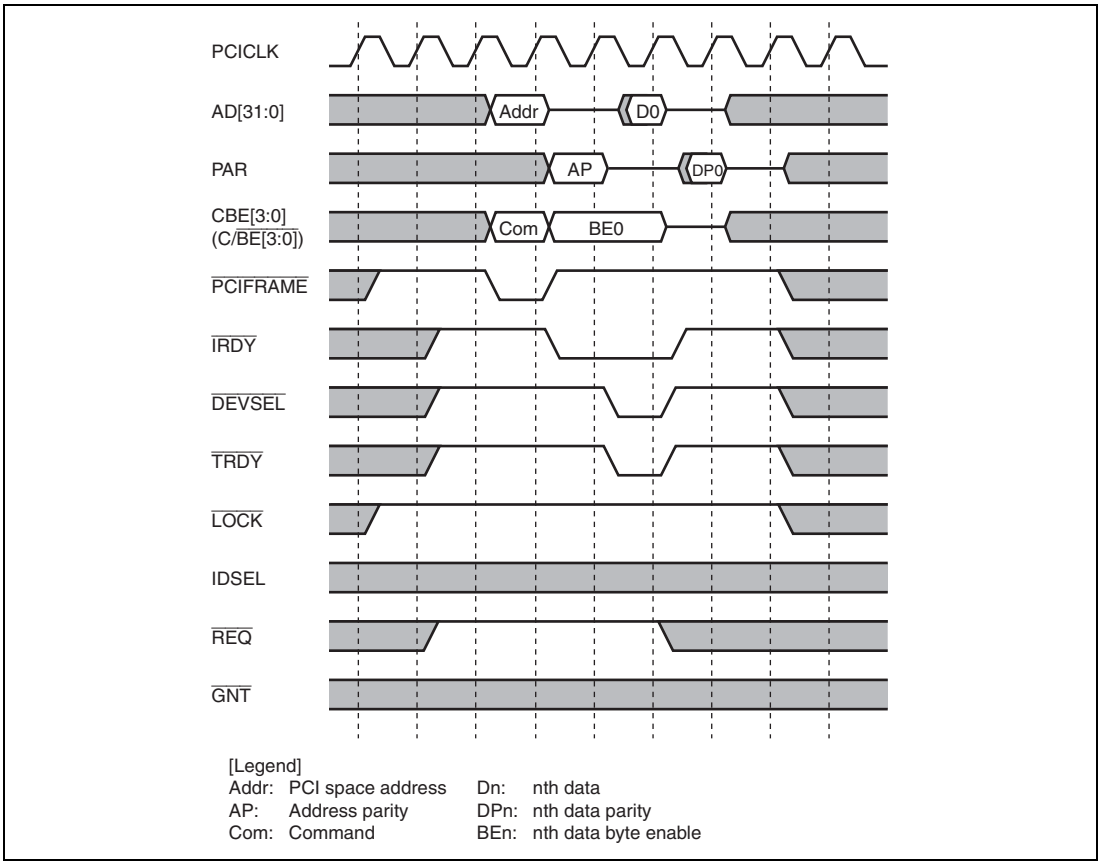
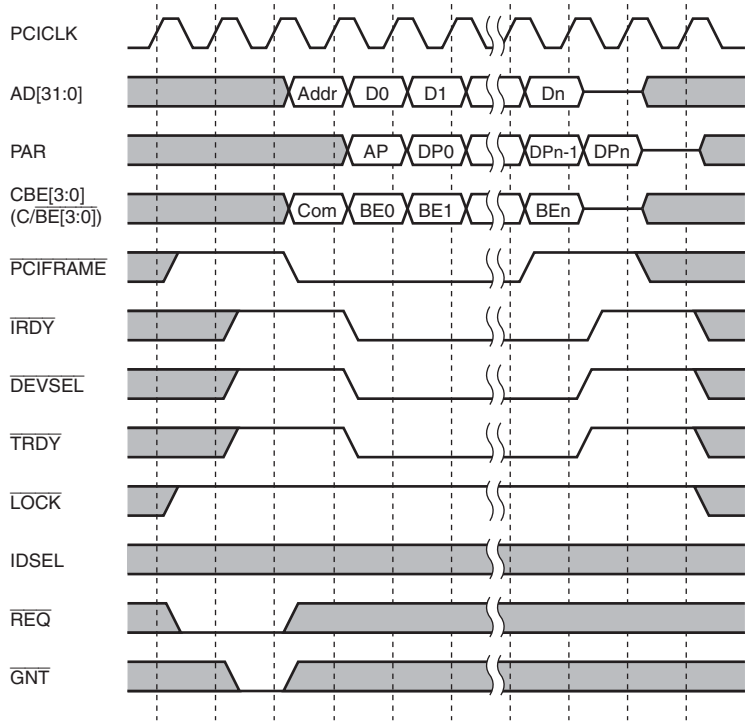


Figure 13.18 Master Read Cycle in Host Bus Bridge Mode (Single)



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.19 Master Write Cycle in Normal Mode (Burst)

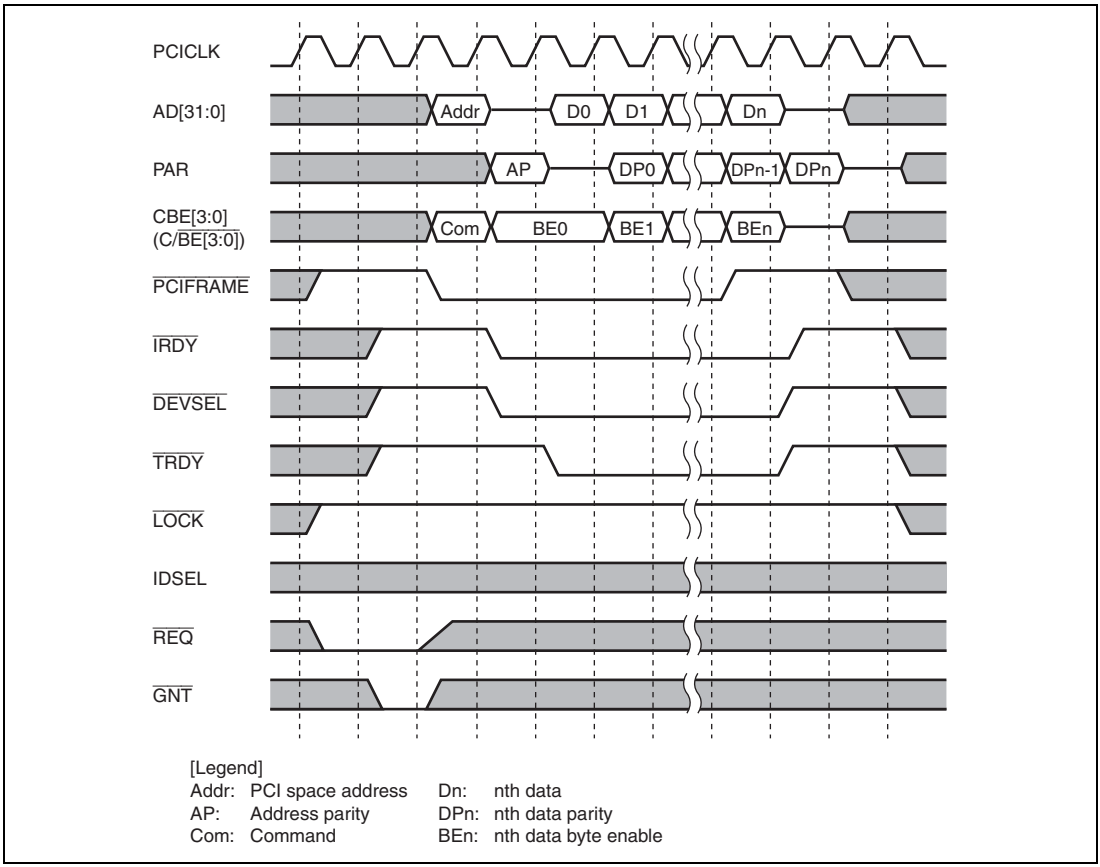


Figure 13.20 Master Read Cycle in Normal Mode (Burst)

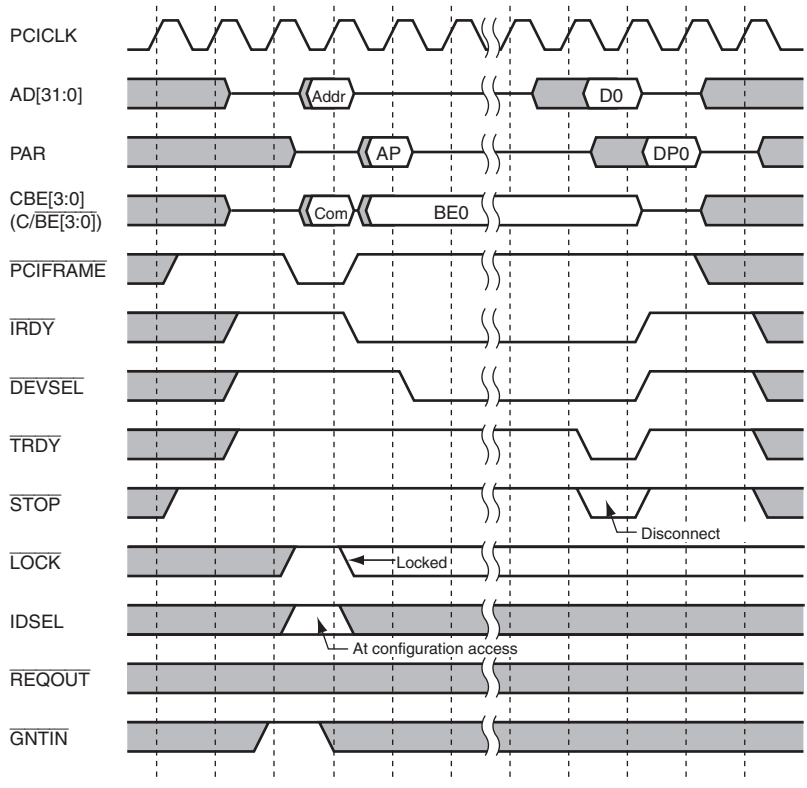
(2) Target Read/Write Cycle Timing

The PCIC responds to target memory burst read accesses from an external master by retries until 8 longword (32-bit) data are prepared in the PCIC's internal FIFO. That is, it always responds to the first target burst read with a retry. For a single read access, the PCIC reaponds as soon as the data is prepared.

Also, when a target memory write access is made, the content of the data is guaranteed until the write data is completely written to the local memory if reading the target write data immediately after write access.

Only single transfers are supported in the case of target accesses of the configuration space and I/O space. If there is a burst access request, the external master is disconnected on completion of the first transfer. Note that the $\overline{\text{DEVSEL}}$ response speed is fixed at 2 clocks (Medium) in the case of target access to the PCIC.

Figure 13.21 shows an example target single read cycle in normal mode. Figure 13.22 shows an example target single write cycle in normal mode. Figure 13.23 is an example of a target burst read cycle in host bus bridge mode. And figure 13.24 is an example of a target burst write cycle in host bus bridge mode.



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.21 Target Read Cycle in Normal Mode (Single)

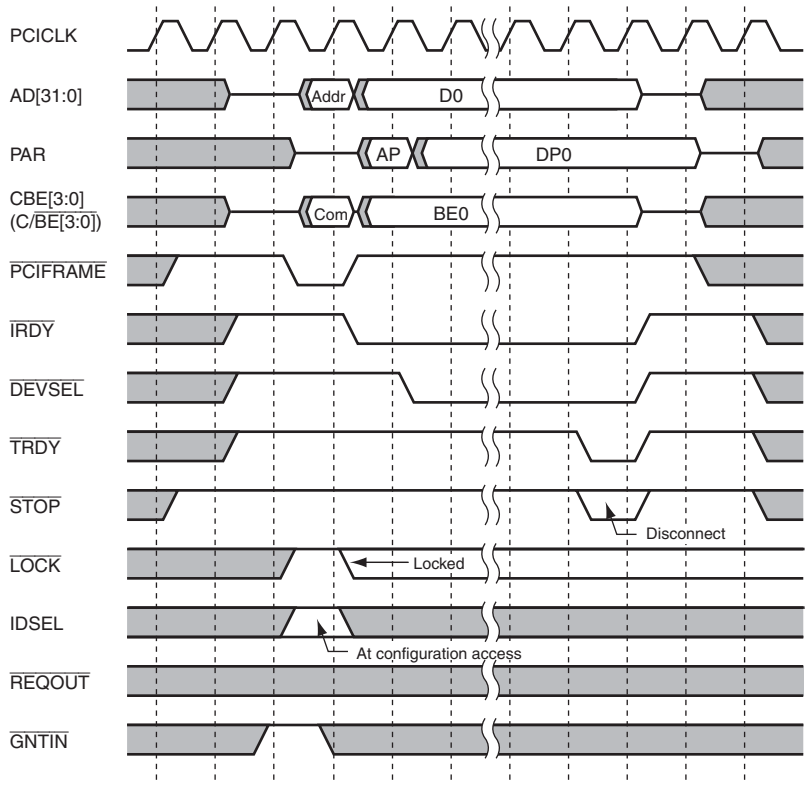


Figure 13.22 Target Write Cycle in Normal Mode (Single)

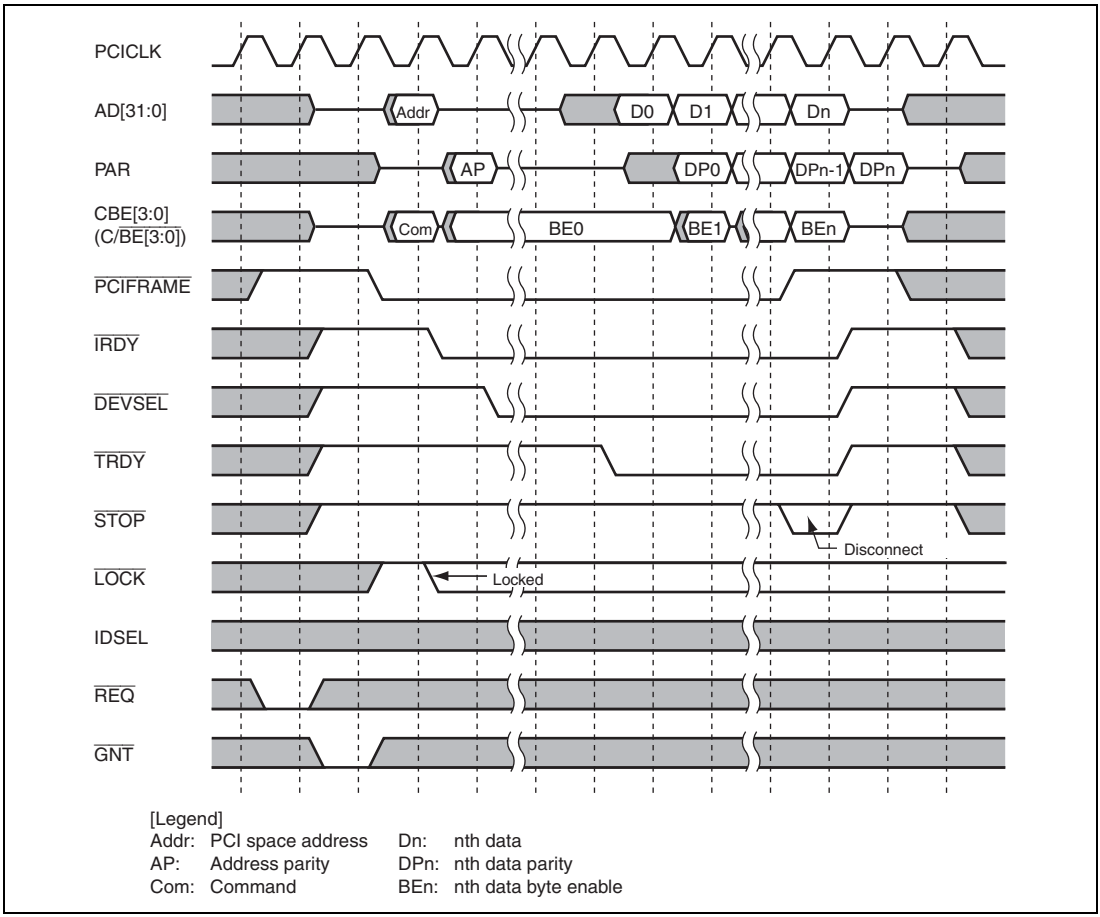


Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burst)

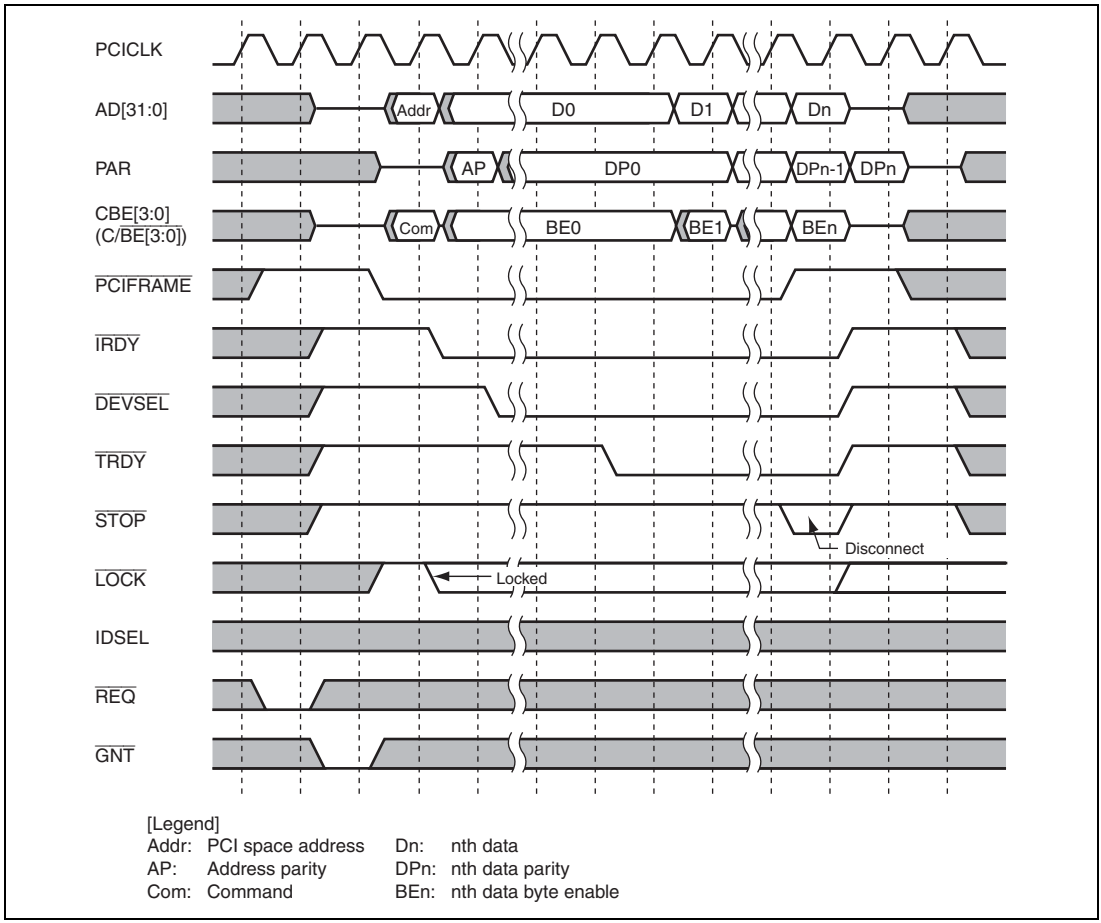


Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst)

(3) Address/Data Stepping Timing

By writing 1 to the SC bit in PCICMD, a wait (stepping) of one clock can be inserted when the PCIC is driving the AD bus. As a result, the PCIC drives the AD bus over 2 clocks. This function can be used when there is a heavy load on the PCI bus and the AD bus does not achieve the stipulated logic level in one clock.

When the PCIC operates as the host bus bridge mode, it is recommended to use this function for the issuance of configuration transfers.

Figure 13.25 is an example of burst memory write cycle with stepping. Figure 13.26 is an example of target burst read cycle with stepping.

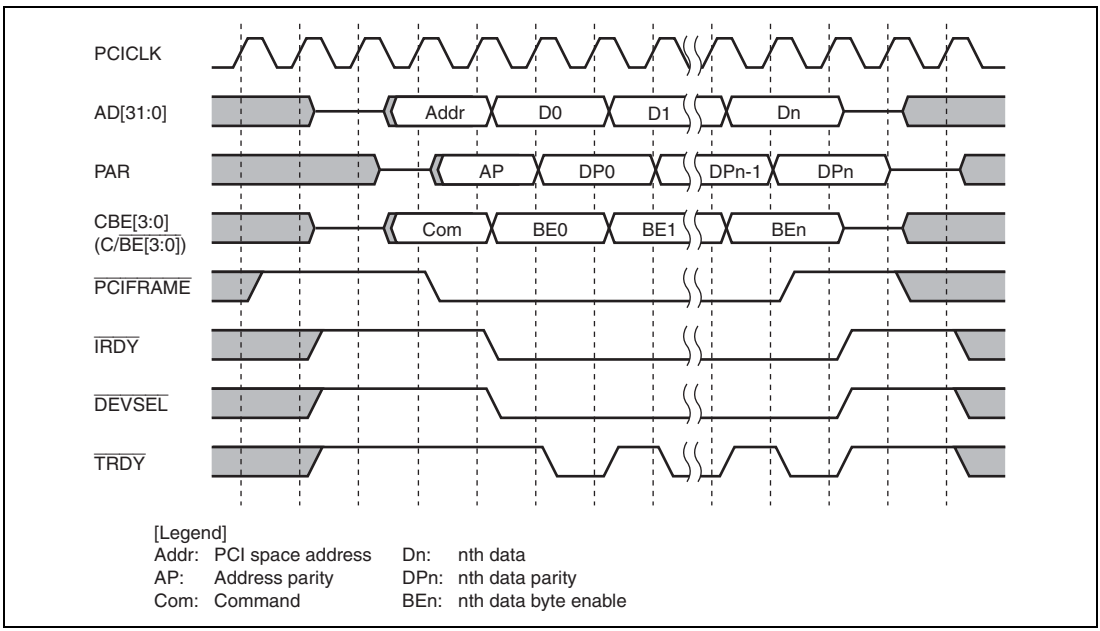


Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with stepping)

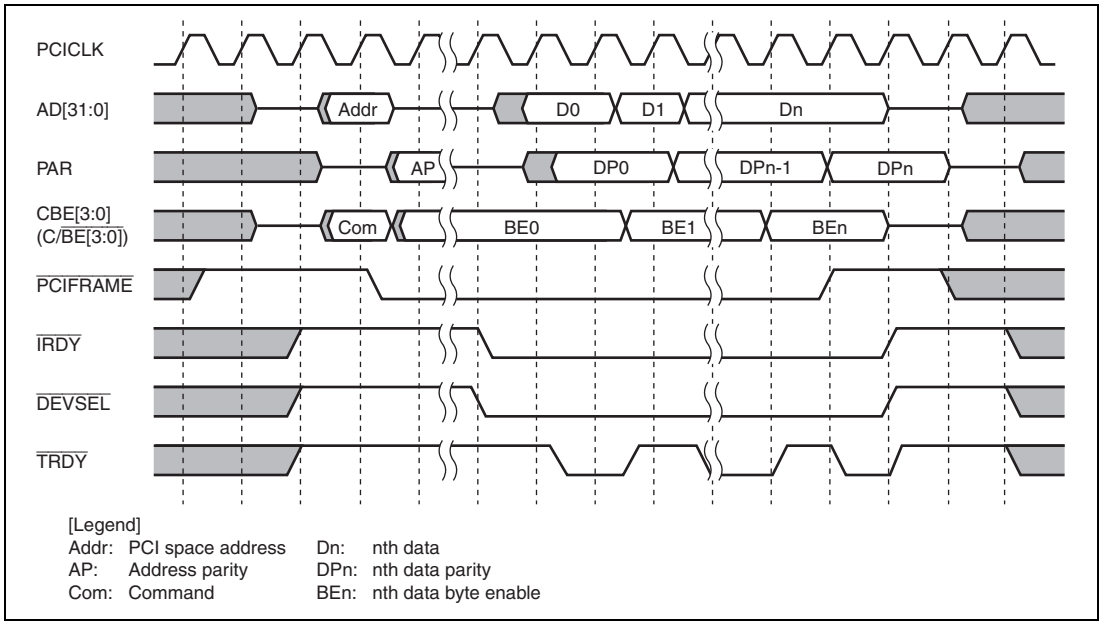


Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with stepping)

13.5 Usage Notes

13.5.1 Notes on PCIC Target Reading

When the PCIC is used in target mode and all the three conditions below are satisfied, data may be lost during a PCIC target read.

1. PFCS bit in PCICR = 1 (32-byte pre-fetch enabled)
2. FTO bit in PCICR = 1 ($\overline{\text{TRDY}}$ control enabled)
3. PFE bit in PCICR = 1 (pre-fetch enabled)

For target reading in target mode, at least one of the above three conditions must be cancelled.

13.5.2 Notes on Host Mode

When the PCIC is used while all the five conditions below are satisfied, $\overline{\text{REQn}}$ (n = 3 to 1) with the lowest priority is masked, thus disabling correct transfers via the PCI bus, which leads to unstable operation of the PCI bus system.

1. Host mode (MD6 = high)
2. PCI bus master arbitration mode is set to fixed mode (BMAM bit in PCICR = 0)
3. In addition to this LSI (with the PCIC in host mode), two or more external PCI devices that can be a bus master are connected to the PCI bus.
4. Among the above external devices, there is at least one device ($\overline{\text{REQm}}$) that does not execute $\overline{\text{REQ}}$ negation and $\overline{\text{FRAME}}$ assertion simultaneously when a single transaction is requested (single or burst transfer).
5. There is an external device ($\overline{\text{REQn}}$; n > m) that can be a bus master with a priority lower than the priority of the external device ($\overline{\text{REQm}}$) satisfying condition 4 above.

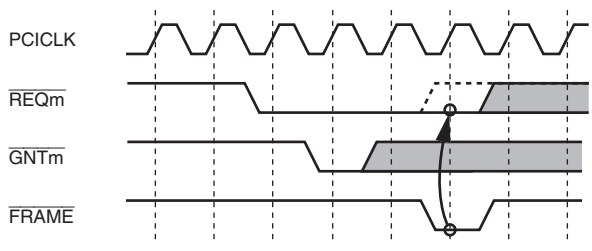


Figure 13.27 Timing Example of Device ($\overline{\text{REQm}}$) Not Executing $\overline{\text{REQ}}$ Negation and $\overline{\text{FRAME}}$ Assertion Simultaneously

To prevent a device that does not execute $\overline{\text{REQ}}$ negation and $\overline{\text{FRAME}}$ assertion simultaneously (figure 13.27) from being a bus master, preventive measure 1 or 2 below should be taken.

1. Use pseudo-round-robin mode.

   Pseudo-round-robin mode should be set (BMAM bit in PCICR = 1) as the PCI bus arbitration scheme.
2. Assign the lowest priority level to the relevant device.

   When there is only one device that does not execute $\overline{\text{REQ}}$ negation and $\overline{\text{FRAME}}$ assertion simultaneously, the device should be connected to the $\overline{\text{REQ}}$n and $\overline{\text{GNT}}$n with the lowest priority.

However, if none of the external devices connected to the PCI has such negation/assertion timing or if none of the external devices with such negation/assertion timing can be a bus master, the above preventive measures are not required.

Section 14 Direct Memory Access Controller (DMAC)

This LSI includes the direct memory access controller (DMAC).

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

14.1 Features

- Six channels (four channels can receive an external request: channel 0 to 3)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), 16 bytes, and 32 bytes
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode
- Transfer requests:

  External request (channel 0 to 3), on-chip peripheral module request (channel 0 to 5), or auto request can be selected.

  The following modules can issue an on-chip peripheral module request.

  — CMT, SCIF0, SCIF1, SCIF2, HAC, USBF, SSI0 to SSI3, MMCIF, SIM, SIOF0 to SIOF2, STIF0 and STIF1
- Selectable bus modes:

  Cycle steal mode (normal mode and intermittent mode) or burst mode can be selected.
- Selectable channel priority levels:

  The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be generated to the CPU after half of the transfers ended, all transfers ended, or an address error occurred.
- External request detection: There are following four types of DREQn input detection. (n = 0 to 3)

  — Low level detection (Initial value)

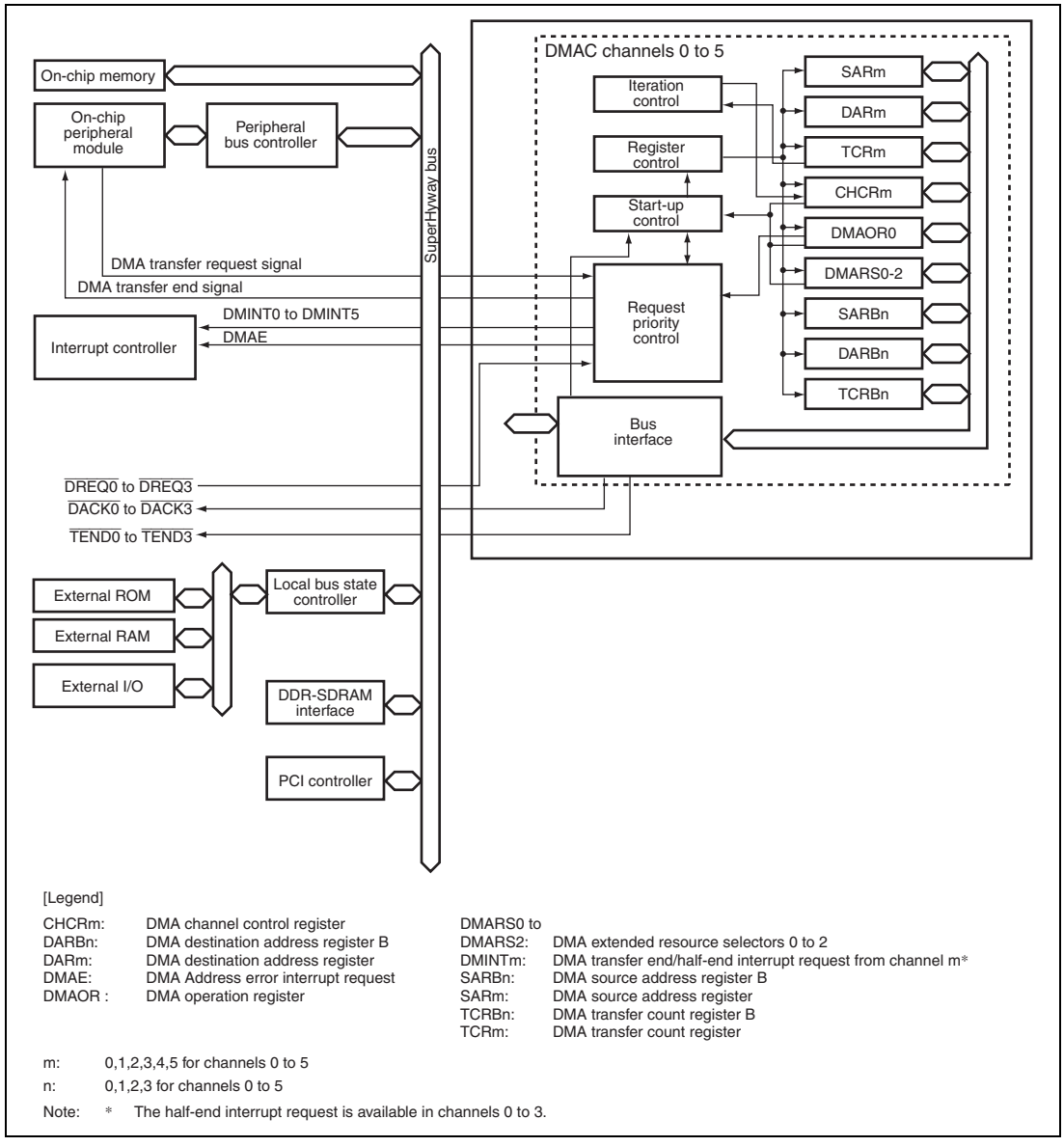
  — High level detection

  — Rising edge detection

  — Falling edge detection

- Active levels for both the DMA transfer request acceptance signal ($\overline{\text{DACKn}}$) and DMA transfer end signal ($\overline{\text{TENDn}}$) can be set. (n = 0 to 3)

Figure 14.1 shows the block diagram of the DMAC.



[Legend]

CHCRm:	DMA channel control register	DMARS0 to DMARS2:	DMA extended resource selectors 0 to 2
DARBn:	DMA destination address register B	DMINTm:	DMA transfer end/half-end interrupt request from channel m*
DARm:	DMA destination address register	SARBn:	DMA source address register B
DMAE:	DMA Address error interrupt request	SARm:	DMA source address register
DMAOR :	DMA operation register	TCRBn:	DMA transfer count register B
		TCRm:	DMA transfer count register

m: 0,1,2,3,4,5 for channels 0 to 5

n: 0,1,2,3 for channels 0 to 5

Note: * The half-end interrupt request is available in channels 0 to 3.

Figure 14.1 Block Diagram of DMAC

14.2 Input/Output Pins

The external pins for the DMAC are described below. Table 14.1 lists the configuration of the pins that are connected to external device. The DMAC has pins for four channels (channel 0 to 3) for external bus use.

The input/output pins of channel 1 are divided in two groups: normal I/O group and mirror I/O group. The input/output operations of pins in two groups are always the same. The pin select register of the GPIO is used to select the Channel 1 pins.

Table 14.1 Pin Configuration

Channel		Pin Name	Function	I/O	Description
0		$\overline{\text{DREQ0}}$*1	DMA transfer request	Input	DMA transfer request input from external device to channel 0
		$\overline{\text{DACK0}}$*2	DMA transfer request acknowledge	Output	Strobe output from channel 0 to external device which has output, regarding DMA transfer request
		$\overline{\text{TEND0}}$*2	DMA transfer end notification	Output	DMA transfer end output from channel 0 to external device
1	Normal I/O Pins	$\overline{\text{DREQ1}}$*1	DMA transfer request	Input	DMA transfer request input from external device to channel 1
		$\overline{\text{DACK1}}$*2	DMA transfer request acknowledge	Output	Strobe output from channel 1 to external device which has output, regarding DMA transfer request
		$\overline{\text{TEND1}}$*2	DMA transfer end notification	Output	DMA transfer end output from channel 1 to external device
	Miller I/O Pins	$\overline{\text{DREQ1M}}$*1	DMA transfer request	Input	DMA transfer request input from external device to channel 1
		$\overline{\text{DACK1M}}$*2	DMA transfer request acknowledge	Output	Strobe output from channel 1 to external device which has output, regarding DMA transfer request
		$\overline{\text{TEND1M}}$*2	DMA transfer end notification	Output	DMA transfer end output from channel 1 to external device

Channel	Pin Name	Function	I/O	Description
2	$\overline{\text{DREQ2}}$*1	DMA transfer request	Input	DMA transfer request input from external device to channel 2
	$\overline{\text{DACK2}}$*2	DMA transfer request acknowledge	Output	Strobe output from channel 2 to external device which has output, regarding DMA transfer request
	$\overline{\text{TEND2}}$*2	DMA transfer end notification	Output	DMA transfer end output from channel 2 to external device
3	$\overline{\text{DREQ3}}$*1	DMA transfer request	Input	DMA transfer request input from external device to channel 3
	$\overline{\text{DACK3}}$*2	DMA transfer request acknowledge	Output	Strobe output from channel 3 to external device which has output, regarding DMA transfer request
	$\overline{\text{TEND3}}$*2	DMA transfer end notification	Output	DMA transfer end output from channel 3 to external device

Notes: 1. The initial value is detected at low level.
2. The initial value is low active.

14.3 Register Descriptions

Table 14.2 shows the configuration of registers of the DMAC. Table 14.3 shows the state of registers in each processing mode.

Table 14.2 Register Configuration of DMAC

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*3
0	DMA source address register 0	SAR0	R/W	H'FF60 8020	H'1F60 8020	32
	DMA destination address register 0	DAR0	R/W	H'FF60 8024	H'1F60 8024	32
	DMA transfer count register 0	TCR0	R/W	H'FF60 8028	H'1F60 8028	32
	DMA channel control register 0	CHCR0	R/W*1	H'FF60 802C	H'1F60 802C	32
1	DMA source address register 1	SAR1	R/W	H'FF60 8030	H'1F60 8030	32
	DMA destination address register 1	DAR1	R/W	H'FF60 8034	H'1F60 8034	32
	DMA transfer count register 1	TCR1	R/W	H'FF60 8038	H'1F60 8038	32
	DMA channel control register 1	CHCR1	R/W*1	H'FF60 803C	H'1F60 803C	32
2	DMA source address register 2	SAR2	R/W	H'FF60 8040	H'1F60 8040	32
	DMA destination address register 2	DAR2	R/W	H'FF60 8044	H'1F60 8044	32
	DMA transfer count register 2	TCR2	R/W	H'FF60 8048	H'1F60 8048	32
	DMA channel control register 2	CHCR2	R/W*1	H'FF60 804C	H'1F60 804C	32
3	DMA source address register 3	SAR3	R/W	H'FF60 8050	H'1F60 8050	32
	DMA destination address register 3	DAR3	R/W	H'FF60 8054	H'1F60 8054	32
	DMA transfer count register 3	TCR3	R/W	H'FF60 8058	H'1F60 8058	32
	DMA channel control register 3	CHCR3	R/W*1	H'FF60 805C	H'1F60 805C	32
0 to 5	DMA operation register	DMAOR	R/W*2	H'FF60 8060	H'1F60 8060	16
4	DMA source address register 4	SAR4	R/W	H'FF60 8070	H'1F60 8070	32
	DMA destination address register 4	DAR4	R/W	H'FF60 8074	H'1F60 8074	32
	DMA transfer count register 4	TCR4	R/W	H'FF60 8078	H'1F60 8078	32
	DMA channel control register 4	CHCR4	R/W*1	H'FF60 807C	H'1F60 807C	32
5	DMA source address register 5	SAR5	R/W	H'FF60 8080	H'1F60 8080	32
	DMA destination address register 5	DAR5	R/W	H'FF60 8084	H'1F60 8084	32
	DMA transfer count register 5	TCR5	R/W	H'FF60 8088	H'1F60 8088	32
	DMA channel control register 5	CHCR5	R/W*1	H'FF60 808C	H'1F60 808C	32

Channel	Name	Abbrev.	R/W	P4 Address	Area 7 Address	Access Size*3
0	DMA source address register B0	SARB0	R/W	H'FF60 8120	H'1F60 8120	32
	DMA destination address register B0	DARB0	R/W	H'FF60 8124	H'1F60 8124	32
	DMA transfer count register B0	TCRB0	R/W	H'FF60 8128	H'1F60 8128	32
1	DMA source address register B1	SARB1	R/W	H'FF60 8130	H'1F60 8130	32
	DMA destination address register B1	DARB1	R/W	H'FF60 8134	H'1F60 8134	32
	DMA transfer count register B1	TCRB1	R/W	H'FF60 8138	H'1F60 8138	32
2	DMA source address register B2	SARB2	R/W	H'FF60 8140	H'1F60 8140	32
	DMA destination address register B2	DARB2	R/W	H'FF60 8144	H'1F60 8144	32
	DMA transfer count register B2	TCRB2	R/W	H'FF60 8148	H'1F60 8148	32
3	DMA source address register B3	SARB3	R/W	H'FF60 8150	H'1F60 8150	32
	DMA destination address register B3	DARB3	R/W	H'FF60 8154	H'1F60 8154	32
	DMA transfer count register B3	TCRB3	R/W	H'FF60 8158	H'1F60 8158	32
0, 1	DMA extended resource selector 0	DMARS0	R/W	H'FF60 9000	H'1F60 9000	16
2, 3	DMA extended resource selector 1	DMARS1	R/W	H'FF60 9004	H'1F60 9004	16
4, 5	DMA extended resource selector 2	DMARS2	R/W	H'FF60 9008	H'1F60 9008	16

Note: 1. Writing 0 after read 1 of HE or TE bit of CHCR is possible to clear the flag.
2. Writing 0 after read 1 of AE or NMIF bit of DMAOR is possible to clear the flag.
3. Accessing with other access sizes is prohibited.

Table 14.3 State of Registers in Each Operating Mode

Channel	Abbreviation	Power-on Reset	Manual Reset	Sleep	Stand by
0	SAR0	Undefined	Undefined	Retained	Retained
	DAR0	Undefined	Undefined	Retained	Retained
	TCR0	Undefined	Undefined	Retained	Retained
	CHCR0	H'4000 0000	H'4000 0000	Retained	Retained
1	SAR1	Undefined	Undefined	Retained	Retained
	DAR1	Undefined	Undefined	Retained	Retained
	TCR1	Undefined	Undefined	Retained	Retained
	CHCR1	H'4000 0000	H'4000 0000	Retained	Retained
2	SAR2	Undefined	Undefined	Retained	Retained
	DAR2	Undefined	Undefined	Retained	Retained
	TCR2	Undefined	Undefined	Retained	Retained
	CHCR2	H'4000 0000	H'4000 0000	Retained	Retained
3	SAR3	Undefined	Undefined	Retained	Retained
	DAR3	Undefined	Undefined	Retained	Retained
	TCR3	Undefined	Undefined	Retained	Retained
	CHCR3	H'4000 0000	H'4000 0000	Retained	Retained
0 to 5	DMAOR	H'0000	H'0000	Retained	Retained
4	SAR4	Undefined	Undefined	Retained	Retained
	DAR4	Undefined	Undefined	Retained	Retained
	TCR4	Undefined	Undefined	Retained	Retained
	CHCR4	H'4000 0000	H'4000 0000	Retained	Retained
5	SAR5	Undefined	Undefined	Retained	Retained
	DAR5	Undefined	Undefined	Retained	Retained
	TCR5	Undefined	Undefined	Retained	Retained
	CHCR5	H'4000 0000	H'4000 0000	Retained	Retained
0	SARB0	Undefined	Undefined	Retained	Retained
	DARB0	Undefined	Undefined	Retained	Retained
	TCRB0	Undefined	Undefined	Retained	Retained

Channel	Abbreviation	Power-on Reset	Manual Reset	Sleep	Stand by
1	SARB1	Undefined	Undefined	Retained	Retained
	DARB1	Undefined	Undefined	Retained	Retained
	TCRB1	Undefined	Undefined	Retained	Retained
2	SARB2	Undefined	Undefined	Retained	Retained
	DARB2	Undefined	Undefined	Retained	Retained
	TCRB2	Undefined	Undefined	Retained	Retained
3	SARB3	Undefined	Undefined	Retained	Retained
	DARB3	Undefined	Undefined	Retained	Retained
	TCRB3	Undefined	Undefined	Retained	Retained
0, 1	DMARS0	H'0000	H'0000	Retained	Retained
2, 3	DMARS1	H'0000	H'0000	Retained	Retained
4, 5	DMARS2	H'0000	H'0000	Retained	Retained

14.3.1 DMA Source Address Registers (SAR0 to SAR5)

SAR is 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.

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

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

14.3.2 DMA Source Address Registers (SARB0 to SARB3)

SARB is 32-bit readable/writable registers that specify the source address of a DMA transfer that is set in SAR again in repeat/reload mode. Data to be written from the CPU to SAR is also written to SARB. To set SARB address that differs from SAR address, write data to SARB after SAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.

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

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

14.3.3 DMA Destination Address Registers (DAR0 to DAR5)

DAR is 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the destination address value. The initial value is undefined.

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

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

14.3.4 DMA Destination Address Registers (DARB0 to DARB3)

DARB is 32-bit readable/writable registers that specify the destination address of a DMA transfer that is set in DAR again in repeat/reload mode. Data to be written from the CPU to DAR is also written to DARB. To set DARB address that differs from DAR address, write data to DARB after DAR.

To transfer data in word or in longword units, specify the address with word or longword address boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte boundary must be set for the source address value. The initial value is undefined.

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

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

14.3.5 DMA Transfer Count Registers (TCR0 to TCR5)

TCR is 32-bit readable/writable registers that specify the DMA transfer count. The number of transfers is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of TCR are always read as 0, and the write value should always be 0. The initial value is undefined.

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

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

14.3.6 DMA Transfer Count Registers (TCRB0 to TCRB3)

TCRB is 32-bit readable/writable registers. Data to be written from the CPU to TCR is also written to TCRB. While the HE function is used, TCRB are used as the initial value hold registers to detect HE. Also, TCRB specify the number of DMA transfers which are set in TCR in repeat mode. TCRB specify the number of DMA transfers and are used as transfer count counters in reload mode.

In reload mode, the lower 16 bits operate as transfer count counters, values of SAR and DAR are updated after the value of the lower 16 bits became 0, and then the value of the upper 16 bits of TCRB are loaded to the lower 16 bits. In upper 16 bits, set the number of transfers which starts reloading. In reload mode, the same number of transfers should be set in both upper and lower 16 bits. Also, set the HIE bit in CHCR to 0 and do not use the HE function.

The initial value of TCRB is undefined.

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

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

14.3.7 DMA Channel Control Registers (CHCR0 to CHCR5)

CHCR is 32-bit readable/writable registers that control the DMA transfer mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	—	RPT[2:0]			—	DO	—	DVMD	TS[2]	HE	HIE	AM	AL
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: Writing 0 is possible to clear the flag.

Bit	Bit Name	Initial Value	R/W	Descriptions
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	LCKN	1	R/W	Bus Lock Signal Disable Specifies whether enable or disable the bus lock signal output when a load instruction is output in dual transfer mode. This bit is effective in cycle steal mode, and should be cleared to 0 in burst mode. To disable the bus lock signal, the bus request from the bus master other than the DMAC could be received, and so improve the bus usage efficiency in total system. 0: Bus lock signal output enabled 1: Bus lock signal output disabled
29, 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
27 to 25	RPT[2:0]	000	R/W	DMA Setting Renewal Specify These bits are enabled in CHCR0 to CHCR3. 000: Normal mode 001: Repeat mode SAR/DAR/TCR used as repeat area 010: Repeat mode DAR/TCR used as repeat area 011: Repeat mode SAR/TCR used as repeat mode 100: Reserved (setting prohibited) 101: Reload mode SAR/DAR/TCR used as reload area 110: Reload mode DAR/TCR used as reload area 111: Reload mode SAR/TCR used as reload area
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR0 to CHCR3. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
21	DVMD	0	R/W	Division Transfer Mode Specification Specifies the execution of the DMA transfer in 16-byte units between the on-chip peripheral module STIF and the external memory. When the STIF is used, always write 1 to this bit. When the STIF is not used, always write 0 to this bit.

Bit	Bit Name	Initial Value	R/W	Descriptions
20	TS2	0	R/W	DMA Transfer Size Specify With TS1 and TS0, this bit specifies the DMA transfer size. When the transfer source or transfer destination is a register of an on-chip peripheral module with a transfer size set, a proper transfer size for the register should be set. For the transfer source or destination address specified by SAR or DAR, an address boundary should be set according to the transfer data size. TS[2:0] 000: Byte units transfer 001: Word (2-byte) units transfer 010: Longword (4-byte) units transfer 011: 16-byte units transfer 100: 32-byte units transfer Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half End Flag After HIE (bit 18) is set to 1 and the number of transfers become half of TCR (1 bit shift to right) which is set before transfer starts, HE becomes 1. This bit is set to 1 when the TCR value is equal to (TCR set before transfer)/2: TCR value is set to even number of times (TCR set before transfer -1)/2: TCR value is set to odd number of times 8,388,608 (H'0080 0000): TCR value is set to the maximum number of times (H'0000 0000) The HE bit is not set when transfers are ended by an NMI interrupt or address error, or by clearing the DE bit or the DME bit in DMAOR before the number of transfers is decreased to half of the TCR value set preceding the transfer. The HE bit is kept set when the transfer ends by an NMI interrupt or address error, or clearing the DE bit (bit 0) or the DME bit in DMAOR after the HE bit is set to 1. To clear the HE bit, write 0 after reading 1 in the HE bit. This bit is valid only in CHCR0 to CHCR3. 0: During the DMA transfer or DMA transfer has been interrupted TCR > (TCR set before transfer)/2 [Clearing condition] Writing 0 after HE = 1 is read. 1: TCR • (TCR set before transfer)/2
18	HIE	0	R/W	Half End Interrupt Enable Specifies whether an interrupt request is generated to the CPU when the number of transfers is decreased to half of the TCR value set preceding the transfer. When the HIE bit is set to 1 and the HE bit is set, an interrupt request is generated to the CPU. Clear this bit to 0 while reload mode is set. This bit is valid in CHCR0 to CHCR3. 0: Half end Interrupt disabled 1: Half end Interrupt enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
17	AM	0	R/W	Acknowledge Mode Selects whether $\overline{\text{DACK}}$ is output in data read cycle or in data write cycle. This bit is valid only in CHCR0 to CHCR3. 0: $\overline{\text{DACK}}$ output in read cycle 1: $\overline{\text{DACK}}$ output in write cycle
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK signal output is high active or low active. This bit is valid only in CHCR0 to CHCR3. 0: Low-active output of DACK and TEND 1: High-active output of DACK and TEND
15, 14	DM[1:0]	00	R/W	Destination Address Mode Specify whether the DMA destination address is incremented, decremented, or left fixed. 00: Fixed destination address 01: Destination address is incremented +1 in byte units transfer +2 in word units transfer +4 in longword units transfer +16 in 16-byte units transfer +32 in 32-byte units transfer 10: Destination address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 16/32-byte units transfer 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
13, 12	SM[1:0]	00	R/W	Source Address Mode Specify whether the DMA source address is incremented, decremented, or left fixed. 00: Fixed source address 01: Source address is incremented +1 in byte units transfer +2 in word units transfer +4 in longword units transfer +16 in 16-byte units transfer +32 in 32-byte units transfer 10: Source address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 16/32-byte units transfer 11: Setting prohibited
11 to 8	RS[3:0]	0000	R/W	Resource Select Specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state that the DMA enable bit (DE) is cleared to 0. 0000: External request, dual address mode 0100: Auto request 1000: Selected by DMA extended resource selector (for on-chip modules) Other than above: Setting prohibited Note: External request specification is valid only in CHCR0 to CHCR3. None of the external request can be selected in CHCR4 and CHCR5.

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 CHCR0 to CHCR3. In channels 0 to 3, 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 at low level 01: DREQ detected at falling edge 10: DREQ detected at 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 Burst mode cannot be used when the on-chip peripheral module is the transfer request source.
4, 3	TS[1:0]	00	R/W	DMA Transfer Size Specify See the description of TS[2] (bit 20).
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 TCR 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 TCR is cleared to 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMAOR. To clear the TE bit, the TE bit should be written to 0 after reading 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been interrupted [Clearing condition] Writing 0 after TE = 1 read 1: DMA transfer ends by the specified count (TCR = 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 To abort the DMA transfer in on-chip peripheral module request mode, clear the DE bit to 0 while the DMA request from the corresponding peripheral module is cleared.

Note: * Writing 0 is possible to clear the flag.

14.3.8 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. DMAOR is a common register for channel 0 to 5.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CMS[1:0]		—	—	PR[1:0]		—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select Select either normal mode or intermittent mode in cycle steal mode. It is necessary that all channel 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.

Bit	Bit Name	Initial Value	R/W	Descriptions
9, 8	PR[1:0]	00	R/W	Priority Mode 1, 0 Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 01: CH0 > CH2 > CH3 > CH1 > CH4 > CH5 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.
2	AE	0	R/(W)*	Address Error Flag Indicates that an address error occurred during DMA transfer. This bit is set under following conditions: • The value set in SAR or DAR does not match to the transfer size boundary. • The transfer source or transfer destination is invalid space. • The transfer source or transfer destination is in module stop mode If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. 0: No DMAC address error [Clearing condition] Writing AE = 0 after AE = 1 read 1: DMAC address error occurs

Bit	Bit Name	Initial Value	R/W	Descriptions
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this bit is set, DMA transfer is disabled even if the DE bit in CHCR and the DME bit in DMAOR are set to 1. When the NMI is input, the DMA transfer in progress can be done in at least one transfer unit. When the DMAC is not in operational, the NMIF bit is set to 1 even if the NMI interrupt was input. 0: No NMI interrupt [Clearing condition] Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs Note: DMA transfer is stopped when an NMI interrupt is input. After returning from the NMI interrupt routine, set all channels again, and then start the DMA transfer.
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: To abort the DMA transfer when the on-chip peripheral module request mode is set for any of the channels specified by DMAOR (channel 0 to 5), clear the DE bit to 0 while the DMA transfer request from the corresponding peripheral module is cleared.

Note: * Writing 0 is possible to clear the flag.

14.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2)

DMARS is 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 specifies for channels 0 and 1, DMARS1 specifies for channels 2 and 3, and DMARS2 specifies for channels 4 and 5. This register can set the transfer request of CMT, SCIF0 to SCIF2. HAC, USBF, SSI0 to SSI3, MMCIF, SIM, SIOF0 to SIOF2, STIF0, AND STIF1.

When MID/RID other than the values listed in table 14.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits RS[3:0] has been set to B'1000 for CHCR0 to CHCR5 registers. Otherwise, even if DMARS has been set, transfer request source is not accepted.

- DMARS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C1MID[5:0]						C1RID[1:0]		C0MID[5:0]						C0RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C1MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 1 (MID) See table 14.4.
9, 8	C1RID[1:0]	00	R/W	Transfer request register ID for DMA channel 1 (RID) See table 14.4.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 0 (MID) See table 14.4
1, 0	C0RID[1:0]	00	R/W	Transfer request register ID for DMA channel 0 (RID) See table 14.4.

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C3MID[5:0]						C3RID[1:0]		C2MID[5:0]						C2RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C3MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 3 (MID) See table 14.4.
9, 8	C3RID[1:0]	00	R/W	Transfer request register ID0 for DMA channel 3 (RID) See table 14.4.
7 to 2	C2MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 2 (MID) See table 14.4.
1, 0	C2RID[1:0]	00	R/W R/W	Transfer request register ID for DMA channel 2 (RID) See table 14.4.

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]						C5RID[1:0]		C4MID[5:0]						C4RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C5MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 5 (MID) See table 14.4.
9, 8	C5RID[1:0]	00	R/W	Transfer request register ID for DMA channel 5 (RID) See table 14.4.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request module ID for DMA channel 4 (MID) See table 14.4.
1, 0	C4RID[1:0]	00	R/W	Transfer request register ID for DMA channel 4 (RID) See table 14.4.

Table 14.4 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID and RID)	MID	RID	Function
CMT channel 0	H'03	B'0000 00	B'11	—
CMT channel 1	H'07	B'0000 01	B'11	—
CMT channel 2	H'0B	B'0000 10	B'11	—
CMT channel 3	H'0F	B'0000 11	B'11	—
CMT channel 4	H'13	B'0001 00	B'11	—
SCIF0	H'21	B'0010 00	B'01	Transmit
	H'22	B'0010 00	B'10	Receive
SCIF1	H'29	B'0010 10	B'01	Transmit
	H'2A	B'0010 10	B'10	Receive
SCIF2	H'41	B'0100 00	B'01	Transmit
	H'42	B'0100 00	B'10	Receive
HAC	H'45	B'0100 01	B'01	Transmit
	H'46	B'0100 01	B'10	Receive
USBF	H'51	B'0101 00	B'01	Transmit
	H'52	B'0101 00	B'10	Receive
SSI0	H'73	B'0111 00	B'11	Transmit and Receive
SSI1	H'77	B'0111 01	B'11	Transmit and Receive
SSI2	H'83	B'1000 00	B'11	Transmit and Receive
SSI3	H'87	B'1000 01	B'11	Transmit and Receive
MMCIF	H'93	B'10 01 00	B'11	Transmit and Receive
SIM	H'A1	B'1010 00	B'01	Transmit
	H'A2	B'1010 00	B'10	Receive

Peripheral Module	Setting Value for One Channel (MID and RID)	MID	RID	Function
SIOF0	H'B1	B'1011 00	B'01	Transmit
	H'B2	B'1011 00	B'10	Receive
SIOF1	H'B5	B'1011 01	B'01	Transmit
	H'B6	B'1011 01	B'10	Receive
SIOF2	H'C1	B'1100 00	B'01	Transmit
	H'C2	B'1100 00	B'10	Receive
STIF0	H'D3	B'1101 00	B'11	Transmit and Receive
STIF1	H'D7	B'1101 01	B'11	Transmit and Receive

14.4 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

14.4.1 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by external devices or 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 bits RS[3:0] in CHCR0 to CHCR5, and DMARS0 to DMARS2.

(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 CHCR0 to CHCR5 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.

(2) External Request Mode

In this mode, a transfer is performed at the request signal ($\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$) of an external device. This mode is valid only in channel 0 to 3. In this mode, the RS[3:0] bits in CHCRn (n = 0 to 3) should be B'0000. 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 14.5 Setting External Request Mode with RS bit

CHCR						
RS[3]	RS[2]	RS[1]	RS[0]	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	Dual Address Mode	Any	Any

Choose to detect $\overline{\text{DREQ}}$ by either the edge or level of the signal input with the DL bit and DS bit in CHCRn (n = 0 to 3) as shown in table 14.6. The source of the transfer request does not have to be the data transfer source or destination.

Table 14.6 Selecting External Request Detection with DL, DS Bits

CHCRn (n=0 to 3)		
DL	**DS**	**Detection of External Request**
0	0	Low level detection (initial value; $\overline{\text{DREQ}}$)
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When $\overline{\text{DREQ}}$ is accepted, the $\overline{\text{DREQ}}$ pin becomes request accept disabled state. After issuing acknowledge signal $\overline{\text{DACK}}$ for the accepted $\overline{\text{DREQ}}$, the $\overline{\text{DREQ}}$ pin again becomes request accept enabled state.

When $\overline{\text{DREQ}}$ is used by level detection, there are following two cases by the timing to detect the next $\overline{\text{DREQ}}$ after outputting $\overline{\text{DACK}}$.

- Overrun 0: Transfer is aborted after the same number of transfer has been performed as requests.
- Overrun 1: Transfer is aborted after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 14.7 Selecting External Request Detection with DO Bit

CHCR	
DO	**External Request**
0	Overrun 0 (initial value)
1	Overrun 1

(3) 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 to SCIF2, HAC USBF, SSI0 to SSI3, MMCIF, SIM, SIOF0 to SIOF2, STIF0, and STIF1 set by DMARS0/1/2, and transfer requests from the CMT.

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, HAC USBF, SSI0 to SSI3, MMCIF, SIM, SIOF0 to SIOF2, STIF0, and STIF1.

Table 14.8 Selecting On-Chip Peripheral Module Request Modes with Bits RS[3:0]

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	000000	11	CMT channel 0	Compare-match transfer request	Any	Any	Cycle steal
	000001	11	CMT channel 1	Compare-match transfer request	Any	Any	Cycle steal
	000010	11	CMT channel 2	Compare-match transfer request	Any	Any	Cycle steal
	000011	11	CMT channel 3	Compare-match transfer request	Any	Any	Cycle steal
	000100	11	CMT channel 4	Compare-match transfer request	Any	Any	Cycle steal
	001000	01	SCI F0 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR0	Cycle steal
		10	SCIF0 receiver	RXI (receive FIFO data full interrupt)	SCFRDR0	Any	Cycle steal
	001010	01	SCI F1 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR1	Cycle steal
		10	SCIF1 receiver	RXI (receive FIFO data full interrupt)	SCFRDR1	Any	Cycle steal
	010000	01	SCIF2 transmitter	TXI (transmit FIFO data empty interrupt)	Any	SCFTDR2	Cycle steal
		10	SCIF2 receiver	RXI (receive FIFO data full interrupt)	SCFRDR2	Any	Cycle steal
	010001	01	HAC transmitter	Transmit data empty request	Any	HACPCML, HACPCMR	Cycle steal
		10	HAC receiver	Receive data is not read	HACPCML, HACPCMR	Any	Cycle steal
	010100	01	USB transmitter	Transmit data empty request	Any	EPDR	Cycle steal
		10	USB receiver	Receive data full request	EPDR	Any	Cycle steal
	011100	11	SSI0 transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSIRDR	Cycle steal
		11	SSI0 receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any	Cycle steal

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Source	Destination	Bus Mode
RS[3:0]	MID	RID					
1000	011101	11	SSI1 transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSITDR	Cycle steal
		11	SSI1 receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any	Cycle steal
	100000	11	SSI2 transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSITDR	Cycle steal
		11	SSI2 receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any	Cycle steal
	100001	11	SSI3 transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSITDR	Cycle steal
		11	SSI3 receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any	Cycle steal
	100100	11	MMCIF data part transmit	FIFO data write request	Any	DR	Cycle steal
			MMCIF data part receive	FIFO data read request	DR	Any	Cycle steal
	101000	01	SIM transmitter	TXT (transmit data empty)	Any	SCTDR	Cycle steal
		10	SIM receiver	RXI (receive data full)	SCRDR	Any	Cycle steal
	101100	01	SIOF0 transmitter	TXI (transmit FIFO data empty)	Any	SITDR0	Cycle steal
		10	SIOF0 receiver	RXI (receive data full)	SIRDR0	Any	Cycle steal
	101101	01	SIOF1 transmitter	TXI1 (transmit FIFO data empty)	Any	SITDR1	Cycle steal
		10	SIOF1 receiver	RXI (receive FIFO data empty)	SIRDR1	Any	Cycle steal
	110000	01	SIOF2 transmitter	TXI (transmit FIFO data empty)	Any	SITDR2	Cycle steal
		10	SIOF2 receiver	RXI (receive FIFO data full)	SIRDR2	Any	Cycle steal
	110100	11	STIF0 transmitter	FIFO data write request	Any	STI0FIFO0	Cycle steal
		11	STIF0 receiver	FIFO data read request	STI0FIFO0	Any	Cycle steal
	110101	11	STIF1 transmitter	FIFO data write request	Any	STI0FIFO1	Cycle steal
		11	STIF1 receiver	FIFO data read request	STI0FIFO1	Any	Cycle steal

14.4.2 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it transfers data according to a predetermined priority. Two modes (fixed mode and round-robin mode) are selected by the bits PR[1:0] in DMAOR.

(1) 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 > CH4 > CH5
- CH0 > CH2 > CH3 > CH1 > CH4 > CH5

These are selected by the bits PR[1:0] in DMAOR

(2) Round-Robin Mode

In round-robin mode each time data of one transfer unit (word, byte, longword, 16-byte, or 32-byte unit) is transferred on one channel, the priority is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority. The round-robin mode operation is shown in figure 14.2. The priority of round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 immediately after reset.

When round-robin mode is specified, do not mix the cycle steal mode and the burst mode in multiple channels' bus modes.

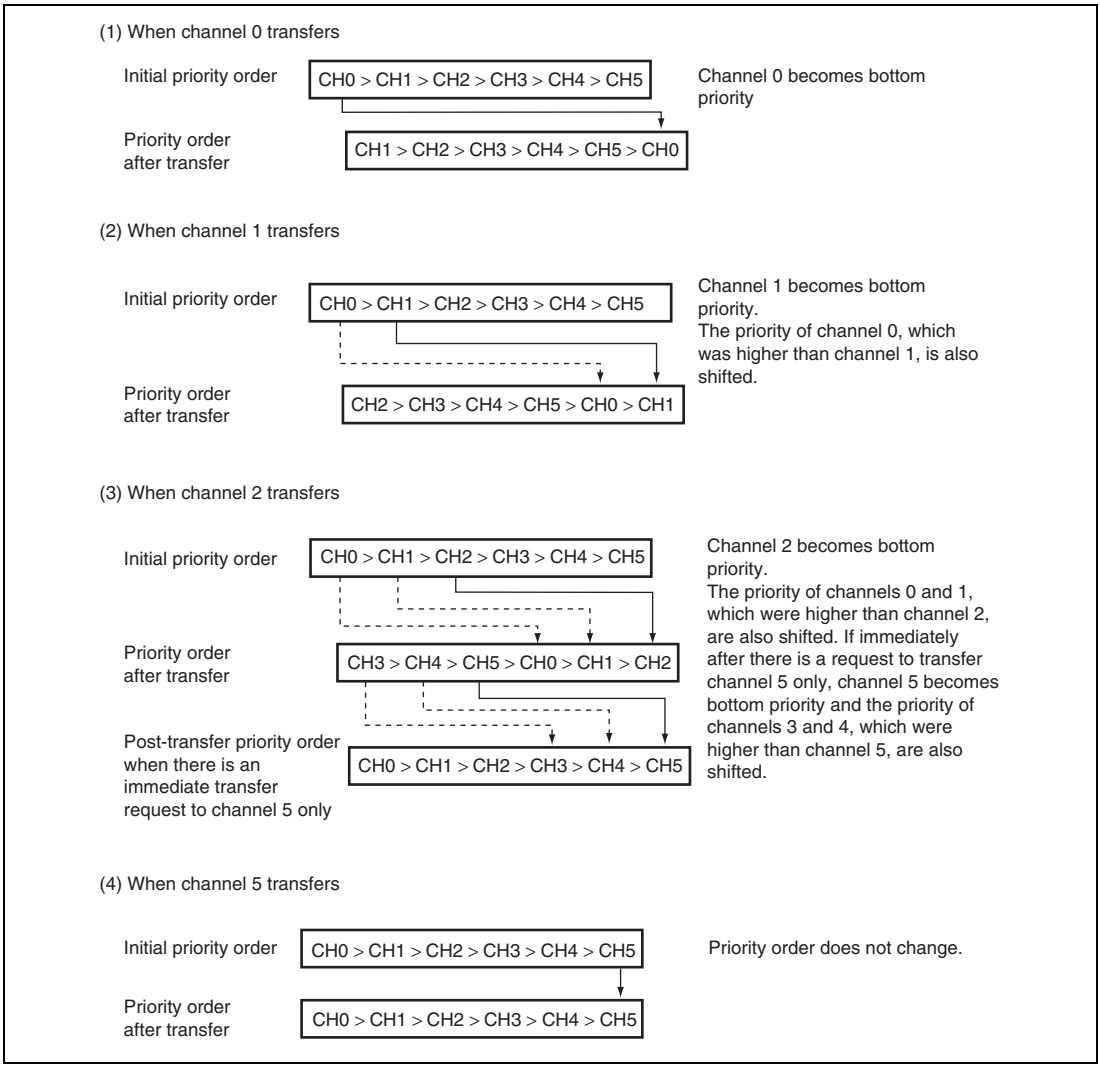


Figure 14.2 Round-Robin Mode

Figure 14.3 shows how the priority changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 becomes lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so that channel 3 becomes the lowest priority.

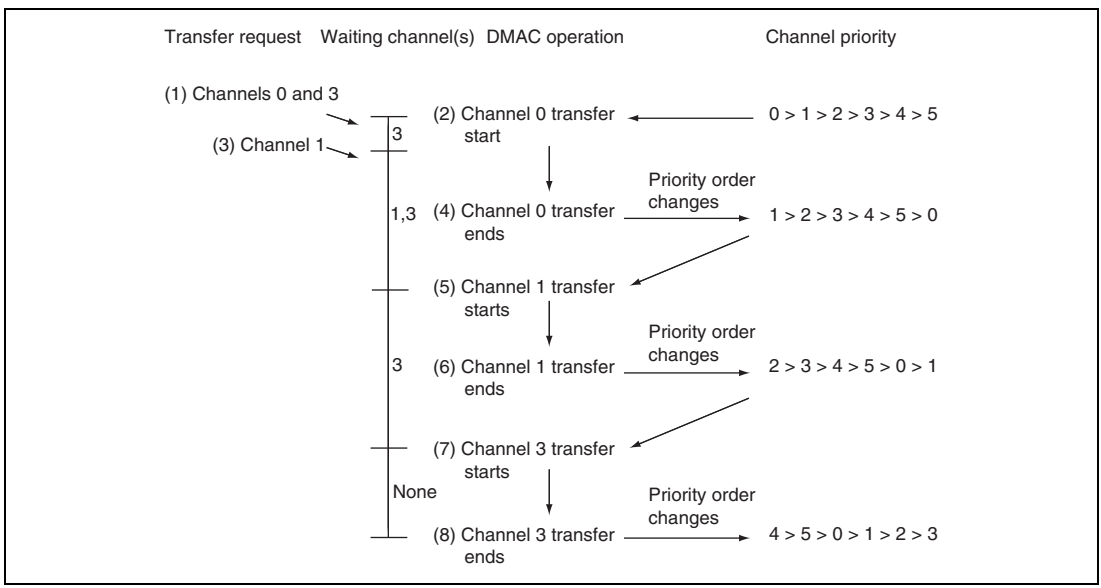


Figure 14.3 Changes in Channel Priority in Round-Robin Mode

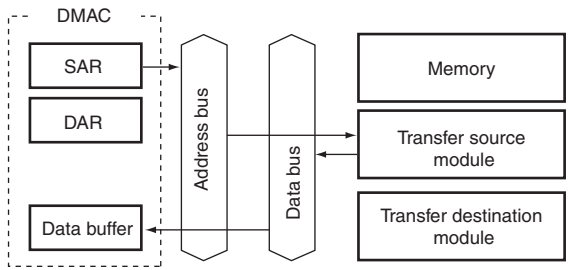
14.4.3 DMA Transfer Types

DMA transfer type is dual address mode transfer. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode.

(1) Dual Address Modes

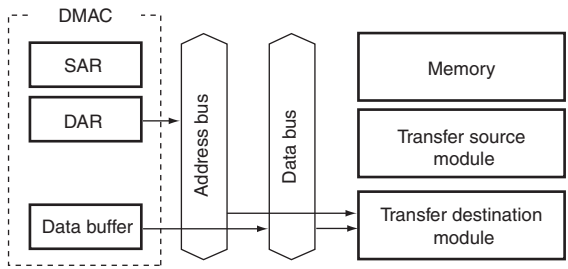
In dual address mode, both the transfer source and destination are accessed by an address. The source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 14.4, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle.



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

First bus cycle



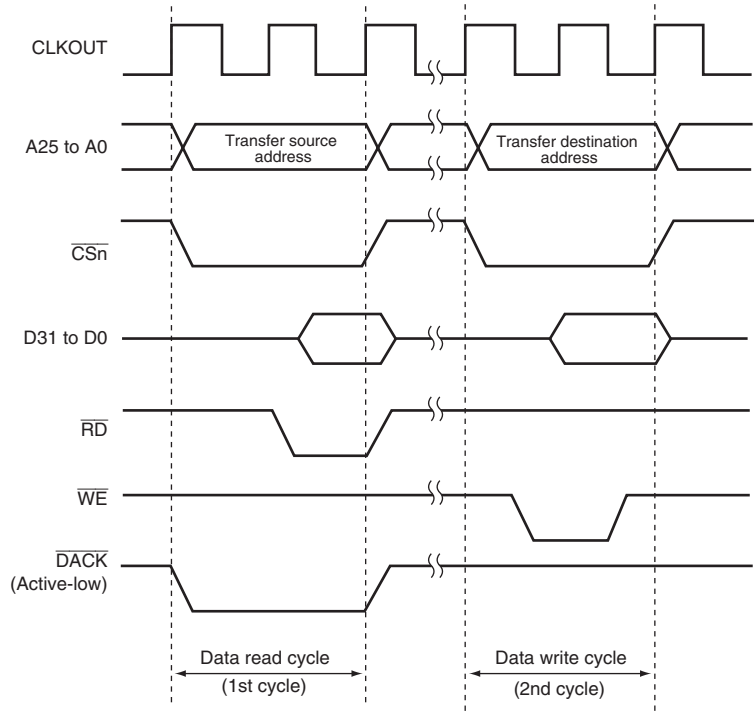
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 14.4 Data Flow of Dual Address Mode

Auto request, external request, and 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. CHCR can specify whether the DACK is output in read cycle or write cycle.

Figure 14.5 shows an example of DMA transfer timing in dual address mode.



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

Figure 14.5 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

(2) Bus Modes

There are two bus modes: cycle steal mode and burst mode. Select the mode in the TB and LCKN bits in CHCR. And cycle steal mode has normal and intermittent modes that are specified by the CMS bits in DMAOR.

- Cycle-Steal Mode
  - — Normal mode1 (DMAOR.CMS = 00, CHCR.LCKN = 0, CHCR.TB = 0)

    In cycle-steal normal mode, the SuperHyway bus mastership is given to another bus master after a one-transfer unit (byte, word, longword, 16-byte, or 32-byte unit) DMA transfer. When the next transfer request occurs, the DMAC issues the next transfer request, the bus mastership is obtained from the other bus master and a transfer is performed for one-transfer unit. When that transfer ends, the bus mastership is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

    In cycle-steal normal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination.

  Figure 14.6 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

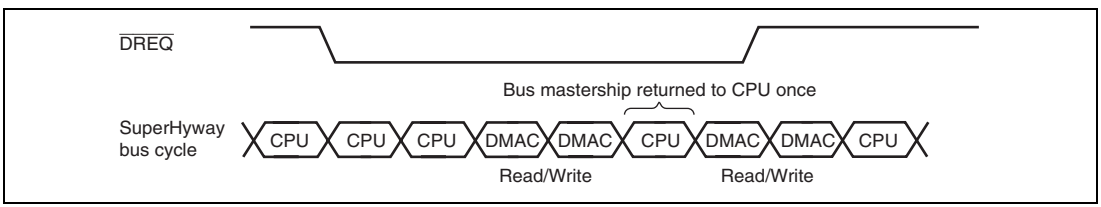


Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection)

  - — Normal mode 2 (DMAOR.CMS = 00, CHCR.LCKN = 1, CHCR.TB = 0)

    In cycle steal normal mode 2, the DMAC does not keep the SuperHyway bus mastership, is to obtain the bus mastership in every one transfer unit of read or write cycle.

    Figure 14.7 shows an example of DMA transfer timing in cycle steal normal mode 2.

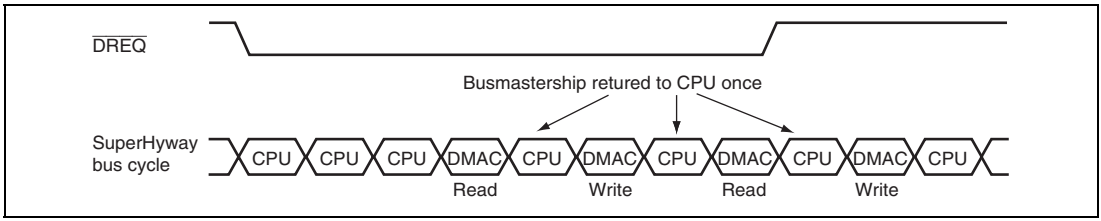


Figure 14.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection)

— Intermittent mode 16 (DMAOR.CMS = 10, CHCR.LCKN = 0 or 1, CHCR.TB = 0), intermittent mode 64 (DMAOR.CMS = 11, CHCR.LCKN = 0 or 1, CHCR.TB = 0)

In intermittent mode of cycle steal, the DMAC returns the SuperHyway bus mastership to other bus master whenever a one-transfer unit (byte, word, longword, or 16-byte or 32-byte unit) is complete. If the next transfer request occurs after that, the DMAC issues the next transfer request after waiting for 16 or 64 clocks in Bck count, and obtains the bus mastership from other bus master. The DMAC then transfers data of one-transfer unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than cycle-steal normal mode.

When the DMAC issues again the transfer request, DMA transfer can be postponed in case of entry updating due to cache miss.

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 14.8 shows an example of DMA transfer timing in cycle steal intermittent mode. Transfer conditions shown in the figure are:

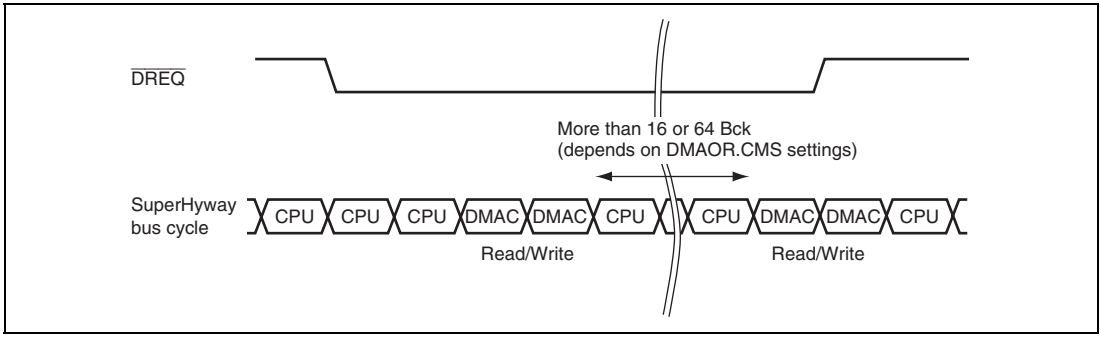


Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection)

- Burst Mode (LCKN = 0, TB = 1)

  In burst mode, once the DMAC obtains the SuperHyway bus mastership, the transfer is performed continuously without releasing the bus mastership until the transfer end condition is satisfied. In external request mode with level detection of the DREQ pin, however, when the DREQ pin is not active, the bus mastership passes to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

  Burst mode cannot be used when the on-chip peripheral module is the transfer request source.

Figure 14.9 shows DMA transfer timing in burst mode.

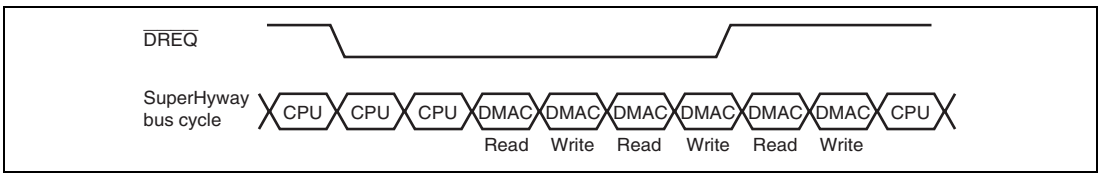


Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection)

(3) DMA Transfer Matrix

Table 14.10 shows the DMA transfer matrix in auto-request mode and table 14.11 shows the DMA transfer matrix in external request mode, and table 14.12 shows the on-chip peripheral module request.

Table 14.9 DMA Transfer Matrix in Auto-Request Mode (all channels)

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**On-chip peripheral module*[1]**	**L RAM**
LBSC space	Yes	Yes	Yes	Yes	Yes
DDRIF space	Yes	Yes	Yes	Yes	Yes
PCIC space	Yes	Yes	Yes	Yes	Yes
On-chip peripheral module*[1]	Yes	Yes	Yes	Yes	Yes
L RAM	Yes	Yes	Yes	Yes	Yes

[Legend]

Yes: Transfer is available.

Note: 1. When the transfer source or destination is on-chip peripheral module register, the transfer size should be the same value of its access size.

Table 14.10 DMA Transfer Matrix in External Request Mode (only channels 0 to 3)

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**On-chip peripheral module*[1]**	**L RAM**
LBSC space	Yes	Yes*[2]	Yes*[2]	Yes	Yes
DDRIF space	Yes*[3]	No	Yes*[4]	Yes*[3]	Yes*[3]
PCIC space	Yes*[3]	Yes*[5]	Yes*[5]	Yes*[3]	Yes*[3]
On-chip peripheral module*[1]	Yes	Yes*[2]	Yes*[2]	Yes	Yes
L RAM	Yes	Yes*[2]	Yes*[2]	Yes	Yes

[Legend]

Yes: Transfer is available.

No: Transfer is not available.

Notes: 1. When the transfer source or destination is on-chip peripheral module register, the transfer size should be the same value of its access size.

2. Transfer is available when the AM bit in CHCR is cleared to 0.

3. Transfer is available when the AM bit in CHCR is set to 1.

4. Transfer is available when the AM bit in CHCR is set to 1 and the destination address of the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).

5. Transfer is available when the AM bit in CHCR is cleared to 0 and the source address of the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).

6. Transfer is available when the source or destination, or both the source and destination address of the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).

   When the transfer source address is H'FD00 0000 to H'FDFF FFFF, the AM bit in CHCR is cleared to 0, when the transfer destination address is H'FD00 0000 to H'FDFF FFFF the AM bit in CHCR is set to 1.

Table 14.11 DMA Transfer Matrix in On-Chip Peripheral module Request Mode

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**On-chip peripheral module*[1]**	**L RAM**
LBSC space	No	No	No	Yes	No
DDRIF space	No	No	No	Yes	No
PCIC space	No	No	No	Yes	No
On-chip peripheral module*[1]	Yes	Yes	Yes	Yes	Yes
L RAM	No	No	No	Yes	No

[Legend]

Yes: Transfer is available.

No: Transfer is not available.

Note: 1. When the transfer source or the destination is an on-chip peripheral module, the transfer size should be the same value of its register access size.

The transfer source or the transfer destination should be a register of request source in on-chip peripheral module request mode. This transfer is available only cycle steal mode, and when the transfer request source is an on-chip peripheral module, the transfer is available in channel 0 to 5.

(4) Bus Mode and Channel Priority

When the priority is set in fixed mode (CH0 > CH1) and channel 1 is transferring in burst mode, if there is a transfer request to channel 0 with a higher priority, the transfer of channel 0 will begin immediately.

At this time, if channel 0 is also operating in burst mode, the channel 1 transfer will continue after the channel 0 transfer has completely finished.

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the transfer of one transfer unit and the channel 1 transfer is continuously performed without releasing the bus mastership. The bus mastership will then switch between the two in the order channel 0, channel 1, channel 0, and channel 1.

This example is shown in figure 14.9. When multiple channels are operating in burst modes, the channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership will not be given to the bus master until all competing burst transfers are complete.

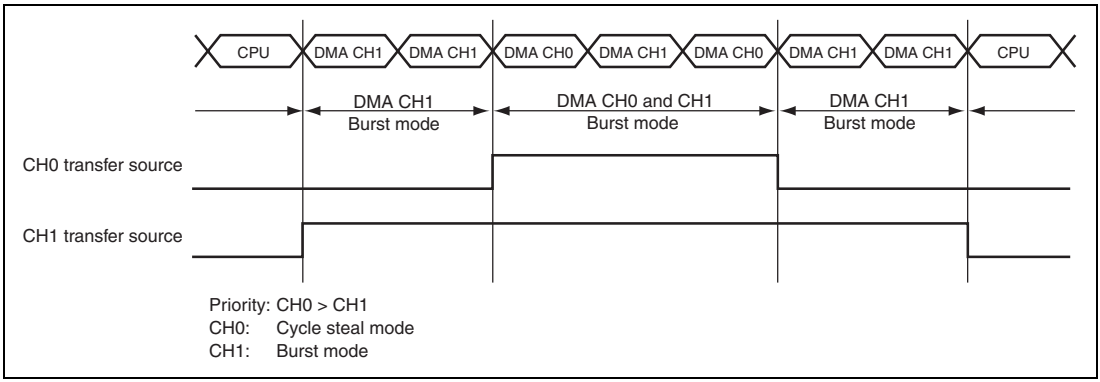


Figure 14.10 Bus State when Multiple Channels are Operating

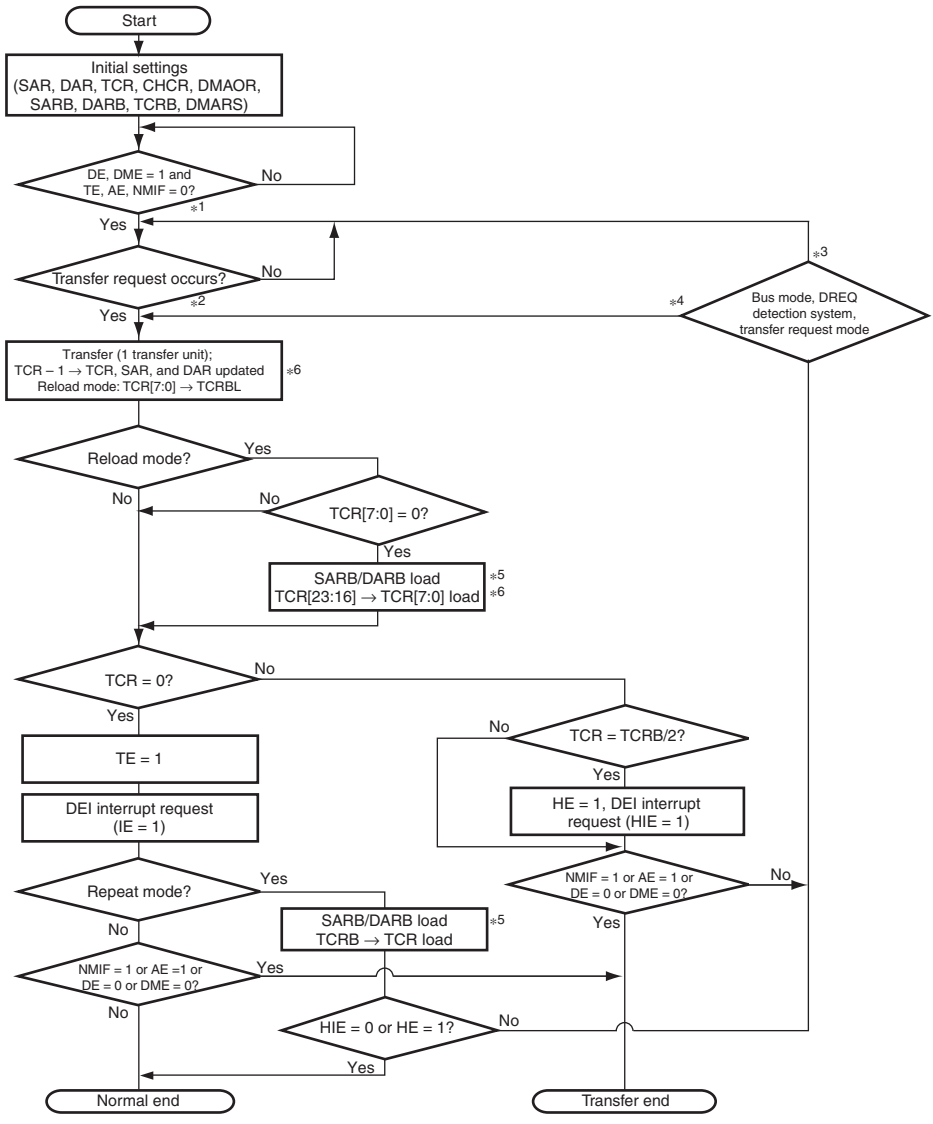
In round-robin mode, the priority changes according to the specification shown in figure 14.2. However, the channel in cycle steal mode cannot be mixed with the channel in burst mode.

14.4.4 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). In auto request mode, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a 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 14.11 shows a flowchart of this procedure.



Notes: 1. In repeat mode, a transfer request is acceptted with TE =1 when HIE = 1 and HE = 0 (half end interrupt is enable and clear the HE to 0 after HE is set to 1).
2. In auto-request mode, transfer starts when bits NMIF, AE, and TE are all 0 or bits TE and HIE are 1 and HE is 0 (in repeat mode), and bits DE and DME are set to 1.
3. DREQ is level detection (external requesrt) in burst mode or cycle-steral mode.
4. DREQ is edge detection (external request) or auto request in burst mode.
5. Loading to SAR and DAR differs according to the operating conditions in each mode.

Figure 14.11 DMA Transfer Flowchart

14.4.5 Repeat Mode Transfer

In a repeat mode transfer, a DMA transfer is repeated without specifying the transfer settings every time before executing a transfer.

Using a repeat mode transfer with the half end function allows a double buffer transfer executed virtually. Following processing can be executed effectively by using a repeat mode transfer. As an example, operation of receiving voice data from the VOICE CODEC and compressing it is explained.

In the following example, processing of compressing 40-word voice data every data reception is explained. In this case, it is assumed that voice data is received by means of SIOF.

1. DMAC settings
- Set address of the SIOF receive data register in SAR
- Set address of an internal memory data store area in DAR
- Set TCR to H'50 (80 times)
- Satisfy the following settings of CHCR
  Bits RPT[2:0] = B'010: Repeat mode (use DAR as a repeat area)
  Bit HIE = B'1: TCR/2 interrupt generated
  Bits DM[1:0] = B'01: DAR incremented
  Bits SM[1:0] = B'00: SAR fixed
  Bit IE = B'1: Interrupt enabled
  Bit DE = B'1: DMA transfer enabled
- Set such as bits TB and TS[2:0] according to use conditions
- Set bits CMS[1:0] and PR[1:0] in DMAOR according to use conditions and set the DME bit to B'1

2. Voice data is received and then transferred by SIOF/DMAC
3. TCR is decreased to half of its initial value and an interrupt is generated

After reading CHCR to confirm that the HE bit is set to 1 by an interrupt processing, clear the HE bit to 0 and compress 40-word voice data from the address set in DAR.

4. TCR is cleared to 0 and an interrupt is generated
   After reading CHCR to confirm that the TE bit is set to 1 by an interrupt processing, clear the TE bit to 0 and compress 40-word voice data from the address set in DAR + 40. After this operation, the value of DARB is copied to DAR in DMAC and initialized, and the value of TCRB is copied to TCR and initialized to 80.

5. Hereafter, steps 2 and 4 are repeated until the DME or DE bit is cleared to 0, or an NMI interrupt is generated. Note that if the HE bit is not cleared in the procedure 3 or if the TE bit is not cleared in the procedure 4, then the transfer is stopped according to the condition of both the HE and the TE bits are set to 1.

As explained above, a repeat mode transfer enables sequential voice compression by changing buffer for storing data received consequentially and a data buffer for processing signals alternately.

14.4.6 Reload Mode Transfer

In a reload mode transfer, according to the settings of bits RPT[2:0] in CHCR, the value set in SARB/DARB is set to SAR/DAR and the value of bits TCRB[23:16] is set in bits TCRB[7:0] at each transfer set in the bits TCRB[7:0], and the transfer is repeated until TCR becomes 0 without specifying the transfer settings again. A reload mode transfer is effective when repeating data transfer with specific area. Figure 14.12 shows the operation of reload mode transfer.

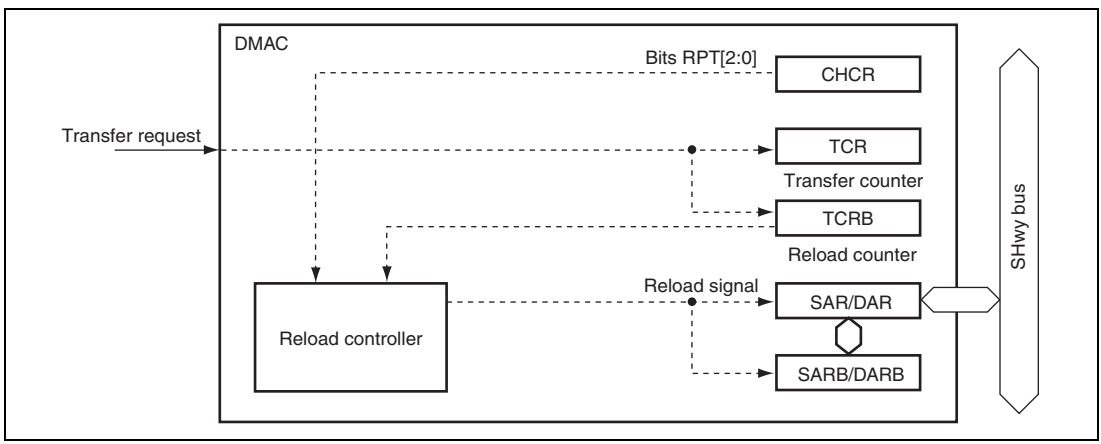


Figure 14.12 Reload Mode Transfer

When a reload mode transfer is executed, TCRB is used as a reload counter. Set TCRB according to section 14.3.6, DMA Transfer Count Registers (TCRB0 to TCRB3).

14.4.7 DREQ Pin Sampling Timing

Figures 14.13 to 14.16 show the sample timing of the DREQ input in each bus mode, respectively.

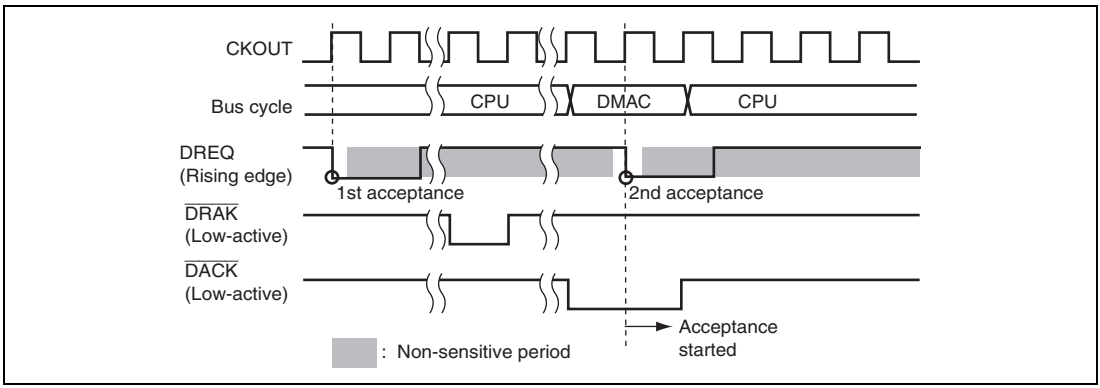


Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

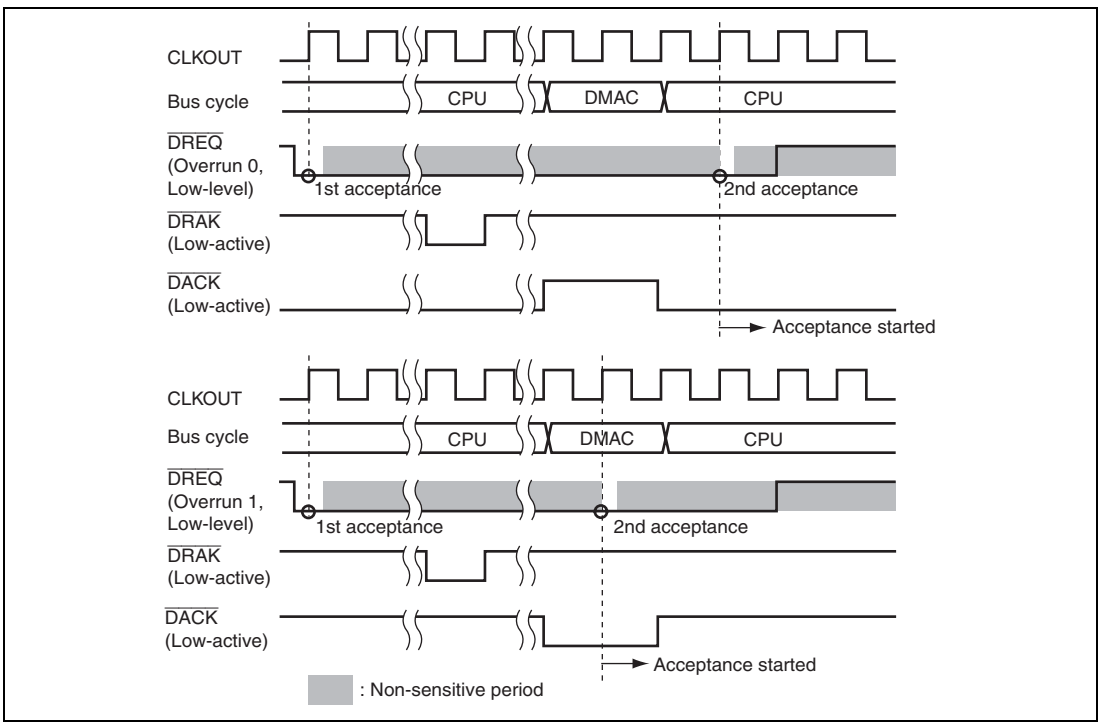


Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

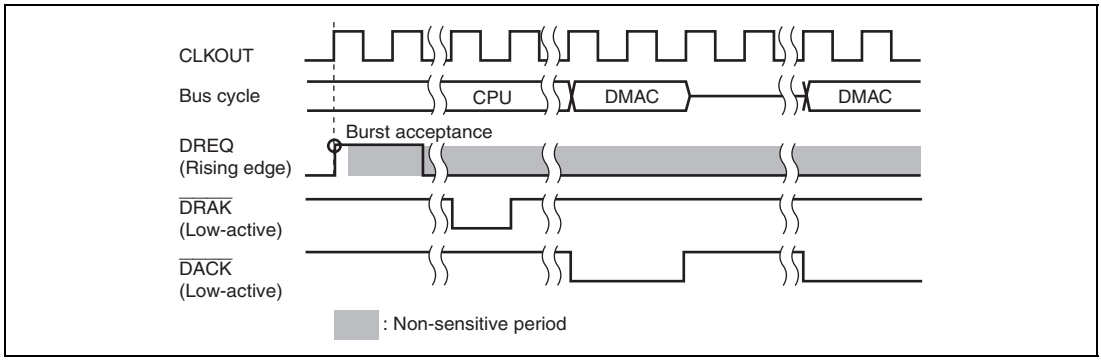


Figure 14.15 Example of DREQ Input Detection in Burst Mode Edge Detection

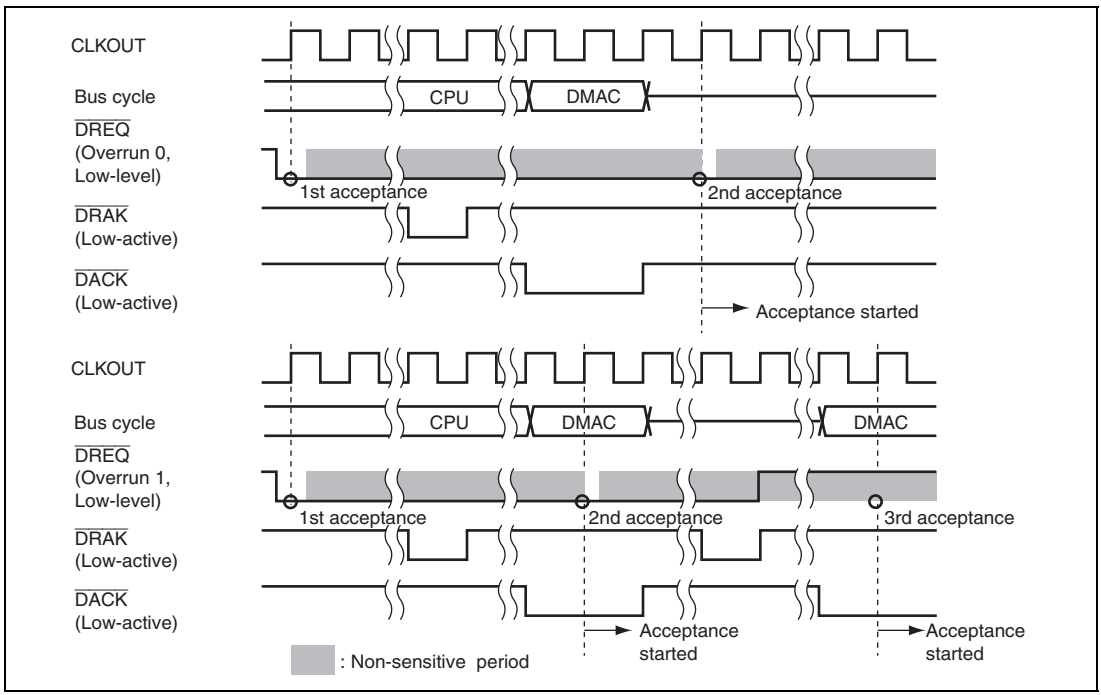


Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 14.17 shows the timing of the TEND output.

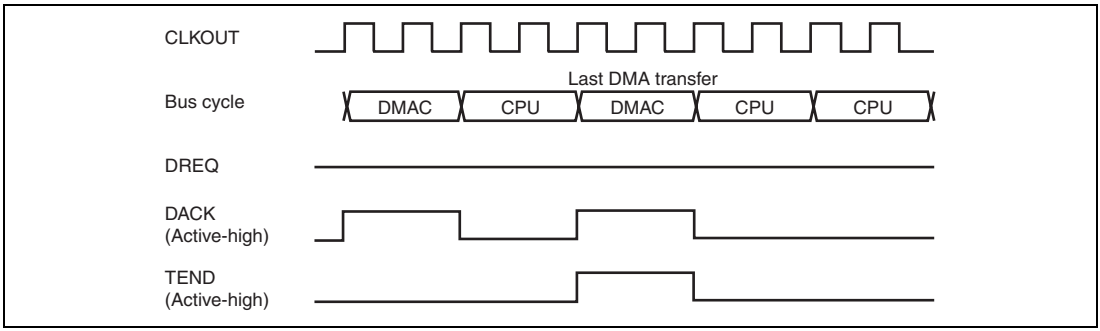
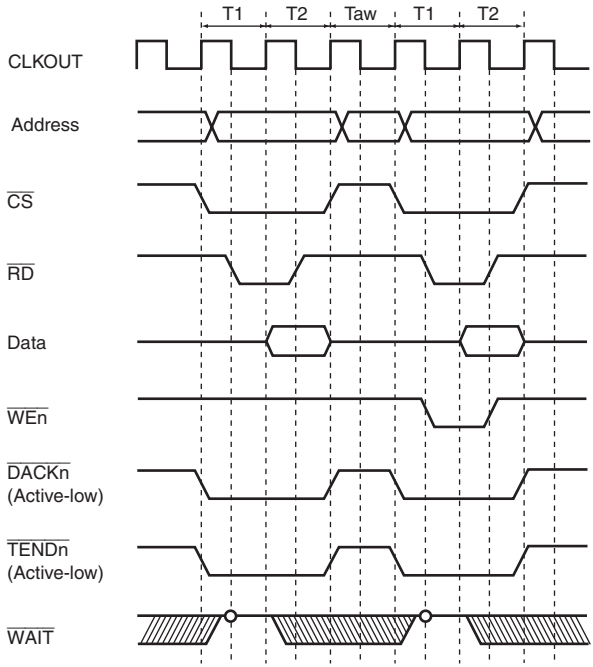


Figure 14.17 DMA Transfer End Signal (Cycle Steal Mode Level Detection)

Note that the DACK output and TEND output are divided to align the data 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. This example is shown in figure 14.18.



Note: TEND is asserted during the last transfer unit of the DMA transfer.
When the transfer unit is divided into several bus cycles and CS is negated between bus cycles, TEND is also divided.

Figure 14.18 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

14.5 Usage Notes

Pay attentions to the following notes when the DMAC is used.

14.5.1 Module Stop

While DMAC is in operation, modules should not be stopped by setting MSTPCR (transition to the module standby state). When modules are stopped, transfer contents cannot be guaranteed.

14.5.2 Address Error

When a DMA address error is occurred, after execute the following procedure, and then start a transfer.

1. Dummy read for the below listed registers.
   BCR (LBSC)
   PCIECR (PCIC)
   MIM (DDRIF)
   INTCB3 (INTC)
2. Issue the SYNCO instruction.
3. Set registers of all channels again.
   If the AE bit in DMAOR is set to 1, channels 0 to 5 should be set again.

14.5.3 Notes on Burst Mode Transfer

During a burst mode transfer, following operation should not be executed until the transfer of corresponding channel has completed.

- Frequency should not be changed.
- Transition to sleep mode should not be made.

14.5.4 DACK and TEND Output Divisions

The DACK and TEND output are divided to align the data unit like the $\overline{CSn}$ output when a DMA transfer unit is divided with multiple bus cycles, for example when an 8-bit or 16-bit external device is accessed in longword units, or when an 8-bit external device is accessed in word units, and the $\overline{CSn}$ output is negated between these bus cycles.

14.5.5 $\overline{CS}$ Output Settings and Transfer Size Larger than External Bus Width

When one DMA transfer is performed by multiple bus cycles*[1], the $\overline{CSn}$ output should be set not to negate between bus cycles*[2]. For detail of settings, refer to tables 11.9 to 11.14. If set the $\overline{CSn}$ output is negated between bus cycles, the DREQ signal is not sampled correctly and malfunction may occur.

Notes: 1. When a DMA transfer is performed with larger transfer size than the bus width. For example, performing the 16-/32-byte transfer to the 8-/16-/32-bit bus width LBSC space, longword (32-bit) transfer to the 8-/16-bit bus width LBSC space, or word (16-bit) transfer to the 8-bit bus width LBSC space. Note that except for a 32-bit access to the MPX interface. This access generates only one bus cycle (burst).

2. When the $\overline{CSn}$ output is negated between bus cycles, then the DACK output is also negated between bus cycles (DACK output is also divided).

14.5.6 DACK and TEND Assertion and DREQ Sampling

The DACK and TEND signals may be asserted ceaselessly during two or more times DMA transfer when the DREQ level detection with overrun 1 and the DREQ edge detection. In this case, the DMA transfer is suspended and do not perform correctly, to avoid this insert one or more idle cycle between the DMA transfer.

The transfer source is the LBSC space and the DACK and TEND are output during the read cycle:

(1) Set B'001 to B'111 (i.e., other than 000) to the IWRRD bits in CSnBCR

(2) Set B'001 to B'111 (i.e., other than 000) to the IWRRS bits in CSnBCR

The transfer destination is the LBSC space and the DACK and TEND are output during the write cycle:

(1) Set B'001 to B'111 (i.e., other than 000) to the IWW bits in CSnBCR

Note: * The transfer source is the LBSC space and the DACK is output during the read cycle or the transfer destination is the LBSC space and the DACK is output during the write cycle. And then specifies no idle cycle (CSnBCR.IWRRD, IWRRS, IWW are cleared to B'000). Note that the case that both the transfer source and the transfer destination are the LBSC spaces, does not apply this.

Tables 14.12 to 14.15 show the number of the bus cycles generated in each DMA transfer and the register settings for the LBSC space. With these settings, $\overline{\text{CSn}}$ is not negated even if multiple bus cycles are generated.

Note that, in the following settings, when either the transfer source or the transfer destination is the LBSC space, to avoid the DACK is asserted ceaselessly during between the two or more times DMA transfer, set B'001 to B'111 to the IWRRD, IWRRS or IWW bits in CSnBCR. In this setting, if the 16-byte DMA transfer is performed, multiple bus cycles are generated and the $\overline{\text{CSn}}$ is negated between bus cycles, the DREQ signal is not sampled correctly and malfunction may occur.

Table 14.12 Register Setting for SRAM, Burst ROM, Byte Control SRAM Interface.

			Register Setting of $\overline{\text{CSn}}$ is not negated	
Bus Width [bit]	**DMA Transfer Access Size**	**Bus Cycle Number**	**CSnBCR.IWRRD, IWRRS or IWW**	**CSnWCR.ADS and ADH**
8	Byte	1	Any	Any
	Word	2	Any	B'000
	Longword	4	Any	B'000
	16-Byte	16	B'000	B'000
	32-Byte	32	Any	B'000
16	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	2	Any	B'000
	16-Byte	8	B'000	B'000
	32-Byte	16	Any	B'000
32	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	1	Any	Any
	16-Byte	4	B'000	B'000
	32-Byte	8	Any	B'000

Table 14.13 Register Setting for PCMCIA Interface

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Setting of CSn is not negated
			CSnWCR.ADS and ADH
8	Byte	1	Any
	Word	2	Any
	Longword	4	Any
	16-Byte	16	B'000
	32-Byte	32	Any
16	Byte	1	Any
	Word	1	Any
	Longword	2	Any
	16-Byte	8	B'000
	32-Byte	16	Any

Table 14.14 Register Setting for MPX Interface (Read Access)

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Setting of CSn is not negated
			CSnWCR.ADS and ADH
32	Byte	1	Any
	Word	1	Any
	Longword	1	Any
	16-Byte	4	Impossible (Negated)
	32-Byte	1	Any

Table 14 15 Register Settings for MPX Interface (Write Access)

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Setting of CSn is not negated	
			CSnBCR.IWW	CSnWCR.IW[1:0]
32	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	1	Any	Any
	16-Byte	4	B'000	B'11 to B'01
	32-Byte	1	Any	Any

14.5.7 DMA Transfer to DMAC Prohibited

Do not perform DMA transfer with the DMAC register specified as the transfer source or transfer destination.

14.5.8 NMI Interrupt

When an NMI interrupt occurs, the DMA transfer is stopped. After returning from the NMI interrupt routine, set all channels again, and then restart the DMA transfer.

Section 15 External CPU Interface (EXCPU)

The DDR-SDRAM space in this LSI and internal registers of this LSI can be accessed by a CPU externally connected to the LSI (hereinafter, simply referred to as “external CPU”). Access by an external CPU is implemented using the MPX protocol.

15.1 Features

- Control of access from an external CPU

  An external CPU is able to access the DDR-SDRAM space and internal registers of this LSI.

  The MPX protocol is used for access from an external CPU.

  The SH7750 Group and the SH7751 Group can be connected as the external CPU via the buses of the SH7750 Group and the SH7751 Group.

Figure 15.1 shows a block diagram of the EXCPU.

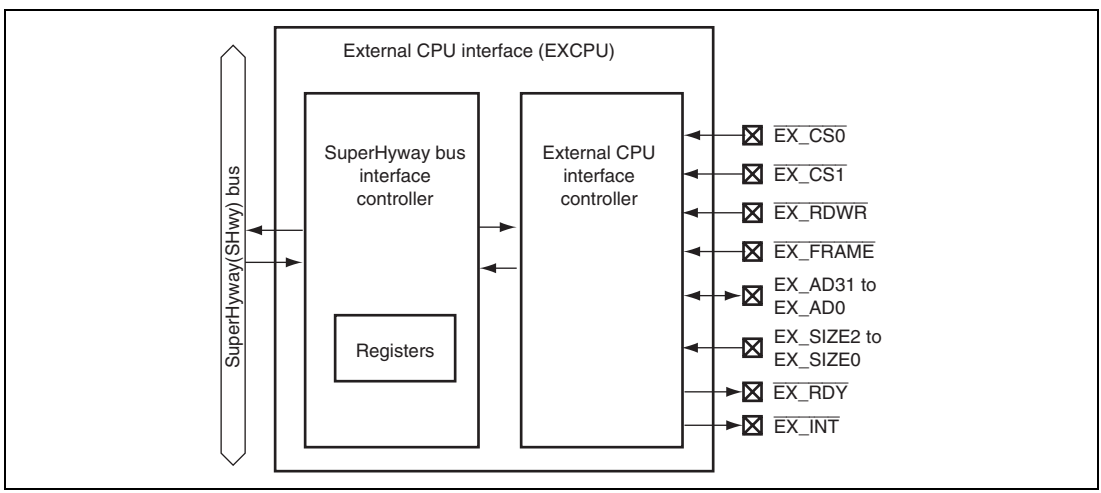


Figure 15.1 EXCPU Block Diagram

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the EXCPU.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Description
$\overline{\text{EX_CS0}}$	Chip select 0	Input	Indicates access to the DDR-SDRAM space
$\overline{\text{EX_CS1}}$	Chip select 1	Input	Indicates access to an internal register of this LSI
$\overline{\text{EX_BS}}$	Bus cycle start	Input	Indicates an address phase
$\overline{\text{EX_FRAME}}$	Access cycle	Input	Indicates an access cycle period
$\overline{\text{EX_RDWR}}$	Read/write	Input	Indicates whether it is data write or read
EX_SIZE2 to EX_SIZE0	Access size	Input	Indicates the access size Note: These pins are used in the SH7750 Group. In the SH7751 Group, pins D31 to D29 act as access size signals.
EX_AD0 EX_AD31	Address/data	Input/ output	During an address phase, signals on EX_AD25 to EX_AD0 are input as an address During a data phase, signals on EX_AD25 to EX_AD0 are input as data
$\overline{\text{EX_RDY}}$	Ready	Output	Wait state request signal
$\overline{\text{EX_INT}}$	External CPU interrupt	Output	Interrupt signal
MD10	Mode control	Input	External CPU connection select

15.3 Register Descriptions

Table 15.2 shows the EXCPU register configuration. Table 15.3 shows the register states in each operating mode.

Table 15.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
External CPU control register	EXCCTRL	R/W	H'FE40 000C	H'1E40 000C	32
External CPU memory space select register	EXCMSETR	R/W	H'FE40 0010	H'1E40 0010	32
External CPU output interrupt control register	EXCINOR	R/W	H'FE40 0014	H'1E40 0014	32

Table 15.3 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
External CPU control register	EXCCTRL	H'0000 0000	H'0000 0000	Retained	Retained
External CPU memory space select register	EXCMSETR	H'0000 0000	H'0000 0000	Retained	Retained
External CPU output interrupt control register	EXCINOR	H'0000 0000	H'0000 0000	Retained	Retained

15.3.1 External CPU Control Register (EXCCTRL)

EXCCTRL indicates whether an external CPU is connected and sets the type of the external CPU.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	EXCMD	Undefined*	R	External CPU Connection Indicator Indicates the state of the MD10 pin. 0: External CPU is not connected 1: External CPU is connected
0	EXCSEL	0	R/W	External CPU Type Selects the type of the CPU to be connected 0: SH7751 Group 1: SH7750 Group

Note: The initial value depends on the state of the MD10 pin.

15.3.2 External CPU Memory Space Select Register (EXCMSETR)

EXCMSETR sets the base address used when the internal memory space of this LSI is accessed by the external CPU.

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

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

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 to 0	EXCMSET	0000	R/W	Internal Memory Space Base Address These bits set the base address of the memory space used for access to the internal memory by the external CPU. The address of the memory space to be accessed is comprised as follows: 31st and 30th bits: Fixed at B'01 29 to 26th bits: EXCMSET[3:0] 25 to 0th bits: Access address from the external CPU The address range of the memory space accessible to the external CPU is from H'4000 0000 to H'7FFF FFFF.

15.3.3 External CPU Interrupt Output Control Register (EXCINOR)

EXCINOR is used to generate an interrupt to the external CPU from this LSI.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EXCINI	0	R/W	Notification of Interrupt to External CPU 1: Asserts the $\overline{\text{EX_INT}}$ pin to generate an interrupt to the external CPU 0: Negates the $\overline{\text{EX_INT}}$ pin to clear an interrupt to the external CPU

15.4 Operation

With this LSI, a CPU externally connected to the LSI (an external CPU) is allowed to access the DDR-SDRAM space or internal registers of the LSI by using the MPX protocol.

The external CPU becomes ready to access the space in this LSI after this sequence: an access request ($\overline{\text{BREQ}}$) from the external CPU is accepted by the LBSC, the local bus is released, and an access acknowledgement ($\overline{\text{BACK}}$) is returned to the external CPU.

The EXCPU determines whether the access is to the DDR-SDRAM space or to an internal register according to the CS signals ($\overline{\text{EX_CS0}}$, $\overline{\text{EX_CS1}}$) from the external CPU and performs processing for the respective access.

In the case of access to the DDR-SDRAM space, the EXCPU implements access to the DDR-SDRAM space in this LSI by converting the signal from the external CPU from the MPX protocol to the SuperHyway bus protocol. In this process, data alignment conversion is performed with the same endian as this LSI according to the access size from the external CPU.

(1) Space Accessible to the External CPU

The DDR-SDRAM space and internal registers of this LSI are accessible to the external CPU. The space to be accessed is selected as shown below using the CS signals.

- $\overline{\text{EX_CS0}}$: DDR-SDRAM space (64 Mbytes)
- $\overline{\text{EX_CS1}}$: Internal registers of this LSI

For DDR-SDRAM space access, however, the size of the space that can be accessed by the external CPU is 64 Mbytes while the entire DDR-SDRAM space in this LSI is 512 Mbytes. So, access to the entire DDR-SDRAM space from the external CPU is enabled by the window method. To access the entire DDR-SDRAM space in this LSI, first designate a 64-Mbyte access space by the EXCMSETR register of the EXCPU, and then create an access to the DDR-SDRAM space.

(2) Data Alignment Conversion for the External CPU

For the external CPU, the EXCPU performs data alignment conversion with the same endian as this LSI. This conversion supports both big endian, where the upper byte is placed at the smaller address, and little endian, where the lower byte is placed at the smaller address. Endian selection is done at a power-on reset by means of an external pin.

Table 15.4 Access and Data Alignment for Little Endian

	EX_AD31 to EX_AD24	EX_AD23 to EX_AD16	EX_AD15 to EX_AD8	EX_AD7 to EX_AD0
Byte access to address 0				Data 7 to data 0
Byte access to address 1			Data 7 to data 0	
Byte access to address 2		Data 7 to data 0		
Byte access to address 3	Data 7 to data 0			
Byte access to address 4				Data 7 to data 0
Byte access to address 5			Data 7 to data 0	
Byte access to address 6		Data 7 to data 0		
Byte access to address 7	Data 7 to data 0			
Word access to address 0			Data 15 to data 8	Data 7 to data 0
Word access to address 2	Data 15 to data 8	Data 7 to data 0		
Word access to address 4			Data 15 to data 8	Data 7 to data 0
Word access to address 6	Data 15 to data 8	Data 7 to data 0		
Longword access to address 0	Data 31 to data 24	Data 23 to data 16	Data 15 to data 8	Data 7 to data 0
Longword access to address 4	Data 31 to data 24	Data 23 to data 16	Data 15 to data 8	Data 7 to data 0

Table 15.5 Access and Data Alignment for Big Endian

	EX_AD31 to EX_AD24	EX_AD23 to EX_AD16	EX_AD15 to EX_AD8	EX_AD7 to EX_AD0
Byte access to address 0	Data 7 to data 0			
Byte access to address 1		Data 7 to data 0		
Byte access to address 2			Data 7 to data 0	
Byte access to address 3				Data 7 to data 0
Byte access to address 4	Data 7 to data 0			
Byte access to address 5		Data 7 to data 0		
Byte access to address 6			Data 7 to data 0	
Byte access to address 7				Data 7 to data 0
Word access to address 0	Data 15 to data 8	Data 7 to data 0		
Word access to address 2			Data 15 to data 8	Data 7 to data 0
Word access to address 4	Data 15 to data 8	Data 7 to data 0		
Word access to address 6			Data 15 to data 8	Data 7 to data 0
Longword access to address 0	Data 31 to data 24	Data 23 to data 16	Data 15 to data 8	Data 7 to data 0
Longword access to address 4	Data 31 to data 24	Data 23 to data 16	Data 15 to data 8	Data 7 to data 0

(3) Timing Charts of External CPU Access

External CPU access through the EXCPU is done through handshaking of the access request ($\overline{\text{BREQ}}$) and access acknowledge ($\overline{\text{BACK}}$) signals.

Figures 15.2 and 15.3 show the access timing of the EXCPU and external CPU.

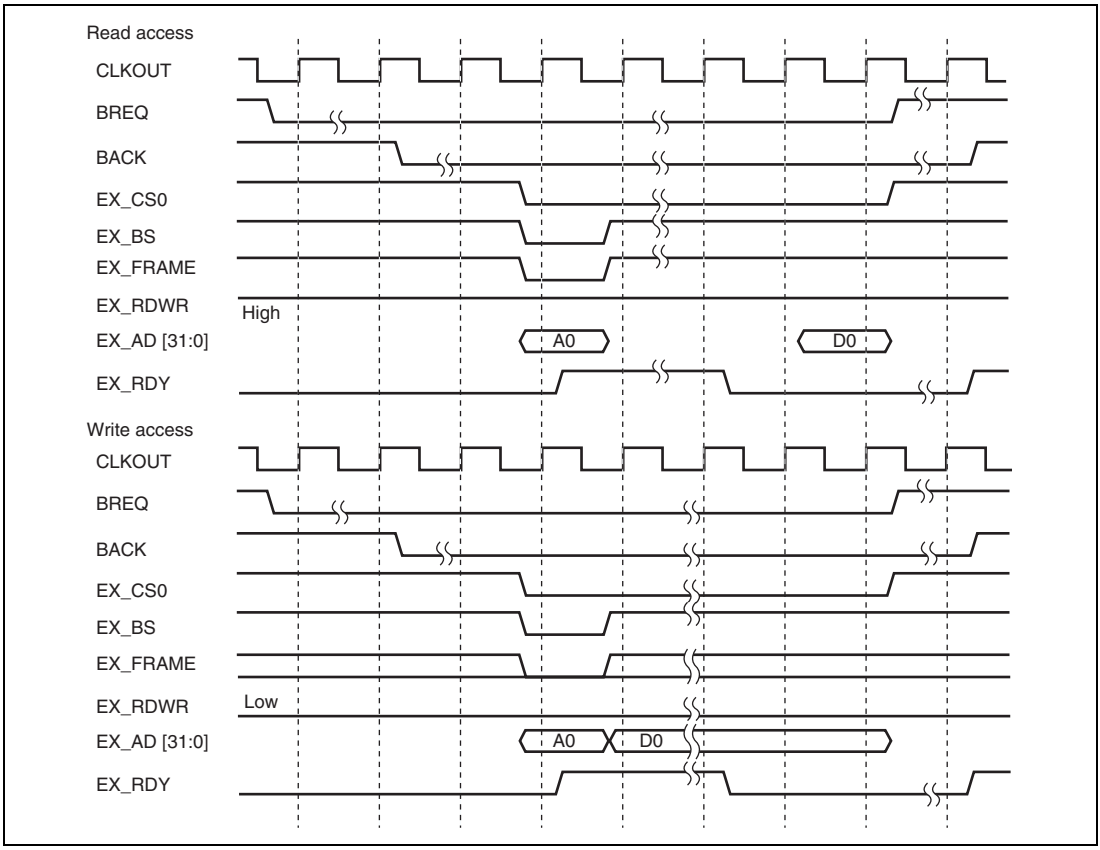


Figure 15.2 External CPU Access (Single Access)

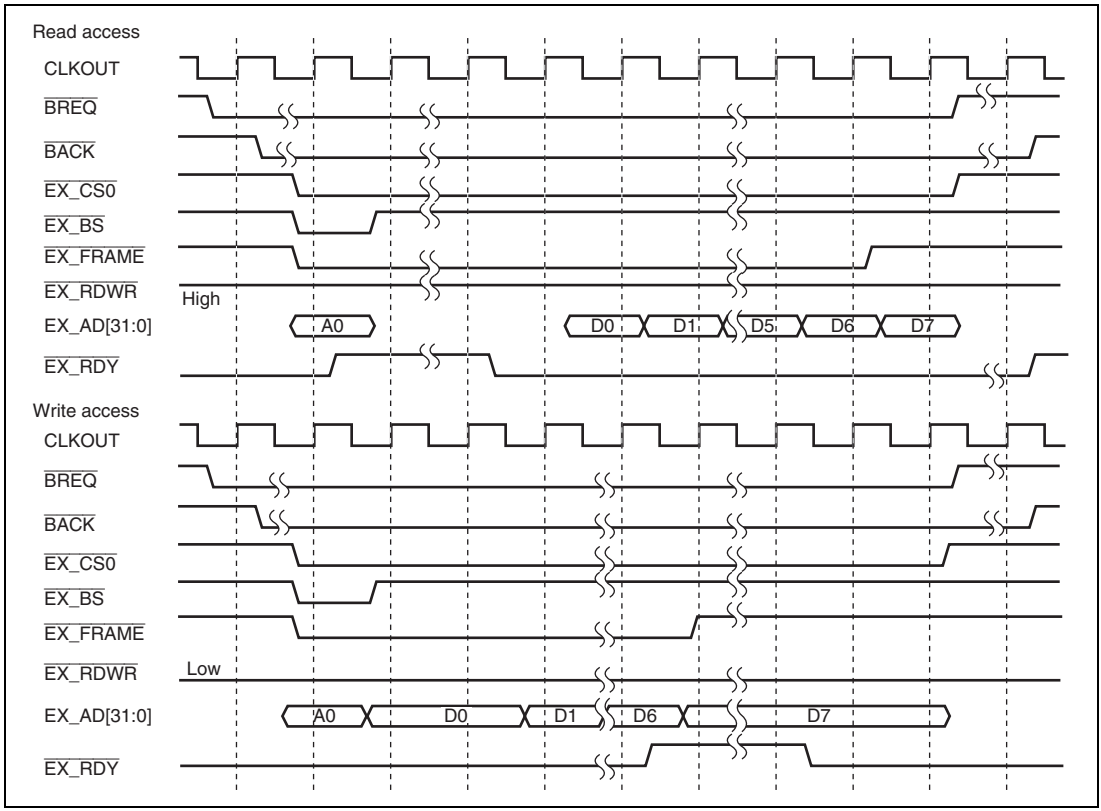


Figure 15.3 External CPU Access (Burst Access)

(4) Configuration of Connection to the External CPU

Figure 15.4 shows the configuration of the connection between the external CPU and this LSI.

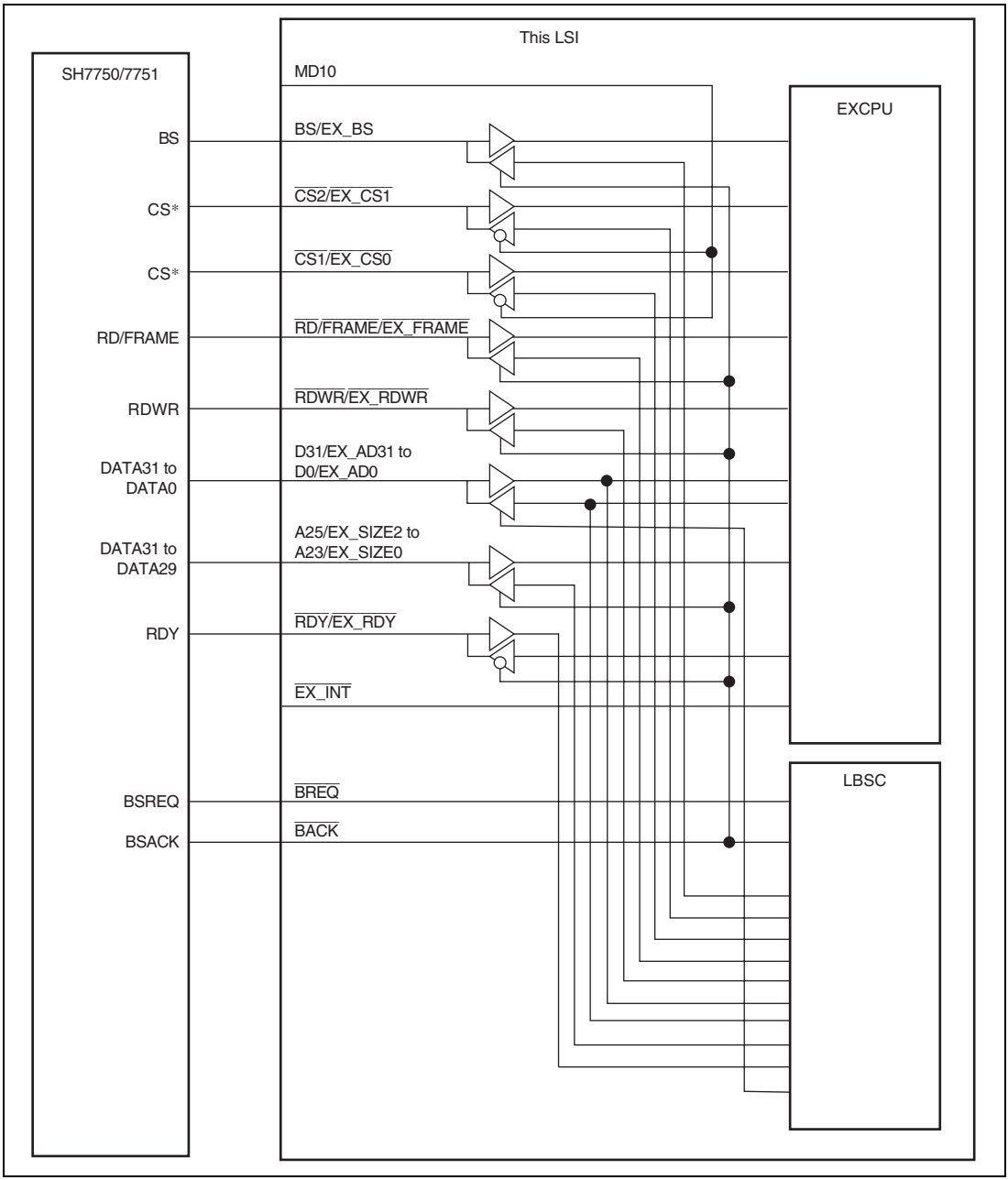


Figure 15.4 Configuration of Connection with External CPU

Section 16 Clock Pulse Generator (CPG)

The CPG generates clocks provided to the on-chip peripheral modules and external bus interface of this LSI, and controls the power-down mode function. The CPG consists of an oscillator, PLL circuits, frequency dividers, and control circuits.

16.1 Features

- Clocks used for LSI internal operation

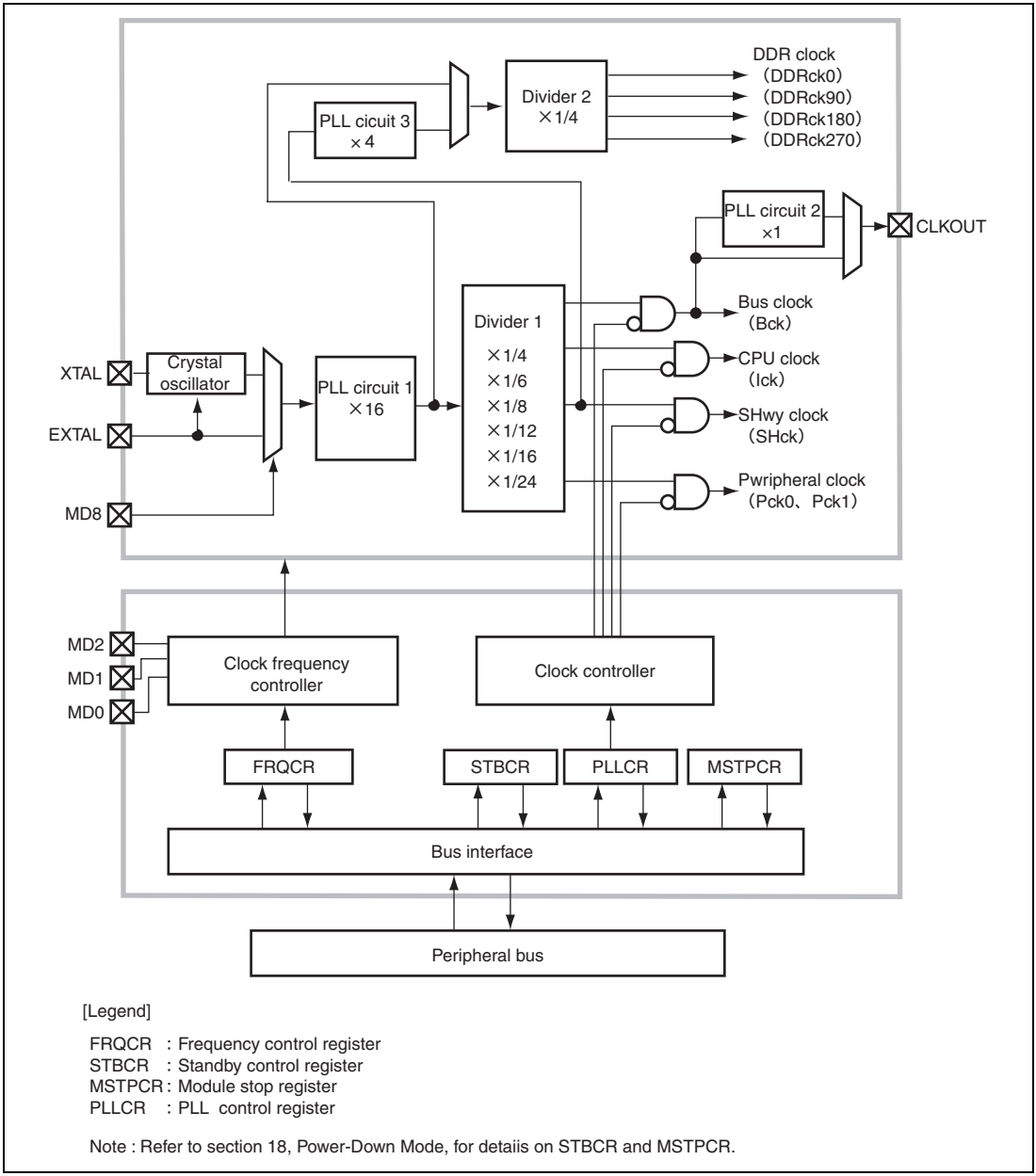
  Generates the CPU clock (Ick) used by the CPU, FPU, cache, and TLB, SHwy clock (SHck) used by the SuperHyway, and peripheral clocks (Pck0, Pck1) supplied to the peripheral modules.
- Clocks supplied to outside modules

  Generates the bus clock (Bck) used by the external bus interface and the memory clocks (DDRck) used by the DDR interface.
- Clock modes

  Either a crystal resonator or an externally input clock can be selected as the CPG clock input. The combination of the division ratios for the CPU clock, SHwy clock, bus clock, peripheral clock, and DDR-memory clock after a power-on reset can be selected from two clock operating modes.
- Power-down mode control

  The clock can be stopped for sleep mode and software standby mode, and specific modules can be stopped in module standby mode.

A block diagram of the CPG is shown in figure 16.1.



[Legend]

FRQCR : Frequency control register
STBCR : Standby control register
MSTPCR : Module stop register
PLLCR : PLL control register

Note : Refer to section 18, Power-Down Mode, for detaiis on STBCR and MSTPCR.

Figure 16.1 Block Diagram of CPG

The functions of the blocks in the CPG are as follows.

(1) PLL Circuit 1

PLL circuit 1 multiples the frequency of the crystal oscillator or the clock input from the EXTAL pin by the ratio of ×16. The multiplication ratio is selected by the combination of mode control pins MD0, MD1, and MD2.

(2) PLL Circuit 2

PLL circuit 2 aligns the phases of the bus clock (Bck) and the clock signal output from the CLKOUT pin that is used by the external peripheral interface.

(3) Crystal Oscillator

The crystal oscillator is a clock pulse generator used when a crystal resonator is connected to the XTAL or EXTAL pin. The crystal oscillator can be enabled by the MD8 pin setting.

(4) Divider 1

Divider 1 generates the CPU clock (Ick), SHwy clock (SHck), peripheral module clocks (Pck0, Pck1), and bus clock (Bck). The division ratio is selected by the combination of mode control pins MD0, MD1, and MD2.

(5) Frequency Control Register (FRQCR)

The frequency control register is a read-only register that depends on the combination of mode control pins MD0, MD1, and MD2.

(6) PLL Circuit 3

PLL circuit 3 multiples the frequency of the SHwy clock (SHck) by the ratio of ×4.

(7) Divider 2

Divider 2 generates the DDR-memory clocks (DDRck0, DDRck90, DDRck180, and DDRck270).

(8) PLL Control Register (PLLCR)

The PLL control register has control bits assigned for enabling or disabling the CLKOUT pin output.

(9) Module Stop Registers 0, 1(MSTPCR0 and MSTPCR1)

The module stop registers have control bits for running/stopping the individual peripheral modules.

(10) Standby Control Register (STBCR)

The standby control register has bits for controlling the power-down modes.

16.2 Input/Output Pins

Table 16.1 lists the CPG pin configuration.

Table 16.1 Pin Configuration and Functions of CPG

Pin Name	Function	I/O	Description
MD0	Mode control pins 0, 1, 2 (Clock operating mode)	Input	Sets the clock operating mode after a power-on reset.
MD1		Input	
MD2		Input	
MD8	Mode control pin 8 (Clock input mode)	Input	Selects the use of the crystal resonator. MD8 = low: External clock is input from the EXTAL pin. MD8 = high: Crystal resonator is connected to the EXTAL and XTAL pins.
XTAL	Clock pins	Output	A crystal resonator is connected.
EXTAL		Input	A crystal resonator is connected, or an external clock is input.
CLKOUT		Output	Used as an external bus clock output pin.

Note: For the guaranteed AC timing of the CLKOUT pin, refer to the section on electrical characteristics. Pay attention to the relationship between the input frequency of the crystal oscillator and the multiplication ratio.

16.3 Clock Operating Mode

Table 16.2 shows the relationship between the mode control pin (MD0, MD1, and MD2) combinations and the clock operating mode after a power-on reset.

Table 16.2 Clock Operating Modes

Clock operating mode	External pin combination*1			PLL 1	PLL 2	PLL 3	EXTAL frequency (MHz)		Clock generated by CPG						Initial value of FRQCR
	MD2	MD1	MD0						Ick	SHck	Bck	Pck0	Pck1	DDRck	
0	0	0	0	ON	ON	OFF	25 to 33.1	Frequency ratio*2	8	4	2	2	1	4	H'1013 0035
								Max. frequency	266	133	66.6	66.6	33.3	133	

Notes: 1. Mode pin (MD0, MD1, and MD2) combinations other than above are prohibited.
2. The ratio of the frequency of each clock to that of the crystal oscillator or the clock input from the EXTAL pin.

16.4 Register Descriptions

Table 16.3 shows the CPG register configuration. Table 16.4 shows the register states in each operating mode.

Table 16.3 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Frequency control register	FRQCR	R	H'FFC8 0000	H'1FC8 0000	32
PLL control register	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32

Table 16.4 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Standby	Sleep
Frequency control register	FRQCR	H'1013 0035	Retained	Retained	Retained
PLL control register	PLLCR	H'0000 0001	Retained	Retained	Retained

16.4.1 Frequency Control Register (FRQCR)

FRQCR is a 32-bit read-only register used to confirm the division ratios for the CPU clock (Ick), SHwy clock (SHck), peripheral clocks (Pck0, Pck1), and the bus clock (Bck) after a power-on reset. For the frequency ratios, refer to table 16.2, Clock Operating Modes. This register can be accessed only in longwords. Operation cannot be guaranteed if this register is written to.

FRQCR is only initialized by a power-on reset caused by the $\overline{\text{PRESET}}$ pin or watchdog timer overflow.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CFC[2:0]			—	BFC[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	P0FC[2:0]			—	P1FC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	0001	R	Reserved These bits are read as B'0001.
27 to 23	—	All 0	R	Reserved These bits are always read as all 0.
22 to 20	CFC[2:0]	001	R	CPU Clock (Ick) and SHwy Clock (SHck) Frequency Division Ratios CFC[2:0] Ick SHck 001: ×1/2 ×1/4
19	—	0	R	Reserved This bit is always read as 0.
18 to 16	BFC[2:0]	011	R	Bus Clock 0 (Bck) Frequency Division Ratio 011: ×1/8
15 to 7	—	All 0	R	Reserved These bits are always read as all 0.
6 to 4	P0FC[2:0]	011	R	Peripheral Clock 0 (Pck0) Frequency Division Ratio 011: ×1/8

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	P1FC[2:0]	101	R	Peripheral Clock 1 (Pck1) Frequency Division Ratio 101: ×1/16

16.4.2 PLL Control Register (PLLCR)

PLLCR is a 32-bit readable/writable register that enables or disables clock output from the CLKOUT pin. PLLCR can be accessed only in longwords.

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

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

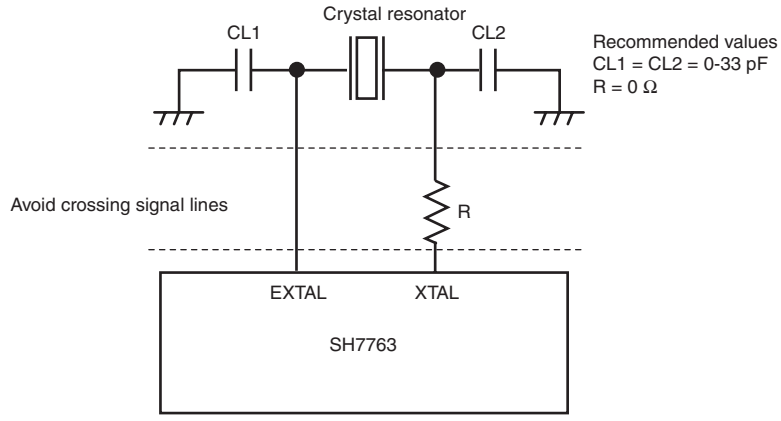
Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved The write value should be the same as the initial values.
14, 13	—	All 1	R	Reserved The write value should be the same as the initial values.
12 to 2	—	All 0	R	Reserved The write value should be the same as the initial values.
1	CKOFF	0	R/W	CLKOUT Output Stop 0: Clock is output from the CLKOUT pin. 1: Clock is not output from the CLKOUT pin. (The pin level is low.)
0	CKONE	1	R/W	Clock Output Enable Selects whether to output clock from the CLKOUT pin or tie the CLKOUT pin to a low level during software standby mode. 0: Tied to a low level 1: Clock is output

Note: * Depends on the clock operating mode that is selected by the MD0 to MD2 pin settings.

16.5 Notes on Board Design

(1) Notes on Using Crystal Resonator

Place the crystal resonator and capacitors as close as possible to the EXTAL and XTAL pins. Do not allow any signal lines to cross the EXTAL or XTAL line to prevent induction from interfering with correct oscillation.



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

Figure 16.2 Notes on Using Crystal Resonator

(2) Notes on Inputting External Clock via EXTAL Pin

Make no connection to the XTAL pin.

(3) Notes on Using PLL or DLL Oscillator Circuit

Separate VDD-PLL and VSS-PLL from the other VDD and VSS lines at the board power supply source, and insert resistors RCB and bypass capacitors CPB near the pins for noise filtering.

VDD-DLL and VSS-DLL should be set to the same level as the VDD and VSS levels, respectively.

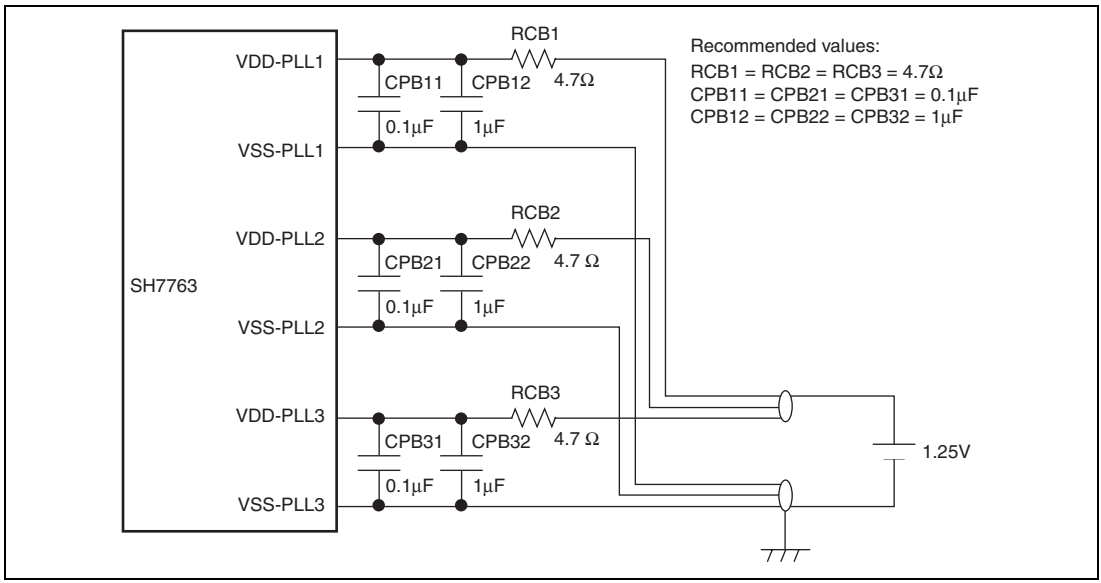


Figure 16.3 Notes on Using PLL or DLL Oscillator Circuit

Section 17 Watchdog Timer and Reset (WDT)

The reset and watchdog timer (WDT) control circuit comprises the reset control unit and WDT control unit which control the power-on reset sequence and a reset for on-chip peripheral modules and external devices.

The WDT is a one-channel timer which can be used as the watchdog timer or interval timer.

17.1 Features

- WDT monitors a system crash using a timer counting at specified intervals.
- WDT supports the watchdog timer mode and the interval timer mode.
- WDT generates an internal reset when a WDT overflow occurs in watchdog timer mode.
  A power-on reset or a manual reset can be selectable.
- WDT generates the interval timer interrupt when counter overflow occurs in interval timer mode.
- The maximum time until the watchdog timer overflows is approximately 21 seconds (when the peripheral clock Pck0 is 50 MHz).
- Writing to WDT-related registers is not normally allowed. A specified code in the upper bits of write data enables writing to the registers.

Figure 17.1 is a system block diagram.

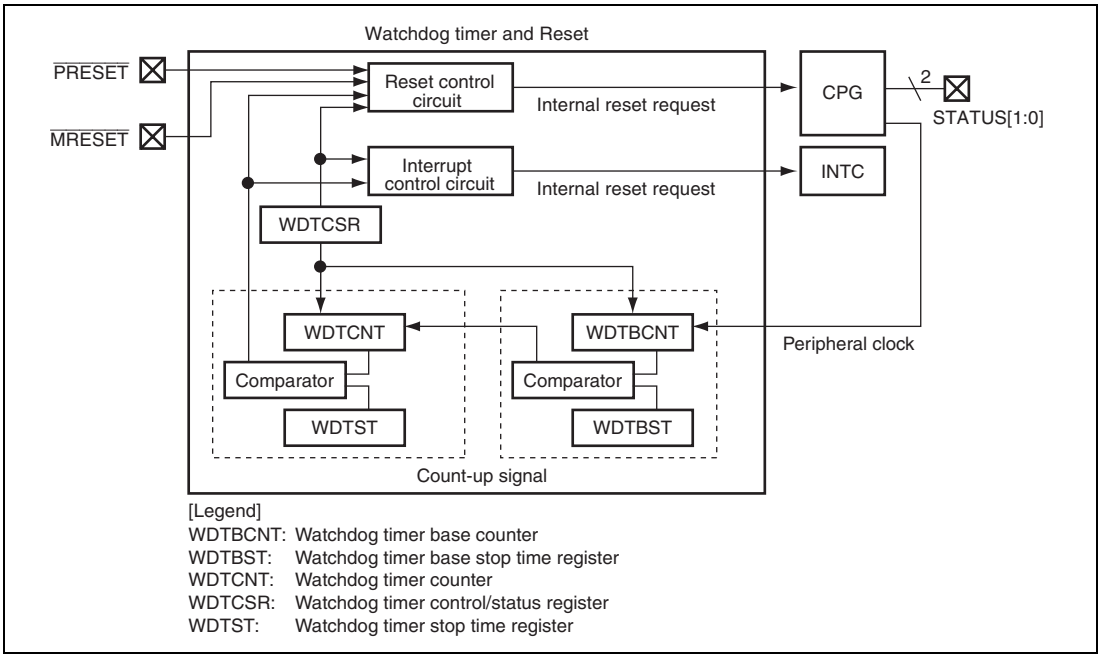


Figure 17.1 System Block Diagram

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the reset control unit.

Table 17.1 Pin Configuration

Pin name	Function	I/O	Description		
$\overline{\text{PRESET}}$	Power-on reset input	Input	Power-on reset occurs at low-level		
$\overline{\text{MRESET}}$	Manual reset input	Input	Manual reset occurs at low-level		
STATUS1*	Processing state 1	Output	Indicate the processor's operating status		
STATUS0*	Processing state 0		STATUS1	STATUS0	Operating Status
			High	High	Reset
			High	Low	Sleep mode
			Low	High	Standby mode
			Low	Low	Normal operation

Note: * These pins are multiplexed with other function pins.

17.3 Register Descriptions

Table 17.2 shows the registers of the reset and watchdog timer. Table 17.3 shows the register state in each operating mode.

Table 17.2 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Access Size
Watchdog timer stop time register	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32
Watchdog timer control/status register	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32
Watchdog timer base stop time register	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32
Watchdog timer counter	WDTCNT	R	H'FFCC 0010	H'1FCC 0010	32
Watchdog timer base counter	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32

Table 17.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin	Power-on Reset by WDT/H-UDI	Manual Reset	Sleep	Standby
Watchdog timer stop time register	WDTST	H'0000 0000	Retained	Retained	Retained	Retained
Watchdog timer control/status register	WDTCSR	H'0000 0000	Retained	Retained	Retained	Retained
Watchdog timer base stop time register	WDTBST	H'0000 0000	Retained	Retained	Retained	Retained
Watchdog timer counter	WDTCNT	H'0000 0000	Retained	Retained	Retained	Retained
Watchdog timer base counter	WDTBCNT	H'0000 0000	Retained	Retained	Retained	Retained

17.3.1 Watchdog Timer Stop Time Register (WDTST)

WDTST is a readable/writable 32-bit register that specifies the time until a watchdog timer overflows. The time until WDTCNT overflows becomes the minimum value when set H'001 to the bits 11 to 0, and the maximum value when set H'000 to the bits 11 to 0. Use a longword access to write to the WDTST, with H'5A in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value must be H'5A.
23 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	WDTST	All 0	R/W	Counter value

17.3.2 Watchdog Timer Control/Status Register (WDTCSR)

WDTCSR is a readable/writable 32-bit register that comprises the timer mode-selecting bit and overflow flags. Use a longword access to write to the WDTCSR, with H'A5 in the bits 31 to 24. The reading value of bits 31 to 24 is always H'00.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TME	WT/IT	RSTS	WOVF	IOVF	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value must be H'A5.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TME	0	R/W	Timer Enable Specifies starting and stopping of timer operation. 0: Stops counting up 1: Starts counting up
6	WT/IT	0	R/W	Timer Mode Select Specifies whether the WDT is used as a watchdog timer or interval timer. Up counting may not be performed correctly if this bit is modified while the WDT is running. 0: Interval timer mode 1: Watchdog timer mode

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	0	R/W	Reset Select Specifies the kind of reset to be performed when WDTCNT overflows in watchdog timer mode. This setting is ignored in interval timer mode. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Flag Indicates that WDTCNT has overflowed in watchdog timer mode. This flag is not set in interval timer mode. 0: An overflow has not occurred 1: An overflow on WDTCNT has occurred
3	IOVF	0	R/W	Interval Timer Overflow Flag Indicates that WDTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: An overflow has not occurred 1: An overflow on WDTCNT has occurred
2 to 0	—	R	All 0	Reserved These bits are always read as 0. The write value should always be 0.

17.3.3 Watchdog timer Base Stop Time Register (WDTBST)

WDTBST is a readable/writable 32-bit register that clears WDTBCNT. Use a longword access to clear the WDTBCNT, with H'A5 in the bits 31 to 24. The reading value of bits WDTBST is always H'0000 0000.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	R/W	H'00	Reserved (Given code for writing) These bits are always read as H'00. To write to this register, the write value must be H'55.
23 to 0	—	R	All 0	Reserved These bits are always read as 0. The write value should always be 0.

17.3.4 Watchdog Timer Counter (WDTCNT)

WDTCNT is a 32-bit read-only register that comprises 12-bit watchdog timer counter and counts up on the WDTBCNT overflow signal. When WDTCNT overflows, a reset is generated in watchdog timer mode, or an interrupt is generated in interval timer mode. Writing to WDTCNT is invalid.

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

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

17.3.5 Watchdog Timer Base Counter (WDTBCNT)

WDTBCNT is a 32-bit read-only register that comprises 18-bit counter and counts up on the peripheral clock (Pck0). When WDTBCNT overflows, WDTCNT is counted up and WDTBCNT is cleared to 0 Writing to WDTCNT is invalid.

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

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

17.4 Operation

17.4.1 Reset request

Power-on reset and manual reset are available. These sources are follows.

(1) Power-on reset

1. Reset sources

- Input low level via $\overline{\text{PRESET}}$ pin.
- The WDTCNT overflows when the WT/IT bit in the WDTCSR is 1, and the RSTS bit is 0.
- The H-UDI reset occurs (For details, see section 42, User Debugging Interface (H-UDI)).

2. Branch destination address: H'A000 0000

3. Operation in branch

Exception code H'000 is set in the EXPEVT register. The VBR and SR registers are initialized, and the program branches to PC =H'A000 0000. By initialization, the VBR register is set to H'0000 0000. In the SR register, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the IMASK3 to IMASK0 bits (interrupt mask level) are set to B'1111.
The CPU and the peripheral modules are also initialized. For details, see the register descriptions in each section.
When the power is turned on, be sure to input a low level to the $\overline{\text{PRESET}}$ pin. The $\overline{\text{TRST}}$ pin should also be brought low level to initialize the H-UDI.

```
Power_on_reset()
{
        EXPEVT = H'0000 0000;
        VBR = H'0000 0000;
        SR.MD = 1;
        SR.RB = 1;
        SR.BL = 1;
        SR.(I0-I3) = B'1111;
        SR.FD = 0;
        Initialize_CPU();
        Initialize_Module(PowerOn);
        PC = H'A000 0000;
}
```

(2) Manual reset

1. Reset sources

- Input low level via $\overline{\text{MRESET}}$ pin.
- When a general exception other than a user break occurs while the BL bit is set to 1 in SR
- When the WDTCNT overflows while the WT/IT bit and the RSTS bit are set to 1 in WTCSR.

2. Branch destination address: H'A000 0000

3. Operation in branch

Exception code H'020 is set in the EXPEVT register. The VBR and SR registers are initialized, and the program branches to PC =H'A000 0000. By initialization, the VBR register is set to H'0000 0000. In the SR register, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the IMASK3 to IMASK0 bits (interrupt mask level) are set to B'1111.
The CPU and the peripheral modules are also initialized. For details, see the register descriptions in each section.

```
Manual_reset()
{
    EXPEVT = H'0000 0020;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.(I0-I3) = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(Manual);
    PC = H'A000 0000;
}
```

17.4.2 Using watchdog timer mode

1. Set the WDTCNT overflow interval value in WDTST.
2. Set the WT/IT bit in WDTCSR to 1, select the type of reset with the RSTS bit.
3. When the TME bit in WDTCSR is set to 1, the WDT count starts.

4. During operation in watchdog timer mode, clear to the WDTCNT or WDTBCNT periodically so that WDTCNT does not overflow. See section 17.4.5, Clearing WDT Counter for WDT counter clear method.
5. When the WDTCNT overflows, the WDT sets the WOVF flag in WDTCSR to 1, and generates a reset of the type specified by the RSTS bit. After reset operation, the WDTCNT and WDTBCNT continue counting again.

17.4.3 Using Interval timer mode

When the WDT is operating in interval timer mode, an interval timer interrupt is generated each time the counter overflows. This enables interrupts to be generated at fixed intervals.

1. Set the WDTCNT overflow time in WDTST.
2. Clear the WT/IT bit in WDTCSR to 0.
3. When the TME bit in WDTCSR is set to 1, the WDT count starts.
4. When the WDTCNT overflows, the WDT sets the IOVF flag in WDTCSR to 1, and sends an interval timer interrupt (ITI) request to INTC. The counter continues counting.

17.4.4 Time for WDT Overflow

Figure 17.2 shows a WDT counting up operation.

In interval timer mode, the WDT continues counting even after the WDTCNT overflow. In watchdog timer mode, the WDT clears the WDTCNT and WDTBCNT and start counting again after reset operation.

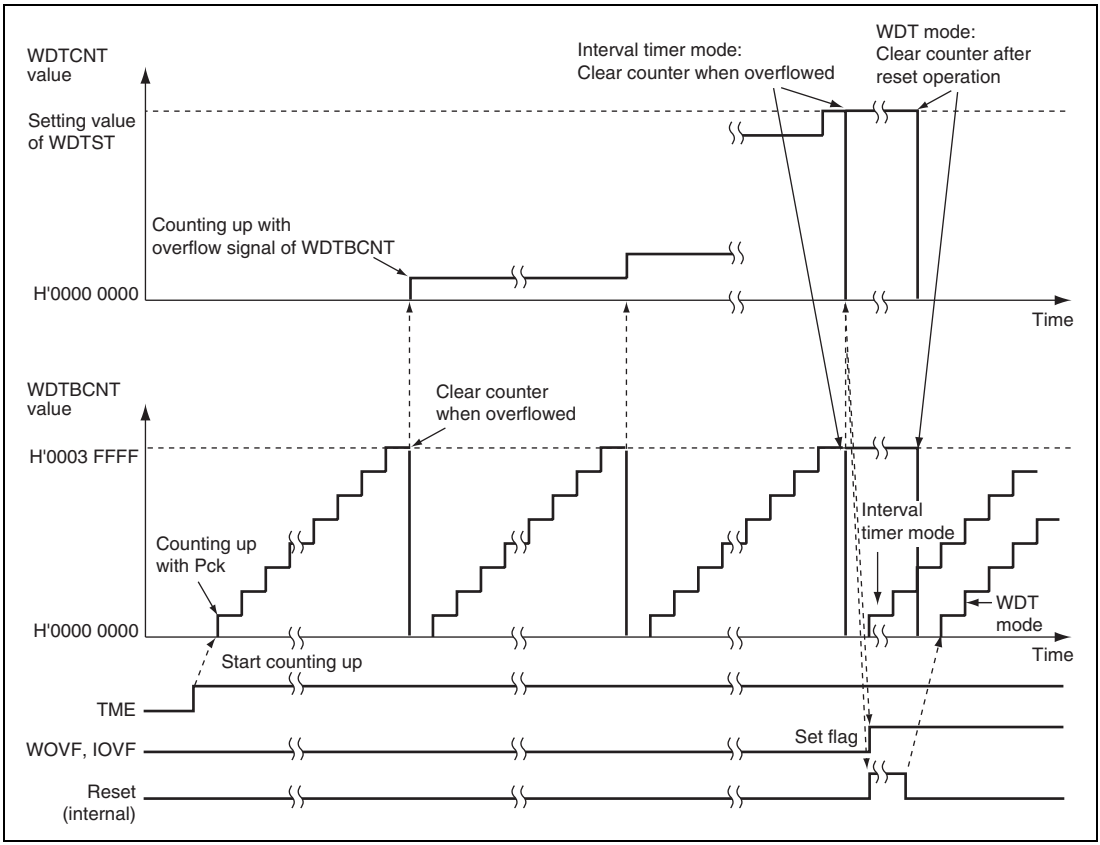


Figure 17.2 WDT Counting Up Operation

WDTBCNT is an 18-bit up-counter operated on the peripheral clock0 (Pck0). WDTBCNT is cleared when H'55 is set to the bits 31 to 24 in WDTBST.

If the peripheral clock frequency is 66.6 MHz, the WDTBCNT overflow time is approximately 3.932 ms (= 2^18 [bit] × 1/66.6 [MHz]).

WDTCNT is a 12-bit counter, starts count up operation when overflow occurs in WDTBCNT. The time until WDTCNT overflows becomes the maximum value when H'000 is set to WDTST.

Where the peripheral clock frequency is 66.6 MHz, the maximum overflow time is approximately 16.105 s (= 2^12 [bit] × 3.932 [ms]).

And the time until WDTCNT overflows becomes the minimum value when H'5A000001 is set to WDTST. The minimum overflow time is approximately 3.932 ms (= 2^1 [bit] × 3.932 [ms]).

17.4.5 Clearing WDT Counter

Writing H'55 to WDTBST with longword access clears WDTBCNT and writing the overflow setting value to WDTST clears WDTCNT.

17.5 Status Pin Change Timing during Reset

17.5.1 Power-On Reset by PRESET

A power-on reset is to initialize the on-chip PLL circuit when this LSI goes to the power-on reset state by the $\overline{\text{PERSET}}$ pin low level input and then it is necessary to ensure the synchronization settling time of the PLL circuit. Therefore, do not input high level to the $\overline{\text{PRESET}}$ pin during the PLL synchronization settling time. The PLL synchronization settling time is the total value of the PLL1 synchronization settling time and the PLL2 synchronization settling time.

After the $\overline{\text{PRESET}}$ pin input level is changed from low level to high level, the reset state is continued during the reset holding time in the LSI. The reset holding time is 20 clock cycles of the EXTAL pin input clock and thereafter equal to or more than 45 clock cycles of the peripheral clock (Pck0).

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and that indicates a normal operation is synchronous with the peripheral clock (Pck0) and asynchronous with both the EXTAL pin input clock and the CLKOUT pin input clock.

(1) Turning On Power Supply

When turning on power supply, the $\overline{\text{PRESET}}$ pin input level should be low level. And the $\overline{\text{TRST}}$ pin input level should be low level to initialize the H-UDI.

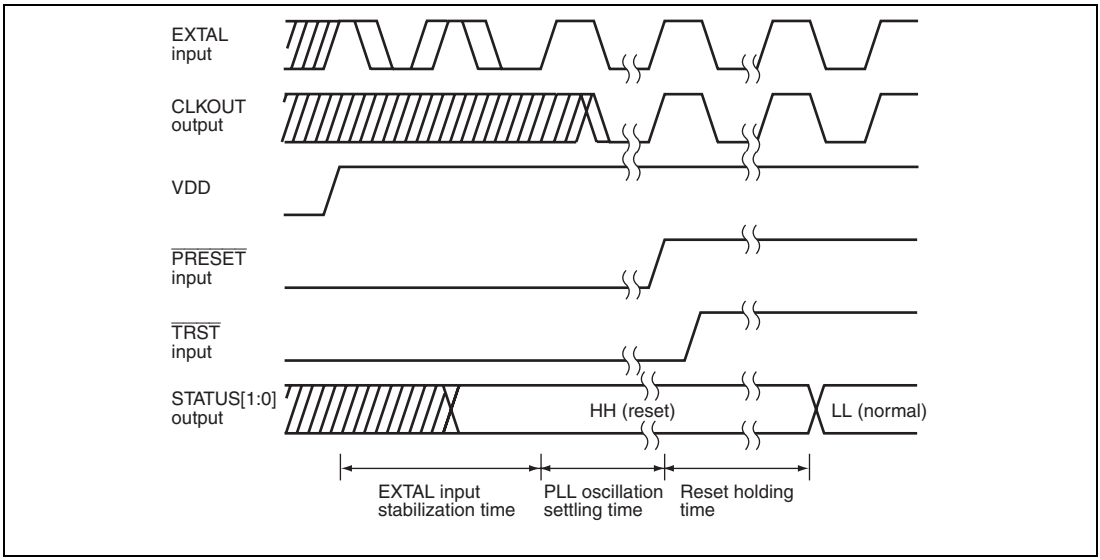


Figure 17.3 STATUS Output during Power-on

(2) $\overline{\text{PRESET}}$ input during normal operation

It is necessary to ensure the PLL oscillation settling time when the $\overline{\text{PRESET}}$ input during normal operation.

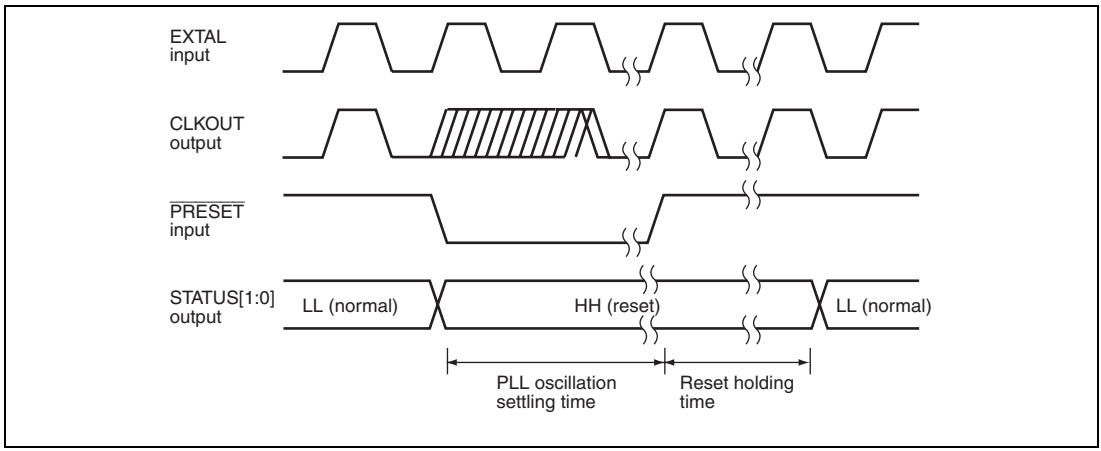


Figure 17.4 STATUS Output by Reset input during Normal Operation

(3) $\overline{\text{PRESET}}$ input during Sleep Mode

It is necessary to ensure the PLL oscillation time when power-on reset generates by the $\overline{\text{PRESET}}$ pin low revel input during sleep mode.

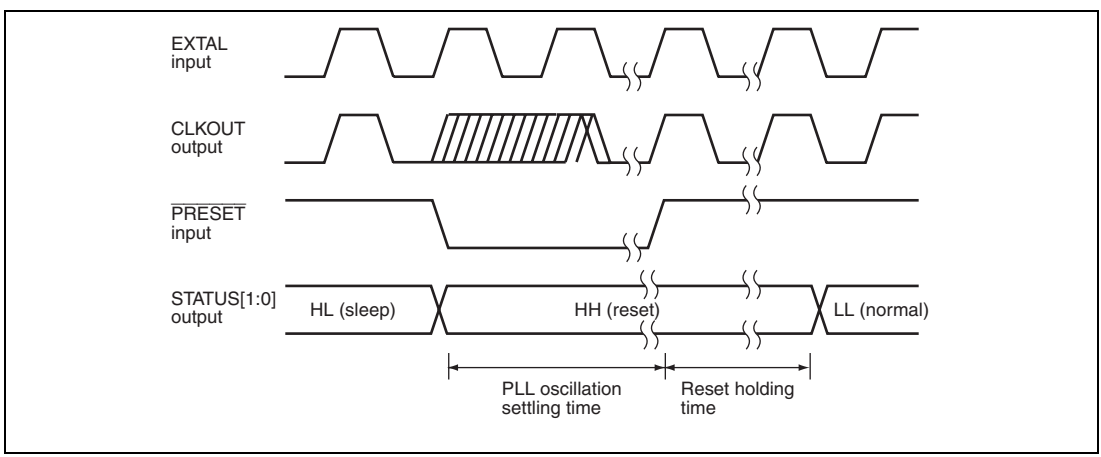


Figure 17.5 STATUS Output by Reset input during Sleep Mode

17.5.2 Power-On Reset by Watchdog Timer Overflow

The power-on reset time (watchdog timer reset holding time) by the watchdog timer overflowed is 3774 clock cycles of the EXTAL pin input clock and thereafter equal to or more than 45 clock cycles of the peripheral clock (Pck0).

The transition time from the watchdog timer overflowed to the power-on reset state (watchdog timer reset setup time) is 1 clock cycle of the EXTAL input clock and thereafter equal to or more than 5 clock cycles of the peripheral clock (Pck0).

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the EXTAL pin input clock and the CLKOUT pin input clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck0).

(1) Power-On Reset by Watchdog timer Overflowed in Normal Operation

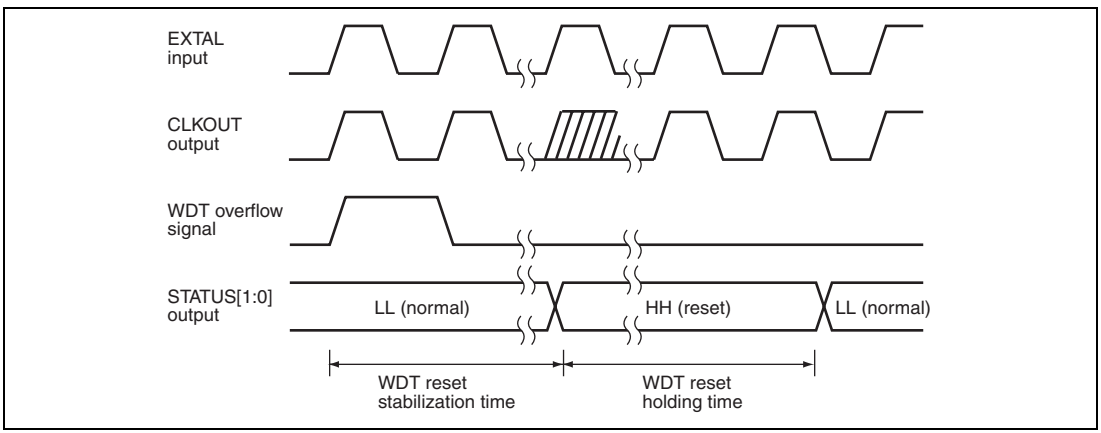


Figure 17.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation

(2) Power-On Reset by Watchdog timer Overflowed in Sleep Mode

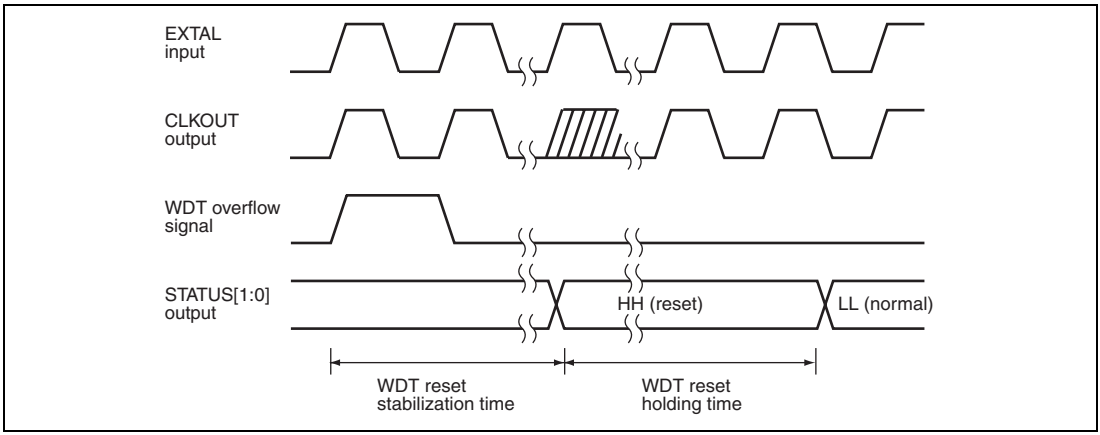


Figure 17.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode

17.5.3 Manual Reset by Watchdog Timer Overflow

The manual reset time (watchdog timer manual reset holding time) by the watchdog timer overflowed is equal to or more than 3774 clock cycles of the EXTAL pin input clock.

The transition time from watchdog timer overflowed to manual reset state (watchdog timer reset setup time) is 1 clock cycle of the EXTAL input and thereafter equal to or more than 5 clock cycles of the peripheral clock (Pck0).

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation is asynchronous with both the EXTAL pin input clock and the CLKOUT pin input clock because the STATUS [1:0] pins output timing is synchronous with the peripheral clock (Pck0).

(1) Manual Reset by Watchdog timer Overflowed in Normal Operation

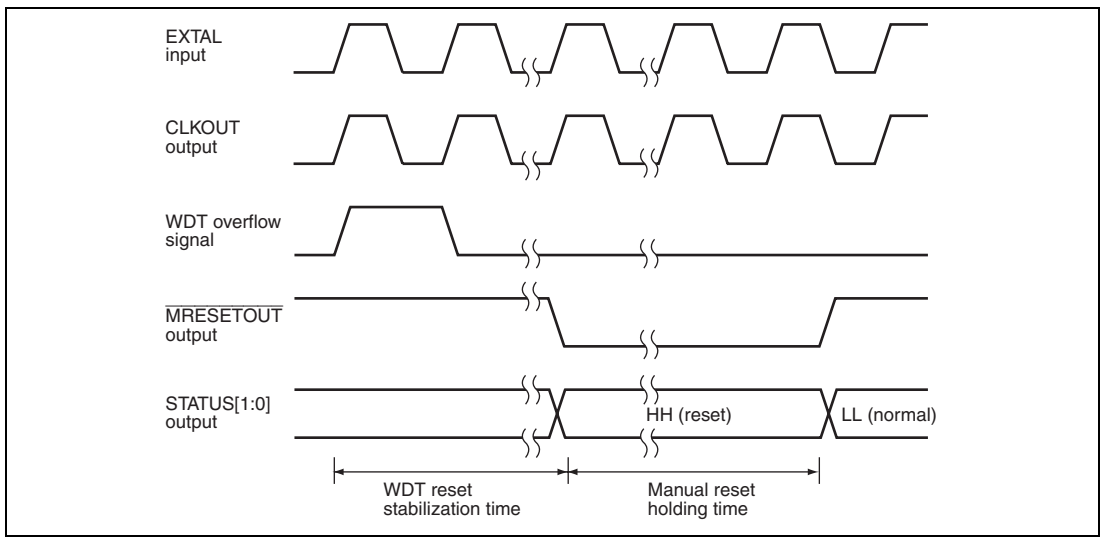


Figure 17.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation

(2) Manual Reset by Watchdog timer Overflowed in Sleep Mode

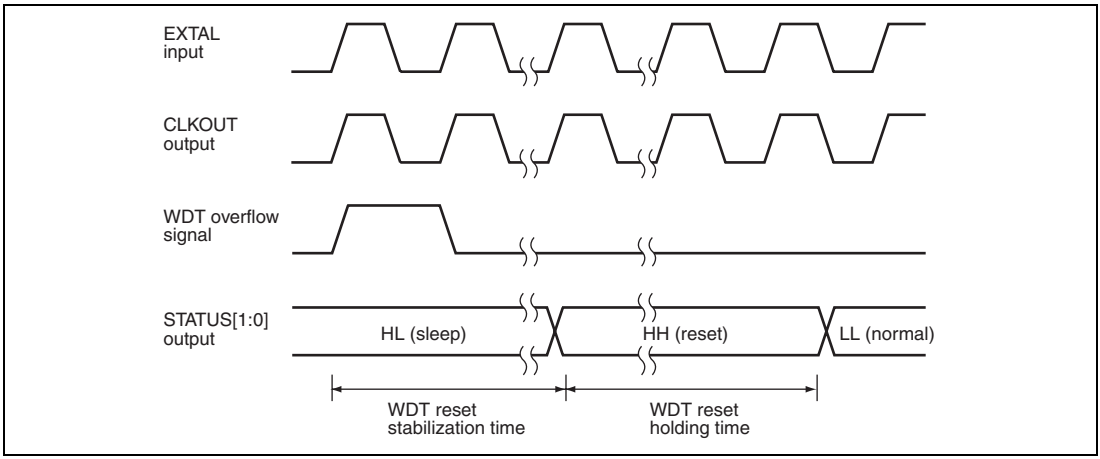


Figure 17.9 STATUS Output by Watchdog timer overflow Manual Reset during Sleep Mode

Section 18 Power-Down Mode

In power-down modes, operations of the CPU and some of the on-chip peripheral modules are stopped to reduce power consumption.

18.1 Features

- Supports sleep mode and module standby mode
- Supports RTC power supply backup mode where the power supply for only the RTC is held and other power supplies are turned off
- Supports DDR-SDRAM power supply backup mode where the power supply for only the 2.5-V power supplied modules are held and other power supplies are turned off

18.1.1 Types of Power-Down Modes

The types and functions of power-down modes are as shown below.

- Sleep mode
- Software standby mode
- Module standby mode
- RTC power supply backup mode
- DDR-SDRAM power supply backup mode

Table 18.1 lists the states of the CPU and on-chip peripheral modules in each mode.

Table 18.1 States in Power-Down Modes

		State									
					On-Chip Peripheral Module						
Power-Down Mode	Transition Condition	CPG	CPU	On-Chip Memory	RTC	Others	Pin	DDR-SDRAM	Cancellation	S1 *7	S0 *7
Sleep	SLEEP instruction executed with STBY = 0 in STBCR	Run	Halt (register contents retained)	Retained	Run	Run	Held	AR or SR*6	- Interrupt - Power-on reset - Manual reset	1	0
Software standby	SLEEP instruction executed with STBY = 1 in STBCR	Halt*8	Halt (register contents retained)	Halt (contents retained)	Run	Halt	Hi-Z	Undefined (refresh not performed)	- NMI or IRQ - Power-on reset - Manual reset	0	1
Module standby	Corresponding bit in MSTPCR0/ MSTPCR1 set to 1	Run	Run	Run	Run	Selected modules halt	Held	AR or SR*6	Clear corresponding bit in MSTPCR0/ MSTPCR1 to 0	0	0
RTC power supply backup *2*3	$\overline{\text{XRTCSTBI}}$ pin driven low	Halt	Halt	Halt	Run	Halt	Hi-Z*4	Undefined (refresh not performed)	Power-on reset	0	1
DDR-SDRAM power supply backup *1*3	See section 18.7, DDR-SDRAM Power Supply Backup.	Halt	Halt	Halt	Halt	Halt	Undefined*5	SR*6	Power-on reset	0	0
Power-on reset	$\overline{\text{PRESET}}$ pin driven low	Initial state	Initial state	Initial state	Counter retained	Initial state	Initial state	Initial state	—	1	1
Manual reset	$\overline{\text{MRESET}}$ pin driven low or software reset	Held	Initial state	Initial state	Counter retained	WDT, GPIO, and debugging interface are held	Held	Initial state	—	1	1
Normal operation		Run	Run	Run	Run	Run	Run	Run	—	0	0

Notes: 1. Because power supplies (1.2 V and 3.3 V) other than the 2.5-V power supply are stopped in DDR-SDRAM power supply backup mode, all modules, except for pads of the DDRIF module, are halted and their register information is not retained.

2. Because power supplies (1.2 V, 2.5 V, and 3.3 V) other than the RTC power supply are stopped in RTC power supply backup mode, all modules other than the RTC module are halted and their register information is not retained.

3. To enter both RTC and DDR-SDRAM power supply backup modes, satisfy the transition conditions for both of them.

4. Hi-Z state, except for the RTC module interface pins
5. Undefined, except for DDR-SDRAM interface pins
6. AR: auto-refresh: SR: self-refresh
7. S1 and S0 are the output states on the STATUS1 and STATUS0 pins, respectively.
8. Although the clock supply to the internal circuitry is stopped, the clock output on the CLKOUT pin continues if the CKONE bit in PLLCR of the CPG is set to 1.

18.2 Input/Output Pins

Table 18.2 lists the pin configuration related to power-down modes.

Table 18.2 Pin Configuration

Pin Name	Function	I/O	Description	
STATUS1	Processing state 1	Output	These pins indicate the operating state of this LSI.	
STATUS0	Processing state 0	Output	STATUS[1:0]	Operating state
			H, H:	Power-on reset or manual reset
			H, L:	Sleep mode
			L, H:	Software standby or RTC power supply backup mode
			L, L:	Normal operation
$\overline{\text{XRTCSTB1}}$	RTC standby	Input	When this pin goes low, the LSI enters RTC power supply backup mode.	

18.3 Register Descriptions

Table 18.3 shows the register configuration for power-down modes. Table 18.4 shows the register states in each operating mode.

Table 18.3 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Standby control register	STBCR	R/W	H'FFC8 0020	H'1FC8 0020	32
Module stop register 0	MSTPCR0	R/W	H'FFC8 0030	H'1FC8 0030	32
Module stop register 1	MSTPCR1	R/W	H'FFC8 0038	H'1FC8 0038	32

Table 18.4 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Standby control register	STBCR	H'0000 0000	Retained	Retained	Retained
Module stop register 0	MSTPCR0	H'0000 0000	Retained	Retained	Retained
Module stop register 1	MSTPCR1	H'0000 0000	Retained	Retained	Retained

18.3.1 Standby Control Register (STBCR)

STBCR is a 32-bit readable/writable register that selects a power-down mode to be entered after a SLEEP instruction is executed.

STBCR can be accessed only in longwords.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	STBY	0	R/W	Standby Selects whether to enter sleep mode or software standby mode after a SLEEP instruction is executed. 0: Sleep mode 1: Software standby mode
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.2 Module Stop Register 0 (MSTPCR0)

MSTPCR0 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR0 can be accessed only in longwords.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	LCDC	0	R/W	LCDC Module Stop Bit When set to 1, the clock supply to the LCDC module is halted. 0: LCDC operates 1: Clock supply to LCDC is halted
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

18.3.3 Module Stop Register 1 (MSTPCR1)

MSTPCR1 is a 32-bit readable/writable register that can individually start or stop the module assigned to each bit.

MSTPCR1 can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PCC	HAC	STIF1	STIF0	SSI3	SSI2	SSI1	SSI0	IIC1	IIC0	SIOF2	SIOF1	SIOF0	SCIF2	SCIF1
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCIF0	SIM	ADC	DAC	CMT	TMU1	TMU0	TPU	—	—	—	—	MMC	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R	R

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	PCC	0	R/W	PCC Module Stop Bit When set to 1, the clock supply to the PCC module is halted. 0: PCC operates 1: Clock supply to PCC is halted
29	HAC	0	R/W	HAC Module Stop Bit When set to 1, the clock supply to the HAC module is halted. 0: HAC operates 1: Clock supply to HAC is halted

Bit	Bit Name	Initial Value	R/W	Description
28	STIF1	0	R/W	STIF1 Module Stop Bit When set to 1, the clock supply to the STIF1 module is halted. 0: STIF1 operates 1: Clock supply to STIF1 is halted
27	STIF0	0	R/W	STIF0 Module Stop Bit When set to 1, the clock supply to the STIF0 module is halted. 0: STIF0 operates 1: Clock supply to STIF0 is halted
26	SSI3	0	R/W	SSI3 Module Stop Bit When set to 1, the clock supply to the SSI3 module is halted. 0: SSI3operates 1: Clock supply to SSI3 is halted
25	SSI2	0	R/W	SSI2 Module Stop Bit When set to 1, the clock supply to the SSI2 module is halted. 0: SSI2 operates 1: Clock supply to SSI2 is halted
24	SSI1	0	R/W	SSI1 Module Stop Bit When set to 1, the clock supply to the SSI1 module is halted. 0: SSI1 operates 1: Clock supply to SSI1 is halted
23	SSI0	0	R/W	SSI0 Module Stop Bit When set to 1, the clock supply to the SSI0 module is halted. 0: SSI0 operates 1: Clock supply to SSI0 is halted

Bit	Bit Name	Initial Value	R/W	Description
22	IIC1	0	R/W	IIC1 Module Stop Bit When set to 1, the clock supply to the IIC1 module is halted. 0: IIC1 operates 1: Clock supply to IIC1 is halted
21	IIC0	0	R/W	IIC0 Module Stop Bit When set to 1, the clock supply to the IIC0 module is halted. 0: IIC0 operates 1: Clock supply to IIC0 is halted
20	SIOF2	0	R/W	SIOF2 Module Stop Bit When set to 1, the clock supply to the SIOF2 module is halted. 0: SIOF2 operates 1: Clock supply to SIOF2 is halted
19	SIOF1	0	R/W	SIOF1 Module Stop Bit When set to 1, the clock supply to the SIOF1 module is halted. 0: SIOF1 operates 1: Clock supply to SIOF1 is halted
18	SIOF0	0	R/W	SIOF0 Module Stop Bit When set to 1, the clock supply to the SIOF0 module is halted. 0: SIOF0 operates 1: Clock supply to SIOF0 is halted
17	SCIF2	0	R/W	SCIF2 Module Stop Bit When set to 1, the clock supply to the SCIF2 module is halted. 0: SCIF2 operates 1: Clock supply to SCIF2 is halted

Bit	Bit Name	Initial Value	R/W	Description
16	SCIF1	0	R/W	SCIF1 Module Stop Bit When set to 1, the clock supply to the SCIF1 module is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 is halted
15	SCIF0	0	R/W	SCIF0 Module Stop Bit When set to 1, the clock supply to the SCIF0 module is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 is halted
14	SIM	0	R/W	SIM Module Stop Bit When set to 1, the clock supply to the SIM module is halted. 0: SIM operates 1: Clock supply to SIM is halted
13	ADC	0	R/W	ADC Module Stop Bit When set to 1, the clock supply to the ADC module is halted. 0: ADC operates 1: Clock supply to ADC is halted
12	DAC	0	R/W	DAC Module Stop Bit When set to 1, the clock supply to the DAC module is halted. 0: DAC operates 1: Clock supply to DAC is halted
11	CMT	0	R/W	CMT Module Stop Bit When set to 1, the clock supply to the CMT module is halted. 0: CMT operates 1: Clock supply to CMT is halted

Bit	Bit Name	Initial Value	R/W	Description
10	TMU1	0	R/W	TMU1 Module Stop Bit When set to 1, the clock supply to the TMU1 module is halted. 0: TMU1 operates 1: Clock supply to TMU1 is halted
9	TMU0	0	R/W	TMU0 Module Stop Bit When set to 1, the clock supply to the TMU0 module is halted. 0: TMU0 operates 1: Clock supply to TMU0 is halted
8	TPU	0	R/W	TPU Module Stop Bit When set to 1, the clock supply to the TPU module is halted. 0: TPU operates 1: Clock supply to TPU is halted
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MMC	0	R/W	MMC Module Stop Bit When set to 1, the clock supply to the MMC module is halted. 0: MMC operates 1: Clock supply to MMC is halted
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When writing to a certain bit in MSTPCR1, read all values in MSTPCR1 first and rewrite the certain bit, then return the renewed values back to MSTPCR1.

18.4 Sleep Mode

18.4.1 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 the CPU registers remain unchanged. On-chip peripheral modules continue to operate, and the clock output on the CLKOUT pin also continues. In sleep mode, a high level is output to the STATUS1 pin and a low level to the STATUS0 pin.

18.4.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ/$\overline{\text{IRL}}$[7:0], or on-chip peripheral module interrupt) or a reset.

Interrupts are accepted in sleep mode even when the BL bit in SR is 1. If necessary, save SPC and SSR to the stack before executing the SLEEP instruction.

(1) Canceling with Interrupt

When an NMI, IRQ/$\overline{\text{IRL}}$[7:0], or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in INTEVT.

(2) Canceling with Reset

Sleep mode is canceled by a power-on reset caused by the $\overline{\text{RESET}}$ pin or watchdog timer overflow or a manual reset.

Note: If an NMI interrupt is used to cancel sleep mode while the LCD is used, the NMIFL bit in the NMIFCR register is set to 1 by the interrupt. This disables the LCD to access to the VRAM used for the display data storage (DDR_SDRAM in area 3).

Moreover, as the LCDC continues to output data stored in the line buffer to the LCD panel data pin, the LCD display will be stopped if the line buffer becomes empty. Accordingly, an NMI interrupt should be disabled and the NMIFL bit should be cleared to 0 before the line buffer becomes empty.

18.5 Software Standby Mode

18.5.1 Transition to Software Standby Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 1 causes a transition from the program execution state to software standby mode. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. However, the clock output from the CLKOUT pin continues when the CKONE bit in the PLLCR register of the CPG is set to 1. When the CKONE bit is 0, a low level is output from the CLKOUT pin.

The contents of the CPU and cache registers remain unchanged. Some registers of the on-chip peripheral modules are initialized.

The procedure for a transition to software standby mode is as follows:

1. Set the STBY bit in STBCR to 1.
2. Execute the SLEEP instruction.
3. Software standby mode is entered and the clocks within the LSI are halted. The output on the STATUS0 pin goes high.

All modules should be stopped before the above procedure is executed.

18.5.2 Canceling Software Standby Mode

Software standby mode is canceled by an interrupt (NMI or IRQ/IRL) or a reset.

(1) Canceling with Interrupt

When an NMI or IRQ, occurs, software standby mode is canceled and the STATUS0 pin goes low. Thereafter, interrupt exception handling is executed and a code indicating the interrupt source is set in INTEVT. After branching to the interrupt service routine, clear the STBY bit in the STBCR register back to 0. Since interrupts are accepted in software standby mode even when the BL bit in SR is 1, save SPC and SSR to the stack before executing the SLEEP instruction if necessary.

Immediately after an interrupt is detected, the clock output on the CLKOUT pin may be unstable until software standby mode is canceled.

18.6 Module Standby Mode

18.6.1 Transition to Module Standby Mode

Setting the bits in the module stop register to 1 halts the clock supply to the corresponding on-chip peripheral modules. This function can be used to reduce power consumption in normal mode.

Modules in module standby mode keep the state immediately before the transition to the module standby mode. The registers retain their contents before the module is halted, and the external pins also hold their states before halted. At waking up from the module standby state, operation starts from the condition immediately before the module was halted.

18.6.2 Canceling Module Standby Mode

The module standby mode can be canceled by clearing the respective bit in the module stop register to 0 or by a power-on reset.

18.7 DDR-SDRAM Power Supply Backup

18.7.1 Control of Self-Refresh and Initialization

To preserve the contents of the DDR-SDRAM with battery backup, make sure that the DDR-SDRAM is in the self-refresh mode before turning off the system power supply. When the system power supply is turned on, initialization of the DDR-SDRAM or cancellation of the self-refresh mode must be performed according to whether the DDR-SDRAM has been in self-refresh mode or has not been initialized. For DDR-SDRAM, both a transition to and a cancellation of the self-refresh mode are done by issuing a command.

(1) RMODE Bit

Bit 33 in the MIM register. The initial value is 0. Setting this bit to 1 after setting the DRE bit in MIM to 1 causes the DDRIF to start the sequence for a transition to the self-refresh mode. For details, see section 12.5.5 (1), Self-Refresh Mode.

(2) Bits SMS2 to SMS0

Bits 2 to 0 in the SCR register. These bits are used to assert the M_CKE signal (high) by setting SMS = B'011 when canceling the self-refresh mode with the DESL command.

(3) $\overline{\text{M_BKPRST}}$ Signal

To prevent the M_CKE signal from being unstable when turning on or off the LSI power supply, the $\overline{\text{M_BKPRST}}$ signal must be input in synchronization with turning the LSI power supply on or off. The $\overline{\text{M_BKPRST}}$ signal must be kept low while the system power supply is turned off.

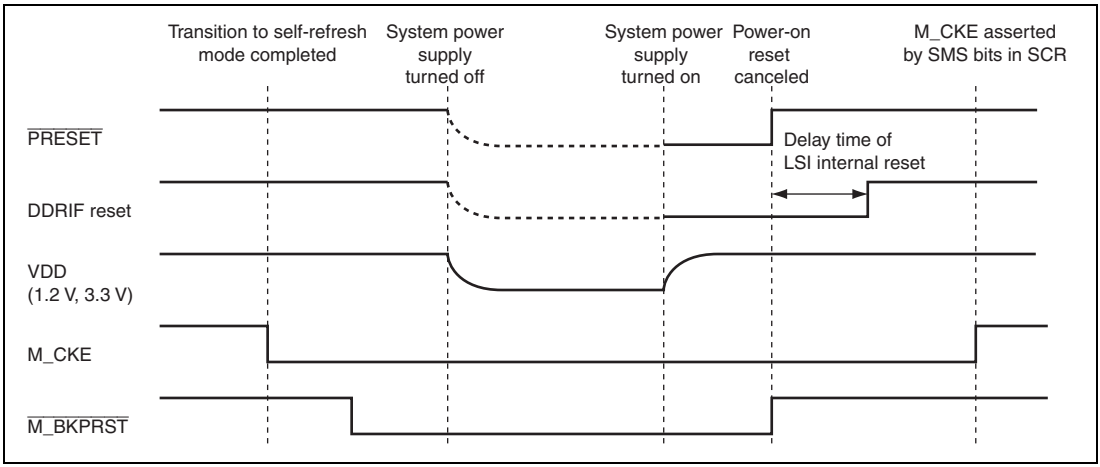


Figure 18.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off

18.7.2 DDR-SDRAM Backup Sequence when Turning Off System Power Supply

The sequence when the system power supply is turned off is shown below.

Figure 18.2 shows the sequence of entering the self-refresh mode and turning off the system power supply.

1. Confirm that all transactions of the DDRIF caused by on-chip peripheral modules are completed.
2. Issue the all bank precharge command (PREALL) with bits SMS2 to SMS0 in SCR by software. Activated banks will be closed. After that, issue the auto-refresh command (REFA) with bits SMS2 to SMS0 in SCR to perform CBR refresh on all rows.
3. Specify the DRE and RMODE bits in the MIM register of the DDRIF to put the SDRAM into the self-refresh mode. At this time, keep the DCE bit set to 1. The DDRIF automatically issues a self-refresh command and drives the M_CKE signal low. After that, the DDR-SDRAM will automatically enter the power-down mode.
4. The SELFS bit in MIM is set to 1.
5. Drive the $\overline{\text{M_BKPRST}}$ pin from high to low.
   The M_CKE output will be unstable immediately after the system power supply is turned off. Therefore, before turning off the system power supply, use the $\overline{\text{M_BKPRST}}$ signal, which is a signal outside the LSI, to keep the M_CKE signal input of the DDR-SDRAM low until the power-on reset is canceled (figure 18.1).
6. Turn off the system power supply (1.2 V and 3.3 V).

After the system power supply is turned on, the M_CKE output may remain unstable until the clock is supplied after the LSI power supply has become stable. Therefore, use the $\overline{\text{M_BKPRST}}$ signal to keep the M_CKE signal input of the DDR-SDRAM low until the power-on reset is canceled.

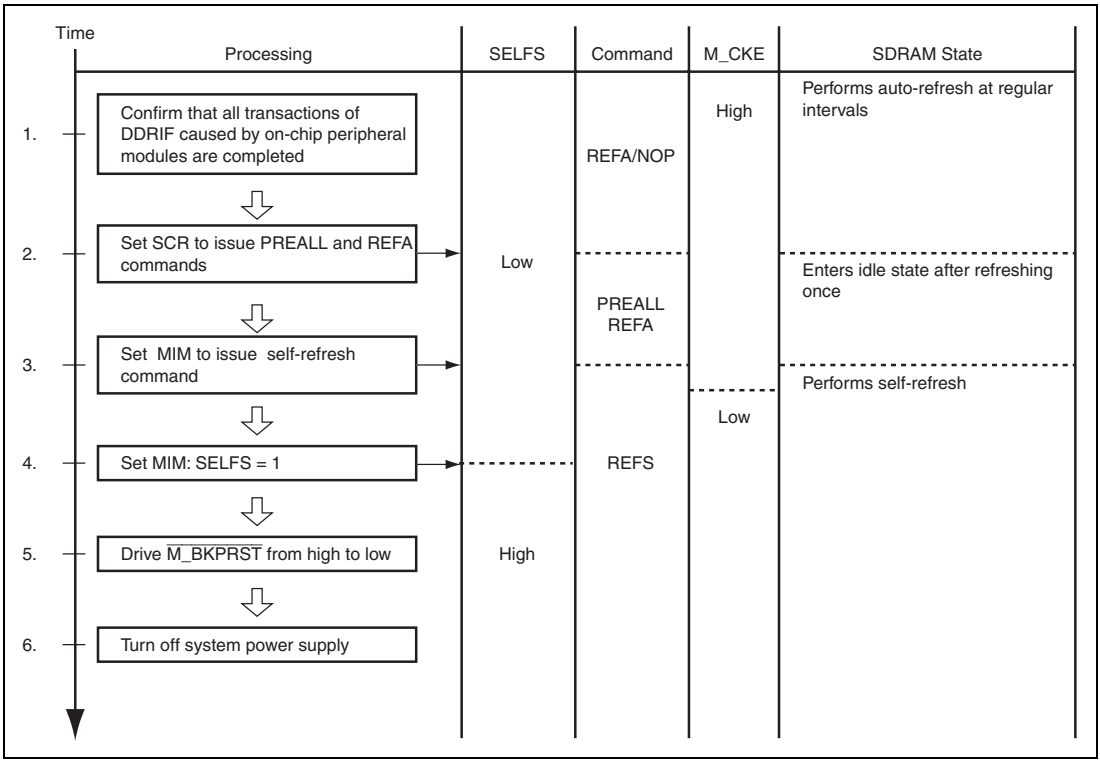


Figure 18.2 Sequence for Turning Off System Power Supply after Entering Self-Refresh Mode

18.8 RTC Power Supply Backup

18.8.1 Transition to RTC Power Supply Backup Mode

When entering the RTC power supply backup mode with the VDD power supply (1.2 V) turned off, the VDD power supply should be turned off while the $\overline{\text{XRTCSTBI}}$ signal is held low. By turning off the VDD power supply, the currents that might be generated in the VDD (1.2 V) operating region can be eliminated to reduce power consumption. If the clock has been supplied from the 32.768-Hz crystal, the RTC continues counting of its second to year counters while the VDD (1.2 V) power supply is turned off.

18.8.2 Canceling RTC Power Supply Backup Mode

The RTC power supply backup mode is canceled by a power-on reset. Even when any interrupt condition is satisfied in the RTC power supply backup mode, the interrupt-generating condition will be canceled by the power-on reset. RTC power supply backup mode can be canceled in the following steps.

1. Turn on the VDD power supply (1.2 V) while holding the $\overline{\text{PRESET}}$ signal low.
2. Since the VDD-RTC (3.3 V), which is exclusively used for the RTC, is supplied, drive the $\overline{\text{XRTCSTBI}}$ signal high only after the power-on oscillation stabilization time has elapsed after the VDD (1.2 V) becomes stable. This is to prevent the LSI from being damaged by the transient current.
3. Hold the $\overline{\text{PRESET}}$ signal low until the RTC is reset by the power-on reset, and then cancel the power-on reset.

Table 18.5 shows the configuration of the pins related to RTC power supply backup mode.

Table 18.5 Pin Configuration

Pin Name	Function	I/O	Description
$\overline{\text{XRTCSTBI}}$	RTC standby	Input	When this pin goes low, the LSI enters RTC power supply backup mode.

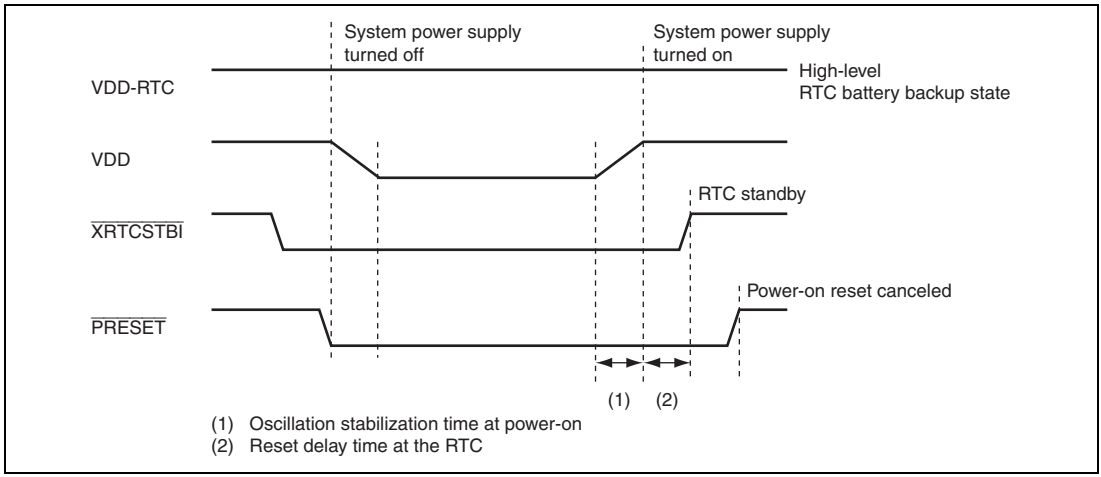


Figure 18.3 Sequence for Turning VDD Power Supply (1.2 V) On/Off

18.9 STATUS Pin Signal Change Timing

18.9.1 Timing at Reset

Refer to section 17.5, Status Pin Change Timing during Reset.

18.9.2 Timing at Sleep Mode Cancellation

(1) When an Interrupt Occurs in Sleep Mode

Figure 18.4 shows the timing of signal changes on the STATUS pins.

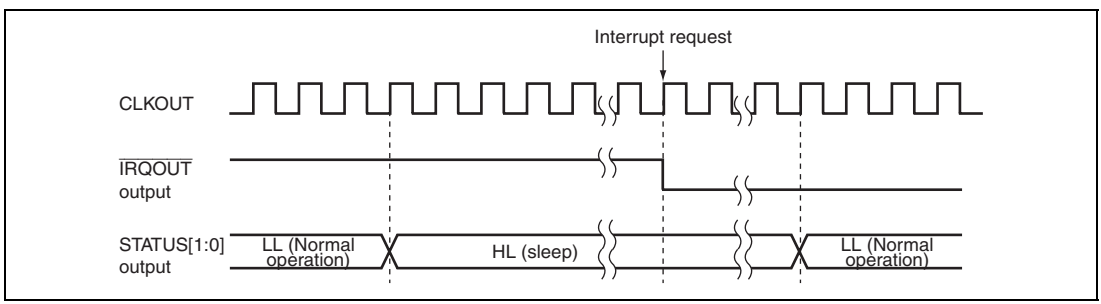


Figure 18.4 STATUS Output when an Interrupt Occurs in Sleep Mode

Section 19 Timer Unit (TMU)

This LSI includes an on-chip 32-bit timer unit (TMU), which has six channels (channels 0 to 5).

19.1 Features

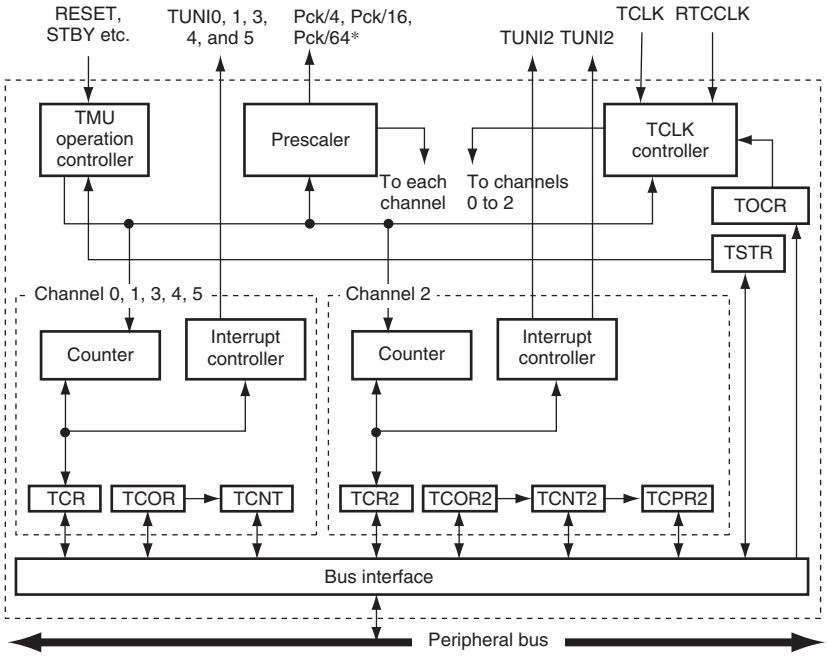
The TMU has the following features.

- Auto-reload type 32-bit down-counter provided for each channel
- Input capture function provided in channel 2
- Selection of rising edge or falling edge as external clock input edge when external clock is selected or input capture function is used for each channel
- 32-bit timer constant register for auto-reload use, readable/writable at any time, and 32-bit down-counter provided for each channel
- Selection of seven counter input clocks:

  External clock (TMU_TCLK) for channel 0 to 2 only, RTC clock (RTCCLK) and five peripheral clocks (Pck0/4, Pck0/16, Pck0/64, Pck0/256, and Pck0/1024) (Pck0 is the peripheral clock0) for each channel.
- Two interrupt sources

  One underflow source (each channel) and one input capture source (channel 2)

Figure 19.1 shows a block diagram of the TMU.



Note: * Internal signals with 1/4, 1/16, or 1/64 of the Pck frequency and supplied to the on-chip peripheral modules.

[Legend]
TSTR: Timer start register
TCOR: Timer constant register
TCNT: Timer counter
TCR: Timer control register
TCPR2: Input capture register 2 (only in channel 2)

Figure 19.1 Block Diagram of TMU

19.2 Input/Output Pins

Table 19.1 shows the TMU pin configuration.

Table 19.1 Pin Configuration

Pin Name	Function	I/O	Description
TMU_TCLK	Clock input	Input	Channel 0, 1 and 2 external clock input pin/channel 2 input capture control input pin

19.3 Register Descriptions

Table 19.2 shows register configuration. Table 19.3 shows the register states in each operating mode.

Table 19.2 Register Configuration

Channel	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size
0,1,2 Common	Timer output control register	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8
	Timer start register 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8
0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32
3,4,5 Common	Timer start register 1	TSTR1	R/W	H'FFD8 8004	H'1FD8 8004	8
3	Timer constant register 3	TCOR3	R/W	H'FFD8 8008	H'1FD8 8008	32
	Timer counter 3	TCNT3	R/W	H'FFD8 800C	H'1FD8 800C	32
	Timer control register 3	TCR3	R/W	H'FFD8 8010	H'1FD8 8010	16
4	Timer constant register 4	TCOR4	R/W	H'FFD8 8014	H'1FD8 8014	32
	Timer counter 4	TCNT4	R/W	H'FFD8 8018	H'1FD8 8018	32
	Timer control register 4	TCR4	R/W	H'FFD8 801C	H'1FD8 801C	16
5	Timer constant register 5	TCOR5	R/W	H'FFD8 8020	H'1FD8 8020	32
	Timer counter 5	TCNT5	R/W	H'FFD8 8024	H'1FD8 8024	32
	Timer control register 5	TCR5	R/W	H'FFD8 8028	H'1FD8 8028	16

Table 19.3 Register States in Each Operating Mode

Channel	Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Standby
0,1,2 Common	Timer output control register	TOCR	H'00	H'00	Retained	Retained
	Timer start register 0	TSTR0	H'00	H'00	Retained	Retained
0	Timer constant register 0	TCOR0	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 0	TCNT0	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 0	TCR0	H'0000	H'0000	Retained	Retained
1	Timer constant register 1	TCOR1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 1	TCNT1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 1	TCR1	H'0000	H'0000	Retained	Retained
2	Timer constant register 2	TCOR2	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 2	TCNT2	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 2	TCR2	H'0000	H'0000	Retained	Retained
	Input capture register 2	TCPR2	H'XXXX XXXX	H'XXXX XXXX	Retained	Retained
3,4,5 Common	Timer start register 1	TSTR1	H'00	H'00	Retained	Retained
3	Timer constant register3	TCOR3	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 3	TCNT3	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 3	TCR3	H'0000	H'0000	Retained	Retained
4	Timer constant register 4	TCOR4	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 4	TCNT4	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 4	TCR4	H'0000	H'0000	Retained	Retained
5	Timer constant register 5	TCOR5	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer counter 5	TCNT5	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
	Timer control register 5	TCR5	H'0000	H'0000	Retained	Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{PRESET}}$ pin.

19.3.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit read-only register that specifies whether external pin TMU_TCLK is used as the external clock or input capture control input pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCOE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCOE	0	R/W	Timer Clock Pin Control (TCOE) Specifies whether timer clock pin (TMU_TCLK) is used as the external clock or input capture control input pin. 0: TMU_TCLK is used as external clock input or input capture control input pin. 1: Invalid

19.3.2 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that specifies whether TCNT in each channel is operated or stopped.

- TSTR0

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Specifies whether TCNT2 is operated or stopped. 0: TCNT2 count operation is stopped 1: TCNT2 performs count operation
1	STR1	0	R/W	Counter Start 1 Specifies whether TCNT1 is operated or stopped. 0: TCNT1 count operation is stopped 1: TCNT1 performs count operation
0	STR0	0	R/W	Counter Start 0 Specifies whether TCNT0 is operated or stopped. 0: TCNT0 count operation is stopped 1: TCNT0 performs count operation

- TSTR1

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR5	STR4	STR3
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR5	0	R/W	Counter Start 5 Specifies whether TCNT5 is operated or stopped. 0: TCNT5 count operation is stopped 1: TCNT5 performs count operation
1	STR4	0	R/W	Counter Start 4 Specifies whether TCNT4 is operated or stopped. 0: TCNT4 count operation is stopped 1: TCNT4 performs count operation
0	STR3	0	R/W	Counter Start 3 Specifies whether TCNT3 is operated or stopped. 0: TCNT3 count operation is stopped 1: TCNT3 performs count operation

19.3.3 Timer Constant Register (TCORn) (n = 0 to 5)

The TCOR registers are 32-bit readable/writable registers. When a TCNT counter underflows while counting down, the TCOR value is set in that TCNT, which continues counting down from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

19.3.4 Timer Counter (TCNTn) (n = 0 to 5)

The TCNT registers are 32-bit readable/writable registers. Each TCNT counts down on the input clock selected by the TPSC2 to TPSC0 bits in TCR.

When a TCNT counter underflows while counting down, the UNF flag is set in TCR of the corresponding channel. At the same time, the TCOR value is set in TCNT, and the count-down operation continues from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

19.3.5 Timer Control Registers (TCRn) (n = 0 to 5)

The TCR registers are 16-bit readable/writable registers. Each TCR selects the count clock, specifies the edge when an external clock is selected, and controls interrupt generation when the flag indicating TCNT underflow is set to 1. TCR2 is also used for input capture control and control of interrupt generation in the event of input capture.

- TCR0, TCR1, TCR3, TCR4 and TCR5

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UNF	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- TCR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ICPF	UNF	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ICPF*[1]	0	R/W	Input Capture Interrupt Flag Status flag, provided in channel 2 only, which indicates the occurrence of input capture. 0: Input capture has not occurred [Clearing condition] When 0 is written to ICPF 1: Input capture has occurred [Setting condition] When input capture occurs*[2]
8	UNF	0	R/W	Underflow Flag Status flag that indicates the occurrence of TCNT underflow. 0: TCNT has not underflowed [Clearing condition] When 0 is written to UNF 1: TCNT has underflowed [Setting condition] When TCNT underflows*[2]

Bit	Bit Name	Initial Value	R/W	Description
7	ICPE1*[1]	0	R/W	Input Capture Control
6	ICPE0*[1]	0	R/W	These bits, provided in channel 2 only, specify whether the input capture function is used, and control enabling or disabling of interrupt generation when the function is used. The CKEG bits specify whether the rising edge or falling edge of the TCLK pin is used to set the TCNT2 value in TCPR2. The TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. When the ICPF bit is 1, TCPR2 is not set in the event of input capture. 00: Input capture function is not used. 01: Setting prohibited 10: Input capture function is used, but interrupt due to input capture (TICPI2) is not enabled. Data transfer request is sent to the DMAC in the event of input capture. 11: Input capture function is used, and interrupt due to input capture (TICPI2) is enabled.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling or disabling of interrupt generation when the UNF status flag is set to 1, indicating TCNT underflow. 0: Interrupt due to underflow (TUNI) is disabled 1: Interrupt due to underflow (TUNI) is enabled
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock input edge when an external clock is selected or the input capture function is used. 00: Count/input capture register set on rising edge 01: Count/input capture register set on falling edge 1X: Count/input capture register set on both rising and falling edges

Bit	Bit Name	Initial Value	R/W	Description
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT count clock.
0	TPSC0	0	R/W	000: Counts on Pck0/4
				001: Counts on Pck0/16
				010: Counts on Pck0/64
				011: Counts on Pck0/256
				100: Counts on Pck0/1024
				101: Setting prohibited
				110: Counts on on-chip RTC output clock (RCTCLK)
				111: Counts on external clock (TCLK) *3

Notes: X: Don't care
1. Reserved bit in channel 0 or 1 (initial value is 0, and can only be read).
2. Writing 1 does not change the value; the previous value is retained.
3. Do not set in channels 3, 4, and 5

19.3.6 Input Capture Register 2 (TCPR2)

TCPR2 is a 32-bit read-only register for use with the input capture function, provided only in channel 2. The input capture function is controlled by means of the ICPE and CKEG bits in TCR2. When input capture occurs, the TCNT2 value is copied into TCPR2. The value is set in TCPR2 only when the ICPF bit in TCR2 is 0.

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

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

19.4 Operation

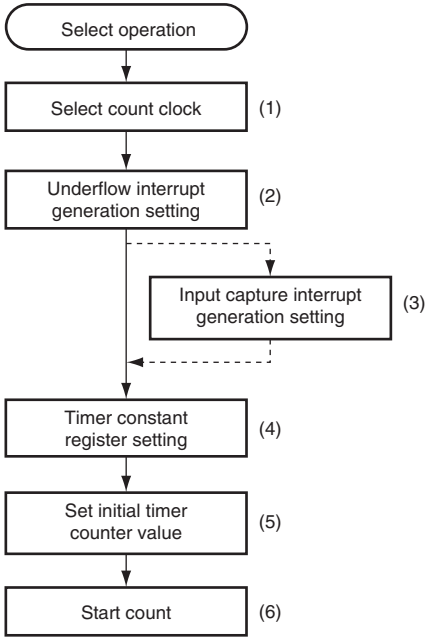
Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). Each TCNT performs count-down operation. The channels have an auto-reload function that allows cyclic count operations, and can also perform external event counting. Channel 2 also has an input capture function.

19.4.1 Counter Operation

When one of bits STR0 to STR2 in TSTR is set to 1, the TCNT for the corresponding channel starts counting. When TCNT underflows, the UNF flag in TCR is set. If the UNIE bit in TCR is set to 1 at this time, an interrupt request is sent to the CPU. At the same time, the value is copied from TCOR into TCNT, and the count-down continues (auto-reload function).

(1) Example of Count Operation Setting Procedure

Figure 19.2 shows an example of the count operation setting procedure.



(1) Select the count clock with the TPSC2 to TPSC0 bits in TCR. When the external clock (TCLK) is selected, specify the external clock edge with the CKEG1 and CKEG0 bits in TCR.

(2) Specify whether an interrupt is to be generated on TCNT underflow with the UNIE bit in TCR.

(3) When the input capture function is used, set the ICPE bits in TCR, including specification of whether the interrupt function is to be used.

(4) Set a value in TCOR.

(5) Set the initial value inTCNT.

(6) Set the STR bit to 1 in TSTR to start the count.

Note: When an interrupt is generated, clear the source flag in the interrupt handler.
If the interrupt enabled state is set without clearing the flag, another interrupt will be generated.

Figure 19.2 Example of Count Operation Setting Procedure

(2) Auto-Reload Count Operation

Figure 19.3 shows the TCNT auto-reload operation.

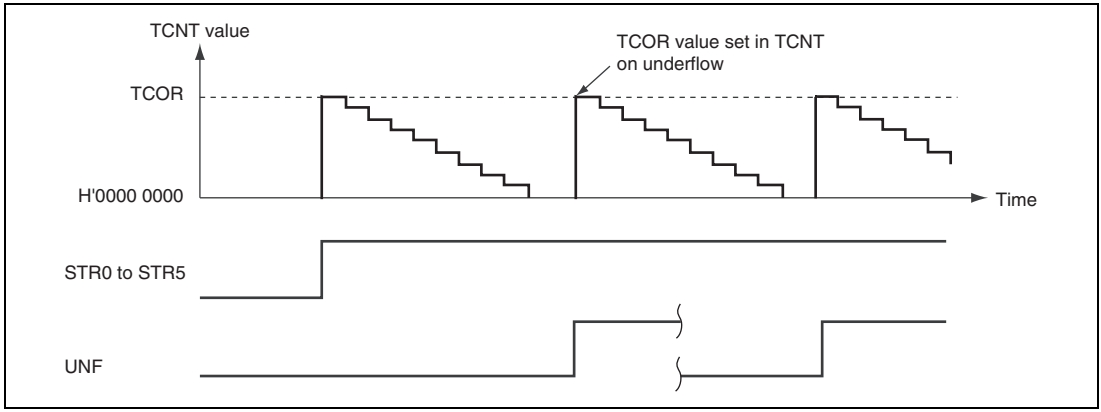


Figure 19.3 TCNT Auto-Reload Operation

(3) TCNT Count Timing

- Operating on internal clock

  Any of five count clocks (Pck0/4, Pck0/16, Pck0/64, Pck0/256, or Pck0/1024) scaled from the peripheral clock can be selected as the count clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 19.4 shows the timing in this case.

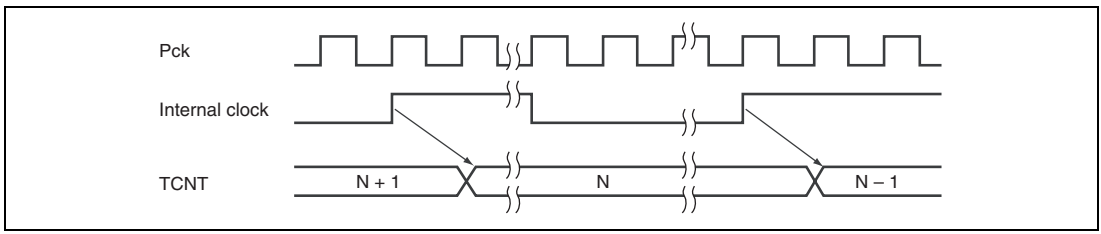


Figure 19.4 Count Timing when Operating on Internal Clock

- Operating on external clock

  In channels 0, 1, and 2, the external clock pin (TCLK) input can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR. The detected edge (rising, falling, or both edges) can be selected with the CKEG1 and CKEG0 bits in TCR.

  Figure 19.5 shows the timing for both-edge detection.

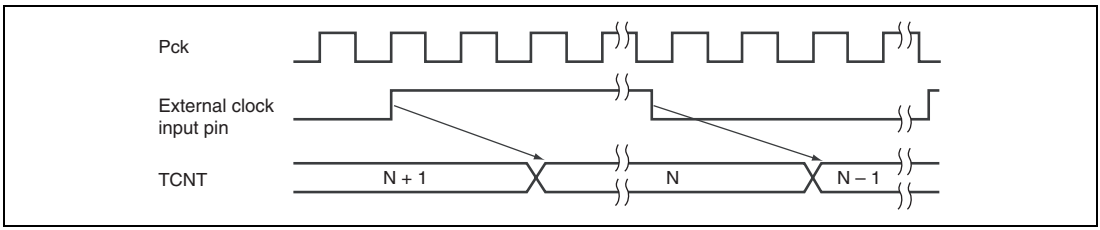


Figure 19.5 Count Timing when Operating on External Clock

- Operating on on-chip RTC output clock

  The on-chip RTC output clock can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 19.6 shows the timing for both-edge detection.

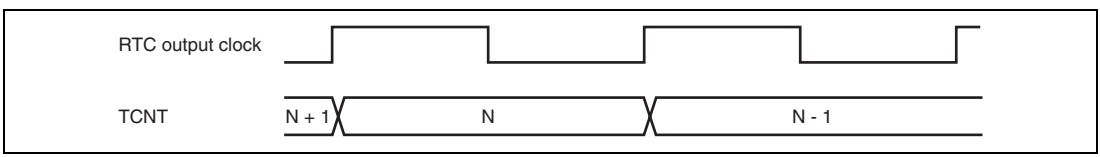


Figure 19.6 Count Timing when Operating on on-chip RTC output Clock

19.4.2 Input Capture Function

Channel 2 has an input capture function.

The procedure for using the input capture function is as follows:

1. Use bits TPSC2 to TPSC0 in TCR to set an internal clock as the timer operating clock.
2. Use bits IPCE1 and IPCE0 in TCR to specify use of the input capture function, and whether interrupts are to be generated when this function is used.
3. Use bits CKEG1 and CKEG0 in TCR to specify whether the rising or falling edge of the TCLK pin is to be used to set the TCNT value in TCPR2.

When input capture occurs, the TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. A new DMAC transfer request is not generated until processing of the previous request is finished.

Figure 19.7 shows the operation timing when the input capture function is used (with TCLK rising edge detection).

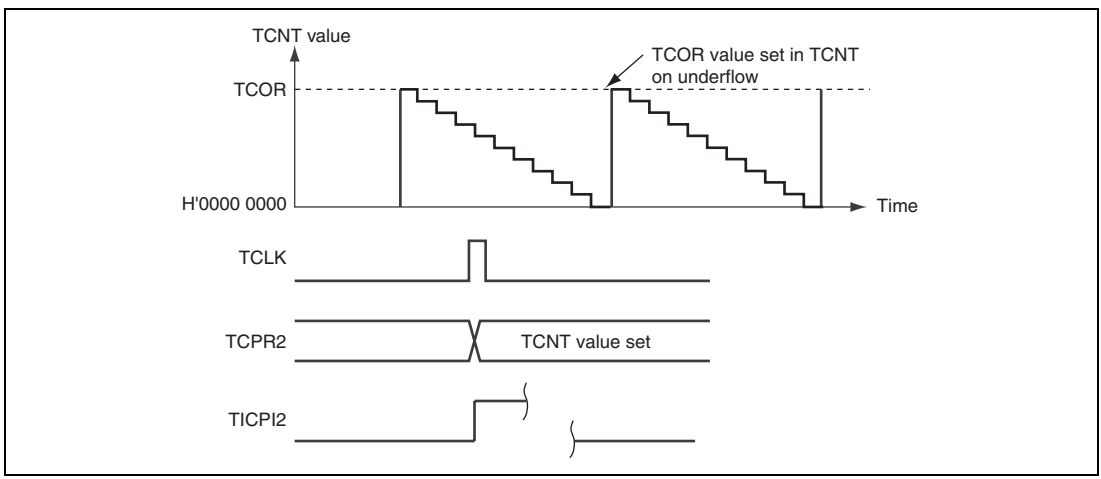


Figure 19.7 Operation Timing when Using Input Capture Function

19.5 Interrupts

There are seven TMU interrupt sources: underflow interrupts and the input capture interrupt when the input capture function is used. Underflow interrupts are generated on each of the channels, and input capture interrupts on channel 2 only.

An underflow interrupt request is generated (for each channel) when both the UNF bit and the interrupt enable bit (UNIE) for that channel are set to 1.

When the input capture function is used and an input capture request is generated, an interrupt is requested if the ICPF bit in TCR2 is 1 and the input capture control bits (ICPE1 and ICPE0) in TCR2 are both set to 11.

The TMU interrupt sources are summarized in Table 19.4.

Table 19.4 TMU Interrupt Sources

Channel	Interrupt Source	Description
0	TUNI0	Underflow interrupt 0
1	TUNI1	Underflow interrupt 1
2	TUNI2	Underflow interrupt 2
	TICPI2	Input capture interrupt 2
3	TUNI3	Underflow interrupt 3
4	TUNI4	Underflow interrupt 4
5	TUNI5	Underflow interrupt 5

19.6 Usage Notes

19.6.1 Register Writes

When writing to a TMU register, timer count operation must be stopped by clearing the start bit (STR5 to STR0) for the relevant channel in TSTR.

Note that TSTR can be written to, and the UNF and ICPF bits in TCR can be cleared while the count is in progress. When the flags (UNF and ICPF) are cleared while the count is in progress, make sure not to change the values of bits other than those being cleared.

19.6.2 Reading from TCNT

Reading from TCNT is performed synchronously with the timer count operation. Note that when the timer count operation is performed simultaneously with reading from a register, the synchronous processing causes the TCNT value before the count-down operation to be read as the TCNT value.

19.6.3 External Clock Frequency

Ensure that the external clock (TMU_TCLK) frequency for channels 0, 1 and 2 does not exceed Pck0/4.

Section 20 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises four 16-bit timer channels.

20.1 Features

- Maximum 4-pulse output
  - A total of 16 timer general registers (TGRA to TGRD × 4 ch.) are provided (four each for channels). TGRA can be set as an output compare register.
  - TGRB, TGRC, and TGRD for each channel can also be used as timer counter clearing registers. TGRC and TGRD can also be used as buffer registers.
- Selection of four counter input clocks for channels 0 and 1, and of six counter input clocks for channels 2 and 3.
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Counter clear operation: Counter clearing possible by compare match
  - PWM mode: Any PWM output duty can be set
    Maximum of 4-phase PWM output possible
- Buffer operation settable for each channel
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 2, and 3
  - Two-phase encoder pulse up/down-count possible
- An interrupt request for each channel
  - For channels 0 and 1, compare match interrupts and overflow interrupts can be requested independently
  - For channels 2, and 3, compare match interrupts, overflow interrupts, and underflow interrupts can be requested independently

Table 20.1 lists the functions of the TPU.

Table 20.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3
Count clock		Pck0/1	Pck0/1	Pck0/1	Pck0/1
		Pck0/4	Pck0/4	Pck0/4	Pck0/4
		Pck0/16	Pck0/16	Pck0/16	Pck0/16
		Pck0/64	Pck0/64	Pck0/64	Pck0/64
		—	—	TPU_TI2A	TPU_TI3A
		—	—	TPU_TI2B	TPU_TI3B
General registers		TGR0A	TGR1A	TGR2A	TGR3A
		TGR0B	TGR1B	TGR2B	TGR3B
General registers/ buffer registers		TGR0C	TGR1C	TGR2C	TGR3C
		TGR0D	TGR1D	TGR2D	TGR3D
Output pins		TPU_TO0	TPU_TO1	TPU_TO2	TPU_TO3
Counter clear function		TGR compare match	TGR compare match	TGR compare match	TGR compare match
Compare match output	0 output	◯	◯	◯	◯
	1 output	◯	◯	◯	◯
	Toggle output	◯	◯	◯	◯
PWM mode		◯	◯	◯	◯
Phase counting mode		—	—	◯	◯
Buffer operation		◯	◯	◯	◯
Interrupt sources		5 sources • Compare match • Overflow	5 sources • Compare match • Overflow	6 sources • Compare match • Overflow • Underflow	6 sources • Compare match • Overflow • Underflow

[Legend]

◯: Possible

—: Not possible

Note: TPU_TI2B and TPU_TI3B are used as count clocks only in phase counting mode.

Figure 20.1 shows a block diagram of the TPU.

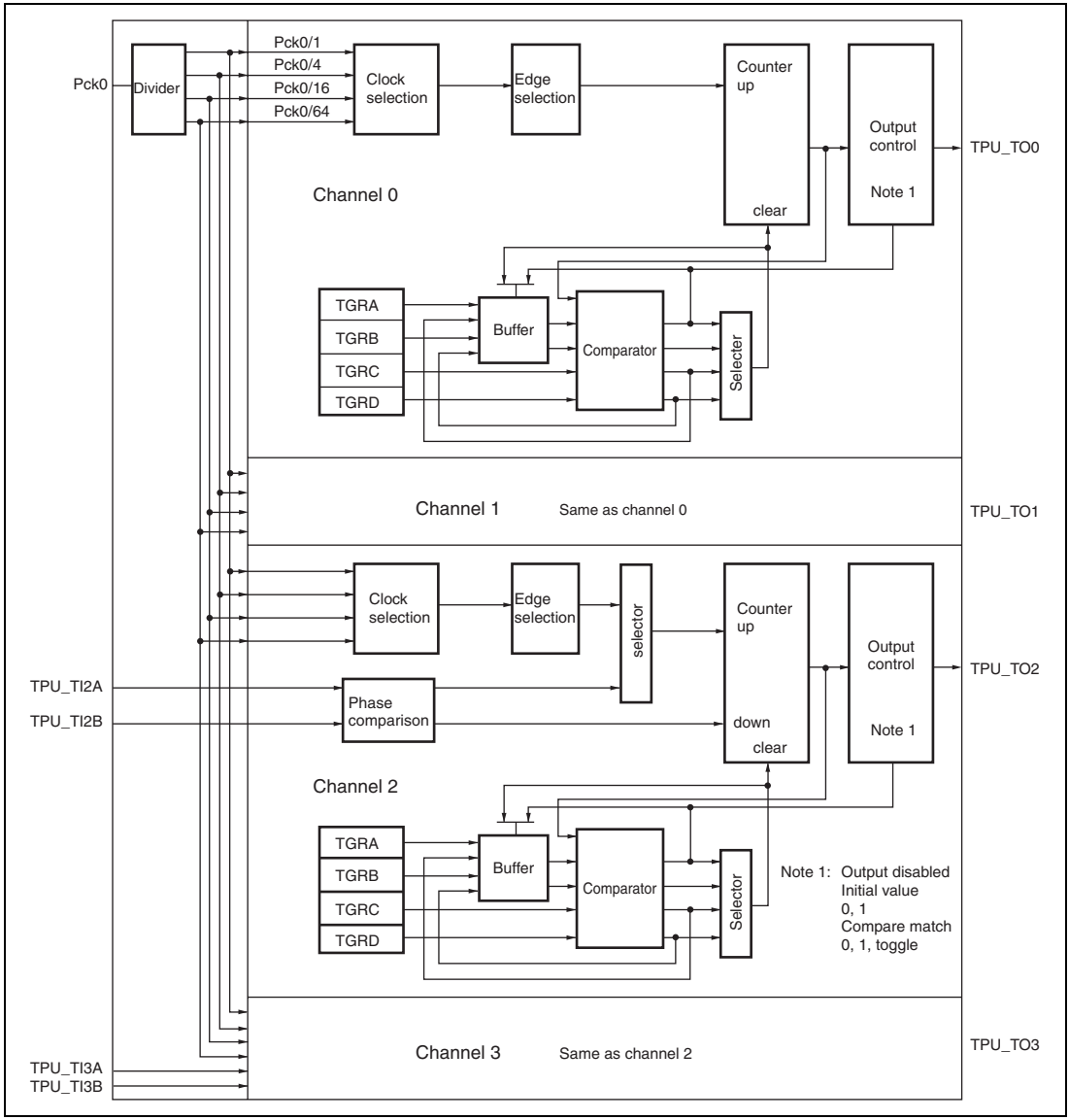


Figure 20.1 Block Diagram of TPU

20.2 Input/Output Pins

Table 20.2 summarizes the TPU related external pins.

Table 20.2 TPU Pin Configurations

Channel	Name	Pin Name	I/O	Function
0	Output compare match 0	TPU_TO0	Output	TGR0A output compare output/PWM output pin
1	Output compare match 1	TPU_TO1	Output	TGR1A output compare output/PWM output pin
2	Output compare match 2A	TPU_TO2	Output	TGR2A output compare output/PWM output pin
	Clock input 2A	TPU_TI2A	Input	External clock channel 2A input pin /channel 2 counting mode A phase input
	Clock input 2B	TPU_TI2B	Input	Channel 2 counting mode B phase input
3	Output compare match 3A	TPU_TO3	Output	TGR3A output compare output/PWM output pin
	Clock input 3A	TPU_TI3A	Input	External clock channel 3A input pin /channel 3 counting mode A phase input
	Clock input 3B	TPU_TI3B	Input	Channel 3 counting mode B phase input

20.3 Register Descriptions

Table 20.3 shows the TPU register configuration. Table 20.4 shows the register state in each operating mode.

Table 20.3 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Timer start register	TSTR	R/W	H'FFE2 8000	H'1FE2 8000	16
Timer control register_0	TCR_0	R/W	H'FFE2 8010	H'1FE2 8010	16
Timer mode register_0	TMDR_0	R/W	H'FFE2 8014	H'1FE2 8014	16
Timer I/O control register_0	TIOR_0	R/W	H'FFE2 8018	H'1FE2 8018	16
Timer interrupt enable register_0	TIER_0	R/W	H'FFE2 801C	H'1FE2 801C	16
Timer status register_0	TSR_0	R/W	H'FFE2 8020	H'1FE2 8020	16
Timer counter_0	TCNT_0	R	H'FFE2 8024	H'1FE2 8024	16
Timer general register A_0	TGRA_0	R	H'FFE2 8028	H'1FE2 8028	16
Timer general register B_0	TGRB_0	R	H'FFE2 802C	H'1FE2 802C	16
Timer general register C_0	TGRC_0	R	H'FFE2 8030	H'1FE2 8030	16
Timer general register D_0	TGRD_0	R	H'FFE2 8034	H'1FE2 8034	16
Timer control register_1	TCR_1	R/W	H'FFE2 8050	H'1FE2 8050	16
Timer mode register_1	TMDR_1	R/W	H'FFE2 8054	H'1FE2 8054	16
Timer I/O control register_1	TIOR_1	R/W	H'FFE2 8058	H'1FE2 8058	16
Timer interrupt enable register_1	TIER_1	R/W	H'FFE2 805C	H'1FE2 805C	16
Timer status register_1	TSR_1	R/W	H'FFE2 8060	H'1FE2 8060	16
Timer counter_1	TCNT_1	R	H'FFE2 8064	H'1FE2 8064	16
Timer general register A_1	TGRA_1	R	H'FFE2 8068	H'1FE2 8068	16
Timer general register B_1	TGRB_1	R	H'FFE2 806C	H'1FE2 806C	16
Timer general register C_1	TGRC_1	R	H'FFE2 8070	H'1FE2 8070	16
Timer general register D_1	TGRD_1	R	H'FFE2 8074	H'1FE2 8074	16
Timer control register_2	TCR_2	R/W	H'FFE2 8090	H'1FE2 8090	16
Timer mode register_2	TMDR_2	R/W	H'FFE2 8094	H'1FE2 8094	16
Timer I/O control register_2	TIOR_2	R/W	H'FFE2 8098	H'1FE2 8098	16

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Timer interrupt enable register_2	TIER_2	R/W	H'FFE2 809C	H'1FE2 809C	16
Timer status register_2	TSR_2	R/W	H'FFE2 80A0	H'1FE2 80A0	16
Timer counter_2	TCNT_2	R	H'FFE2 80A4	H'1FE2 80A4	16
Timer general register A_2	TGRA_2	R	H'FFE2 80A8	H'1FE2 80A8	16
Timer general register B_2	TGRB_2	R	H'FFE2 80AC	H'1FE2 80AC	16
Timer general register C_2	TGRC_2	R	H'FFE2 80B0	H'1FE2 80B0	16
Timer general register D_2	TGRD_2	R	H'FFE2 80B4	H'1FE2 80B4	16
Timer control register_3	TCR_3	R/W	H'FFE2 80D0	H'1FE2 80D0	16
Timer mode register_3	TMDR_3	R/W	H'FFE2 80D4	H'1FE2 80D4	16
Timer I/O control register_3	TIOR_3	R/W	H'FFE2 80D8	H'1FE2 80D8	16
Timer interrupt enable register_3	TIER_3	R/W	H'FFE2 80DC	H'1FE2 80DC	16
Timer status register_3	TSR_3	R/W	H'FFE2 80E0	H'1FE2 80E0	16
Timer counter_3	TCNT_3	R	H'FFE2 80E4	H'1FE2 80E4	16
Timer general register A_3	TGRA_3	R	H'FFE2 80E8	H'1FE2 80E8	16
Timer general register B_3	TGRB_3	R	H'FFE2 80EC	H'1FE2 80EC	16
Timer general register C_3	TGRC_3	R	H'FFE2 80F0	H'1FE2 80F0	16
Timer general register D_3	TGRD_3	R	H'FFE2 80F4	H'1FE2 80F4	16

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 20.4 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Timer start register	TSTR	H'0000	H'0000	Retained	Retained
Timer control register_0	TCR_0	H'0000	H'0000	Retained	Retained
Timer mode register_0	TMDR_0	H'0000	H'0000	Retained	Retained
Timer I/O control register_0	TIOR_0	H'0000	H'0000	Retained	Retained
Timer interrupt enable register_0	TIER_0	H'0000	H'0000	Retained	Retained
Timer status register_0	TSR_0	H'0000	H'0000	Retained	Retained
Timer counter_0	TCNT_0	H'0000	H'0000	Retained	Retained
Timer general register A_0	TGRA_0	H'FFFF	H'FFFF	Retained	Retained
Timer general register B_0	TGRB_0	H'FFFF	H'FFFF	Retained	Retained
Timer general register C_0	TGRC_0	H'FFFF	H'FFFF	Retained	Retained
Timer general register D_0	TGRD_0	H'FFFF	H'FFFF	Retained	Retained
Timer control register_1	TCR_1	H'0000	H'0000	Retained	Retained
Timer mode register_1	TMDR_1	H'0000	H'0000	Retained	Retained
Timer I/O control register_1	TIOR_1	H'0000	H'0000	Retained	Retained
Timer interrupt enable register_1	TIER_1	H'0000	H'0000	Retained	Retained
Timer status register_1	TSR_1	H'0000	H'0000	Retained	Retained
Timer counter_1	TCNT_1	H'0000	H'0000	Retained	Retained
Timer general register A_1	TGRA_1	H'FFFF	H'FFFF	Retained	Retained
Timer general register B_1	TGRB_1	H'FFFF	H'FFFF	Retained	Retained
Timer general register C_1	TGRC_1	H'FFFF	H'FFFF	Retained	Retained
Timer general register D_1	TGRD_1	H'FFFF	H'FFFF	Retained	Retained
Timer control register_2	TCR_2	H'0000	H'0000	Retained	Retained
Timer mode register_2	TMDR_2	H'0000	H'0000	Retained	Retained
Timer I/O control register_2	TIOR_2	H'0000	H'0000	Retained	Retained
Timer interrupt enable register_2	TIER_2	H'0000	H'0000	Retained	Retained
Timer status register_2	TSR_2	H'0000	H'0000	Retained	Retained
Timer counter_2	TCNT_2	H'0000	H'0000	Retained	Retained
Timer general register A_2	TGRA_2	H'FFFF	H'FFFF	Retained	Retained
Timer general register B_2	TGRB_2	H'FFFF	H'FFFF	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Timer general register C_2	TGRC_2	H'FFFF	H'FFFF	Retained	Retained
Timer general register D_2	TGRD_2	H'FFFF	H'FFFF	Retained	Retained
Timer control register_3	TCR_3	H'0000	H'0000	Retained	Retained
Timer mode register_3	TMDR_3	H'0000	H'0000	Retained	Retained
Timer I/O control register_3	TIOR_3	H'0000	H'0000	Retained	Retained
Timer interrupt enable register_3	TIER_3	H'0000	H'0000	Retained	Retained
Timer status register_3	TSR_3	H'0000	H'0000	Retained	Retained
Timer counter_3	TCNT_3	H'0000	H'0000	Retained	Retained
Timer general register A_3	TGRA_3	H'FFFF	H'FFFF	Retained	Retained
Timer general register B_3	TGRB_3	H'FFFF	H'FFFF	Retained	Retained
Timer general register C_3	TGRC_3	H'FFFF	H'FFFF	Retained	Retained
Timer general register D_3	TGRD_3	H'FFFF	H'FFFF	Retained	Retained

20.3.1 Timer Control Registers (TCR)

The TCR registers are 16-bit registers that control the TCNT channels. The TPU has four TCR registers, one for each of channels 0 to 3. The TCR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TCR register settings should be made only when TCNT operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
7 to 5	CCLR[2:0]	000	R/W	Counter Clear These bits select the TCNT counter clearing source. 000: TCNT clearing disabled 001: TCNT cleared by TGRA compare match 010: TCNT cleared by TGRB compare match 011: Reserved (setting prohibited) 100: TCNT clearing disabled 101: TCNT cleared by TGRC compare match 110: TCNT cleared by TGRD compare match 111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
4, 3	CKEG[1:0]	00	R/W	Clock Edge These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used this setting is ignored. 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges* [Legend] X: Don't care Note: * If Pck0/1 is selected for the input clock, operation is disabled.
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 20.5 shows the clock sources that can be set for each channel. For more in formation on count clock selection, see table 20.6.

Table 20.5 TPU Clock Sources

	Internal Clock				External Clock	
Channel	Pck0/1	Pck0/4	Pck0/16	Pck0/64	TPU_TI2A	TPU_TI3A
0	◯	◯	◯	◯		
1	◯	◯	◯	◯		
2	◯	◯	◯	◯	◯	
3	◯	◯	◯	◯		◯

[Legend]

◯ : Setting

Blank : No setting

Table 20.6 TPSC[2:0] (1)

Channel	TPSC[2]	TPSC[1]	TPSC[0]	Description
0	0	0	0	Internal clock: counts on Pck0/1 (Initial value)
			1	Internal clock: counts on Pck0/4
		1	0	Internal clock: counts on Pck0/16
			1	Internal clock: counts on Pck0/64
	1	*	*	Reserved (setting prohibited)

Table 20.6 TPSC[2:0] (2)

Channel	TPSC[2]	TPSC[1]	TPSC[0]	Description
1	0	0	0	Internal clock: counts on Pck0/1 (Initial value)
			1	Internal clock: counts on Pck0/4
		1	0	Internal clock: counts on Pck0/16
			1	Internal clock: counts on Pck0/64
	1	*	*	Reserved (setting prohibited)

Table 20.6 TPSC[2:0] (3)

Channel	TPSC2	TPSC1	TPSC0	Description
2	0	0	0	Internal clock: counts on Pck0/1 (Initial value)
			1	Internal clock: counts on Pck0/4
		1	0	Internal clock: counts on Pck0/16
			1	Internal clock: counts on Pck0/64
	1	0	0	External clock: counts on TPU_TI2A pin input
			1	Reserved (setting prohibited)
		1	*	

Table 20.6 TPSC[2:0] (4)

Channel	TPSC2	TPSC1	TPSC0	Description
3	0	0	0	Internal clock: counts on Pck0/1 (Initial value)
			1	Internal clock: counts on Pck0/4
		1	0	Internal clock: counts on Pck0/16
			1	Internal clock: counts on Pck0/64
	1	0	0	External clock: counts on TPU_TI3A pin input
			1	Reserved (setting prohibited)
		1	*	

Note: * Don't care

20.3.2 Timer Mode Registers (TMDR)

The TMDR registers are 16-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has four TMDR registers, one for each channel. The TMDR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TMDR register settings should be made only when TCNT operation is stopped.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFWT	BFB	BFA	—	MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
6	BFWT	0	R/W	Buffer Write Timing Specifies TGRA and TGRB update timing when TGRC and TGRD are used as a compare match buffer. When TGRC and TGRD are not used as a compare match buffer register, this bit does not function. 0: TGRA and TGRB are rewritten at compare match of each register. 1: TGRA and TGRB are rewritten in counter clearing.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation*
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2 to 0	MD[2:0]	000	R/W	Modes These bits are used to set the timer operating mode. 000: Normal operation 001: Reserved (setting prohibited) 010: PWM mode 011: Reserved (setting prohibited) 100: Phase counting mode 1 101: Phase counting mode 2 110: Phase counting mode 3 111: Phase counting mode 4

Note: * Operation when setting (BFWT, BFB, BFA) = (1, 1, 0) is the same as when setting (BFWT, BFB, BFA) = (1, 0, 1). However, when the BFB bit is set to 1 (TGRB and TGRD used together for buffer operation), the setting of (BFWT, BFB, BFA) = (1, 1, 1) should be made. In this case, the value set in TGRA should also be set in TGRC because TGRA and TGRC are also used together for buffer operation.

20.3.3 Timer I/O Control Registers (TIOR)

The TIOR registers are 16-bit registers that control the TPU_TO pin. The TPU has four TIOR registers, one for each channel. The TIOR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

TIOR register settings should be made only when TCNT operation is halted.

Care is required since TIOR is affected by the TMDR setting.

If the counting operation is halted, the initial value set by this register is output from the TPU_TO pin.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2 to 0	IOA[2:0]	000	R/W	I/O Control Bits IOA3 to IOA0 specify the functions of TGRA and the TPU_TO pin. For details, see table 20.7.

Table 20.7 IOA[2:0]

Channels	IOA[2]	IOA[1]	IOA[0]	Description	
0 to 3	0	0	0	Always 0 output (Initial value)	
			1	Initial output is 0 output for TPU_TO pin	0 output at TGRA compare match*
		1	0		1 output at TGRA compare match
			1		Toggle output TGRA at compare match*
	1	0	0	Always 1 output	
			1	Initial output is 1 output for TPU_TO pin	0 output at TGRA compare match
		1	0		1 output at TGRA compare match*
			1		Toggle output at TGRA compare match*

Note: * This setting is invalid in PWM mode.

20.3.4 Timer Interrupt Enable Registers (TIER)

The TIER registers are 16-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has four TIER registers, one for each channel. The TIER registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode or module standby.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TC1EU	TC1EV	TG1ED	TG1EC	TG1EB	TG1EA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	0	R	Reserved These bits are always read as 0 and cannot be modified.
5	TC1EU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests by the TCFU bit when the TCFU bit in TSR is set to 1 in phase counting mode of channels 2, and 3 (TCNT underflow). In channels 0 and 1, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests by TCFU disabled 1: Interrupt requests by TCFU enabled
4	TC1EV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests by the TCFV bit when the TCFV bit in TSR is set to 1 (TCNT overflow). 0: Interrupt requests by TCFV disabled 1: Interrupt requests by TCFV enabled
3	TG1ED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests by the TGFD bit when the TGFD bit in TSR is set to (TCNT and TGRD compare match). 0: Interrupt requests by TGFD disabled 1: Interrupt requests by TGFD enabled

Bit	Bit Name	Initial Value	R/W	Description
2	TG1EC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests by the TGFC bit when the TGFC bit in TSR is set to 1 (TCNT and TGRC compare match). 0: Interrupt requests by TGFC disabled 1: Interrupt requests by TGFC enabled
1	TG1EB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests by the TGFB bit when the TGFB bit in TSR is set to 1 (TCNT and TGRB compare match). 0: Interrupt requests by TGFB disabled 1: Interrupt requests by TGFB enabled
0	TG1EA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests by the TGFA bit when the TGFA bit in TSR is set to 1 (TCNT and TGRA compare match). 0: Interrupt requests by TGFA disabled 1: Interrupt requests by TGFA enabled

20.3.5 Timer Status Registers (TSR)

The TSR registers are 16-bit registers that indicate the status of each channel. The TPU has four TSR registers, one for each channel. The TSR registers are initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode or module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
7	TCFD	0	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in phase counting mode of channels 2, and 3. In channels 0 and 1, bit 7 is reserved. It is always read as 0 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 2, and 3 are set to phase counting mode. In channels 0 and 1, bit 5 is reserved. It is always read as 0 and cannot be modified. [Clearing condition] (Initial value) When 0 is written to TCFU after reading TCFU = 1 [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Clearing condition] When 0 is written to TCFV after reading TCFV = 1 [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)
3	TGFD	0	R/(W)*	Compare Flag D Status flag that indicates the occurrence of TGRD compare match. [Clearing conditions] When 0 is written to TGFD after reading TGFD = 1 [Setting conditions] When TCNT = TGRD
2	TGFC	0	R/(W)*	Compare Flag C Status flag that indicates the occurrence of TGRC compare match. [Clearing conditions] When 0 is written to TGFC after reading TGFC = 1 [Setting conditions] When TCNT = TGRC
1	TGFB	0	R/(W)*	Compare Flag B Status flag that indicates the occurrence of TGRB compare match. [Clearing conditions] When 0 is written to TGFB after reading TGFB = 1 [Setting conditions] When TCNT = TGRB

Bit	Bit Name	Initial Value	R/W	Description
0	TGFA	0	R/(W)*	Output Compare Flag A Status flag that indicates the occurrence of TGRA compare match. [Clearing conditions] When 0 is written to TGFA after reading TGFA = 1 [Setting conditions] When TCNT = TGRA

Note: * Only 0 can be written, to clear the flag.

20.3.6 Timer Counters (TCNT)

The TCNT registers are 16-bit counters. The TPU has four TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset.

The TCNT counters are not initialized in standby mode, sleep mode, or module standby.

20.3.7 Timer General Registers (TGR)

The TGR registers are 16-bit registers. The TPU has 16 TGR registers, four each for channels 0 and 3. TGRC and TGRD can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset. These registers are not initialized in standby mode, sleep mode, or module standby.

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

20.3.8 Timer Start Register (TSTR)

TSTR is a 16-bit readable/writable register that selects TCNT operation/stoppage for channels 0 to 3. TSTR is initialized to H'0000 by a reset, but not initialized in standby mode, sleep mode, or module standby.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CST3	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	CST3	0	R/W	Counter Start
2	CST2	0	R/W	These bits select operation or stoppage for TCNT.
1	CST1	0	R/W	0: TCNTn count operation is stopped)
0	CST0	0	R/W	1: TCNTn performs count operation n = 3 to 0

20.4 Operation

20.4.1 Overview

Operation in each mode is outlined below.

(1) Normal Operation

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

(2) Buffer Operation

When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR. For update timing from a buffer register, rewriting on compare match occurrence or on counter clearing can be selected.

(3) PWM Mode

In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

(4) Phase Counting Mode

In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins (TPU_TI2A and TPU_TI2B, or TPU_TI3A and TPU_TI3B) in channels 2, and 3. When phase counting mode is set, the corresponding TI pin functions as the clock pin, and TCNT performs up/down-counting.

This can be used for two-phase encoder pulse input.

20.4.2 Basic Functions

(1) Counter Operation

When one of bits CST[0:3] is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

(a) Example of count operation setting procedure

Figure 20.2 shows an example of the count operation setting procedure.

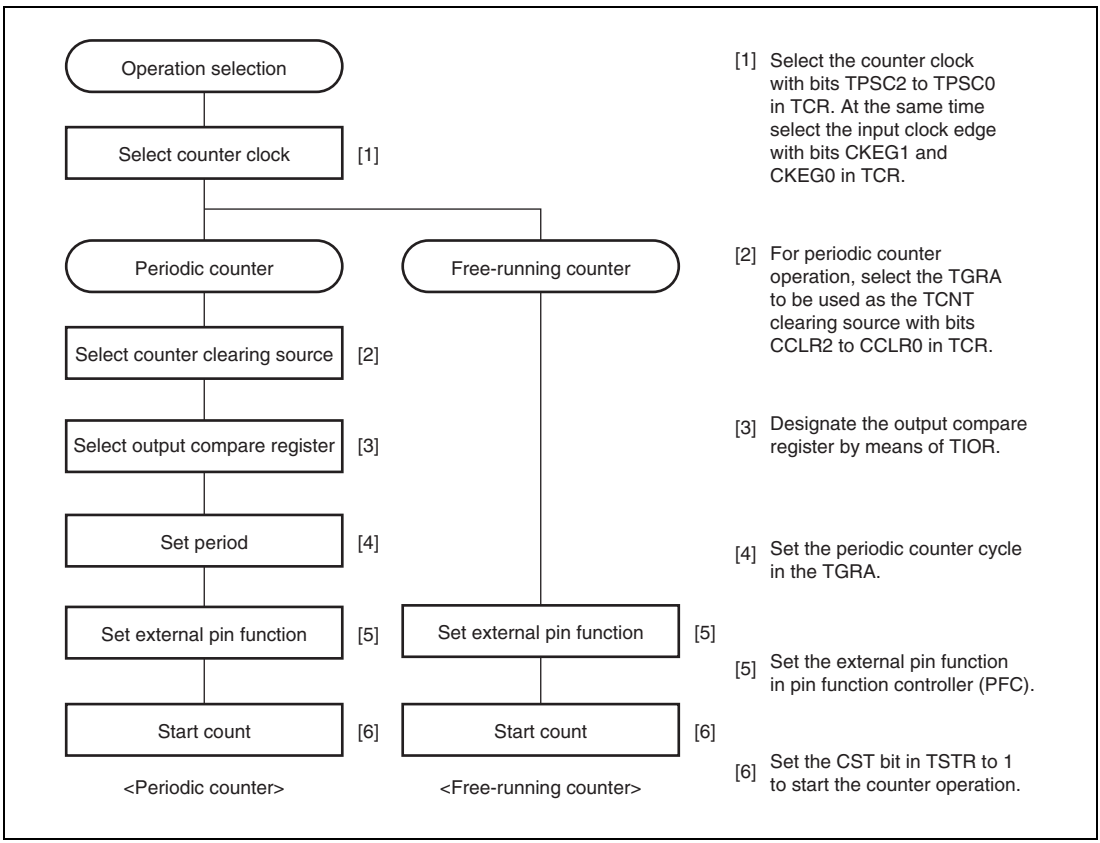


Figure 20.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. After overflow, TCNT starts counting up again from H'0000.

Figure 20.3 illustrates free-running counter operation.

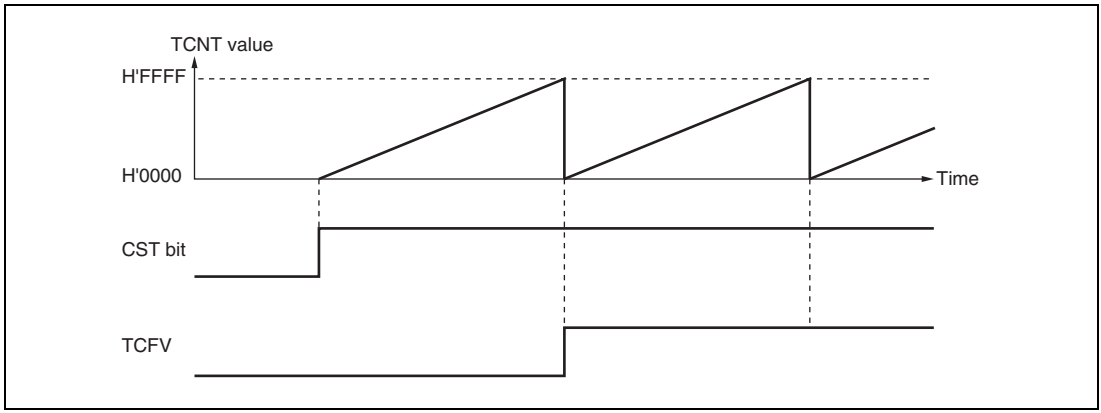


Figure 20.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

After a compare match, TCNT starts counting up again from H'0000.

Figure 20.4 illustrates periodic counter operation.

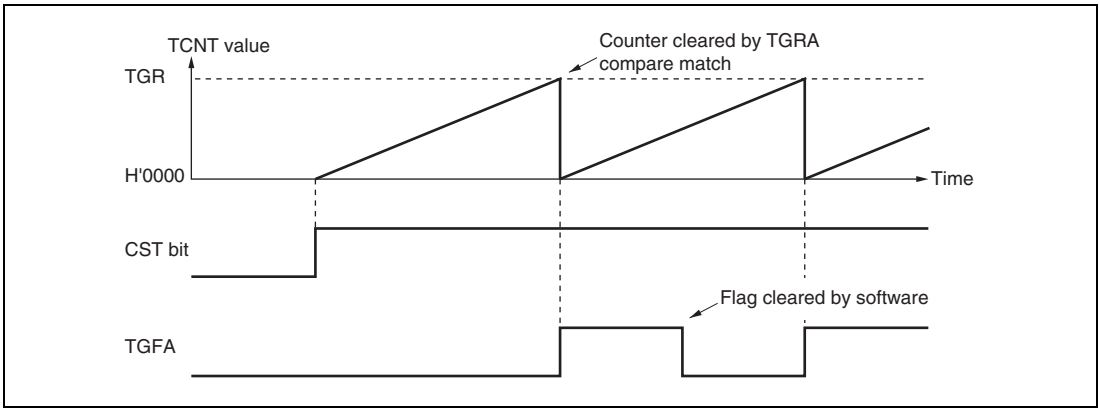


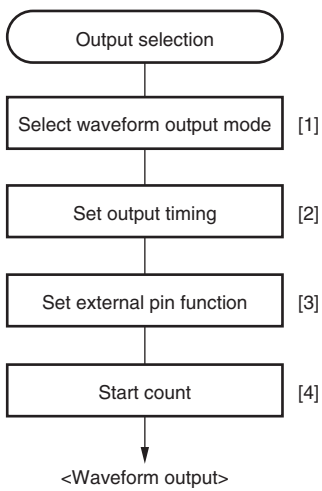
Figure 20.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin (TPU_TO pin) using TGRA compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 20.5 shows an example of the setting procedure for waveform output by compare match.



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TPU_TO pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGRA.

[3] Set the external pin function in pin function controller (PFC).

[4] Set the CST bit in TSTR to 1 to start the count operation.

Figure 20.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 20.6 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

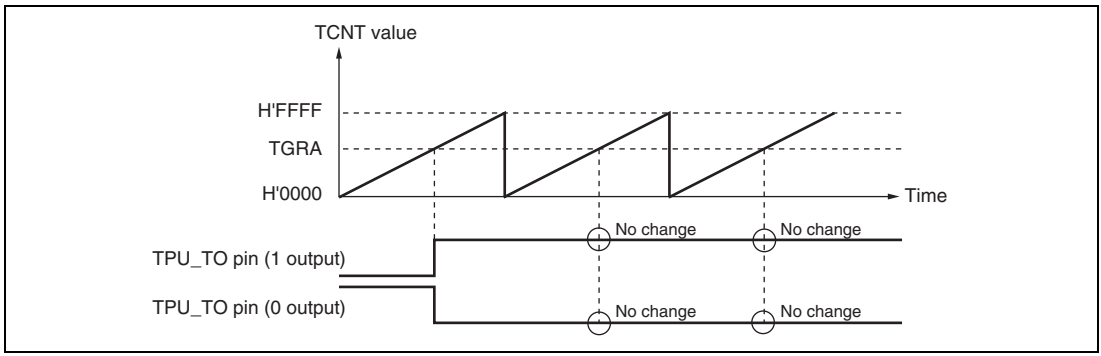


Figure 20.6 Example of 0 Output/1 Output Operation

Figure 20.7 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by compare match A.

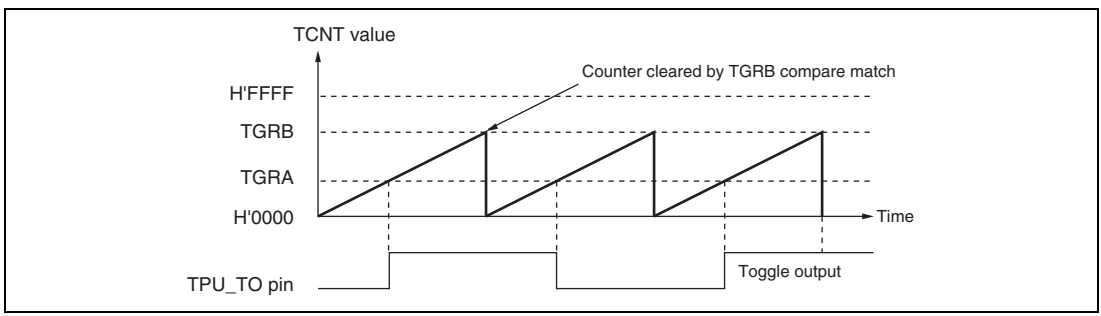


Figure 20.7 Example of Toggle Output Operation

20.4.3 Buffer Operation

Buffer operation, enables TGRC and TGRD to be used as buffer registers.

Table 20.8 shows the register combinations used in buffer operation.

Table 20.8 Register Combinations in Buffer Operation

Timer General Register	Buffer Register
TGRA	TGRC
TGRB	TGRD

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. For update timing from a buffer register, rewriting on compare match occurrence or on counter cleaning can be selected.

This operation is illustrated in figure 20.8.

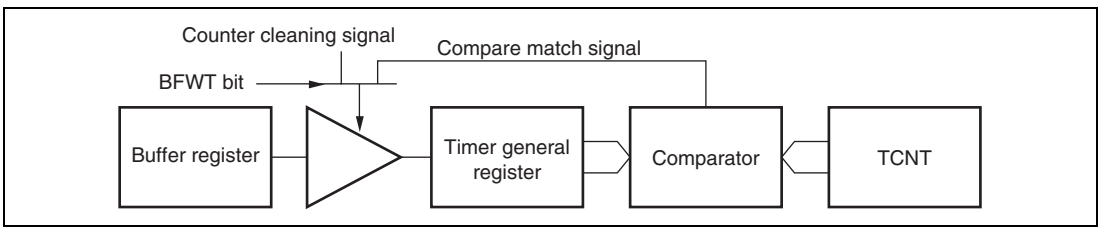


Figure 20.8 Compare Match Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 20.9 shows an example of the buffer operation setting procedure.

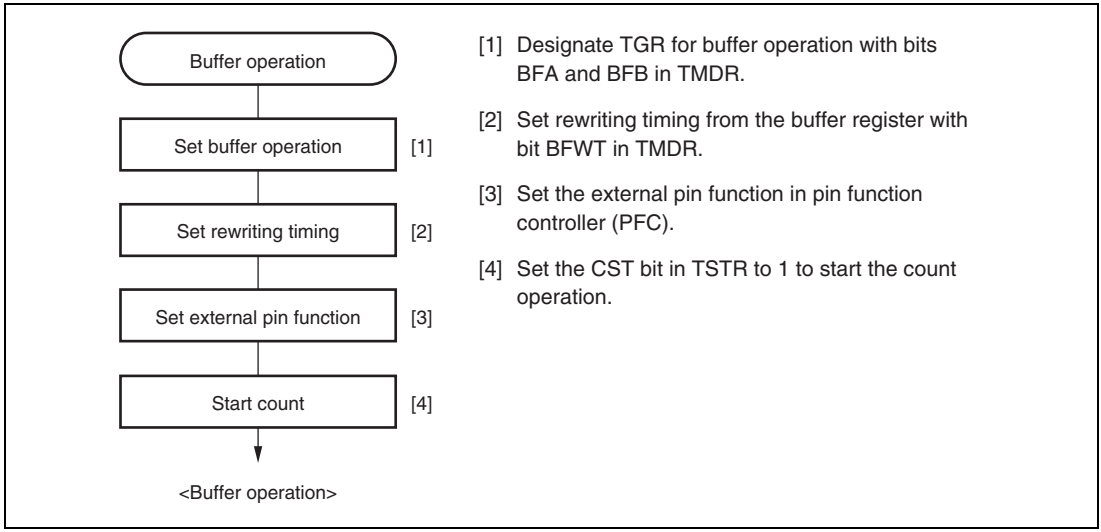


Figure 20.9 Example of Buffer Operation Setting Procedure

(2) Example of Buffer Operation

Figure 20.10 shows an operation example in which PWM mode has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A (TPU_TO pin), and 0 output at counter clearing. Rewriting timing from the buffer register is set at counter clearing.

As buffer operation has been set, when compare match A occurs the output changes. When counter clearing occurs by TGRB, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 20.4.4, PWM Modes.

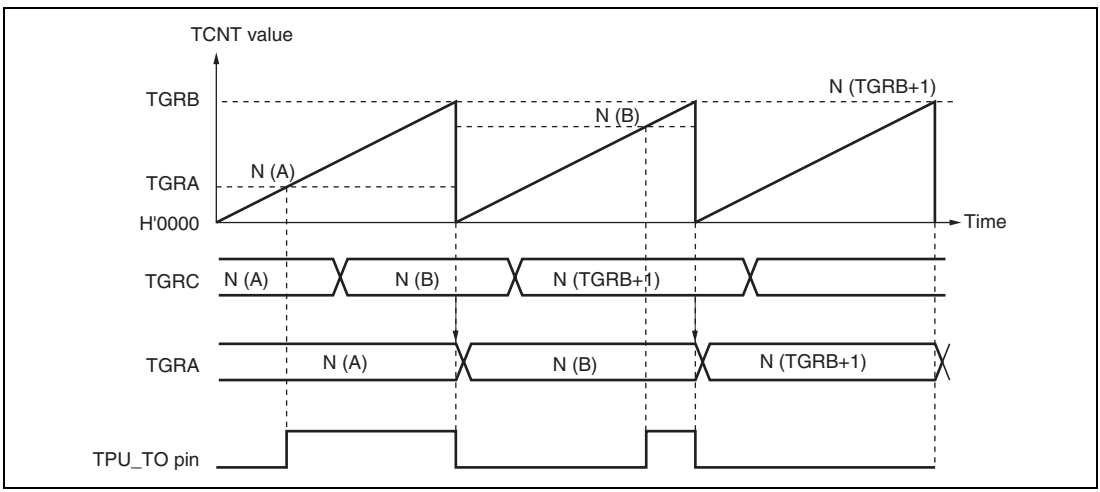


Figure 20.10 Example of Buffer Operation

20.4.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, or 1, output can be selected as the output level in response to compare match of each TGRA.

Designating TGRB compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently.

PWM output is generated from the TPU_TO pin using TGRB as the period register and TGRA as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a period register compare match, the output value of each pin is the initial value set in TIOR. Set TIOR so that the initial output and an output value by compare match are different. If the same levels or toggle outputs are selected, operation is disabled.

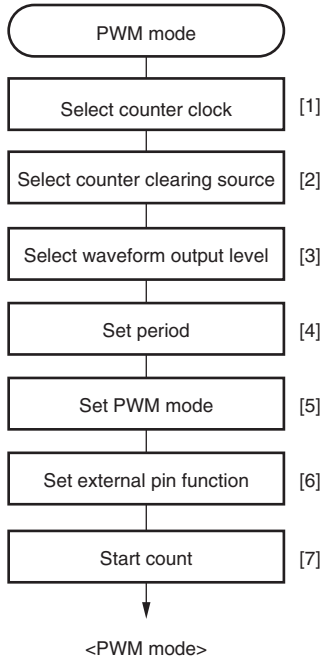
Conditions of duty 0% and 100% are shown below.

- Duty 0% : The set value of the period register (TGRB) is TGRA + 1 for the duty register (TGRA).
- Duty 100% : The set value of the duty register (TGRA) is 0.

In PWM mode 1, a maximum 4-phase PWM output is possible.

(1) Example of PWM Mode Setting Procedure

Figure 20.11 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGRB to be used as the TCNT clearing source.

[3] Use TIOR to select the initial value and output value.

[4] Set the period in TGRB, and set the duty in TGRA.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the external pin function in pin function controller (PFC).

[7] Set the CST bit in TSTR to 1 to start the count operation.

Figure 20.11 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 20.12 shows an example of PWM mode operation.

In this example, TGRB compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRA output value.

In this case, the value set in TGRB is used as the period, and the value set in TGRA as the duty.

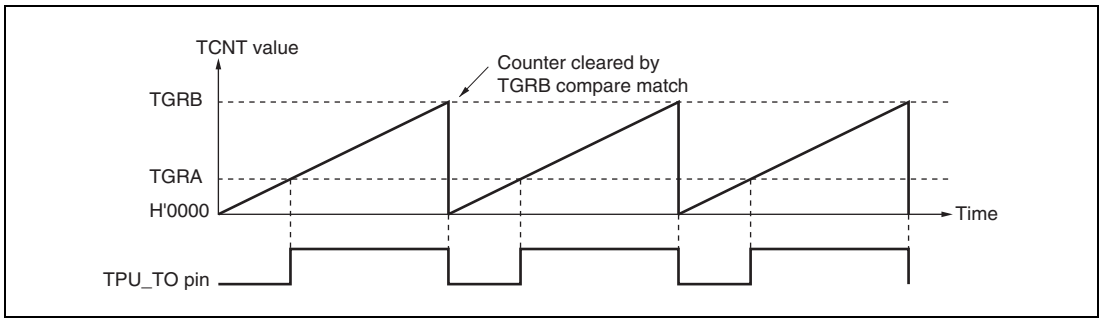


Figure 20.12 Example of PWM Mode Operation (1)

Figure 20.13 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

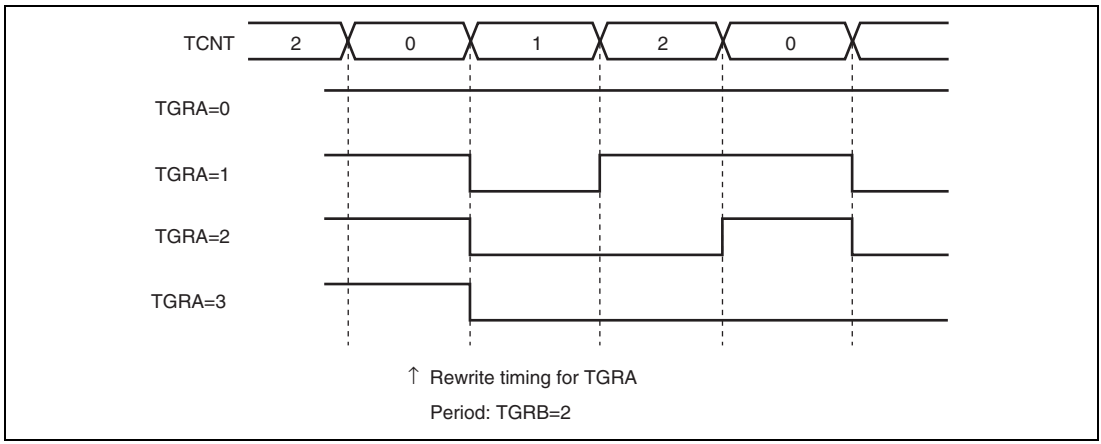


Figure 20.13 Examples of PWM Mode Operation (2)

20.4.5 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 2, and 3.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and compare match and interrupt functions can be used.

The previous set value (initial output value set before the timer was started in phase counting mode) is output from the TPU_TO pin in TIOR.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 20.9 shows the correspondence between external clock pins and channels.

Table 20.9 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 2 is set to phase counting mode	TPU_TI2A	TPU_TI2B
When channel 3 is set to phase counting mode	TPU_TI3A	TPU_TI3B

(1) Example of Phase Counting Mode Setting Procedure

Figure 20.14 shows an example of the phase counting mode setting procedure.

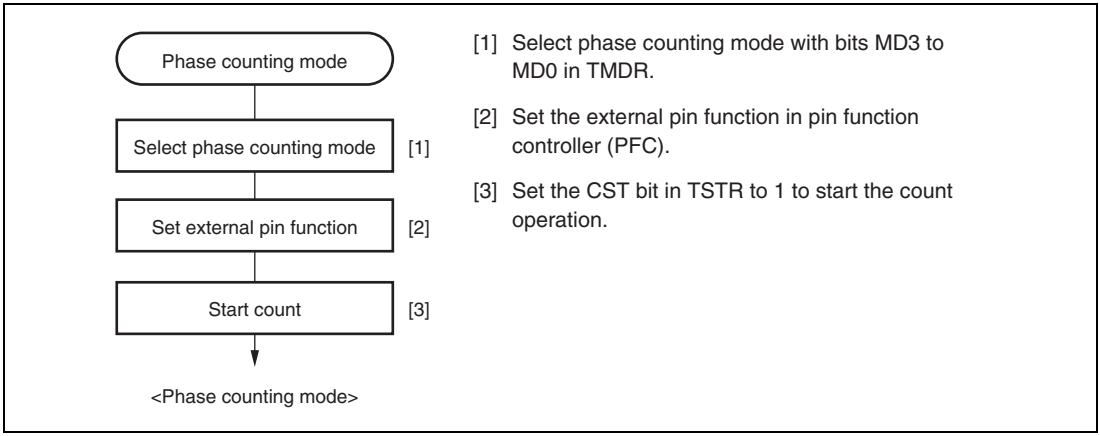


Figure 20.14 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 20.15 shows an example of phase counting mode 1 operation, and table 20.10 summarizes the TCNT up/down-count conditions.

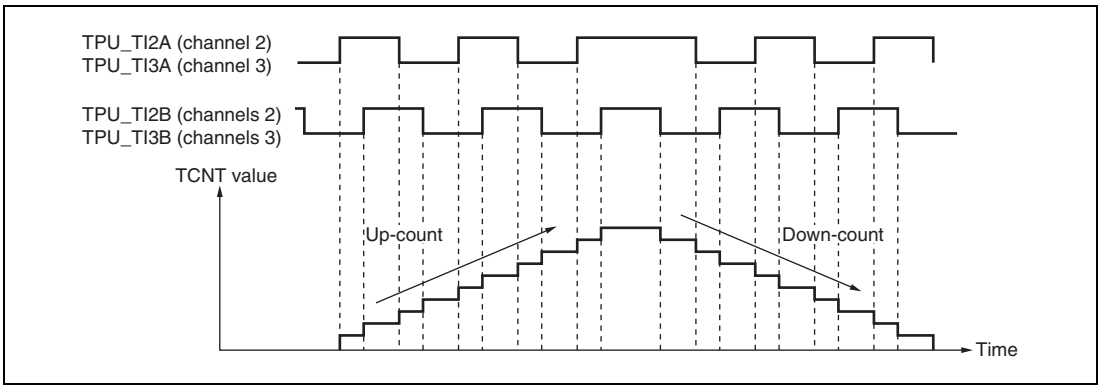


Figure 20.15 Example of Phase Counting Mode 1 Operation

Table 20.10 Up/Down-Count Conditions in Phase Counting Mode 1

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

(b) Phase counting mode 2

Figure 20.16 shows an example of phase counting mode 2 operation, and table 20.11 summarizes the TCNT up/down-count conditions.

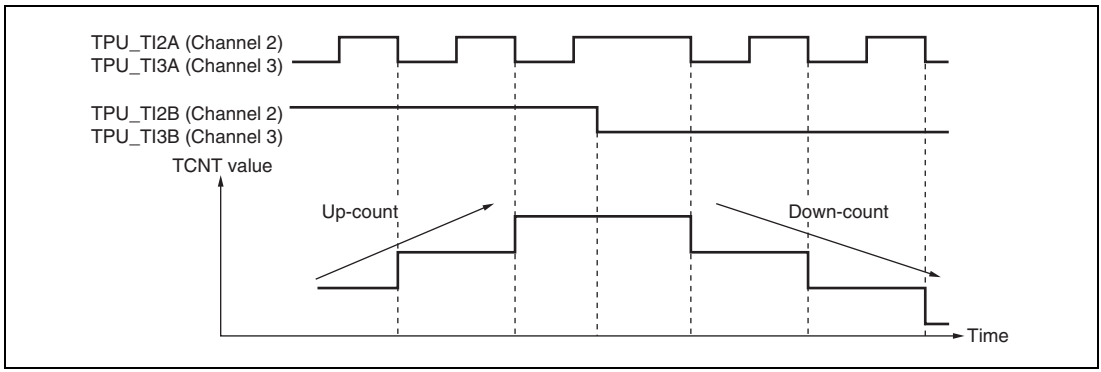


Figure 20.16 Example of Phase Counting Mode 2 Operation

Table 20.11 Up/Down-Count Conditions in Phase Counting Mode 2

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

[Legend]
↑ : Rising edge
↓ : Falling edge

(c) Phase counting mode 3

Figure 20.17 shows an example of phase counting mode 3 operation, and table 20.12 summarizes the TCNT up/down-count conditions.

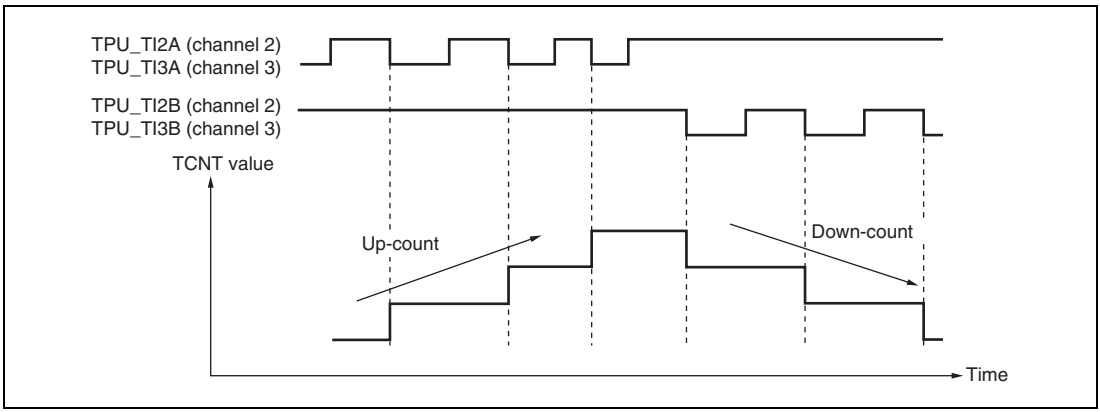


Figure 20.17 Example of Phase Counting Mode 3 Operation

Table 20.12 Up/Down-Count Conditions in Phase Counting Mode 3

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

(d) Phase counting mode 4

Figure 20.18 shows an example of phase counting mode 4 operation, and table 20.13 summarizes the TCNT up/down-count conditions.

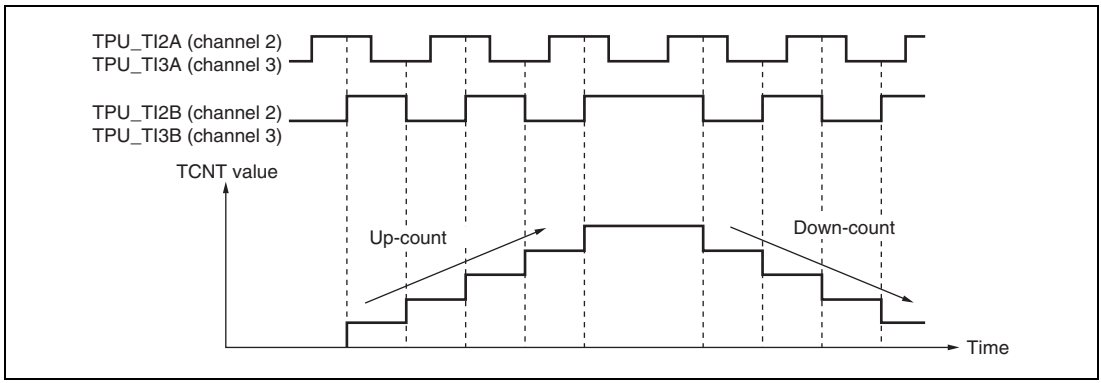


Figure 20.18 Example of Phase Counting Mode 4 Operation

Table 20.13 Up/Down-Count Conditions in Phase Counting Mode 4

TPU_TI2A (Channel 2) TPU_TI3A (Channel 3)	TPU_TI2B (Channel 2) TPU_TI3B (Channel 3)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

20.5 Usage Notes

Note that the kinds of operation and contention described below can occur during TPU operation.

(1) Input Clock Restrictions

The input clock pulse width must be at least 2 states in the case of single-edge detection, and at least 3 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 2 states, and the pulse width must be at least 3 states. Figure 20.19 shows the input clock conditions in phase counting mode.

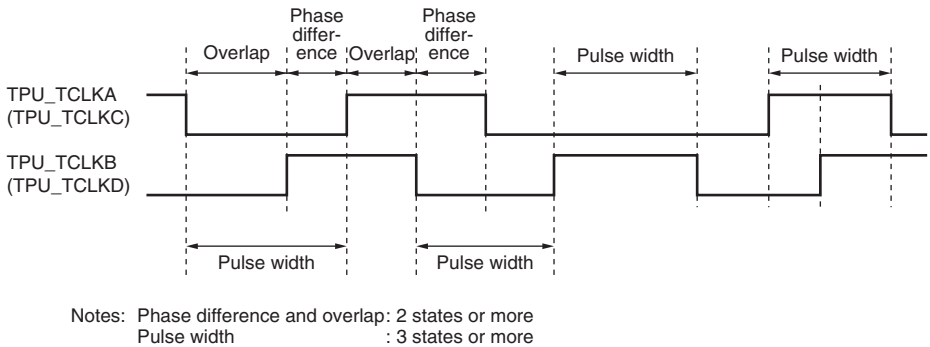


Figure 20.19 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Section 21 Compare Match Timer (CMT)

This LSI includes a 32-bit compare match timer (CMT) of five channels (channel 0 to channel 4).

21.1 Features

- 16 bits/32 bits can be selected.
- Each channel is provided with an auto-reload up counter.
- All channels are provided with 32-bit constant registers and 32-bit up counters that can be written or read at any time.
- Allows selection among three counter input clocks for channel 0 to channel 4:
  - Peripheral clock (Pck0): 1/8, 1/32, and 1/128
- One-shot operation and free-running operation are selectable.
- Allows selection of compare match or overflow for the interrupt source.
- Generate a DMA transfer request when compare match or overflow occurs in channels 0 to 4.
- Module standby mode can be set.

Figure 21.1 shows a block diagram of the CMT.

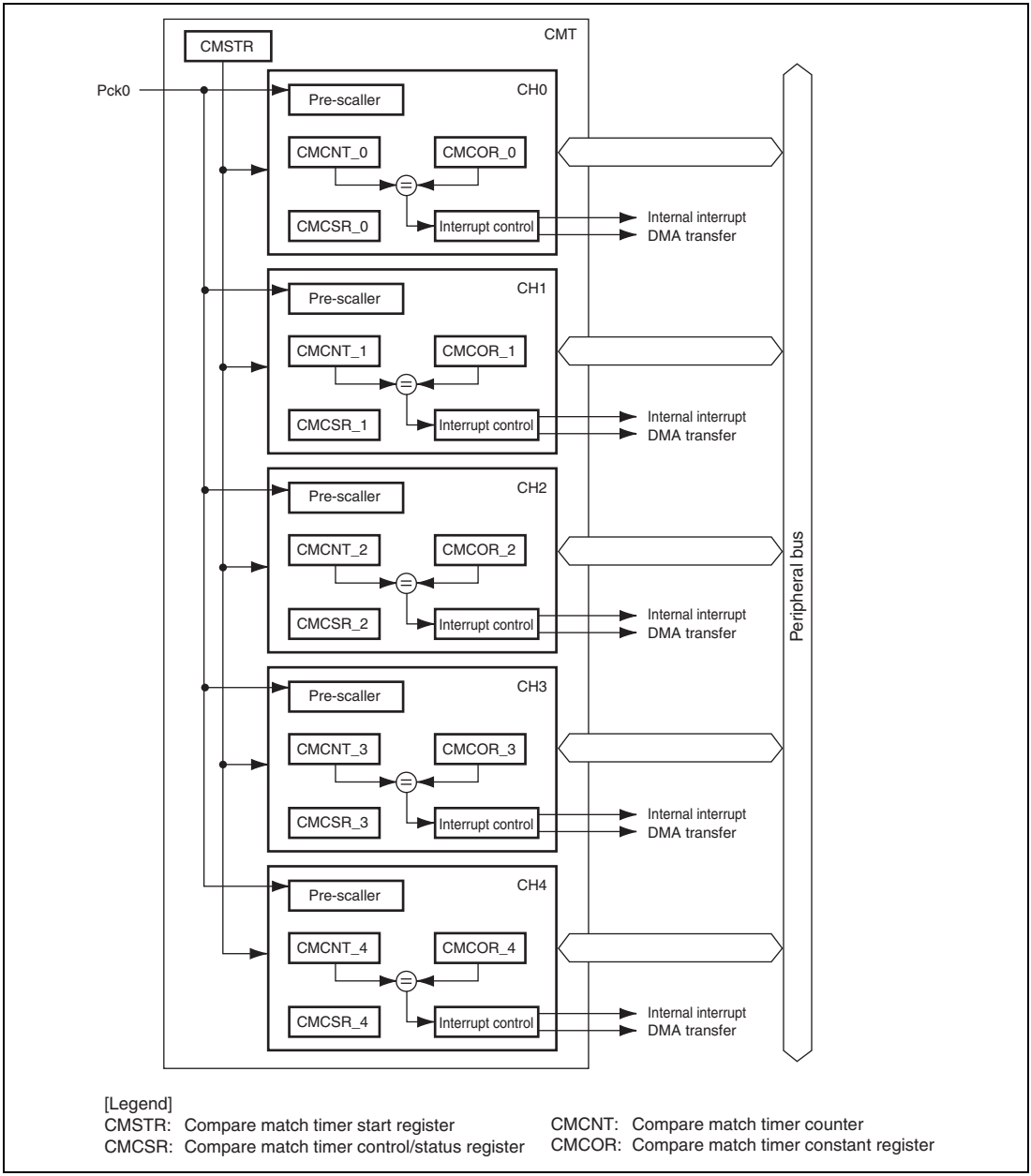


Figure 21.1 Block Diagram of CMT

21.2 Register Descriptions

Table 21.2 shows the CMT register configuration. Table 21.3 shows the register state in each operating mode.

Table 21.1 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Compare match timer start register	CMSTR	R/W	H'FFE2 0000	H'1FE2 0000	16
Compare match timer control/status register_0	CMCSR_0	R/W	H'FFE2 0010	H'1FE2 0010	16
Compare match timer counter_0	CMCNT_0	R/W	H'FF20 0014	H'1F20 0014	32
Compare match timer constant register_0	CMCOR_0	R/W	H'FF20 0018	H'1F20 0018	32
Compare match timer control/status register_1	CMCSR_1	R/W	H'FFE2 0020	H'1FE2 0020	16
Compare match timer counter_1	CMCNT_1	R/W	H'FF20 0024	H'1F20 0024	32
Compare match timer constant register_1	CMCOR_1	R/W	H'FF20 0028	H'1F20 0028	32
Compare match timer control/status register_2	CMCSR_2	R/W	H'FFE2 0030	H'1FE2 0030	16
Compare match timer counter_2	CMCNT_2	R/W	H'FF20 0034	H'1F20 0034	32
Compare match timer constant register_2	CMCOR_2	R/W	H'FF20 0038	H'1F20 0038	32
Compare match timer control/status register_3	CMCSR_3	R/W	H'FFE2 0040	H'1FE2 0040	16
Compare match timer counter_3	CMCNT_3	R/W	H'FF20 0044	H'1F20 0044	32
Compare match timer constant register_3	CMCOR_3	R/W	H'FF20 0048	H'1F20 0048	32
Compare match timer control/status register_4	CMCSR_4	R/W	H'FFE2 0050	H'1FE2 0050	16
Compare match timer counter_4	CMCNT_4	R/W	H'FF20 0054	H'1F20 0054	32
Compare match timer constant register_4	CMCOR_4	R/W	H'FF20 0058	H'1F20 0058	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 21.2 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Compare match timer start register	CMSTR	H'0000	H'0000	Retained	Retained
Compare match timer control/status register_0	CMCSR_0	H'0000	H'0000	Retained	Retained
Compare match timer counter_0	CMCNT_0	H'0000 0000	H'0000 0000	Retained	Retained
Compare match timer constant register_0	CMCOR_0	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Compare match timer control/status register_1	CMCSR_1	H'0000	H'0000	Retained	Retained
Compare match timer counter_1	CMCNT_1	H'0000 0000	H'0000 0000	Retained	Retained
Compare match timer constant register_1	CMCOR_1	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Compare match timer control/status register_2	CMCSR_2	H'0000	H'0000	Retained	Retained
Compare match timer counter_2	CMCNT_2	H'0000 0000	H'0000 0000	Retained	Retained
Compare match timer constant register_2	CMCOR_2	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Compare match timer control/status register_3	CMCSR_3	H'0000	H'0000	Retained	Retained
Compare match timer counter_3	CMCNT_3	H'0000 0000	H'0000 0000	Retained	Retained
Compare match timer constant register_3	CMCOR_3	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained
Compare match timer control/status register_4	CMCSR_4	H'0000	H'0000	Retained	Retained
Compare match timer counter_4	CMCNT_4	H'0000 0000	H'0000 0000	Retained	Retained
Compare match timer constant register_4	CMCOR_4	H'FFFF FFFF	H'FFFF FFFF	Retained	Retained

21.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether the compare match timer counter (CMCNT) is operated or halted.

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

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	STR[4:0]	All 0	R/W	Count Start Selects whether to operate or halt the compare match timer counter for each channel (CMCNT_4 to CMCNT_0). 0: CMCNTn count operation halted 1: CMCNTn count operation n: 4 to 0 (corresponds to each channel)

21.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates the occurrence of compare matches, enables interrupts and DMA transfer request, and sets the counter input clocks.

Do not change bits other than bits CMF and OVF during the compare match timer counter (CMCNT) operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMF	OVF	—	—	—	—	CMS	CMM	—	—	CMR[1:0]		—	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CMF	0	R/(W)*[1]	Compare Match Flag This flag indicates whether or not values of the compare match timer counter (CMCNT) and compare match timer constant register (CMCOR) have matched. Software cannot write 1 to the bit. When one-shot is selected for the counter operation, counting resumes by clearing this bit. 0: CMCNT and CMCOR values have not matched [Clearing condition] • Write 0 to CMF after reading CMF=1 1: CMCNT and CMCOR values have matched
14	OVF	0	R/(W)*[1]	Overflow Flag This flag indicates whether or not the compare match timer counter (CMCNT) has overflowed and been cleared to 0. Software cannot write 1 to this bit. 0: CMCNT has not overflowed [Clearing condition] • Write 0 to OVF after reading OVF=1 1: CMCNT has overflowed

Bit	Bit Name	Initial Value	R/W	Description
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	CMS	0	R/W	Compare Match Timer Counter Size Selects whether the compare match timer counter (CMCNT) is used as a 16-bit counter or a 32-bit counter. This setting becomes the valid size for the compare match timer constant register (CMCOR). 0: Operates as a 32-bit counter 1: Operates as a 16-bit counter
8	CMM	0	R/W	Compare Match Mode Selects one-shot operation or free-running operation of the counter. 0: One-shot operation 1: Free-running operation
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	CMR[1:0]	00	R/W	Compare Match Request Selects enable or disable for a DMA transfer request or internal interrupt request in a compare match. 00: Disables a DMA transfer request and internal interrupt request 01: Enables DMA transfer request 10: Enables an internal interrupt request 11: Setting prohibited
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 to 0	CKS[2:0]	All 0	R/W	Clock Select These bits select the clock input to CMCNT. When the STRn (n: 4 to 0) bit in CMSTR is set to 1, CMCNT begins incrementing with the clock selected by these bits. 000: Pck0/8 001: Pck0/32 010: Pck0/128 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * Only 0 can be written to clear the flag.

21.2.3 Compare Match Timer Counter (CMCNT)

CMCNT is a 32-bit register that is used as an up-counter.

A counter operation is set by the compare match timer control/status register (CMCSR). Therefore, set CMCSR first, before starting a channel operation corresponding to the compare match timer start register (CMSTR). When the 16-bit counter operation is selected by the CMS bit, bits 15 to 0 of this register become valid. When the register should be written to, write the data that is added H'0000 to the upper half in a 32-bit operation. The contents of this register are initialized to H'00000000.

21.2.4 Compare Match Timer Constant Register (CMCOR)

CMCOR is a 32-bit register that sets the compare match period with CMCNT for each channel.

When the 16-bit counter operation is selected by the CMS bit in CMCSR, bits 15 to 0 of this register become valid. When the register should be written to, write the data that is added H'0000 to the upper half in a 32-bit operation.

An overflow is detected when CMCNT is cleared to 0 and this register is H'FFFFFFFF. The contents of this register are initialized to H'FFFFFFFF.

21.3 Operation

21.3.1 Counter Operation

The CMT starts the operation of the counter by writing a 1 to the STRn bit in CMSTR of a channel that has been selected for operation. Complete all of the settings before starting the operation. Do not change the register settings other than by clearing flag bits.

The counter operates in one of two ways.

- One-Shot Operation

  One-shot operation is selected by setting the CMM bit in CMCSR to 0. When the value in CMCNT matches the value in CMCOR, the value in CMCNT is cleared to H'00000000 and the CMF bit in CMCSR is set to 1. Counting by CMCNT stops after it has been cleared.

  To detect an overflow interrupt, set the value in CMCOR to H'FFFFFFFF. When the value in CMCNT matches the value in CMCOR, CMCNT is cleared to H'00000000 and bits CMF and OVF in CMCSR are set to 1.

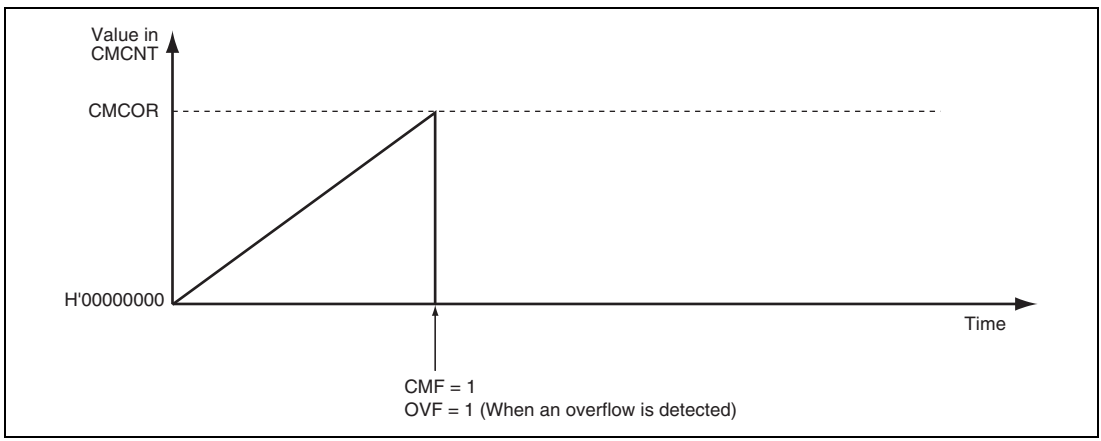


Figure 21.2 Counter Operation (One-Shot Operation)

- Free-Running Operation

  Free-running operation is selected by setting the CMM bit in CMCSR to 1. When the value in CMCNT matches the value in CMCOR, CMCNT is cleared to H'00000000 and the CMF bit in CMCSR is set to 1. CMCNT resumes counting-up after it has been cleared.

  To detect an overflow interrupt, set CMCOR to H'FFFFFFFF. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'00000000 and bits CMF and OVF in CMCSR are set to 1.

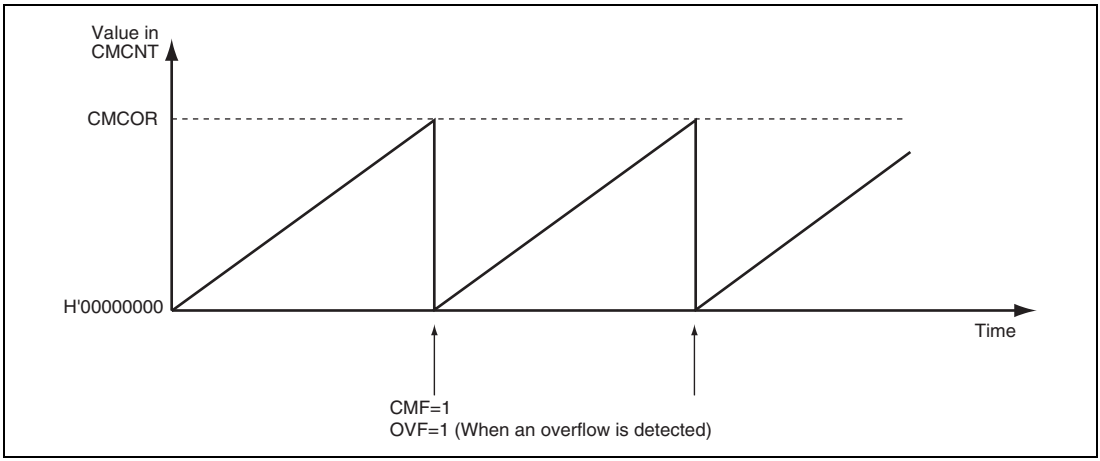


Figure 21.3 Counter Operation (Free-Running Operation)

21.3.2 Counter Size

In this module, the size of the counter is selectable as either 16 or 32 bits. This is selected by the CMS bit in CMCSR.

When the 16-bit size is selected, use a 32-bit value which has H'0000 as its upper half to set CMCOR.

To detect an overflow interrupt, the value must be set to H'0000FFFF.

21.3.3 Timing for Counting by CMCNT

In this module, the clock for the counter can be selected from among the following:

- For channels 0 to 4:
  — Peripheral clock (Pck0): 1/8, 1/32, or 1/128

The clock for the counter is selected by bits CKS2 to CKS0 in CMCSR. CMCNT is incremented at the rising edge of the selected clock.

21.3.4 DMA Transfer Requests and Internal Interrupt Requests to CPU

The setting of bits CMR1 and CMR0 in CMCSR selects the sending of a request for a DMA transfer or for an internal interrupt to the CPU at a compare match.

A DMA transfer request has different specifications according to the CMT channel as described below.

1. For channels 0 and 1, a single DMA transfer request is output at a compare match.
2. For channels 2 to 4, a DMA transfer request continues until the amount of data transferred has reached the value set in the DMAC, and the output of the request then automatically stops.

To clear the interrupt request, the CMF bit should be set to 0. Set the CMF bit to 0 in the handling routine for the CMT interrupt.

21.3.5 Compare Match Flag Set Timing (All Channels)

The CMF bit in CMCSR is set to 1 by the compare match signal generated when CMCOR and CMCNT match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT value is updated to H'0000). Consequently, after CMCOR and CMCNT match, a compare match signal will not be generated until a CMCNT counter clock is input. Figure 21.4 shows the set timing of the CMF bit.

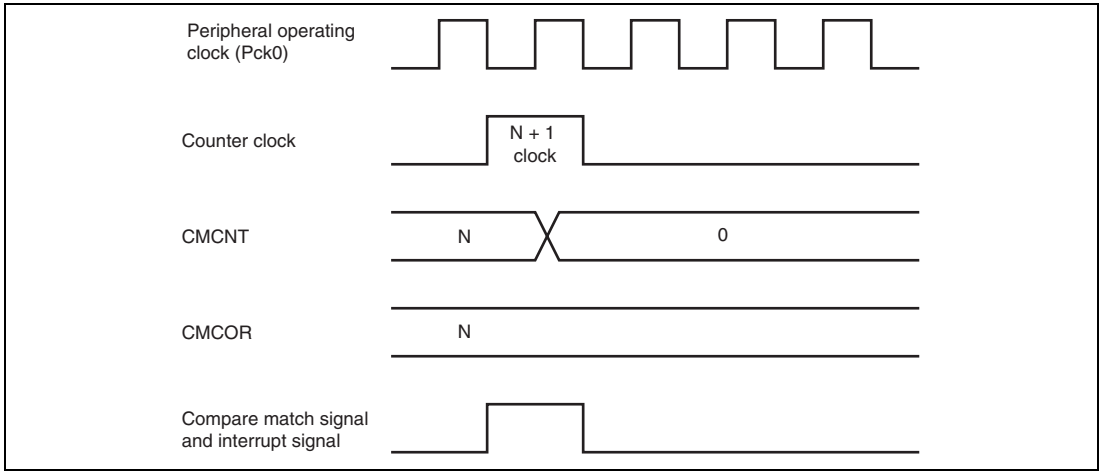


Figure 21.4 CMF Set Timing

Section 22 Realtime Clock (RTC)

This LSI includes an on-chip realtime clock (RTC) and a 32.768 kHz crystal oscillator for use by the RTC.

22.1 Features

The RTC has the following features.

- Clock and calendar functions (BCD display)
  Counts seconds, minutes, hours, day-of-week, days, months, and years.
- 1 to 64 Hz timer (binary display)
  The 64 Hz counter register indicates a state of 64 Hz to 1 Hz within the RTC frequency divider
- Start/stop function
- 30-second adjustment function
- Alarm interrupts
  Comparison with second, minute, hour, day-of-week, day, month, or year can be selected as the alarm interrupt condition
- Periodic interrupts
  An interrupt period of 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds can be selected
- Carry interrupt
  Carry interrupt function indicating a second counter carry, or a 64 Hz counter carry when the 64 Hz counter is read
- Automatic leap year adjustment

22.1.1 Block Diagram

Figure 22.1 shows a block diagram of the RTC.

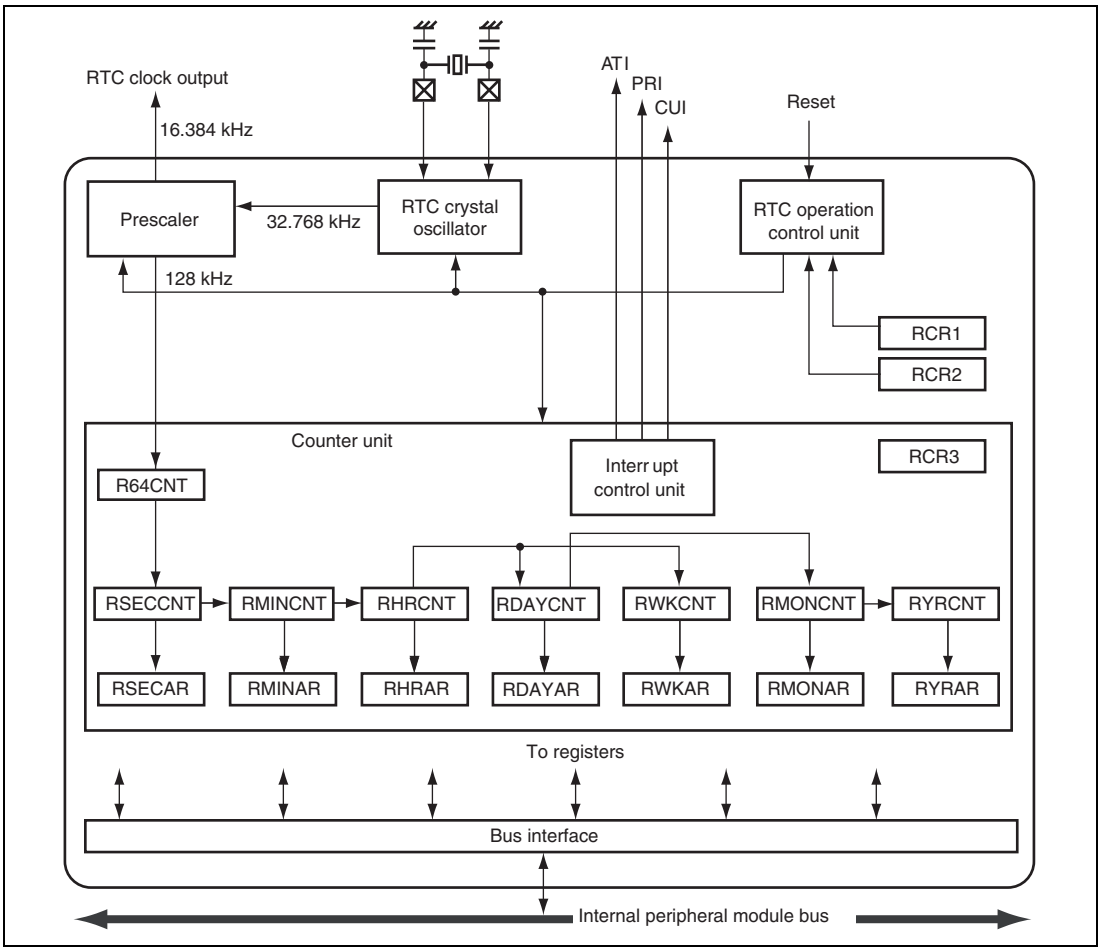


Figure 22.1 Block Diagram of RTC

22.2 Input/Output Pins

Table 22.1 shows the RTC pins.

Table 22.1 RTC Pins

Pin Name	Abbreviation	I/O	Function
RTC oscillator crystal pin	EXTAL2	Input	Connects crystal to RTC oscillator
RTC oscillator crystal pin	XTAL2	Output	Connects crystal to RTC oscillator
Dedicated RTC power supply	Vdd-RTC	—	RTC oscillator power supply pin*
Dedicated RTC GND pin	Vss-RTC	—	RTC oscillator GND pin*
RTC standby	$\overline{\text{XRTCSTBI}}$	Input	RTC standby

Note: * Power must be supplied to the RTC power supply pins even when the RTC is not used.

22.3 Register Descriptions

Table 22.2 shows the RTC register configuration. Table 22.3 shows the register state in each operating mode.

Table 22.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
64 Hz counter	R64CNT	R	H'FFF8 0000	H'1FF8 0000	8
Second counter	RSECCNT	R/W	H'FFF8 0004	H'1FF8 0004	8
Minute counter	RMINCNT	R/W	H'FFF8 0008	H'1FF8 0008	8
Hour counter	RHRCNT	R/W	H'FFF8 000C	H'1FF8 000C	8
Day-of-week counter	RWKCNT	R/W	H'FFF8 0010	H'1FF8 0010	8
Day counter	RDAYCNT	R/W	H'FFF8 0014	H'1FF8 0014	8
Month counter	RMONCNT	R/W	H'FFF8 0018	H'1FF8 0018	8
Year counter	RYRCNT	R/W	H'FFF8 001C	H'1FF8 001C	16
Second alarm register	RSECAR	R/W	H'FFF8 0020	H'1FF8 0020	8
Minute alarm register	RMINAR	R/W	H'FFF8 0024	H'1FF8 0024	8
Hour alarm register	RHRAR	R/W	H'FFF8 0028	H'1FF8 0028	8
Day-of-week alarm register	RWKAR	R/W	H'FFF8 002C	H'1FF8 002C	8
Day alarm register	RDAYAR	R/W	H'FFF8 0030	H'1FF8 0030	8
Month alarm register	RMONAR	R/W	H'FFF8 0034	H'1FF8 0034	8
RTC control register 1	RCR1	R/W	H'FFF8 0038	H'1FF8 0038	8
RTC control register 2	RCR2	R/W	H'FFF8 003C	H'1FF8 003C	8
RTC control register 3	RCR3	R/W	H'FFF8 0050	H'1FF8 0050	8
Year alarm register	RYRAR	R/W	H'FFF8 0054	H'1FF8 0054	16

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 22.3 Register State in Each Operating Mode

Name	Abbreviation	Initial Value	Power-On Reset	Manual Reset	Sleep	Standby
64 Hz counter	R64CNT	Undefined	Counts	Counts	Counts	Counts
Second counter	RSECCNT	Undefined	Counts	Counts	Counts	Counts
Minute counter	RMINCNT	Undefined	Counts	Counts	Counts	Counts
Hour counter	RHRCNT	Undefined	Counts	Counts	Counts	Counts
Day-of-week counter	RWKCNT	Undefined	Counts	Counts	Counts	Counts
Day counter	RDAYCNT	Undefined	Counts	Counts	Counts	Counts
Month counter	RMONCNT	Undefined	Counts	Counts	Counts	Counts
Year counter	RYRCNT	Undefined	Counts	Counts	Counts	Counts
Second alarm register	RSECAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
Minute alarm register	RMINAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
Hour alarm register	RHRAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
Day-of-week alarm register	RWKAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
Day alarm register	RDAYAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
Month alarm register	RMONAR	Undefined*[1]	Initialized*[1]	Retained	Retained	Retained
RTC control register 1	RCR1	H'00*[3]	Initialized	Initialized	Retained	Retained
RTC control register 2	RCR2	H'09*[4]	Initialized	Initialized*[2]	Retained	Retained
RTC control register 3	RCR3	H'00	Initialized	Retained	Retained	Retained
Year alarm register	RYRAR	Undefined	Retained	Retained	Retained	Retained

Notes: 1. The ENB bit in each register is initialized.
2. Bits other than the RTCEN bit and START bit are initialized.
3. The value of the CF bit, CRF bit and AF bit is undefined.
4. The value of the PEF bit is undefined.

22.4 Register Descriptions

22.4.1 64 Hz Counter (R64CNT)

R64CNT is an 8-bit read-only register that indicates a state of 64 Hz to 1 Hz within the RTC frequency divider.

If this register is read when a carry is generated from the 128 kHz frequency division stage, bit 7 (CF) in RTC control register 1 (RCR1) is set to 1, indicating the simultaneous occurrence of the carry and the 64 Hz counter read. In this case, the read value is not valid, and so R64CNT must be read again after first writing 0 to the CF bit in RCR1 to clear it.

When the RESET bit or ADJ bit in RTC control register 2 (RCR2) is set to 1, the RTC frequency divider is initialized and R64CNT is initialized to H'00.

R64CNT is not initialized by a power-on or manual reset.

Bit 7 is always read as 0 and cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

22.4.2 Second Counter (RSECCNT)

RSECCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded second value in the RTC. It counts on the carry (transition of the R64CNT.1Hz bit from 1 to 0) generated once per second by the 64 Hz counter.

The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RSECCNT is not initialized by a power-on or manual reset.

Bit 7 is always read as 0. A write to this bit is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	10-second units			1-second units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.4.3 Minute Counter (RMINCNT)

RMINCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded minute value in the RTC. It counts on the carry generated once per minute by the second counter.

The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RMINCNT is not initialized by a power-on or manual reset.

Bit 7 is always read as 0. A write to this bit is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	10-minute units			1-minute units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.4.4 Hour Counter (RHRCNT)

RHRCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded hour value in the RTC. It counts on the carry generated once per hour by the minute counter.

The setting range is decimal 00 to 23. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RHRCNT is not initialized by a power-on or manual reset.

Bits 7 and 6 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	10-hour units		1-hour units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

22.4.5 Day-of-Week Counter (RWKCNT)

RWKCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded day-of-week value in the RTC. It counts on the carry generated once per day by the hour counter.

The setting range is decimal 0 to 6. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RWKCNT is not initialized by a power-on or manual reset.

Bits 7 to 3 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Day-of-week code	0	1	2	3	4	5	6
Day of week	Sun	Mon	Tue	Wed	Thu	Fri	Sat

22.4.6 Day Counter (RDAYCNT)

RDAYCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded day value in the RTC. It counts on the carry generated once per day by the hour counter.

The setting range is decimal 01 to 31. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RDAYCNT is not initialized by a power-on or manual reset.

The setting range for RDAYCNT depends on the month and whether the year is a leap year, so care is required when making the setting. Taking the year counter (RYRCNT) value as the year, leap year calculation is performed according to whether or not the value is divisible by 400, 100, and 4.

Bits 7 and 6 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	10-day units		1-day units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

22.4.7 Month Counter (RMONCNT)

RMONCNT is an 8-bit readable/writable register used as a counter for setting and counting the BCD-coded month value in the RTC. It counts on the carry generated once per month by the day counter.

The setting range is decimal 01 to 12. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RMONCNT is not initialized by a power-on or manual reset.

Bits 7 to 5 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	10-month unit	1-month units			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

22.4.8 Year Counter (RYRCNT)

RYRCNT is a 16-bit readable/writable register used as a counter for setting and counting the BCD-coded year value in the RTC. It counts on the carry generated once per year by the month counter.

The setting range is decimal 0000 to 9999. The RTC will not operate normally if any other value is set. Write processing should be performed after stopping the count with the START bit in RCR2, or by using the carry flag.

RYRCNT is not initialized by a power-on or manual reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1000-year units				100-year units				10-year units				1-year units			
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.4.9 Second Alarm Register (RSECAR)

RSECAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded second value counter, RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 59 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The other fields in RSECAR are not initialized by a power-on or manual reset.

Bit:	7	6	5	4	3	2	1	0
	ENB	10-second units			1-second units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.4.10 Minute Alarm Register (RMINAR)

RMINAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded minute value counter, RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 59 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The other fields in RMINAR are not initialized by a power-on or manual reset.

Bit:	7	6	5	4	3	2	1	0
	ENB	10-minute units			1-minute units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.4.11 Hour Alarm Register (RHRAR)

RHRAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded hour value counter, RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 00 to 23 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The other fields in RHRAR are not initialized by a power-on or manual reset.

Bit 6 is always read as 0. A write to this bit is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-hour units		1-hour units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

22.4.12 Day-of-Week Alarm Register (RWKAR)

RWKAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded day-of-week value counter, RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 0 to 6 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The other fields in RWKAR are not initialized by a power-on or manual reset.

Bits 6 to 3 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Day-of-week code	0	1	2	3	4	5	6
Day of week	Sun	Mon	Tue	Wed	Thu	Fri	Sat

22.4.13 Day Alarm Register (RDAYAR)

RDAYAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded day value counter, RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 01 to 31 + ENB bit. The RTC will not operate normally if any other value is set. The setting range for RDAYAR depends on the month and whether the year is a leap year, so care is required when making the setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The other fields in RDAYAR are not initialized by a power-on or manual reset.

Bit 6 is always read as 0. A write to this bit is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-day units		1-day units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

22.4.14 Month Alarm Register (RMONAR)

RMONAR is an 8-bit readable/writable register used as an alarm register for the RTC's BCD-coded month value counter, RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. Comparison between the counter and the alarm register is performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values all match.

The setting range is decimal 01 to 12 + ENB bit. The RTC will not operate normally if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The other fields in RMONAR are not initialized by a power-on or manual reset.

Bits 6 and 5 are always read as 0. A write to these bits is invalid, but the write value should always be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	10-month unit	1-month units			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

22.4.15 RTC Control Register 1 (RCR1)

RCR1 is an 8-bit readable/writable register containing a carry flag and alarm flag, plus flags to enable or disable interrupts for these flags.

The CIE and AIE bits are initialized to 0 by a power-on or manual reset; the value of bits other than CIE and AIE is undefined.

Bit:	7	6	5	4	3	2	1	0
	CF	—	—	CIE	AIE	—	—	AF
Initial value:	—	—	—	0	0	—	—	—
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag This flag is set to 1 on generation of a second counter carry, or a 64 Hz counter carry when the 64 Hz counter is read. The count register value read at this time is not guaranteed, and so the count register must be read again. 0: No second counter carry, or 64 Hz counter carry when 64 Hz counter is read [Clearing condition] When 0 is written to CF 1: Second counter carry, or 64 Hz counter carry when 64 Hz counter is read [Setting conditions] Generation of a second counter carry, or a 64 Hz counter carry when the 64 Hz counter is read When 1 is written to CF
6 to 5	—	Undefined	R	Reserved The initial value of these bits is undefined. A write to these bits is invalid, but the write value should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag Enables or disables interrupt generation when the carry flag (CF) is set to 1. 0: Carry interrupt is not generated when CF flag is set to 1 1: Carry interrupt is generated when CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag Enables or disables interrupt generation when the alarm flag (AF) is set to 1. 0: Alarm interrupt is not generated when AF flag is set to 1 1: Alarm interrupt is generated when AF flag is set to 1
2 to 1	—	Undefined	R	Reserved The initial value of these bits is undefined. A write to these bits is invalid, but the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	AF	Undefined	R/W	Alarm Flag Set to 1 when the alarm time set in those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR in which the ENB bit is set to 1 matches the respective counter values. 0: Alarm registers and counter values do not match (Initial value) [Clearing condition] When 0 is written to AF 1: Alarm registers and counter values match* [Setting condition] When alarm registers in which the ENB bit is set to 1 and counter values match* Note: * Writing 1 does not change the value.

22.4.16 RTC Control Register 2 (RCR2)

RCR2 is an 8-bit readable/writable register used for periodic interrupt control, 30-second adjustment, and frequency divider RESET and RTC count control.

RCR2 is basically initialized to H'09 by a power-on reset, except that the value of the PEF bit is undefined. In a manual reset, bits other than RTCEN and START are initialized, while the value of the PEF bit is undefined. In standby mode RCR2 is not initialized, and retains its current value.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	—	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	Undefined	R/W	Periodic Interrupt Flag Indicates interrupt generation at the interval specified by bits PES2–PES0. When this flag is set to 1, a periodic interrupt is generated. 0: Interrupt is not generated at interval specified by bits PES2–PES0 [Clearing condition] When 0 is written to PEF 1: Interrupt is generated at interval specified by bits PES2–PES0 [Setting conditions] Generation of interrupt at interval specified by bits PES2–PES0 When 1 is written to PEF
6 to 4	PES[2:0]	All 0	R/W	Periodic Interrupt Enable These bits specify the period for periodic interrupts. 000: No periodic interrupt generation 001: Periodic interrupt generated at 1/256-second intervals 010: Periodic interrupt generated at 1/64-second intervals 011: Periodic interrupt generated at 1/16-second intervals 100: Periodic interrupt generated at 1/4-second intervals 101: Periodic interrupt generated at 1/2-second intervals 110: Periodic interrupt generated at 1-second intervals 111: Periodic interrupt generated at 2-second intervals
3	RTCEN	1	R/W	Oscillator Enable Controls the operation of the RTC's crystal oscillator. 0: RTC crystal oscillator is halted 1: RTC crystal oscillator is operated

Bit	Bit Name	Initial Value	R/W	Description
2	ADJ	0	R/W	30-Second Adjustment Used for 30-second adjustment. When 1 is written to this bit, a value up to 29 seconds is rounded down to 00 seconds, and a value of 30 seconds or more is rounded up to 1 minute. The frequency divider circuits (RTC prescaler and R64CNT) are also reset at this time. This bit always returns 0 if read. 0: Normal clock operation 1: 30-second adjustment performed
1	RESET	0	R/W	Reset The frequency divider circuits are initialized by writing 1 to this bit. When 1 is written to the RESET bit, the frequency divider circuits (RTC prescaler and R64CNT) are reset and the RESET bit is automatically cleared to 0 (i.e. does not need to be written with 0). 0: Normal clock operation 1: Frequency divider circuits are reset
0	START	1	R/W	Start Bit Stops and restarts counter (clock) operation. 0: Second, minute, hour, day, day-of-week, month, and year counters are stopped* 1: Second, minute, hour, day, day-of-week, month, and year counters operate normally* Note: * The 64 Hz counter continues to operate unless stopped by means of the RTCEN bit.

22.4.17 RTC Control Register (RCR3) and Year-Alarm Register (RYRAR)

RCR3 and RYRAR are readable/writable registers. RYRAR is the alarm register for the RTC's BCD-coded year-value counter RYRCNT. When the YENB bit of RCR3 is set to 1, the RYRCNT value is compared with the RYRAR value. Comparison between the counter and the alarm register only takes place with the alarm registers in which the ENB and YENB bits are set to 1. The alarm flag of RCR1 is only set to 1 when the respective values all match.

The setting range of RYRAR is decimal 0000 to 9999, and normal operation is not obtained if a value beyond this range is set here.

RCR3 is initialized by a power-on reset, but RYRAR will not be initialized by a power-on or manual reset.

Bits 6 to 0 of RCR3 are always read as 0. A write to these bits is invalid. If a value is written to these bits, it should always be 0.

- RCR3

Bit:	7	6	5	4	3	2	1	0
	YENB	—	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

- RYRAR

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1000 years				100 years				10 years				1 year			
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.5 Operation

Examples of the use of the RTC are shown below.

22.5.1 Time Setting Procedures

Figure 22.2 shows examples of the time setting procedures.

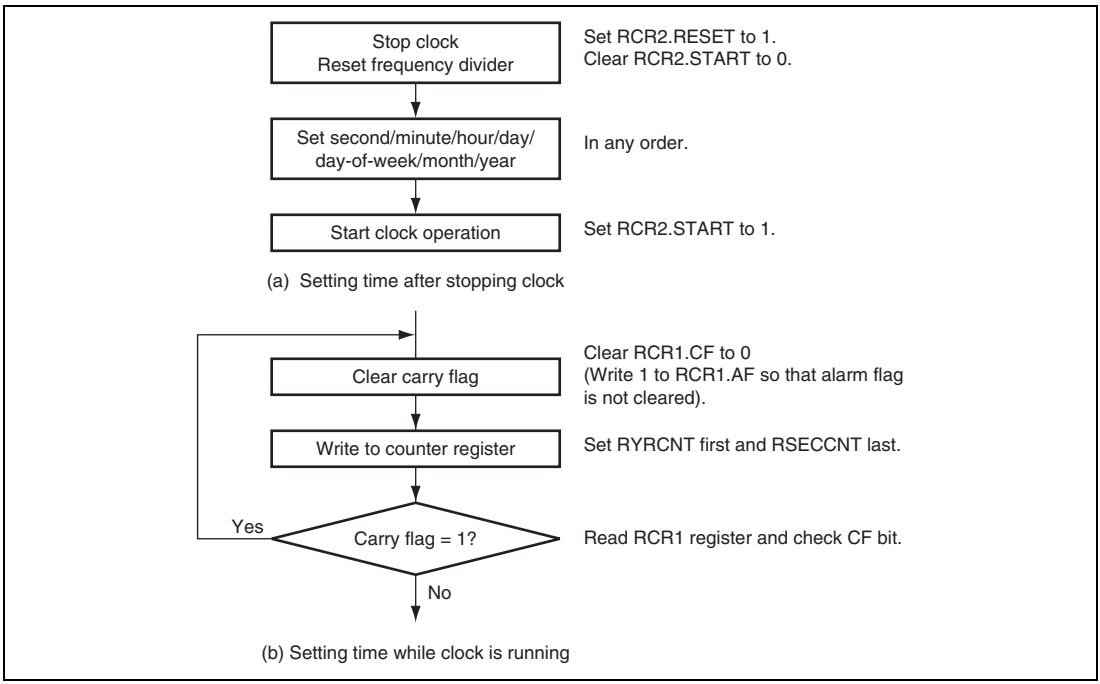


Figure 22.2 Examples of Time Setting Procedures

The procedure for setting the time after stopping the clock is shown in figure 22.2 (a). The programming for this method is simple, and it is useful for setting all the counters, from second to year.

The procedure for setting the time while the clock is running is shown in figure 22.2 (b). This method is useful for modifying only certain counter values (for example, only the second data or hour data). If a carry occurs during the write operation, the write data is automatically updated and there will be an error in the set data. The carry flag should therefore be used to check the write status. If the carry flag (RCR1.CF) is set to 1, the write must be repeated.

The interrupt function can also be used to determine the carry flag status.

22.5.2 Time Reading Procedures

Figure 22.3 shows examples of the time reading procedures.

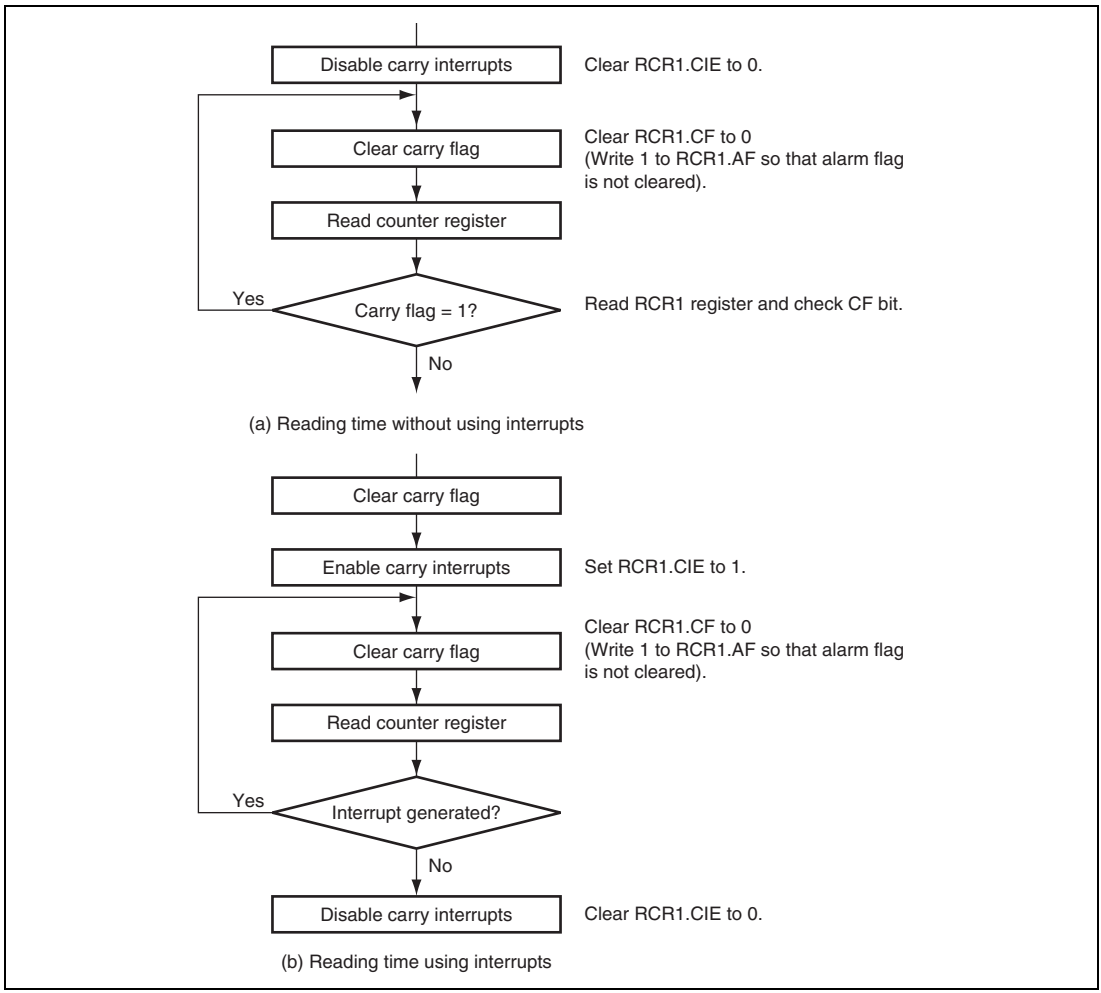


Figure 22.3 Examples of Time Reading Procedures

If a carry occurs while the time is being read, the correct time will not be obtained and the read must be repeated. The procedure for reading the time without using interrupts is shown in figure 22.3 (a), and the procedure using carry interrupts in figure 22.3 (b). The method without using interrupts is normally used to keep the program simple.

22.5.3 Alarm Function

The use of the alarm function is illustrated in figure 22.4.

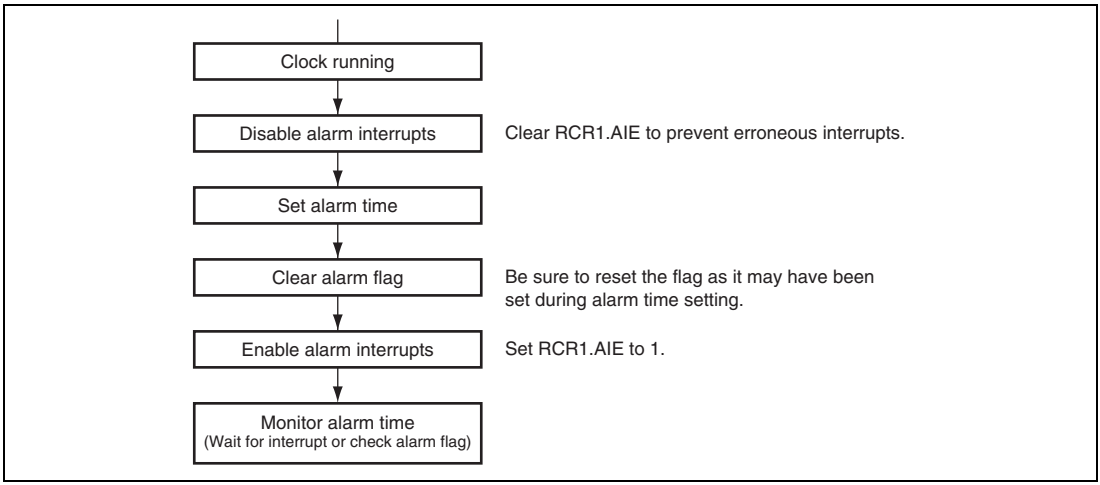


Figure 22.4 Example of Use of Alarm Function

An alarm can be generated by the second, minute, hour, day-of-week, day, month, or year value, or a combination of these. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

When the counter and the alarm time match, RCR1.AF is set to 1. Alarm detection can be confirmed by reading this bit, but normally an interrupt is used. If 1 has been written to RCR1.AIE, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

22.6 Interrupts

There are three kinds of RTC interrupt: alarm interrupts, periodic interrupts, and carry interrupts.

An alarm interrupt request (ATI) is generated when the alarm flag (AF) in RCR1 is set to 1 while the alarm interrupt enable bit (AIE) is also set to 1.

A periodic interrupt request (PRI) is generated when the periodic interrupt enable bits (PES2–PES0) in RCR2 are set to a value other than 000 and the periodic interrupt flag (PEF) is set to 1.

A carry interrupt request (CUI) is generated when the carry flag (CF) in RCR1 is set to 1 while the carry interrupt enable bit (CIE) is also set to 1.

22.7 Usage Notes

22.7.1 Register Initialization

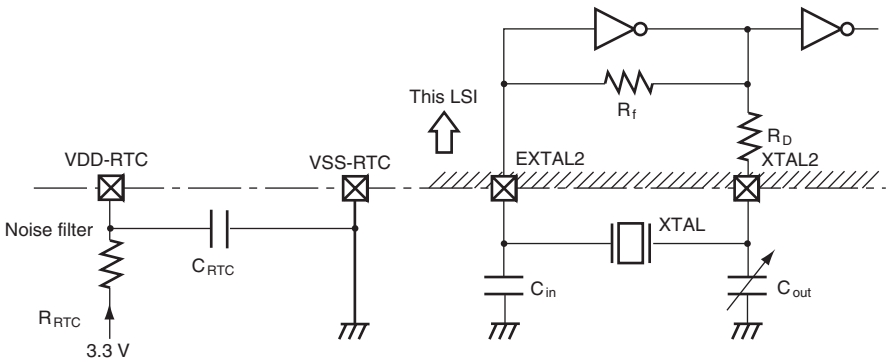
After powering on and making the RCR1 register settings, reset the frequency divider (by setting RCR2.RESET to 1) and make initial settings for all the other registers.

22.7.2 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 22.4, and the RTC crystal oscillator circuit in figure 22.5.

Table 22.4 Crystal Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10–22 pF	10–22 pF



Notes: 1. Select either the C_{in} or C_{out} side for the frequency adjustment variable capacitor according to requirements such as the adjustment range, degree of stability, etc.
2. Built-in resistance value R_f (typ. value) = 10 MΩ, R_D (typ. value) = 400 kΩ
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a solidearth board.
4. The crystal oscillation stabilization time depends on the mounted circuit constants, floating capacitance, etc., and should be decided after consultation with the crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip. (Correct oscillation may not be possible if there is externally induced noise in the EXTAL2 and XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2 and XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.
7. Insert a noise filter in the RTC power supply.

Figure 22.5 Example of Crystal Oscillator Circuit Connection

22.7.3 Interrupt source and request generating order

If it occurs two or three interrupt source of alarm interrupts(ATI), periodic interrupts(PFI), and carry interrupts(CUI) at the same time, RTC generates interrupt request as the order of table 22.5.

Table 22.5 Interrupt source and request generating order

Interrupt source	Interrupt request generating order Fast ←		
ATI, PFI and CUI	PFI	CUI	ATI
ATI and PFI	PFI	ATI	—
PFI and CUI	PFI	CUI	—
ATI and CUI	CUI	ATI	—

Section 23 Gigabit Ethernet Controller (GETHER)

This LSI has an on-chip Gigabit Ethernet controller (GETHER) 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 GETHER to perform transmission and reception of Ethernet/IEEE802.3 frames. The LSI has two MAC layer interface ports (hereafter referred to as port 0 and port 1), both of which can be made to perform transmission and reception independently.

The GETHER can transfer the transmitted or received Ethernet frame data to and from the transmit/receive buffer in the memory at high speed using a dedicated direct memory access controller (E-DMAC).

The GETHER also has an on-chip TSU (Transfer Switching Unit) which controls transferring, allowing mutual transfer of data between MAC layer controllers of ports 0 and 1.

23.1 Features

- MAC (Media Access Control) function
  Constructs/deconstructs data frames (frame format conforming to IEEE802.3, 2000 Edition)
  Supports transfer at 10, 100, and 1000 Mbps
  Supports full-duplex and half-duplex modes
  Two channels (GETHER0 and GETHER1)
  Flow control conforming to IEEE802.3x
  Supports three PHY interfaces conforming to IEEE802.3
  — GMII (Gigabit Media Independent Interface)
  — MII (Media Independent Interface)
  — RMII (Reduced Media Independent Interface)
  Upward protocol support (checksum) function
- Switching unit for data transfer between channels (relay FIFO: 6 Kbytes)
- E-DMAC (Direct Memory Access Controller for Ethernet controller) function
  Data transfer between GETHER and external/internal memory
  Four channels
  32-byte burst transfer
  Supports single-frame/single-descriptor operation and single-frame/multi-descriptor (multi-buffer) operation
  Transfer data width: 32 bits

Transmit/receive FIFO (for transmission: 2 Kbytes, for reception: 8 Kbytes)

Figure 23.1 shows the configuration of the GETHER.

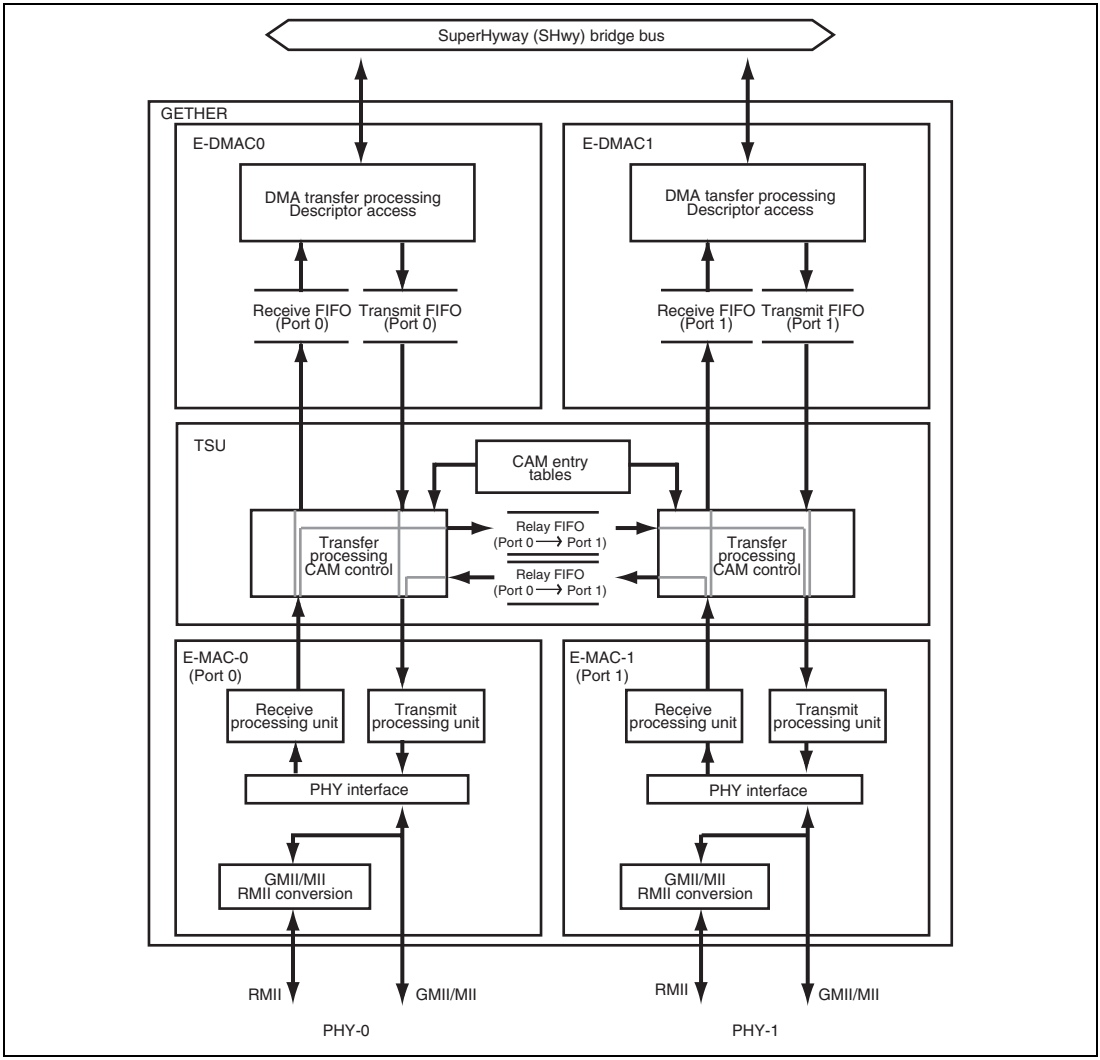


Figure 23.1 Configuration of GETHER

23.2 Input/Output Pins

Table 23.1 lists the pin configuration of the GETHER.

Table 23.1 Pin Configuration

Name	Port	Abbreviation	I/O	Function
Transmit clock	0	ET0_TX-CLK	Input	ET0_TX-EN, ET0_ETXD3 to ET0_ETXD0, ET0_TX-ER timing reference signal
Transmit enable		ET0_TX-EN	Output	Indicates that transmit data is ready on ET0_ETXD3 to ET0_ETXD0
MII/GMII transmit data		ET0_ETXD3 to ET0_ETXD0	Output	4-bit MII transmit data or lower four bits of GMII transmit data
GMII transmit data		GET0_ETXD7 to GET0_ETXD4	Output	Upper four bits of GMII transmit data
Collision detection		ET0_COL	Input	Collision detection signal
Transmit error		ET0_TX-ER	Output	Notifies PHY-LSI of error during transmission
Receive clock		ET0_RX-CLK	Input	ET0_RX-DV, ET0_ERXD3 to ET0_ERXD0, ET0_RX-ER timing reference signal
Receive data valid		ET0_RX-DV	Input	Indicates that valid receive data is on ET0_ERXD3 to ET0_ERXD0
MII/GMII receive data		ET0_ERXD3 to ET0_ERXD0	Input	4-bit MII receive data or lower four bits of GMII receive data (MII and GMII)
GMII receive data		GET0_ERXD7 to GET0_ERXD4	Input	Upper four bits of GMII receive data
Receive error		ET0_RX-ER	Input	Identifies error state occurred during data reception
Carrier detection		ET0_CRS	Input	Carrier detection signal
Management data clock		ET0_MDC	Output	Reference clock signal for information transfer via ET0_MDIO
Management data I/O		ET0_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY

Name	Port	Abbreviation	I/O	Function
RMII management data clock	0	RMII0_MDC	Output	Reference clock signal for information transfer via RMII0_MDIO in RMII mode
RMII management data I/O		RMII0_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY in RMII mode
RMII management data clock (mirror 0 pin)		RMII0M0_MDC	Output	Reference clock signal for information transfer via RMII0M0_MDIO in RMII mode (mirror 0 pin)
RMII management data I/O (mirror 0 pin)		RMII0M0_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY in RMII mode (mirror 0 pin)
RMII management data clock (mirror 1 pin)		RMII0M1_MDC	Output	Reference clock signal for information transfer via RMII0M1_MDIO in RMII mode (mirror 1 pin)
RMII management data I/O (mirror 1 pin)		RMII0M1_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY in RMII mode (mirror 1 pin)
Link status		ET0_LINKSTA	Input	Inputs link status from PHY-LSI
Wake-On-LAN		ET0_WOL	Output	Signal indicating reception of Magic Packet
PHY interrupt		ET0_PHY-INT	Input	Interrupt signal from PHY
GMII transmit clock		GET0_GTX-CLK	Output	Transmit signal timing reference signal in GMII mode
RMII carrier detection		RMII0_CRS_DV	Input	Carrier detection signal in RMII mode
RMII receive error		RMII0_RX_ER	Input	Identifies error state occurred during data reception in RMII mode
RMII receive data		RMII0_RXD0	Input	2-bit receive data in RMII mode
RMII receive data		RMII0_RXD1	Input	2-bit receive data in RMII mode
RMII transmit enable		RMII0_TXD_EN	Output	Indicates that transmit data is ready on RMII0_TXD0 and RMII0_TXD1 in RMII mode

Name	Port	Abbreviation	I/O	Function
RMII transmit data	0	RMII0_TXD0	Output	2-bit transmit data in RMII mode
RMII transmit data		RMII0_TXD1	Output	2-bit transmit data in RMII mode
Transmit clock	1	ET1_TX-CLK	Input	ET1_TX-EN, ET1_ETXD3 to ET1_ETXD0, ET1_TX-ER timing reference signal
Receive clock		ET1_RX-CLK	Input	ET1_RX-DV, ET1_ERXD3 to ET1_ERXD0, ET1_RX-ER timing reference signal
Transmit enable		ET1_TX-EN	Output	Indicates that transmit data is ready on ET1_ETXD3 to ET1_ETXD0
MII/GMII transmit data		ET1_ETXD3 to ET1_ETXD0	Output	4-bit MII transmit data or lower four bits of GMII transmit data
GMII transmit data		GET1_ETXD7 to GET1_ETXD4	Output	Upper four bits of GMII transmit data
Transmit error		ET1_TX-ER	Output	Notifies PHY-LSI of error during transmission
Receive data valid		ET1_RX-DV	Input	Indicates that valid receive data is on ET1_ERXD3 to ET1_ERXD0
MII/GMII receive data		ET1_ERXD3 to ET1_ERXD0	Input	4-bit MII receive data or lower four bits of GMII receive data (MII and GMII)
GMII receive data		GET1_ERXD7 to GET1_ERXD4	Input	Upper four bits of GMII receive data
Receive error		ET1_RX-ER	Input	Identifies error state occurred during data reception
Carrier detection		ET1_CRS	Input	Carrier detection signal
Collision detection		ET1_COL	Input	Collision detection signal
Management data clock		ET1_MDC	Output	Reference clock signal for information transfer via ET1_MDIO
Management data I/O		ET1_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY
RMII management data clock		RMII1_MDC	Output	Reference clock signal for information transfer via RMII1_MDIO in RMII mode

Name	Port	Abbreviation	I/O	Function
RMII management data I/O	1	RMII1_MDIO	I/O	Bidirectional signal for exchange of management information between STA and PHY in RMII mode
Link status		ET1_LINKSTA	Input	Inputs link status from PHY-LSI
Wake-On-LAN		ET1_WOL	Output	Signal indicating reception of Magic Packet
PHY interrupt		ET1_PHY-INT	Input	Interrupt signal from PHY
GMII transmit clock		GET1_GTX-CLK	Output	Transmit signal timing reference signal in GMII mode
RMII carrier detection		RMII1_CRS_DV	Input	Carrier detection signal in RMII mode
RMII receive error		RMII1_RX_ER	Input	Identifies error state occurred during data reception in RMII mode
RMII receive data		RMII1_RXD0	Input	2-bit receive data in RMII mode
RMII receive data		RMII1_RXD1	Input	2-bit receive data in RMII mode
RMII transmit enable		RMII1_TXD_EN	Output	Indicates that transmit data is ready on RMII1_TXD0 and RMII1_TXD1 in RMII mode
RMII transmit data		RMII1_TXD0	Output	2-bit transmit data in RMII mode
RMII transmit data		RMII1_TXD1	Output	2-bit transmit data in RMII mode
RMII carrier detection (mirror pin)		RMII1M_CRS_DV	Input	Carrier detection signal in RMII mode (mirror pin)
RMII receive error (mirror pin)		RMII1M_RX_ER	Input	Identifies error state occurred during data reception in RMII mode (mirror pin)
RMII receive data (mirror pin)		RMII1M_RXD0	Input	2-bit receive data in RMII mode (mirror pin)
RMII receive data (mirror pin)		RMII1M_RXD1	Input	2-bit receive data in RMII mode (mirror pin)
RMII transmit enable (mirror pin)		RMII1M_TXD_EN	Output	Indicates that transmit data is ready on RMII1_TXD0 and RMII1_TXD1 in RMII mode (mirror pin)

Name	Port	Abbreviation	I/O	Function
RMII transmit data (mirror pin)	1	RMII1M_TXD0	Output	2-bit transmit data in RMII mode(mirror pin)
RMII transmit data (mirror pin)		RMII1_TXD1	Output	2-bit transmit data in RMII mode (mirror pin)
125-MHz reference clock	Common	REF125CK	Input	Transmit clock generation signal in GMII mode
50-MHz reference clock	Common	REF50CK	Input	Transmit clock generation signal in RMII mode

Note: * MII signal conforming to IEEE802.3u

23.3 Register Descriptions

Table 23.2 shows the configuration of registers of the GETHER. Table 23.3 shows the state of registers in each processing mode. The last number of the abbreviation of a register, except for registers related to the CAM entry tables, corresponds to the number of the two Ethernet interface ports (port 0 or port 1). Some numbers have been omitted in the text.

Table 23.2 Register Configuration

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Software reset register	ARSTR	R/W	H'FEE0 1800	H'1EE0 1800	32
E-MAC mode register	ECMR0	R/W	H'FEE0 0500	H'1EE0 0500	32
E-MAC status register	ECSR0	R/W	H'FEE0 0510	H'1EE0 0510	32
E-MAC interrupt permission register	ECSIPR0	R/W	H'FEE0 0518	H'1EE0 0518	32
PHY interface register	PIR0	R/W	H'FEE0 0520	H'1EE0 0520	32
MAC address high register	MAHR0	R/W	H'FEE0 05C0	H'1EE0 05C0	32
MAC address low register	MALR0	R/W	H'FEE0 05C8	H'1EE0 05C8	32
Receive frame length register	RFLR0	R/W	H'FEE0 0508	H'1EE0 0508	32
PHY status register	PSR0	R	H'FEE0 0528	H'1EE0 0528	32
PHY_INT polarity register	PIPR0	R/W	H'FEE0 052C	H'1EE0 052C	32
Transmit retry over counter register	TROCR0	R/W	H'FEE0 0700	H'1EE0 0700	32
Delayed collision detect counter register	CDCR0	R/W	H'FEE0 0708	H'1EE0 0708	32
Lost carrier counter register	LCCR0	R/W	H'FEE0 0710	H'1EE0 0710	32
CRC error frame receive counter register	CEFCR0	R/W	H'FEE0 0740	H'1EE0 0740	32
Frame receive error counter register	FRECR0	R/W	H'FEE0 0748	H'1EE0 0748	32
Too-short frame receive counter register	TSFRCR0	R/W	H'FEE0 0750	H'1EE0 0750	32
Too-long frame receive counter register	TLFRCR0	R/W	H'FEE0 0758	H'1EE0 0758	32
Residual-bit frame receive counter register	RFCR0	R/W	H'FEE0 0760	H'1EE0 0760	32
Carrier extension loss counter register	CERCR0	R/W	H'FEE0 0768	H'1EE0 0768	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Carrier extension error counter register	CEECR0	R/W	H'FEE0 0770	H'1EE0 0770	32
Multicast address frame receive counter register	MAFCR0	R/W	H'FEE0 0778	H'1EE0 0778	32
Automatic PAUSE frame register	APR0	R/W	H'FEE0 0554	H'1EE0 0554	32
Manual PAUSE frame register	MPR0	R/W	H'FEE0 0558	H'1EE0 0558	32
Automatic PAUSE frame retransmit count register	TPAUSER0	R/W	H'FEE0 0564	H'1EE0 0564	32
PAUSE frame transmit counter register	PFTCR0	R	H'FEE0 055C	H'1EE0 055C	32
PAUSE frame receive counter register	PFRCR0	R	H'FEE0 0560	H'1EE0 0560	32
GETHER mode register	GECMR0	R/W	H'FEE0 05B0	H'1EE0 05B0	32
Burst cycle count upper-limit register	BCULR0	R/W	H'FEE0 05B4	H'1EE0 05B4	32
E-MAC mode register	ECMR1	R/W	H'FEE0 0D00	H'1EE0 0D00	32
E-MAC status register	ECSR1	R/W	H'FEE0 0D10	H'1EE0 0D10	32
E-MAC interrupt permission register	ECSIPR1	R/W	H'FEE0 0D18	H'1EE0 0D18	32
PHY interface register	PIR1	R/W	H'FEE0 0D20	H'1EE0 0D20	32
PHY_INT polarity register	PIPR1	R/W	H'FEE0 0D2C	H'1EE0 0D2C	32
MAC address high register	MAHR1	R/W	H'FEE0 0DC0	H'1EE0 0DC0	32
MAC address low register	MALR1	R/W	H'FEE0 0DC8	H'1EE0 0DC8	32
Receive frame length register	RFLR1	R/W	H'FEE0 0D08	H'1EE0 0D08	32
PHY status register	PSR1	R	H'FEE0 0D28	H'1EE0 0D28	32
Transmit retry over counter register	TROCR1	R/W	H'FEE0 0F00	H'1EE0 0F00	32
Delayed collision detect counter register	CDCR1	R/W	H'FEE0 0F08	H'1EE0 0F08	32
Lost carrier counter register	LCCR1	R/W	H'FEE0 0F10	H'1EE0 0F10	32
CRC error frame receive counter register	CEFCR1	R/W	H'FEE0 0F40	H'1EE0 0F40	32
Frame receive error counter register	FRECR1	R/W	H'FEE0 0F48	H'1EE0 0F48	32
Too-short frame receive counter register	TSFRCR1	R/W	H'FEE0 0F50	H'1EE0 0F50	32
Too-long frame receive counter register	TLFRCR1	R/W	H'FEE0 0F58	H'1EE0 0F58	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Residual-bit frame receive counter register	RFCR1	R/W	H'FEE0 0F60	H'1EE0 0F60	32
Carrier extension loss counter register	CERCR1	R/W	H'FEE0 0F68	H'1EE0 0F68	32
Carrier extension error counter register	CEECR1	R/W	H'FEE0 0F70	H'1EE0 0F70	32
Multicast address frame receive counter register	MAFCR1	R/W	H'FEE0 0F78	H'1EE0 0F78	32
Automatic PAUSE frame register	APR1	R/W	H'FEE0 0D54	H'1EE0 0D54	32
Manual PAUSE frame register	MPR1	R/W	H'FEE0 0D58	H'1EE0 0D58	32
Automatic PAUSE frame retransmit count register	TPAUSER1	R/W	H'FEE0 0D64	H'1EE0 0D64	32
PAUSE frame transmit counter register	PFTCR1	R	H'FEE0 0D5C	H'1EE0 0D5C	32
PAUSE frame receive counter register	PFRCR1	R	H'FEE0 0D60	H'1EE0 0D60	32
GETHER mode register	GECMR1	R/W	H'FEE0 0DB0	H'1EE0 0DB0	32
Burst cycle count upper-limit register	BCULR1	R/W	H'FEE0 0DB4	H'1EE0 0DB4	32
TSU counter reset register	TSU_CTRST	R/W	H'FEE0 1804	H'1EE0 1804	32
Relay enable register (Port 0 to 1)	TSU_FWEN0	R/W	H'FEE0 1810	H'1EE0 1810	32
Relay enable register (Port 1 to 0)	TSU_FWEN1	R/W	H'FEE0 1814	H'1EE0 1814	32
Relay FIFO size select register	TSU_FCM	R/W	H'FEE0 1818	H'1EE0 1818	32
Relay FIFO overflow alert set register (port 0)	TSU_BSYSL0	R/W	H'FEE0 1820	H'1EE0 1820	32
Relay FIFO overflow alert set register (port 1)	TSU_BSYSL1	R/W	H'FEE0 1824	H'1EE0 1824	32
Transmit/relay priority control mode register (port 0)	TSU_PRISL0	R/W	H'FEE0 1828	H'1EE0 1828	32
Transmit/relay priority control mode register (port 1)	TSU_PRISL1	R/W	H'FEE0 182C	H'1EE0 182C	32
Receive/relay function set register (port 0 to 1)	TSU_FWSL0	R/W	H'FEE0 1830	H'1EE0 1830	32
Receive/relay function set register (port 1 to 0)	TSU_FWSL1	R/W	H'FEE0 1834	H'1EE0 1834	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Relay function set register (common)	TSU_FWSLC	R/W	H'FEE0 1838	H'1EE0 1838	32
Qtag addition/deletion set register (port 0 to 1)	TSU_QTAG0	R/W	H'FEE0 1840	H'1EE0 1840	32
Qtag addition/deletion set register (port 1 to 0)	TSU_QTAG1	R/W	H'FEE0 1844	H'1EE0 1844	32
Relay status register	TSU_FWSR	R/W	H'FEE0 1850	H'1EE0 1850	32
Relay status interrupt mask register	TSU_FWINMK	R/W	H'FEE0 1854	H'1EE0 1854	32
Added Qtag value set register (port 0 to 1)	TSU_ADQT0	R/W	H'FEE0 1848	H'1EE0 1848	32
Added Qtag value set register (port 1 to 0)	TSU_ADQT1	R/W	H'FEE0 184C	H'1EE0 184C	32
VLANtag set register (port 0)	TSU_VTAG0	R/W	H'FEE0 1858	H'1EE0 1858	32
VLANtag set register (port 1)	TSU_VTAG1	R/W	H'FEE0 185C	H'1EE0 185C	32
CAM entry table busy register	TSU_ADSBSY	R	H'FEE0 1860	H'1EE0 1860	32
CAM entry table enable register	TSU_TEN	R/W	H'FEE0 1864	H'1EE0 1864	32
CAM entry table POST1 register	TSU_POST1	R/W	H'FEE0 1870	H'1EE0 1870	32
CAM entry table POST2 register	TSU_POST2	R/W	H'FEE0 1874	H'1EE0 1874	32
CAM entry table POST3 register	TSU_POST3	R/W	H'FEE0 1878	H'1EE0 1878	32
CAM entry table POST4 register	TSU_POST4	R/W	H'FEE0 187C	H'1EE0 187C	32
CAM entry table 0H register	TSU_ADRH0	R/W	H'FEE0 1900	H'1EE0 1900	32
CAM entry table 1H register	TSU_ADRH1	R/W	H'FEE0 1908	H'1EE0 1908	32
CAM entry table 2H register	TSU_ADRH2	R/W	H'FEE0 1910	H'1EE0 1910	32
CAM entry table 3H register	TSU_ADRH3	R/W	H'FEE0 1918	H'1EE0 1918	32
CAM entry table 4H register	TSU_ADRH4	R/W	H'FEE0 1920	H'1EE0 1920	32
CAM entry table 5H register	TSU_ADRH5	R/W	H'FEE0 1928	H'1EE0 1928	32
CAM entry table 6H register	TSU_ADRH6	R/W	H'FEE0 1930	H'1EE0 1930	32
CAM entry table 7H register	TSU_ADRH7	R/W	H'FEE0 1938	H'1EE0 1938	32
CAM entry table 8H register	TSU_ADRH8	R/W	H'FEE0 1940	H'1EE0 1940	32
CAM entry table 9H register	TSU_ADRH9	R/W	H'FEE0 1948	H'1EE0 1948	32
CAM entry table 10H register	TSU_ADRH10	R/W	H'FEE0 1950	H'1EE0 1950	32
CAM entry table 11H register	TSU_ADRH11	R/W	H'FEE0 1958	H'1EE0 1958	32
CAM entry table 12H register	TSU_ADRH12	R/W	H'FEE0 1960	H'1EE0 1960	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
CAM entry table 13H register	TSU_ADRH13	R/W	H'FEE0 1968	H'1EE0 1968	32
CAM entry table 14H register	TSU_ADRH14	R/W	H'FEE0 1970	H'1EE0 1970	32
CAM entry table 15H register	TSU_ADRH15	R/W	H'FEE0 1978	H'1EE0 1978	32
CAM entry table 16H register	TSU_ADRH16	R/W	H'FEE0 1980	H'1EE0 1980	32
CAM entry table 17H register	TSU_ADRH17	R/W	H'FEE0 1988	H'1EE0 1988	32
CAM entry table 18H register	TSU_ADRH18	R/W	H'FEE0 1990	H'1EE0 1990	32
CAM entry table 19H register	TSU_ADRH19	R/W	H'FEE0 1998	H'1EE0 1998	32
CAM entry table 20H register	TSU_ADRH20	R/W	H'FEE0 19A0	H'1EE0 19A0	32
CAM entry table 21H register	TSU_ADRH21	R/W	H'FEE0 19A8	H'1EE0 19A8	32
CAM entry table 22H register	TSU_ADRH22	R/W	H'FEE0 19B0	H'1EE0 19B0	32
CAM entry table 23H register	TSU_ADRH23	R/W	H'FEE0 19B8	H'1EE0 19B8	32
CAM entry table 24H register	TSU_ADRH24	R/W	H'FEE0 19C0	H'1EE0 19C0	32
CAM entry table 25H register	TSU_ADRH25	R/W	H'FEE0 19C8	H'1EE0 19C8	32
CAM entry table 26H register	TSU_ADRH26	R/W	H'FEE0 19D0	H'1EE0 19D0	32
CAM entry table 27H register	TSU_ADRH27	R/W	H'FEE0 19D8	H'1EE0 19D8	32
CAM entry table 28H register	TSU_ADRH28	R/W	H'FEE0 19E0	H'1EE0 19E0	32
CAM entry table 29H register	TSU_ADRH29	R/W	H'FEE0 19E8	H'1EE0 19E8	32
CAM entry table 30H register	TSU_ADRH30	R/W	H'FEE0 19F0	H'1EE0 19F0	32
CAM entry table 31H register	TSU_ADRH31	R/W	H'FEE0 19F8	H'1EE0 19F8	32
CAM entry table 0L register	TSU_ADRL0	R/W	H'FEE0 1904	H'1EE0 1904	32
CAM entry table 1L register	TSU_ADRL1	R/W	H'FEE0 190C	H'1EE0 190C	32
CAM entry table 2L register	TSU_ADRL2	R/W	H'FEE0 1914	H'1EE0 1914	32
CAM entry table 3L register	TSU_ADRL3	R/W	H'FEE0 191C	H'1EE0 191C	32
CAM entry table 4L register	TSU_ADRL4	R/W	H'FEE0 1924	H'1EE0 1924	32
CAM entry table 5L register	TSU_ADRL5	R/W	H'FEE0 192C	H'1EE0 192C	32
CAM entry table 6L register	TSU_ADRL6	R/W	H'FEE0 1934	H'1EE0 1934	32
CAM entry table 7L register	TSU_ADRL7	R/W	H'FEE0 193C	H'1EE0 193C	32
CAM entry table 8L register	TSU_ADRL8	R/W	H'FEE0 1944	H'1EE0 1944	32
CAM entry table 9L register	TSU_ADRL9	R/W	H'FEE0 194C	H'1EE0 194C	32
CAM entry table 10L register	TSU_ADRL10	R/W	H'FEE0 1954	H'1EE0 1954	32
CAM entry table 11L register	TSU_ADRL11	R/W	H'FEE0 195C	H'1EE0 195C	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
CAM entry table 12L register	TSU_ADRL12	R/W	H'FEE0 1964	H'1EE0 1964	32
CAM entry table 13L register	TSU_ADRL13	R/W	H'FEE0 196C	H'1EE0 196C	32
CAM entry table 14L register	TSU_ADRL14	R/W	H'FEE0 1974	H'1EE0 1974	32
CAM entry table 15L register	TSU_ADRL15	R/W	H'FEE0 197C	H'1EE0 197C	32
CAM entry table 16L register	TSU_ADRL16	R/W	H'FEE0 1984	H'1EE0 1984	32
CAM entry table 17L register	TSU_ADRL17	R/W	H'FEE0 198C	H'1EE0 198C	32
CAM entry table 18L register	TSU_ADRL18	R/W	H'FEE0 1994	H'1EE0 1994	32
CAM entry table 19L register	TSU_ADRL19	R/W	H'FEE0 199C	H'1EE0 199C	32
CAM entry table 20L register	TSU_ADRL20	R/W	H'FEE0 19A4	H'1EE0 19A4	32
CAM entry table 21L register	TSU_ADRL21	R/W	H'FEE0 19AC	H'1EE0 19AC	32
CAM entry table 22L register	TSU_ADRL22	R/W	H'FEE0 19B4	H'1EE0 19B4	32
CAM entry table 23L register	TSU_ADRL23	R/W	H'FEE0 19BC	H'1EE0 19BC	32
CAM entry table 24L register	TSU_ADRL24	R/W	H'FEE0 19C4	H'1EE0 19C4	32
CAM entry table 25L register	TSU_ADRL25	R/W	H'FEE0 19CC	H'1EE0 19CC	32
CAM entry table 26L register	TSU_ADRL26	R/W	H'FEE0 19D4	H'1EE0 19D4	32
CAM entry table 27L register	TSU_ADRL27	R/W	H'FEE0 19DC	H'1EE0 19DC	32
CAM entry table 28L register	TSU_ADRL28	R/W	H'FEE0 19E4	H'1EE0 19E4	32
CAM entry table 29L register	TSU_ADRL29	R/W	H'FEE0 19EC	H'1EE0 19EC	32
CAM entry table 30L register	TSU_ADRL30	R/W	H'FEE0 19F4	H'1EE0 19F4	32
CAM entry table 31L register	TSU_ADRL31	R/W	H'FEE0 19FC	H'1EE0 19FC	32
Transmit frame counter register (port 0) (normal transmission only)	TXNLCR0	R	H'FEE0 1880	H'1EE0 1880	32
Transmit frame counter register (port 0) (normal and erroneous transmission)	TXALCR0	R	H'FEE0 1884	H'1EE0 1884	32
Receive frame counter register (port 0) (normal reception only)	RXNLCR0	R	H'FEE0 1888	H'1EE0 1888	32
Receive frame counter register (port 0) (normal and erroneous reception)	RXALCR0	R	H'FEE0 188C	H'1EE0 188C	32
Relay frame counter register (port 1 to 0) (normal relay only)	FWNLCR0	R	H'FEE0 1890	H'1EE0 1890	32
Relay frame counter register (port 1 to 0) (normal and erroneous relay)	FWALCR0	R	H'FEE0 1894	H'1EE0 1894	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Transmit frame counter register (port 1) (normal transmission only)	TXNLCR1	R	H'FEE0 18A0	H'1EE0 18A0	32
Transmit frame counter register (port 1) (normal and erroneous transmission)	TXALCR1	R	H'FEE0 18A4	H'1EE0 18A4	32
Receive frame counter register (port 1) (normal reception only)	RXNLCR1	R	H'FEE0 18A8	H'1EE0 18A8	32
Receive frame counter register (port 1) (normal and erroneous reception)	RXALCR1	R	H'FEE0 18AC	H'1EE0 18AC	32
Relay frame counter register (port 0 to 1) (normal relay only)	FWNLCR1	R	H'FEE0 18B0	H'1EE0 18B0	32
Relay frame counter register (port 0 to 1) (normal and erroneous relay)	FWALCR1	R	H'FEE0 18B4	H'1EE0 18B4	32
E-DMAC start register	EDSR0	W	H'FEE0 0000	H'1EE0 0000	32
E-DMAC mode register	EDMR0	R/W	H'FEE0 0400	H'1EE0 0400	32
E-DMAC transmit request register	EDTRR0	R/W	H'FEE0 0408	H'1EE0 0408	32
E-DMAC receive request register	EDRRR0	R/W	H'FEE0 0410	H'1EE0 0410	32
Transmit descriptor list start address register	TDLAR0	R/W	H'FEE0 0010	H'1EE0 0010	32
Receive descriptor list start address register	RDLAR0	R/W	H'FEE0 0030	H'1EE0 0030	32
E-MAC/E-DMAC status register	EESR0	R/W	H'FEE0 0428	H'1EE0 0428	32
E-MAC/E-DMAC status interrupt permission register	EESIPR0	R/W	H'FEE0 0430	H'1EE0 0430	32
Transmit/receive status copy enable register	TRSCER0	R/W	H'FEE0 0438	H'1EE0 0438	32
Receive missed-frame counter register	RMFCR0	R/W	H'FEE0 0440	H'1EE0 0440	32
Transmit FIFO threshold register	TFTR0	R/W	H'FEE0 0448	H'1EE0 0448	32
FIFO depth register	FDR0	R/W	H'FEE0 0450	H'1EE0 0450	32
Receiving method control register	RMCR0	R/W	H'FEE0 0458	H'1EE0 0458	32
Receive descriptor fetch address register	RDFAR0	R/W	H'FEE0 0034	H'1EE0 0034	32
Receive descriptor finished address register	RDFXR0	R/W	H'FEE0 0038	H'1EE0 0038	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Receive descriptor final flag register	RDFFR0	R/W	H'FEE0 003C	H'1EE0 003C	32
Transmit descriptor fetch address register	TDFAR0	R/W	H'FEE0 0014	H'1EE0 0014	32
Transmit descriptor finished address register	TDFXR0	R/W	H'FEE0 0018	H'1EE0 0018	32
Transmit descriptor final flag register	TDFFR0	R/W	H'FEE0 001C	H'1EE0 001C	32
Overflow alert FIFO threshold register	FCFTR0	R/W	H'FEE0 0468	H'1EE0 0468	32
Receive data padding insert register	RPADIR0	R/W	H'FEE0 0460	H'1EE0 0460	32
E-DMAC start register	EDSR1	W	H'FEE0 0800	H'1EE0 0800	32
E-DMAC mode register	EDMR1	R/W	H'FEE0 0C00	H'1EE0 0C00	32
E-DMAC transmit request register	EDTRR1	R/W	H'FEE0 0C08	H'1EE0 0C08	32
E-DMAC receive request register	EDRRR1	R/W	H'FEE0 0C10	H'1EE0 0C10	32
Transmit descriptor list start address register	TDLAR1	R/W	H'FEE0 0810	H'1EE0 0810	32
Receive descriptor list start address register	RDLAR1	R/W	H'FEE0 0830	H'1EE0 0830	32
E-MAC/E-DMAC status register	EESR1	R/W	H'FEE0 0C28	H'1EE0 0C28	32
E-MAC/E-DMAC status interrupt permission register	EESIPR1	R/W	H'FEE0 0C30	H'1EE0 0C30	32
Transmit/receive status copy enable register	TRSCER1	R/W	H'FEE0 0C38	H'1EE0 0C38	32
Receive missed-frame counter register	RMFCR1	R/W	H'FEE0 0C40	H'1EE0 0C40	32
Transmit FIFO threshold register	TFTR1	R/W	H'FEE0 0C48	H'1EE0 0C48	32
FIFO depth register	FDR1	R/W	H'FEE0 0C50	H'1EE0 0C50	32
Receiving method control register	RMCR1	R/W	H'FEE0 0C58	H'1EE0 0C58	32
Receive descriptor fetch address register	RDFAR1	R/W	H'FEE0 0834	H'1EE0 0834	32
Receive descriptor finished address register	RDFXR1	R/W	H'FEE0 0838	H'1EE0 0838	32
Receive descriptor final flag register	RDFFR1	R/W	H'FEE0 083C	H'1EE0 083C	32
Transmit descriptor fetch address register	TDFAR1	R/W	H'FEE0 0814	H'1EE0 0814	32

Name	Abbreviation	R/W	P4 Area Address	Area 7 Address	Access Size
Transmit descriptor finished address register	TDFXR1	R/W	H'FEE0 0818	H'1EE0 0818	32
Transmit descriptor final flag register	TDFFR1	R/W	H'FEE0 081C	H'1EE0 081C	32
Overflow alert FIFO threshold register	FCFTR1	R/W	H'FEE0 0C68	H'1EE0 0C68	32
Receive data padding insert register	RPADIR1	R/W	H'FEE0 0C60	H'1EE0 0C60	32

Table 23.3 Register States in Each Operating Mode

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Software reset register	ARSTR	H'00000000	H'00000000	Retained	Retained
E-MAC mode register	ECMR0	H'00000000	H'00000000	Retained	Retained
E-MAC status register	ECSR0	H'00000000	H'00000000	Retained	Retained
E-MAC interrupt permission register	ECSIPR0	H'00000000	H'00000000	Retained	Retained
PHY interface register	PIR0	H'0000000x	H'0000000x	Retained	Retained
MAC address high register	MAHR0	H'00000000	H'00000000	Retained	Retained
MAC address low register	MALR0	H'00000000	H'00000000	Retained	Retained
Receive frame length register	RFLR0	H'00000000	H'00000000	Retained	Retained
PHY status register	PSR0	H'00000000	H'00000000	Retained	Retained
PHY_INT polarity register	PIPR0	H'00000000	H'00000000	Retained	Retained
Transmit retry over counter register	TROCR0	H'00000000	H'00000000	Retained	Retained
Delayed collision detect counter register	CDCR0	H'00000000	H'00000000	Retained	Retained
Lost carrier counter register	LCCR0	H'00000000	H'00000000	Retained	Retained
CRC error frame receive counter register	CEFCR0	H'00000000	H'00000000	Retained	Retained
Frame receive error counter register	FRECR0	H'00000000	H'00000000	Retained	Retained
Too-short frame receive counter register	TSFRCR0	H'00000000	H'00000000	Retained	Retained
Too-long frame receive counter register	TLFRCR0	H'00000000	H'00000000	Retained	Retained
Residual-bit frame receive counter register	RFCR0	H'00000000	H'00000000	Retained	Retained
Carrier extension loss counter register	CERCR0	H'00000000	H'00000000	Retained	Retained
Carrier extension error counter register	CEECR0	H'00000000	H'00000000	Retained	Retained
Multicast address frame receive counter register	MAFCR0	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Automatic PAUSE frame register	APR0	H'00000000	H'00000000	Retained	Retained
Manual PAUSE frame register	MPR0	H'00000000	H'00000000	Retained	Retained
Automatic PAUSE frame retransmit count register	TPAUSER0	H'00000000	H'00000000	Retained	Retained
PAUSE frame transmit counter register	PFTCR0	H'00000000	H'00000000	Retained	Retained
PAUSE frame receive counter register	PFRCR0	H'00000000	H'00000000	Retained	Retained
GETHER mode register	GECMR0	H'00000000	H'00000000	Retained	Retained
Burst cycle count upper-limit register	BCULR0	H'00000000	H'00000000	Retained	Retained
E-MAC mode register	ECMR1	H'00000000	H'00000000	Retained	Retained
E-MAC status register	ECSR1	H'00000000	H'00000000	Retained	Retained
E-MAC interrupt permission register	ECSIPR1	H'00000000	H'00000000	Retained	Retained
PHY interface register	PIR1	H'0000000x	H'0000000x	Retained	Retained
PHY_INT polarity register	PIPR1	H'00000000	H'00000000	Retained	Retained
MAC address high register	MAHR1	H'00000000	H'00000000	Retained	Retained
MAC address low register	MALR1	H'00000000	H'00000000	Retained	Retained
Receive frame length register	RFLR1	H'00000000	H'00000000	Retained	Retained
PHY status register	PSR1	H'00000000	H'00000000	Retained	Retained
Transmit retry over counter register	TROCR1	H'00000000	H'00000000	Retained	Retained
Delayed collision detect counter register	CDCR1	H'00000000	H'00000000	Retained	Retained
Lost carrier counter register	LCCR1	H'00000000	H'00000000	Retained	Retained
CRC error frame receive counter register	CEFCR1	H'00000000	H'00000000	Retained	Retained
Frame receive error counter register	FRECR1	H'00000000	H'00000000	Retained	Retained
Too-short frame receive counter register	TSFRCR1	H'00000000	H'00000000	Retained	Retained
Too-long frame receive counter register	TLFRCR1	H'00000000	H'00000000	Retained	Retained
Residual-bit frame receive counter register	RFCR1	H'00000000	H'00000000	Retained	Retained
Carrier extension loss counter register	CERCR1	H'00000000	H'00000000	Retained	Retained
Carrier extension error counter register	CEECR1	H'00000000	H'00000000	Retained	Retained
Multicast address frame receive counter register	MAFCR1	H'00000000	H'00000000	Retained	Retained
Automatic PAUSE frame register	APR1	H'00000000	H'00000000	Retained	Retained
Manual PAUSE frame register	MPR1	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Automatic PAUSE frame retransmit count register	TPAUSER1	H'00000000	H'00000000	Retained	Retained
PAUSE frame transmit counter register	PFTCR1	H'00000000	H'00000000	Retained	Retained
PAUSE frame receive counter register	PFRCR1	H'00000000	H'00000000	Retained	Retained
GETHER mode register	GECMR1	H'00000000	H'00000000	Retained	Retained
Burst cycle count upper-limit register	BCULR1	H'00000000	H'00000000	Retained	Retained
TSU counter reset register	TSU_CTRST	H'00000000	H'00000000	Retained	Retained
Relay enable register (Port 0 to 1)	TSU_FWEN0	H'00000000	H'00000000	Retained	Retained
Relay enable register (Port 1 to 0)	TSU_FWEN1	H'00000000	H'00000000	Retained	Retained
Relay FIFO size select register	TSU_FCM	H'00000000	H'00000000	Retained	Retained
Relay FIFO overflow alert set register (port 0)	TSU_BSYSL0	H'0000003F	H'0000003F	Retained	Retained
Relay FIFO overflow alert set register (port 1)	TSU_BSYSL1	H'0000003F	H'0000003F	Retained	Retained
Transmit/relay priority control mode register (port 0)	TSU_PRISL0	H'00000000	H'00000000	Retained	Retained
Transmit/relay priority control mode register (port 1)	TSU_PRISL1	H'00000000	H'00000000	Retained	Retained
Receive/relay function set register (port 0 to 1)	TSU_FWSL0	H'00000000	H'00000000	Retained	Retained
Receive/relay function set register (port 1 to 0)	TSU_FWSL1	H'00000000	H'00000000	Retained	Retained
Relay function set register (common)	TSU_FWSLC	H'00000000	H'00000000	Retained	Retained
Qtag addition/deletion set register (port 0 to 1)	TSU_QTAG0	H'00000000	H'00000000	Retained	Retained
Qtag addition/deletion set register (port 1 to 0)	TSU_QTAG1	H'00000000	H'00000000	Retained	Retained
Relay status register	TSU_FWSR	H'00000000	H'00000000	Retained	Retained
Relay status interrupt mask register	TSU_FWINMK	H'00000000	H'00000000	Retained	Retained
Added Qtag value set register (port 0 to 1)	TSU_ADQT0	H'81000000	H'81000000	Retained	Retained
Added Qtag value set register (port 1 to 0)	TSU_ADQT1	H'81000000	H'81000000	Retained	Retained
VLANtag set register (port 0)	TSU_VTAG0	H'00000000	H'00000000	Retained	Retained
VLANtag set register (port 1)	TSU_VTAG1	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
CAM entry table busy register	TSU_ ADSBSY	H'00000000	H'00000000	Retained	Retained
CAM entry table enable register	TSU_TEN	H'00000000	H'00000000	Retained	Retained
CAM entry table POST1 register	TSU_POST1	H'00000000	H'00000000	Retained	Retained
CAM entry table POST2 register	TSU_POST2	H'00000000	H'00000000	Retained	Retained
CAM entry table POST3 register	TSU_POST3	H'00000000	H'00000000	Retained	Retained
CAM entry table POST4 register	TSU_POST4	H'00000000	H'00000000	Retained	Retained
CAM entry table 0H register	TSU_ADRH0	H'00000000	H'00000000	Retained	Retained
CAM entry table 1H register	TSU_ADRH1	H'00000000	H'00000000	Retained	Retained
CAM entry table 2H register	TSU_ADRH2	H'00000000	H'00000000	Retained	Retained
CAM entry table 3H register	TSU_ADRH3	H'00000000	H'00000000	Retained	Retained
CAM entry table 4H register	TSU_ADRH4	H'00000000	H'00000000	Retained	Retained
CAM entry table 5H register	TSU_ADRH5	H'00000000	H'00000000	Retained	Retained
CAM entry table 6H register	TSU_ADRH6	H'00000000	H'00000000	Retained	Retained
CAM entry table 7H register	TSU_ADRH7	H'00000000	H'00000000	Retained	Retained
CAM entry table 8H register	TSU_ADRH8	H'00000000	H'00000000	Retained	Retained
CAM entry table 9H register	TSU_ADRH9	H'00000000	H'00000000	Retained	Retained
CAM entry table 10H register	TSU_ ADRH10	H'00000000	H'00000000	Retained	Retained
CAM entry table 11H register	TSU_ ADRH11	H'00000000	H'00000000	Retained	Retained
CAM entry table 12H register	TSU_ ADRH12	H'00000000	H'00000000	Retained	Retained
CAM entry table 13H register	TSU_ADRH13	H'00000000	H'00000000	Retained	Retained
CAM entry table 14H register	TSU_ ADRH14	H'00000000	H'00000000	Retained	Retained
CAM entry table 15H register	TSU_ ADRH15	H'00000000	H'00000000	Retained	Retained
CAM entry table 16H register	TSU_ ADRH16	H'00000000	H'00000000	Retained	Retained
CAM entry table 17H register	TSU_ ADRH17	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
CAM entry table 18H register	TSU_ ADRH18	H'00000000	H'00000000	Retained	Retained
CAM entry table 19H register	TSU_ ADRH19	H'00000000	H'00000000	Retained	Retained
CAM entry table 20H register	TSU_ ADRH20	H'00000000	H'00000000	Retained	Retained
CAM entry table 21H register	TSU_ ADRH21	H'00000000	H'00000000	Retained	Retained
CAM entry table 22H register	TSU_ ADRH22	H'00000000	H'00000000	Retained	Retained
CAM entry table 23H register	TSU_ ADRH23	H'00000000	H'00000000	Retained	Retained
CAM entry table 24H register	TSU_ ADRH24	H'00000000	H'00000000	Retained	Retained
CAM entry table 25H register	TSU_ ADRH25	H'00000000	H'00000000	Retained	Retained
CAM entry table 26H register	TSU_ ADRH26	H'00000000	H'00000000	Retained	Retained
CAM entry table 27H register	TSU_ ADRH27	H'00000000	H'00000000	Retained	Retained
CAM entry table 28H register	TSU_ ADRH28	H'00000000	H'00000000	Retained	Retained
CAM entry table 29H register	TSU_ ADRH29	H'00000000	H'00000000	Retained	Retained
CAM entry table 30H register	TSU_ ADRH30	H'00000000	H'00000000	Retained	Retained
CAM entry table 31H register	TSU_ ADRH31	H'00000000	H'00000000	Retained	Retained
CAM entry table 0L register	TSU_ADRL0	H'00000000	H'00000000	Retained	Retained
CAM entry table 1L register	TSU_ADRL1	H'00000000	H'00000000	Retained	Retained
CAM entry table 2L register	TSU_ADRL2	H'00000000	H'00000000	Retained	Retained
CAM entry table 3L register	TSU_ADRL3	H'00000000	H'00000000	Retained	Retained
CAM entry table 4L register	TSU_ADRL4	H'00000000	H'00000000	Retained	Retained
CAM entry table 5L register	TSU_ADRL5	H'00000000	H'00000000	Retained	Retained
CAM entry table 6L register	TSU_ADRL6	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
CAM entry table 7L register	TSU_ADRL7	H'00000000	H'00000000	Retained	Retained
CAM entry table 8L register	TSU_ADRL8	H'00000000	H'00000000	Retained	Retained
CAM entry table 9L register	TSU_ADRL9	H'00000000	H'00000000	Retained	Retained
CAM entry table 10L register	TSU_ ADRL10	H'00000000	H'00000000	Retained	Retained
CAM entry table 11L register	TSU_ ADRL11	H'00000000	H'00000000	Retained	Retained
CAM entry table 12L register	TSU_ ADRL12	H'00000000	H'00000000	Retained	Retained
CAM entry table 13L register	TSU_ ADRL13	H'00000000	H'00000000	Retained	Retained
CAM entry table 14L register	TSU_ ADRL14	H'00000000	H'00000000	Retained	Retained
CAM entry table 15L register	TSU_ ADRL15	H'00000000	H'00000000	Retained	Retained
CAM entry table 16L register	TSU_ ADRL16	H'00000000	H'00000000	Retained	Retained
CAM entry table 17L register	TSU_ ADRL17	H'00000000	H'00000000	Retained	Retained
CAM entry table 18L register	TSU_ ADRL18	H'00000000	H'00000000	Retained	Retained
CAM entry table 19L register	TSU_ ADRL19	H'00000000	H'00000000	Retained	Retained
CAM entry table 20L register	TSU_ ADRL20	H'00000000	H'00000000	Retained	Retained
CAM entry table 21L register	TSU_ ADRL21	H'00000000	H'00000000	Retained	Retained
CAM entry table 22L register	TSU_ ADRL22	H'00000000	H'00000000	Retained	Retained
CAM entry table 23L register	TSU_ ADRL23	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
CAM entry table 24L register	TSU_ ADRL24	H'00000000	H'00000000	Retained	Retained
CAM entry table 25L register	TSU_ ADRL25	H'00000000	H'00000000	Retained	Retained
CAM entry table 26L register	TSU_ ADRL26	H'00000000	H'00000000	Retained	Retained
CAM entry table 27L register	TSU_ ADRL27	H'00000000	H'00000000	Retained	Retained
CAM entry table 28L register	TSU_ ADRL28	H'00000000	H'00000000	Retained	Retained
CAM entry table 29L register	TSU_ ADRL29	H'00000000	H'00000000	Retained	Retained
CAM entry table 30L register	TSU_ ADRL30	H'00000000	H'00000000	Retained	Retained
CAM entry table 31L register	TSU_ ADRL31	H'00000000	H'00000000	Retained	Retained
Transmit frame counter register (port 0) (normal transmission only)	TXNLCR0	H'00000000	H'00000000	Retained	Retained
Transmit frame counter register (port 0) (normal and erroneous transmission)	TXALCR0	H'00000000	H'00000000	Retained	Retained
Receive frame counter register (port 0) (normal reception only)	RXNLCR0	H'00000000	H'00000000	Retained	Retained
Receive frame counter register (port 0) (normal and erroneous reception)	RXALCR0	H'00000000	H'00000000	Retained	Retained
Relay frame counter register (port 1 to 0) (normal relay only)	FWNLCR0	H'00000000	H'00000000	Retained	Retained
Relay frame counter register (port 1 to 0) (normal and erroneous relay)	FWALCR0	H'00000000	H'00000000	Retained	Retained
Transmit frame counter register (port 1) (normal transmission only)	TXNLCR1	H'00000000	H'00000000	Retained	Retained
Transmit frame counter register (port 1) (normal and erroneous transmission)	TXALCR1	H'00000000	H'00000000	Retained	Retained
Receive frame counter register (port 1) (normal reception only)	RXNLCR1	H'00000000	H'00000000	Retained	Retained
Receive frame counter register (port 1) (normal and erroneous reception)	RXALCR1	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Relay frame counter register (port 0 to 1) (normal relay only)	FWNLCR1	H'00000000	H'00000000	Retained	Retained
Relay frame counter register (port 0 to 1) (normal and erroneous relay)	FWALCR1	H'00000000	H'00000000	Retained	Retained
E-DMAC start register	EDSR0	H'00000000	H'00000000	Retained	Retained
E-DMAC mode register	EDMR0	H'00000000	H'00000000	Retained	Retained
E-DMAC transmit request register	EDTRR0	H'00000000	H'00000000	Retained	Retained
E-DMAC receive request register	EDRRR0	H'00000000	H'00000000	Retained	Retained
Transmit descriptor list start address register	TDLAR0	H'00000000	H'00000000	Retained	Retained
Receive descriptor list start address register	RDLAR0	H'00000000	H'00000000	Retained	Retained
E-MAC/E-DMAC status register	EESR0	H'00000000	H'00000000	Retained	Retained
E-MAC/E-DMAC status interrupt permission register	EESIPR0	H'00000000	H'00000000	Retained	Retained
Transmit/receive status copy enable register	TRSCER0	H'00000000	H'00000000	Retained	Retained
Receive missed-frame counter register	RMFCR0	H'00000000	H'00000000	Retained	Retained
Transmit FIFO threshold register	TFTR0	H'00000000	H'00000000	Retained	Retained
FIFO depth register	FDR0	H'00000000	H'00000000	Retained	Retained
Receiving method control register	RMCR0	H'00000000	H'00000000	Retained	Retained
Receive descriptor fetch address register	RDFAR0	H'00000000	H'00000000	Retained	Retained
Receive descriptor finished address register	RDFXR0	H'00000000	H'00000000	Retained	Retained
Receive descriptor final flag register	RDFFR0	H'00000000	H'00000000	Retained	Retained
Transmit descriptor fetch address register	TDFAR0	H'00000000	H'00000000	Retained	Retained
Transmit descriptor finished address register	TDFXR0	H'00000000	H'00000000	Retained	Retained
Transmit descriptor final flag register	TDFFR0	H'00000000	H'00000000	Retained	Retained
Overflow alert FIFO threshold register	FCFTR0	H'001F00FF	H'001F00FF	Retained	Retained
Receive data padding insert register	RPADIR0	H'00000000	H'00000000	Retained	Retained
E-DMAC start register	EDSR1	H'00000000	H'00000000	Retained	Retained
E-DMAC mode register	EDMR1	H'00000000	H'00000000	Retained	Retained

Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
E-DMAC transmit request register	EDTRR1	H'00000000	H'00000000	Retained	Retained
E-DMAC receive request register	EDRRR1	H'00000000	H'00000000	Retained	Retained
Transmit descriptor list start address register	TDLAR1	H'00000000	H'00000000	Retained	Retained
Receive descriptor list start address register	RDLAR1	H'00000000	H'00000000	Retained	Retained
E-MAC/E-DMAC status register	EESR1	H'00000000	H'00000000	Retained	Retained
E-MAC/E-DMAC status interrupt permission register	EESIPR1	H'00000000	H'00000000	Retained	Retained
Transmit/receive status copy enable register	TRSCER1	H'00000000	H'00000000	Retained	Retained
Receive missed-frame counter register	RMFCR1	H'00000000	H'00000000	Retained	Retained
Transmit FIFO threshold register	TFTR1	H'00000000	H'00000000	Retained	Retained
FIFO depth register	FDR1	H'00000000	H'00000000	Retained	Retained
Receiving method control register	RMCR1	H'00000000	H'00000000	Retained	Retained
Receive descriptor fetch address register	RDFAR1	H'00000000	H'00000000	Retained	Retained
Receive descriptor finished address register	RDFXR1	H'00000000	H'00000000	Retained	Retained
Receive descriptor final flag register	RDFFR1	H'00000000	H'00000000	Retained	Retained
Transmit descriptor fetch address register	TDFAR1	H'00000000	H'00000000	Retained	Retained
Transmit descriptor finished address register	TDFXR1	H'00000000	H'00000000	Retained	Retained
Transmit descriptor final flag register	TDFFR1	H'00000000	H'00000000	Retained	Retained
Overflow alert FIFO threshold register	FCFTR1	H'001F00FF	H'001F00FF	Retained	Retained
Receive data padding insert register	RPADIR1	H'00000000	H'00000000	Retained	Retained

23.3.1 Software Reset Register (ARSTR)

ARSTR resets all blocks (E-MAC, TSU, and E-DMAC) in the GETHER. By writing 1 to the ARST bit in this register, a software reset is issued to all blocks of the GETHER (for 256 cycles of external bus clock Bck). The ARST bit is always read as 0. While a software reset is issued, register access to all blocks of the GETHER is prohibited.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ARST	0	R/W	Software Reset When 1 is written to this bit, a software reset is issued to all blocks of the GETHER (for 256 cycles of external bus clock Bck). Writing 0 does not affect this bit. This bit is always read as 0. While a software reset is issued, register access to all blocks of the GETHER is prohibited. The following registers are not initialized by a software reset. TSU_ADRH0 to TSU_ADRH31, TSU_ADRL0 to TSU_ADRL31, TXNLCR0, TXNLCR1, TXALCR0, TXALCR1, RXNLCR0, RXNLCR1, RXALCR0, RXALCR1, FWNLCR0, FWNLCR1, FWALCR0, FWALCR1 When relay operations from the E-MAC-1 to E-MAC-0 or from the E-MAC-0 to E-MAC-1 are enabled, a reset must be issued using this bit. A software reset issued by the SWRT and SWRR bits in EDMR does not reset the transfer switching unit (TSU) performing data transfer between the E-MAC-1 and E-MAC-0.

23.3.2 E-MAC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register that specifies the operating mode of the GETHER. 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 E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits in EDMR before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MCT	—	—	—	MPDE	—	—	RE	TE	—	ILB	—	DM	PRM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TRCCM	0	R/W	Counter Clear Mode Sets the method for clearing the counter register. Refer to the description of each register. 0: Cleared to 0 by writing H'11111111 to the relevant register 1: Cleared to 0 when the relevant register is read
25, 24	—	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
23	RCSC	0	R/W	Checksum Calculation Specifies whether to perform automatic calculation (hardware calculation) of the checksum of the receive frame data unit. 0: Checksum is not automatically calculated 1: Checksum is automatically calculated Note that the checksum calculation of a frame with a VLAN tag is not supported.
22	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
21	DPAD	0	R/W	Data Padding 0: Padding is inserted to data less than 60 bytes so it is transmitted as 60-byte data 1: Padding is not inserted to data less than 60 bytes and it is transmitted without changes
20	RZPF	0	R/W	PAUSE Frame Reception with TIME = 0 0: Reception of a PAUSE frame whose TIME parameter value is 0 is disabled 1: Reception of a PAUSE frame whose TIME parameter value is 0 is enabled

Bit	Bit Name	Initial Value	R/W	Description
19	ZPF	0	R/W	PAUSE Frame Usage with TIME = 0 Enable/Lost Carrier Error Detection Enable. PAUSE Frame Usage with TIME = 0 Enable (In full-duplex mode) 0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled. The next frame is not transmitted until the time specified by the Timer value has elapsed. If a PAUSE frame whose time specified by the Timer value is 0 is received, that PAUSE frame is discarded. 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled. When the data size in the receive FIFO becomes smaller than the FCFTR setting before the time specified by the Timer value elapses, an automatic PAUSE frame with a Timer value of 0 is transmitted. On receiving a PAUSE frame with a Timer value of 0, the transmission wait state is canceled. Lost carrier Error Detection Enable (In half-duplex mode) 0: A lost carrier error is checked during frame transmission. 1: A lost carrier error is not checked during frame transmission Lost carrier error detection can be enabled only when the time period from the EX_TX_EN signal activation (high-active) to the ET_CRS = 1 detection is 63BT* or less. If the time period from the EX_TX_EN signal activation (high-active) to the ET_CRS = 1 detection is greater than 63BT*, or if the ET_CRS signal timing is undefined, this bit should not be cleared to 0. Note*: 1BT = 1ns (1000Mbps), 1BT = 10ns (100bps), 1BT = 100nS (10Mbps)
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to E-DMAC 1: PAUSE frame is transferred to E-DMAC

Bit	Bit Name	Initial Value	R/W	Description
17	RXF	0	R/W	Operating Mode for Receiving Port Flow Control 0: PAUSE frame detection is disabled 1: Flow control for the receiving port is enabled
16	TXF	0	R/W	Operating Mode for Transmitting Port Flow Control 0: Flow control for the transmitting port is disabled (Automatic PAUSE frame is not transmitted) 1: Flow control for the transmitting port is enabled (Automatic PAUSE frame is transmitted as required)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	MCT	0	R/W	Multicast Address Frame Receive Mode 0: Frames other than the multicast address set by the CAM entry table 0 to 31 (H/L) registers are received. However, if the on-chip CAM entry table reference is disabled, all multicast address frames are received. 1: Only the multicast address set by the CAM entry table 0 to 31 (H/L) registers is received.
12 to 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.

Bit	Bit Name	Initial Value	R/W	Description
6	RE	0	R/W	Reception Enable If a switch is made from receiving function enabled (RE = 1) to disabled (RE = 0) while a frame is being received, the receiving function will be enabled until reception of the corresponding frame is completed. 0: Receiving function is disabled 1: Receiving function is enabled
5	TE	0	R/W	Transmission Enable If a switch is made from transmitting function enabled (TE = 1) to disabled (TE = 0) while a frame is being transmitted, the transmitting function will be enabled until transmission of the corresponding frame is completed. 0: Transmitting function is disabled 1: Transmitting 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 GETHER. 0: Normal data transmission/reception is performed 1: Data loopback is performed inside the E-MAC in the GETHER when DM = 1
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	DM	0	R/W	Duplex Mode Specifies the GETHER 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: GETHER performs normal operation 1: GETHER performs promiscuous mode operation

Note: All bits, except for TE and RE, should be changed while the transmitting function is disabled (TE = 0) and the receiving function is disabled (RE = 0).

23.3.3 E-MAC Status Register (ECSR)

ECSR is a 32-bit readable/writable register that indicates the status in the E-MAC. This status can be notified to the CPU by interrupts. When 1 is written to the PFROI, LCHNG, MPD, and ICD bits, the corresponding flags can be cleared. Writing 0 does not affect the flag. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in ECSIPR. Writing 1 or 0 to the PHYI bit does not change its value.

The interrupts generated due to this status register are indicated in each ECI bit in EESR of the E-DMAC0 for port 0 and the E-DMAC1 for port 1.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PFROI	PHYI	LCHNG	MPD	ICD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PFROI	0	R/W	PAUSE Frame Retransmit Retry Over Indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper-limit set in the automatic PAUSE frame retransmit count register (TPAUSER). 0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit

Bit	Bit Name	Initial Value	R/W	Description
3	PHYI	0	R	ET_PHY-INT Interrupt Indicates the state of the ET_PHY-INT pin input from the PHY-LSI. 0: ET_PHY-INT pin is not asserted 1: ET_PHY-INT pin is asserted The signal polarity of the ET_PHY-INT pin can be set by PIPR.
2	LCHNG	0	R/W	Link Signal Change Indicates that the ET_LNKSTA signal input from the PHY-LSI has changed from high to low or low to high. However, signal changes may be detected at the timing at which the ET_LNKSTA function was selected using PACR of the GPIO. To check the current Link state, refer to the LMON bit in the PHY status register (PSR). 0: Change in the ET_LNKSTA signal has not been detected 1: Change in the ET_LNKSTA signal has been detected (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detected on the line. 0: Magic Packet has not been detected 1: Magic Packet has been detected
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used. 0: PHY-LSI has not detected an illegal carrier on the line 1: PHY-LSI has detected an illegal carrier on the line

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PFROIP	PHYIP	LCHNGIP	MPDIP	ICDIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PFROIP	0	R/W	PAUSE Frame Retransmit Interrupt Enable 0: Interrupt notification by the PFROI bit is disabled 1: Interrupt notification by the PFROI bit is enabled
3	PHYIP	0	R/W	ET_PHY-INT Pin Interrupt Enable 0: Interrupt notification by the PHYI bit is disabled 1: Interrupt notification by the PHYI bit is enabled
2	LCHNGIP	0	R/W	LINK Signal Change 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 Detect 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 Detect Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

23.3.5 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY-LSI internal registers via the GMII/MII/RMII.

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

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

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	GMII/MII/RMII Management Data-In Indicates the level of the ET_MDIO pin.
2	MDO	0	R/W	GMII/MII/RMII Management Data-Out Outputs the value set in this bit from the ET_MDIO pin when the MMD bit is 1.
1	MMD	0	R/W	GMII/MII/RMII Management Mode Specifies the data read/write direction with respect to the GMII/MII/RMII. 0: Read direction is specified 1: Write direction is specified
0	MDC	0	R/W	GMII/MII/RMII Management Data Clock Outputs the value set in this bit from the ET_MDC pin and supplies the GMII/MII/RMII with the management data clock. For the method of accessing the GMII/MII/RMII registers, see section 23.5.4, Accessing MII Registers.

23.3.6 MAC Address High Register (MAHR)

MAHR is a 32-bit readable/writable register that specifies the upper 32 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits in EDMR before making settings again.

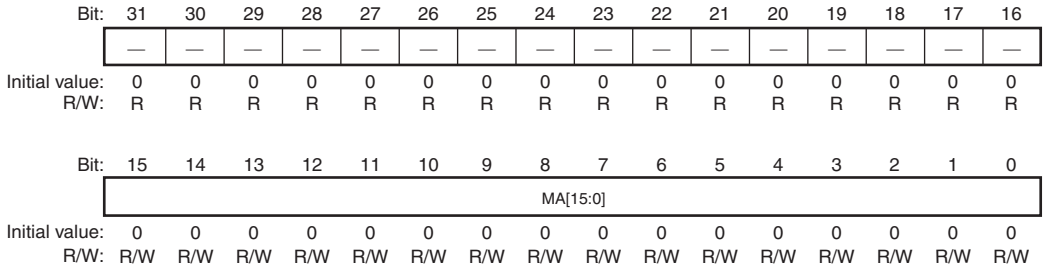
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MA[47:16]	All 0	R/W	MAC Address Bits 47 to 16 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), set H'01234567 in this register.

23.3.7 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Return the E-MAC and E-DMAC to their initial states by means of the SWRT and SWRR bits 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	MA[15:0]	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), set H'000089AB in this register.

23.3.8 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register that 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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 to 0	RFL[17:0]	All 0	R/W	Receive Frame Length The frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data is not included in the transfer. When data that exceeds the specified value is received, the part of data that exceeds the specified value is discarded. H'00000 to H'005EE: 1,518 bytes H'005EF: 1,519 bytes H'005F0: 1,520 bytes : : H'007FF: 2,047 bytes H'00800: 2,048 bytes : : H'01000: 4,096 bytes : : H'10000: 65,536 bytes : : H'20000 to H'3FFFF: 131,072 bytes

23.3.9 PHY Status Register (PSR)

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LMON	0	R	ET_LNKSTA Pin Status The Link status can be read by connecting the Link signal output from the PHY-LSI to the ET_LNKSTA pin. For the polarity, refer to the specifications of the PHY-LSI to be connected.

23.3.10 PHY_INT Polarity Register (PIPR)

PIPR is used to set the polarity of the ET_PHY-INT pin.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PHYIP	0	R/W	ET_PHY-INT Input Pin Polarity 0: ET_PHY-INT pin is low-active (enters the interrupt state at low) 1: ET_PHY-INT pin is high-active (enters the interrupt state at high) For the polarity, refer to the specifications of the PHY-LSI to be connected.

23.3.11 Transmit Retry Over Counter Register (TROCR)

TROCR is a 16-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, this register is incremented by 1. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TROC[15:0]	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.

23.3.12 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 16-bit counter that indicates the number of all delayed collisions that occurred on the line after the start of data transmission. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	COSDC[15:0]	All 0	R/W	Delayed Collision Detect Count These bits indicate the number of all delayed collisions after the start of data transmission.

23.3.13 Lost Carrier Counter Register (LCCR)

LCCR is a 16-bit counter that indicates the number of times the carrier was lost during data transmission. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	LCC[15:0]	All 0	R/W	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

23.3.14 CRC Error Frame Receive Counter Register (CEFCR)

CEFCR is a 16-bit counter that indicates the number of times a frame with a CRC error was received. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

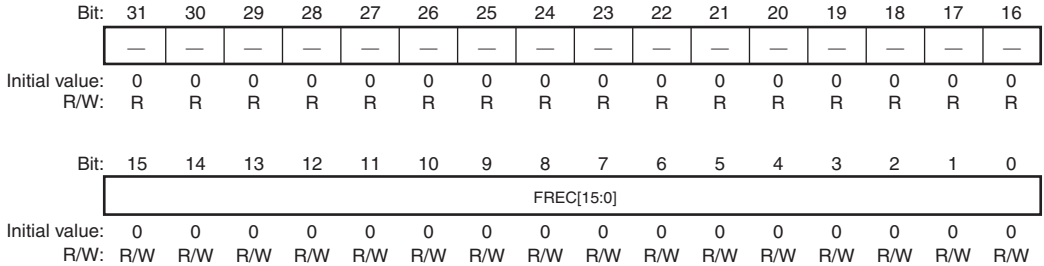
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	CEFC[15:0]	All 0	R/W	CRC Error Frame Count These bits indicate the number of CRC error frames received.

23.3.15 Frame Receive Error Counter Register (FRECR)

FRECR is a 16-bit counter that indicates the number of frames for which a receive error was generated by the ET_RX-ER pin input from the PHY-LSI. FRECR is incremented each time the ET_RX-ER pin becomes active. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.



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	FREC[15:0]	All 0	R/W	Frame Receive Error Count These bits indicate the number of errors during frame reception.

23.3.16 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 16-bit counter that indicates the number of frames received with a length fewer than 64 bytes. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TSFC[15:0]	All 0	R/W	Too-Short Frame Receive Count These bits indicate the number of frames received with a length of less than 64 bytes.

23.3.17 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 16-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'0000FFFF, count-up is halted. This register 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 receive counter register (RFCR). This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TLFC[15:0]	All 0	R/W	Too-Long Frame Receive Count These bits indicate the number of frames received with a length exceeding the value in RFLR.

23.3.18 Residual-Bit Frame Receive Counter Register (RFCR)

RFCR is a 16-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'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	RFC[15:0]	All 0	R/W	Residual-Bit Frame Receive Count These bits indicate the number of frames received containing residual bits.

23.3.19 Carrier Extension Loss Counter Register (CERCR)

CERCR is a 16-bit counter that indicates the number of frames received with the carrier extension lost. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	CERC[15:0]	All 0	R/W	Carrier Extension Loss Frame Receive Count These bits indicate the number of frames received with the carrier extension lost.

23.3.20 Carrier Extension Error Counter Register (CEECR)

CEECR is a 16-bit counter that indicates the number of frames received with an illegal carrier extension. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	CEEC[15:0]	All 0	R/W	Carrier Extension Error Count These bits indicate the number of frames received with an illegal carrier extension.

23.3.21 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 16-bit counter that indicates the number of frames received with a specified multicast address. When the value in this register reaches H'0000FFFF, count-up is halted. This register is cleared to 0 when it is read with the TRCCM bit in ECMR set to 1. When the TRCCM bit in ECMR is 0, this register is cleared to 0 by writing H'11111111.

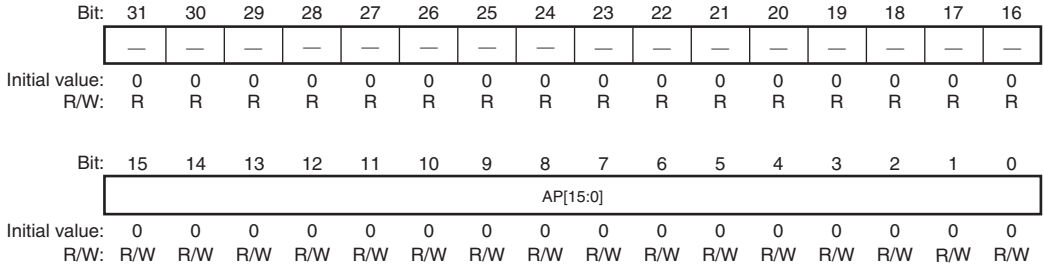
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MAFC[15:0]	All 0	R/W	Multicast Address Frame Count These bits indicate the number of multicast frames received.

23.3.22 Automatic PAUSE Frame Register (APR)

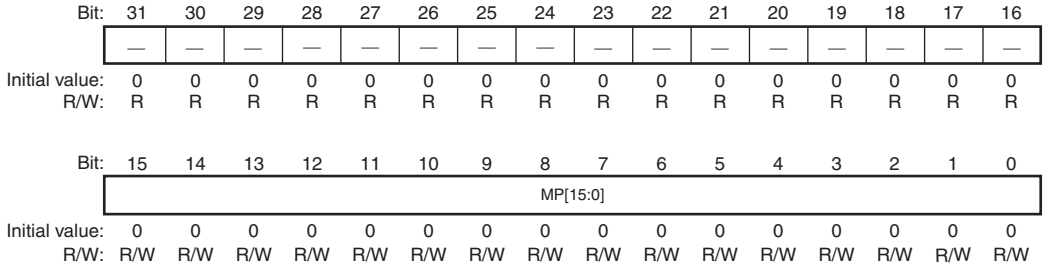
APR is used to set the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.



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	AP[15:0]	All 0	R/W	Automatic PAUSE These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512 bit-time. When flow control is enabled in transmission (PAUSE frame transmission) (TXF bit in ECMR = 1), set a value other than H'0000 in these bits. H'0000: — H'0001: 512 × 1 bit-time H'0002: 512 × 2 bit-time : : H'FFFF: 512 × 65,535 bit-time Note: The bit-time becomes as follows according to the transfer speed. 1000 Mbps: 1 bit-time = 1 ns 100 Mbps: 1 bit-time = 10 ns 10 Mbps: 1 bit-time = 100 ns

23.3.23 Manual PAUSE Frame Register (MPR)

MPR is used to set the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, 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	MP[15:0]	All 0	R/W	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512 bit-time. H'0000: — H'0001: 512 × 1 bit-time H'0002: 512 × 2 bit-time : : H'FFFF: 512 × 65,535 bit-time Note: The bit-time becomes as follows according to the transfer speed. 1000 Mbps: 1 bit-time = 1 ns 100 Mbps: 1 bit-time = 10 ns 10 Mbps: 1 bit-time = 100 ns

23.3.24 Automatic PAUSE Frame Retransmit Count Register (TPAUSER)

TPAUSER is used to set the upper limit for the number of times to retransmit an automatic PAUSE frame. The settings in this register must not be changed while the transmitting function is enabled.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE[15:0]	All 0	R/W	Upper Limit for Automatic PAUSE Frame Retransmission H'0000: Retransmit count is unlimited H'0001: Retransmit count is 1 : : H'FFFF: Retransmit count is 65,535

23.3.25 PAUSE Frame Transmit Counter Register (PFTCR)

PFTCR is a 16-bit counter that indicates the number of times a PAUSE frame is transmitted. This register is cleared to 0 when it is read.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	PFTXC[15:0]	All 0	R	PAUSE Frame Transmit Count These bits indicate the total number of automatic PAUSE frames and manual PAUSE frames transmitted.

23.3.26 PAUSE Frame Receive Counter Register (PFRCR)

PFRCR is a 16-bit counter that indicates the number of times a PAUSE frame is received. This register is cleared to 0 when it is read.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	PFRXC[15:0]	All 0	R	PAUSE Frame Receive Count These bits indicate the number of PAUSE frames received when flow control is enabled in reception (RXF bit in ECMR = 1).

23.3.27 GETHER Mode Register (GECMR)

GECMR is used to set the operating mode of the GETHER.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SPEED[1]	0	R/W	Transfer Speed Sets the transfer speed in combination with the SPEED[0] bit. Refer to the SPEED[0] bit.
1	BSE	0	R/W	Burst Transfer Enable 0: Burst transfer is not performed 1: Burst transfer is performed when the transfer speed is 1 Gpbs in half-duplex transfer (DM bit in ECMR = 0).
0	SPEED[0]	0	R/W	Transfer Speed The transfer speed is specified by a combination of the SPEED[1] and SPEED[0] bits. SPEED[1:0] 00: 10-Mbps transfer 01: 1-Gbps transfer 10: 100-Mbps transfer 11: Setting prohibited

23.3.28 Burst Cycle Count Upper-Limit Register (BCULR)

BCULR sets the upper limit for the number of burst cycles.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	BSTLMT[11:0]	All 0	R/W	Burst Cycle Upper-Limit These bits set the upper limit for burst cycles. Burst transfer is finished when the burst timer exceeds the value set in this register. If the burst timer exceeds the value set in this register while a frame is being transferred, burst transfer is continued until transfer of the corresponding frame is completed. H'000 to H'100: Burst cycle count is 256 cycles H'101: Burst cycle count is 257 cycles : : H'FFE: Burst cycle count is 4,094 cycles H'FFF: Burst cycle count is 4,095 cycles Note: 1 cycle = 32 ns

23.3.29 TSU Counter Reset Register (TSU_CTRST)

TSU_CTRST clears the transmit, receive, and relay frame counters to 0.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CTRST	0	R/W	TSU Counter Reset When 1 is written to this bit, the values of registers TXNLCR0/TXNLCR1, TXALCR0/TXALCR1, RXNLCR0/RXNLCR1, RXALCR0/RXALCR1, FWNLCR0/FWNLCR1, and FWALCR0/FWALCR1 are cleared to 0. Writing 0 does not affect this bit. This bit is always read as 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.30 Relay Enable Register (Port 0 to 1) (TSU_FWEN0)

TSU_FWEN0 enables or disables relay operations from the E-MAC-0 to E-MAC-1 (writing to the relay FIFO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FWEN0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	FWEN0	0	R/W	Port 0 to 1 Relay Operation Enable 0: Port 0 to 1 relay is disabled 1: Port 0 to 1 relay is enabled When the value of bits FCM[2:0] in the relay FIFO size select register (TSU_FCM) is set to H'4, setting this bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.31 Relay Enable Register (Port 1 to 0) (TSU_FWEN1)

TSU_FWEN1 enables or disables relay operations from the E-MAC-1 to E-MAC-0 (writing to the relay FIFO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FWEN1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	FWEN1	0	R/W	Port 1 to 0 Relay Operation Enable 0: Port 1 to 0 relay is disabled 1: Port 1 to 0 relay is enabled When the value of bits FCM[2:0] in the relay FIFO size select register (TSU_FCM) is set to H'3, setting this bit to 1 is prohibited.
30 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.32 Relay FIFO Size Select Register (TSU_FCM)

TSU_FCM selects the size of the relay FIFO in the TSU, used for relay operations between the E-MAC-0 and E-MAC-1.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	FCM[2:0]	All 0	R/W	Relay FIFO Size H'0: Port 0 to 1: 3 Kbytes Port 1 to 0: 3 Kbytes H'1: Port 0 to 1: 4 Kbytes Port 1 to 0: 2 Kbytes H'2: Port 0 to 1: 5 Kbytes Port 1 to 0: 1 Kbyte H'3: Port 0 to 1: 6 Kbytes Port 1 to 0: Not used H'4: Port 0 to 1: Not used Port 1 to 0: 6 Kbytes H'5: Port 0 to 1: 1 Kbyte Port 1 to 0: 5 Kbytes H'6: Port 0 to 1: 2 Kbytes Port 1 to 0: 4 Kbytes H'7: Setting prohibited This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1). When data equal to or greater than the specified size of 64 bytes is stored in the relay FIFO, an overflow is detected and the frame being transferred is discarded.

23.3.33 Relay FIFO Overflow Alert Set Register (Port 0) (TSU_BSYSL0)

The TSU has an alert function, which informs the E-MAC-0 and E-MAC-1 that writing to the relay FIFO will be disabled when the data volume written in the relay FIFO during relay operations exceeds a certain threshold. TSU_BSYSL0 sets the threshold of the relay FIFO when the TSU alerts the E-MAC-0 that writing in the relay FIFO will be disabled during relay operations.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BSYSL0[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	BSYSL0[5:0]	111111	R/W	These bits set the threshold of the port 0-to-1 relay FIFO size in 256-byte units when the TSU alerts the E-MAC-0 that writing in the relay FIFO will be disabled during relay operations. H'00: 0 byte H'01: 256 bytes H'02: 512 bytes : : H'29: 12,032 bytes H'30: 12,288 bytes Settings are disabled for H'31 to H'3F. (Alert is not always carried out.) When the data volume written in the relay FIFO exceeds the threshold set in these bits, the TSU alerts the E-MAC-0 that writing in the relay FIFO will be disabled. Thereafter, alerting will be stopped when the data volume written in the relay FIFO becomes 16 bytes smaller than this threshold. When H'00 is set, the TSU always alerts the E-MAC-0 that writing to the relay FIFO will be disabled. When the value set is equal to or higher than the port 0-to-1 relay FIFO size set by bits FCM[2:0] in TSU_FCM, the TSU does not alert the E-MAC-0 that writing in the relay FIFO will be disabled. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1). When the enable bit of relay operations (FWEN0 bit in TSU_FWEN0 or FWEN1 bit in TSU_FWEN1) is cleared to 0, the TSU stops alerting the E-MAC-0 that writing in the relay FIFO will be disabled.

23.3.34 Relay FIFO Overflow Alert Set Register (Port 1) (TSU_BSYSL1)

The TSU has an alert function, which informs the E-MAC-0 and E-MAC-1 that writing to the relay FIFO will be disabled when the data volume written in the relay FIFO during relay operations exceeds a certain threshold. TSU_BSYSL1 sets the threshold of the relay FIFO when the TSU alerts the E-MAC-1 that writing in the relay FIFO will be disabled during relay operations.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BSYSL1[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	BSYSL1[5:0]	All 1	R/W	These bits set the threshold of the port 1-to-0 relay FIFO size in 256-byte units when the TSU alerts the E-MAC-1 that writing in the relay FIFO will be disabled during relay operations. H'00: 0 byte H'01: 256 bytes H'02: 512 bytes : : H'16: 5632 bytes H'17: 5888 bytes Settings are disabled for H'18 to H'3F. (Alert is not always carried out.) When the data volume written in the relay FIFO exceeds the threshold set in these bits, the TSU alerts the E-MAC-1 that writing in the relay FIFO will be disabled. Thereafter, alerting will be stopped when the data volume written in the relay FIFO becomes 16 bytes smaller than this threshold. When H'00 is set, the TSU always alerts the E-MAC-1 that writing to the relay FIFO will be disabled. When the value set is equal to or higher than the port 1-to-0 relay FIFO size set by bits FCM[2:0] in TSU_FCM, the TSU does not alert the E-MAC-1 that writing in the relay FIFO will be disabled. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1). When the enable bit of relay operations (FWEN0 bit in TSU_FWEN0 or FWEN1 bit in TSU_FWEN1) is cleared to 0, the TSU stops alerting the E-MAC-1 that writing in the relay FIFO will be disabled.

23.3.35 Transmit/Relay Priority Control Mode Register (Port 0) (TSU_PRISL0)

TSU_PRISL0 sets the priority control mode when the transmission request from the E-DMAC to E-MAC-0 comes into collision with port 1 to 0 relay operations. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PRIMD0[2:0]			—	—	—	—	PRISL0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 12	PRIMD0[2:0]	All 0	R/W	These bits set the priority control mode of E-MAC-0 transmission and port 1 to 0 relay operations. H'0: Round robin H'1: Transmission priority H'2: Relay priority H'4: Round robin, however switched to relay priority when relay FIFO use amount exceeds the PRISL0[7:0] setting H'5: Transmission priority, however switched to relay priority when relay FIFO use amount exceeds the PRISL0[7:0] setting Others: Setting prohibited
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PRISL0[7:0]	All 0	R/W	These bits set the threshold of the port 1-to-0 relay FIFO size in 64-byte units in the event of switching to relay priority when bits PRIMD0[2:0] are set to H'4 or H'5. H'00: 0 byte H'01: 64 bytes H'02: 128 bytes : : H'5E: 6,016 bytes H'5F: 6,080 bytes Settings are disabled for H'60 to H'FF. When H'00 is set in these bits, relay always takes priority. When the value set is equal to or above the port 1-to-0 relay FIFO size set by bits FCM[2:0] in TSU_FCM, if bits PRIMD0[2:0] are H'4, round robin will always be set. If bits PRIMD0[2:0] are H'5, transmission always takes priority.

23.3.36 Transmit/Relay Priority Control Mode Register (Port 1) (TSU_PRISL1)

TSU_PRISL1 sets the priority control mode when the transmission request from the E-DMAC to E-MAC-1 comes into collision with port 0 to 1 relay operations. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PRIMD1[2:0]			—	—	—	—	PRISL1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 12	PRIMD1[2:0]	All 0	R/W	These bits set the priority control mode of E-MAC-1 transmission and port 0 to 1 relay operations. H'0: Round robin H'1: Transmission priority H'2: Relay priority H'4: Round robin, however switched to relay priority when relay FIFO use amount exceeds the PRISL1[7:0] setting H'5: Transmission priority, however switched to relay priority when relay FIFO use amount exceeds the PRISL1[7:0] setting Others: Setting prohibited
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PRISL1[7:0]	All 0	R/W	These bits set the threshold of the port 0-to-1 relay FIFO size in 64-byte units in the event of switching to relay priority when bits PRIMD1[2:0] are set to H'4 or H'5. H'00: 0 byte H'01: 64 bytes H'02: 128 bytes : : H'5E: 6,016 bytes H'5F: 6,080 bytes Settings are disabled for H'60 to H'FF. When H'00 is set in these bits, relay always takes priority. When the value set is equal to or above the port 0-to-1 relay FIFO size set by bits FCM[2:0] in TSU_FCM, if bits PRIMD1[2:0] are H'4, round robin will always be set. If bits PRIMD1[2:0] are H'5, transmission always takes priority.

23.3.37 Receive/Relay Function Set Register (Port 0 to 1) (TSU_FWSL0)

TSU_FWSL0 sets the processing method (enable or disable relay operation) of each frame in port 0 reception and port 0 to 1 relay operations. For multicast frames and frames whose destinations are other than this LSI, the processing method in relay operations can be determined by referring to the CAM evaluation results. (For details, refer to section 23.4.5, CAM Function.) This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FW50	FW40	FW30	FW20	FW10	—	—	—	RMSA0	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R

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	FW50	0	R/W	Sets the processing method when frames from port 0 are MAC control frames. 0: Frames are not relayed 1: Frames are relayed to port 1
11	FW40	0	R/W	Sets the processing method when frames from port 0 are addressed to this LSI. 0: Frames are not relayed 1: Frames are relayed to port 1
10	FW30	0	R/W	Sets the processing method when frames from port 0 are Broadcast frames. 0: Frames are not relayed 1: Frames are relayed to port 1

Bit	Bit Name	Initial Value	R/W	Description
9	FW20	0	R/W	Sets the processing method when frames from port 0 are multicast frames. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1
8	FW10	0	R/W	Sets the processing method when frames from port 0 are addressed to other than this LSI. 0: CAM hit: Frames are relayed to port 1 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 1
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RMSA0	0	R/W	Sets the processing method when the SA (source address) of a frame received from port 0 is not registered in the entry table. 0: Frame is not received 1: Frame is received However, a frame discarded because of the VLANtag evaluation result is not received. If a frame whose source address is not registered in the entry table has been received, a carrier extension error is issued regardless of whether the frame is received or not.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.38 Receive/Relay Function Set Register (Port 1 to 0) (TSU_FWSL1)

TSU_FWSL1 sets the processing method (enable or disable relay operation) of each frame in port 1 reception and port 1 to 0 relay operations. For multicast frames and frames whose destinations are other than this LSI, the processing method in relay operations can be determined by referring to the CAM evaluation results. (For details, refer to section 23.4.5, CAM Function.) This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FW51	FW41	FW31	FW21	FW11	—	—	—	RMSA1	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R

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	FW51	0	R/W	Sets the processing method when frames from port 1 are MAC control frames. 0: Frames are not relayed 1: Frames are relayed to port 0
11	FW41	0	R/W	Sets the processing method when frames from port 1 are addressed to this LSI. 0: Frames are not relayed 1: Frames are relayed to port 0
10	FW31	0	R/W	Sets the processing method when frames from port 1 are Broadcast frames. 0: Frames are not relayed 1: Frames are relayed to port 0

Bit	Bit Name	Initial Value	R/W	Description
9	FW21	0	R/W	Sets the processing method when frames from port 1 are multicast frames. 0: CAM hit: Frames are relayed to port 0 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 0
8	FW11	0	R/W	Sets the processing method when frames from port 1 are addressed to other than this LSI. 0: CAM hit: Frames are relayed to port 0 CAM mishit: Frames are not relayed 1: CAM hit: Frames are not relayed CAM mishit: Frames are relayed to port 0
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RMSA1	0	R/W	Sets the processing method when the SA (source address) of a frame received from port 1 is not registered in the entry table. 0: Frame is not received 1: Frame is received However, a frame discarded because of the VLANtag evaluation result is not received. If a frame whose source address is not registered in the entry table has been received, a carrier extension error is issued regardless of whether the frame is received or not.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.39 Relay Function Set Register (Common) (TSU_FWSLC)

When the CAM is used, the referred area in the CAM entry table (partially or wholly) can be specified by the TSU_POST1 to TSU_POST4 registers. TSU_FWSLC enables settings by the TSU_POST1 to TSU_POST4 registers. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	POSTENU	0	R/W	Enables the settings of the POST field of CAM entry tables 0 to 15 (settings by the TSU_POST1 and TSU_POST2 registers). 0: Disables the settings of the POST field. (The CAM entry table is referred to only in port 0 reception.) 1: Enables the settings of the POST field. (The CAM entry table reference conditions follow the POST field settings.)
12	POSTENL	0	R/W	Enables the settings of the POST field of CAM entry tables 16 to 31 (settings by the TSU_POST3 and TSU_POST4 registers). 0: Disables the settings of the POST field. (The CAM entry table is referred to only in port 1 reception.) 1: Enables the settings of the POST field. (The CAM entry table reference conditions follow the POST field settings.)

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

23.3.40 Qtag Addition/Deletion Set Register (Port 0 to 1) (TSU_QTAG0)

TSU_QTAG0 sets the functions adding Qtag to the normal Ethernet frames (no Qtag) to convert them into IEEE802.1Q frames (with Qtag) and deleting Qtag from IEEE802.1Q frames (with Qtag) to convert them into normal Ethernet frames (no Qtag) during port 0 to 1 relay operations. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	QTAG0[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	QTAG0[2:0]	All 0	R/W	These bits set Qtag adding and deleting functions during port 0 to 1 relay operations. H'0: Disables Qtag adding and deleting functions H'1: Disables Qtag adding and deleting functions H'2: Deletes Qtag from frames with Qtag H'3: Adds Qtag to frames with no Qtag (Does not add Qtag to MAC control frames) H'4: Disables Qtag adding and deleting functions H'5: Setting prohibited H'6: Setting prohibited H'7: Adds Qtag to frames with no Qtag (Adds Qtag to MAC control frames) This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

23.3.41 Qtag Addition/Deletion Set Register (Port 1 to 0) (TSU_QTAG1)

TSU_QTAG1 sets the functions adding Qtag to the normal Ethernet frames (no Qtag) to convert them into IEEE802.1Q frames (with Qtag) and deleting Qtag from IEEE802.1Q frames (with Qtag) to convert them into normal Ethernet frames (no Qtag) during port 1 to 0 relay operations. This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	QTAG1[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	QTAG1[2:0]	All 0	R/W	These bits set Qtag adding and deleting functions during port 1 to 0 relay operations. H'0: Disables Qtag adding and deleting functions H'1: Disables Qtag adding and deleting functions H'2: Deletes Qtag from frames with Qtag H'3: Adds Qtag to frames with no Qtag (Does not add Qtag to MAC control frames) H'4: Disables Qtag adding and deleting functions H'5: Setting prohibited H'6: Setting prohibited H'7: Adds Qtag to frames with no Qtag (Adds Qtag to MAC control frames) This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).

23.3.42 Relay Status Register (TSU_FWSR)

TSU_FWSR is a 32-bit readable/writable register that indicates the status during relay operations. By setting the relay status interrupt mask register (TSU_FWINMK), this status can be notified to the CPU as an interrupt source. The status bit set to 1 will be cleared to 0 by writing 1 to the corresponding bit. (The status bit retains the value until it is cleared to 0.)

Interrupts generated due to this status register are identified as EINT2. For details on the priority order of interrupts, see section 9.4.6, Interrupt Exception Handling and Priority in section 9, Interrupt Controller (INTC).

If an error other than RBSY1 or RBSY0 occurs during relay operations, the corresponding relay frame is discarded.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	OVF0	RBSY 0	RINT 60	RINT 50	RINT 40	RINT 30	RINT 20	RINT 10
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OVF1	RBSY 1	RINT 61	RINT 51	RINT 41	RINT 31	RINT 21	RINT 11
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	OVF0	0	R/W	Port 0-to-1 Relay FIFO Overflow Detect Set to 1 when the port 0-to-1 relay FIFO overflows.
22	RBSY0	0	R/W	E-MAC-0 Overflow Alert Signal Output Set to 1 when the threshold of TSU_BSYSL0 is valid and exceeded.
21	RINT60	0	R/W	E-MAC-0 Carrier Extension Loss Error Detect Set to 1 when a frame with the carrier extension lost is received in the E-MAC-0.

Bit	Bit Name	Initial Value	R/W	Description
20	RINT50	0	R/W	E-MAC-0 Residual-Bit Frame Receive Set to 1 when a frame containing residual bits (less than an 8-bit unit) is received in the E-MAC-0.
19	RINT40	0	R/W	E-MAC-0 Too-Long Frame Receive Set to 1 when a frame exceeding the value set by RFLR0 is received in the E-MAC-0.
18	RINT30	0	R/W	E-MAC-0 Too-Short Frame Receive Set to 1 when a frame with a length of less than 64 bytes is received in the E-MAC-0.
17	RINT20	0	R/W	E-MAC-0 Frame Receive Error Set to 1 when a receive error is detected on the ET0_RX-ER pin input from the PHY-LSI in the E-MAC-0.
16	RINT10	0	R/W	E-MAC-0 CRC Error Frame Receive Set to 1 when a receive frame results in a CRC error in the E-MAC-0.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	OVF1	0	R/W	Port 1-to-0 Relay FIFO Overflow Detect Set to 1 when the port 1-to-0 relay FIFO overflows.
6	RBSY1	0	R/W	E-MAC-1 Overflow Alert Signal Output Set to 1 when the threshold of TSU_BSYSL1 is valid and exceeded.
5	RINT61	0	R/W	E-MAC-1 Carrier Extension Loss Error Detect Set to 1 when a frame with the carrier extension lost is received in the E-MAC-1.
4	RINT51	0	R/W	E-MAC-1 Residual-Bit Frame Receive Set to 1 when a frame containing residual bits (less than an 8-bit unit) is received in the E-MAC-1.
3	RINT41	0	R/W	E-MAC-1 Too-Long Frame Receive Set to 1 when a frame exceeding the value set by RFLR1 is received in the E-MAC-1.

Bit	Bit Name	Initial Value	R/W	Description
2	RINT31	0	R/W	E-MAC-1 Too-Short Frame Receive Set to 1 when a frame with a length of less than 64 bytes is received in the E-MAC-1.
1	RINT21	0	R/W	E-MAC-1 Frame Receive Error Set to 1 when a receive error is detected on the ET1_RX-ER pin input from the PHY-LSI in the E-MAC-1.
0	RINT11	0	R/W	E-MAC-1 CRC Error Frame Receive Set to 1 when a receive frame results in a CRC error in the E-MAC-1.

23.3.43 Relay Status Interrupt Mask Register (TSU_FWINMK)

TSU_FWINMK is a 32-bit readable/writable register that sets the interrupt mask for status bits in TSU_FWSR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	OVFM0	RBSYM0	RINTM60	RINTM50	RINTM40	RINTM30	RINTM20	RINTM10
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OVFM1	RBSYM1	RINTM61	RINTM51	RINTM41	RINTM31	RINTM21	RINTM11
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

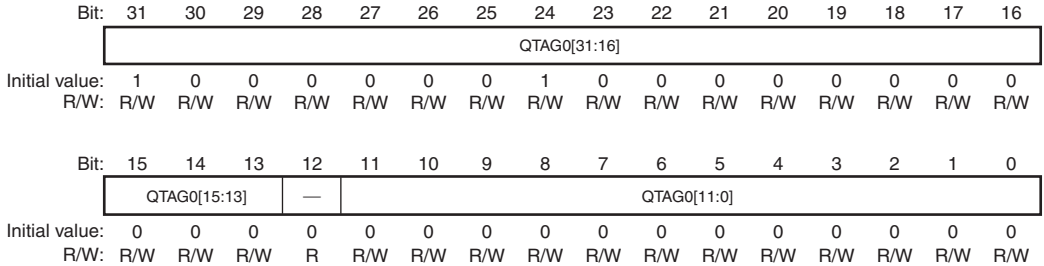
Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	OVFM0	0	R/W	Port 0-to-1 Relay FIFO Overflow Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
22	RBSYM0	0	R/W	E-MAC-0 Overflow Alert Signal Output Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
21	RINTM60	0	R/W	E-MAC-0 Carrier Extension Loss Error Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
20	RINTM50	0	R/W	E-MAC-0 Residual-Bit Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
19	RINTM40	0	R/W	E-MAC-0 Too-Long Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

Bit	Bit Name	Initial Value	R/W	Description
18	RINTM30	0	R/W	E-MAC-0 Too-Short Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
17	RINTM20	0	R/W	E-MAC-0 Frame Receive Error Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
16	RINTM10	0	R/W	E-MAC-0 CRC Error Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	OVFM1	0	R/W	Port 1-to-0 Relay FIFO Overflow Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
6	RBSYM1	0	R/W	E-MAC-1 Overflow Alert Signal Output Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
5	RINTM61	0	R/W	E-MAC-1 Carrier Extension Loss Error Detect Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
4	RINTM51	0	R/W	E-MAC-1 Residual-Bit Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
3	RINTM41	0	R/W	E-MAC-1 Too-Long Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
2	RINTM31	0	R/W	E-MAC-1 Too-Short Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

Bit	Bit Name	Initial Value	R/W	Description
1	RINTM21	0	R/W	E-MAC-1 Frame Receive Error Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled
0	RINTM11	0	R/W	E-MAC-1 CRC Error Frame Receive Interrupt Mask 0: Interrupts disabled 1: Interrupts enabled

23.3.44 Added Qtag Value Set Register (Port 0 to 1) (TSU_ADQT0)

TSU_ADQT0 sets the Qtag data to be added in the conversion of normal Ethernet frames (no Qtag) to IEEE802.1Q frames (with Qtag) in port 0 to 1 relay operations (if bits QTAG0[2:0] in TSU_QTAG0 are set to H'3 or H'7 when using the Qtag adding function). This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).



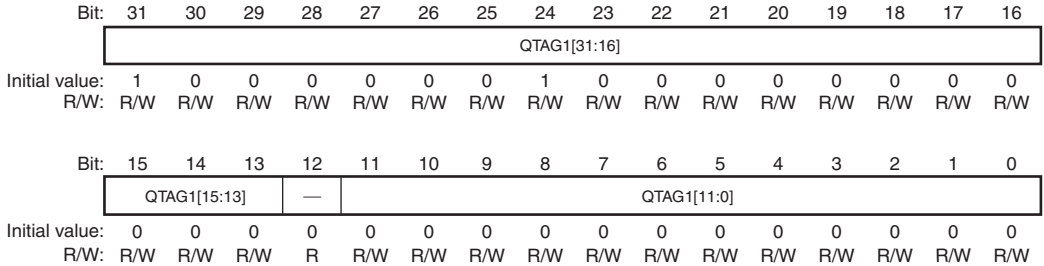
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	QTAG0[31:16]															
Initial value:	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QTAG0[15:13]			—	QTAG0[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	QTAG0[31:16]	H'8100	R/W	Be sure to set the value of the upper 16 bits (QTAG0[31:16]) as H'8100 (indicates the Qtag extension frame format is used). The value read is H'8100.
15 to 13	QTAG0[15:13]	H'0	R/W	Priority Setting (PRT) These bits set the processing priority of frames with Qtag. For details on the settings, refer to the specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	QTAG0[11:0]	H'000	R/W	V-LAN ID Setting (VID) These bits should be set when frames with Qtag are to be used in systems supporting V-LAN. For details on settings, refer to the specifications on Qtag control specified in IEEE802.1Q.

23.3.45 Added Qtag Value Set Register (Port 1 to 0) (TSU_ADQT1)

TSU_ADQT1 sets the Qtag data to be added in the conversion of normal Ethernet frames (no Qtag) to IEEE802.1Q frames (with Qtag) in port 1 to 0 relay operations (if bits QTAG1[2:0] in TSU_QTAG1 are set to H'3 or H'7 when using the Qtag adding function). This register must not be written to once after relay operations have been enabled (after the FWEN0 bit in TSU_FWEN0 or the FWEN1 bit in TSU_FWEN1 is set to 1).



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	QTAG1[31:16]															
Initial value:	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QTAG1[15:13]			—	QTAG1[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	QTAG1[31:16]	H'8100	R/W	Be sure to set the value of the upper 16 bits (QTAG1[31:16]) as H'8100 (indicates the Qtag extension frame format is used). The value read is H'8100.
15 to 13	QTAG1[15:13]	H'0	R/W	Priority Setting (PRT) These bits set the processing priority of frames with Qtag. For details on the settings, refer to the specifications on Qtag control specified in IEEE802.1Q.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	QTAG1[11:0]	H'000	R/W	V-LAN ID Setting (VID) These bits should be set when frames with Qtag are to be used in systems supporting V-LAN. For details on settings, refer to the specifications on Qtag control specified in IEEE802.1Q.

23.3.46 VLANtag Set Register (Port 0) (TSU_VTAG0)

TSU_VTAG0 enables or disables the frame receive/discard evaluation function based on the VLAN number in port 0 relay operations, and also sets the VLAN number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VTAG0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VID0[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	VTAG0	0	R/W	Port 0 VLANtag Evaluation Function 0: Disables receive/discard evaluation for frames based on the VLAN number 1: Enables receive/discard evaluation for frames based on the VLAN number
30 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	VID0[11:0]	All 0	R/W	V-LAN ID Setting (VID) These bits set the VLAN number received by port 0 receive frames.

23.3.47 VLANtag Set Register (Port 1) (TSU_VTAG1)

TSU_VTAG1 enables or disables the frame receive/discard evaluation function based on the VLAN number in port 1 relay operations, and also sets the VLAN number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VTAG1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	VID1[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	VTAG1	0	R/W	Port 1 VLANtag Evaluation Function 0: Disables receive/discard evaluation for frames based on the VLAN number 1: Enables receive/discard evaluation for frames based on the VLAN number
30 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	VID1[11:0]	All 0	R/W	V-LAN ID Setting (VID) These bits set the VLAN number received by port 1 receive frames.

23.3.48 CAM Entry Table Busy Register (TSU_ADSBSY)

When CAM entry table registers (TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31) are set by register writing, the ADSBSY bit in this register is set to 1 (when the process of reflecting the contents of the CAM entry table registers in the CAM controller is completed inside the TSU, the ADSBSY bit is automatically restored to 0).

Access to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while the ADSBSY bit in this register is set to 1. This register is a read-only status register, which must not be written to.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADSBSY	0	R	CAM Entry Table Setting Busy When TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 are set by register writing, this bit is set to 1. When the process of reflecting the contents of the CAM entry table registers in the CAM controller is completed inside the TSU, this bit is automatically restored to 0. Access to TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31 is prohibited, while this bit is set to 1. Writing to this register is also prohibited.

23.3.49 CAM Entry Table Enable Register (TSU_TEN)

TSU_TEN enables or disables the settings of TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEN0	TEN1	TEN2	TEN3	TEN4	TEN5	TEN6	TEN7	TEN8	TEN9	TEN10	TEN11	TEN12	TEN13	TEN14	TEN15
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEN16	TEN17	TEN18	TEN19	TEN20	TEN21	TEN22	TEN23	TEN24	TEN25	TEN26	TEN27	TEN28	TEN29	TEN30	TEN31
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TEN0	0	R/W	CAM Entry Table 0 (TSU_ADRH0 and TSU_ADRL0) Setting 0: Disabled 1: Enabled
30	TEN1	0	R/W	CAM Entry Table 1 (TSU_ADRH1 and TSU_ADRL1) Setting 0: Disabled 1: Enabled
29	TEN2	0	R/W	CAM Entry Table 2 (TSU_ADRH2 and TSU_ADRL2) Setting 0: Disabled 1: Enabled
28	TEN3	0	R/W	CAM Entry Table 3 (TSU_ADRH3 and TSU_ADRL3) Setting 0: Disabled 1: Enabled
27	TEN4	0	R/W	CAM Entry Table 4 (TSU_ADRH4 and TSU_ADRL4) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
26	TEN5	0	R/W	CAM Entry Table 5 (TSU_ADRH5 and TSU_ADRL5) Setting 0: Disabled 1: Enabled
25	TEN6	0	R/W	CAM Entry Table 6 (TSU_ADRH6 and TSU_ADRL6) Setting 0: Disabled 1: Enabled
24	TEN7	0	R/W	CAM Entry Table 7 (TSU_ADRH7 and TSU_ADRL7) Setting 0: Disabled 1: Enabled
23	TEN8	0	R/W	CAM Entry Table 8 (TSU_ADRH8 and TSU_ADRL8) Setting 0: Disabled 1: Enabled
22	TEN9	0	R/W	CAM Entry Table 9 (TSU_ADRH9 and TSU_ADRL9) Setting 0: Disabled 1: Enabled
21	TEN10	0	R/W	CAM Entry Table 10 (TSU_ADRH10 and TSU_ADRL10) Setting 0: Disabled 1: Enabled
20	TEN11	0	R/W	CAM Entry Table 11 (TSU_ADRH11 and TSU_ADRL11) Setting 0: Disabled 1: Enabled
19	TEN12	0	R/W	CAM Entry Table 12 (TSU_ADRH12 and TSU_ADRL12) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
18	TEN13	0	R/W	CAM Entry Table 13 (TSU_ADRH13 and TSU_ADRL13) Setting 0: Disabled 1: Enabled
17	TEN14	0	R/W	CAM Entry Table 14 (TSU_ADRH14 and TSU_ADRL14) Setting 0: Disabled 1: Enabled
16	TEN15	0	R/W	CAM Entry Table 15 (TSU_ADRH15 and TSU_ADRL15) Setting 0: Disabled 1: Enabled
15	TEN16	0	R/W	CAM Entry Table 16 (TSU_ADRH16 and TSU_ADRL16) Setting 0: Disabled 1: Enabled
14	TEN17	0	R/W	CAM Entry Table 17 (TSU_ADRH17 and TSU_ADRL17) Setting 0: Disabled 1: Enabled
13	TEN18	0	R/W	CAM Entry Table 18 (TSU_ADRH18 and TSU_ADRL18) Setting 0: Disabled 1: Enabled
12	TEN19	0	R/W	CAM Entry Table 19 (TSU_ADRH19 and TSU_ADRL19) Setting 0: Disabled 1: Enabled
11	TEN20	0	R/W	CAM Entry Table 20 (TSU_ADRH20 and TSU_ADRL20) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
10	TEN21	0	R/W	CAM Entry Table 21 (TSU_ADRH21 and TSU_ADRL21) Setting 0: Disabled 1: Enabled
9	TEN22	0	R/W	CAM Entry Table 22 (TSU_ADRH22 and TSU_ADRL22) Setting 0: Disabled 1: Enabled
8	TEN23	0	R/W	CAM Entry Table 23 (TSU_ADRH23 and TSU_ADRL23) Setting 0: Disabled 1: Enabled
7	TEN24	0	R/W	CAM Entry Table 24 (TSU_ADRH24 and TSU_ADRL24) Setting 0: Disabled 1: Enabled
6	TEN25	0	R/W	CAM Entry Table 25 (TSU_ADRH25 and TSU_ADRL25) Setting 0: Disabled 1: Enabled
5	TEN26	0	R/W	CAM Entry Table 26 (TSU_ADRH26 and TSU_ADRL26) Setting 0: Disabled 1: Enabled
4	TEN27	0	R/W	CAM Entry Table 27 (TSU_ADRH27 and TSU_ADRL27) Setting 0: Disabled 1: Enabled
3	TEN28	0	R/W	CAM Entry Table 28 (TSU_ADRH28 and TSU_ADRL28) Setting 0: Disabled 1: Enabled

Bit	Bit Name	Initial Value	R/W	Description
2	TEN29	0	R/W	CAM Entry Table 29 (TSU_ADRH29 and TSU_ADRL29) Setting 0: Disabled 1: Enabled
1	TEN30	0	R/W	CAM Entry Table 30 (TSU_ADRH30 and TSU_ADRL30) Setting 0: Disabled 1: Enabled
0	TEN31	0	R/W	CAM Entry Table 31 (TSU_ADRH31 and TSU_ADRL31) Setting 0: Disabled 1: Enabled

23.3.50 CAM Entry Table POST1 Register (TSU_POST1)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST1 specifies the conditions for referring to TSU_ADRH0 to TSU_ADRH7 and TSU_ADRL0 to TSU_ADRL7. The settings of this register are valid when the POSTENU bit in TSU_FWSLC is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST0[3:0]				POST1[3:0]				POST2[3:0]				POST3[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST4[3:0]				POST5[3:0]				POST6[3:0]				POST7[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST0[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 0. By setting multiple bits to 1, multiple conditions can be selected. POST0[3]: CAM entry table 0 is referred to in port 0 reception. POST0[2]: CAM entry table 0 is referred to in port 0 to 1 relay. POST0[1]: CAM entry table 0 is referred to in port 1 reception. POST0[0]: CAM entry table 0 is referred to in port 1 to 0 relay.
27 to 24	POST1[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 1. By setting multiple bits to 1, multiple conditions can be selected. POST1[3]: CAM entry table 1 is referred to in port 0 reception. POST1[2]: CAM entry table 1 is referred to in port 0 to 1 relay. POST1[1]: CAM entry table 1 is referred to in port 1 reception. POST1[0]: CAM entry table 1 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST2[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 2. By setting multiple bits to 1, multiple conditions can be selected. POST2[3]: CAM entry table 2 is referred to in port 0 reception. POST2[2]: CAM entry table 2 is referred to in port 0 to 1 relay. POST2[1]: CAM entry table 2 is referred to in port 1 reception. POST2[0]: CAM entry table 2 is referred to in port 1 to 0 relay.
19 to 16	POST3[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 3. By setting multiple bits to 1, multiple conditions can be selected. POST3[3]: CAM entry table 3 is referred to in port 0 reception. POST3[2]: CAM entry table 3 is referred to in port 0 to 1 relay. POST3[1]: CAM entry table 3 is referred to in port 1 reception. POST3[0]: CAM entry table 3 is referred to in port 1 to 0 relay.
15 to 12	POST4[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 4. By setting multiple bits to 1, multiple conditions can be selected. POST4[3]: CAM entry table 4 is referred to in port 0 reception. POST4[2]: CAM entry table 4 is referred to in port 0 to 1 relay. POST4[1]: CAM entry table 4 is referred to in port 1 reception. POST4[0]: CAM entry table 4 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST5[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 5. By setting multiple bits to 1, multiple conditions can be selected. POST5[3]: CAM entry table 5 is referred to in port 0 reception. POST5[2]: CAM entry table 5 is referred to in port 0 to 1 relay. POST5[1]: CAM entry table 5 is referred to in port 1 reception. POST5[0]: CAM entry table 5 is referred to in port 1 to 0 relay.
7 to 4	POST6[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 6. By setting multiple bits to 1, multiple conditions can be selected. POST6[3]: CAM entry table 6 is referred to in port 0 reception. POST6[2]: CAM entry table 6 is referred to in port 0 to 1 relay. POST6[1]: CAM entry table 6 is referred to in port 1 reception. POST6[0]: CAM entry table 6 is referred to in port 1 to 0 relay.
3 to 0	POST7[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 7. By setting multiple bits to 1, multiple conditions can be selected. POST7[3]: CAM entry table 7 is referred to in port 0 reception. POST7[2]: CAM entry table 7 is referred to in port 0 to 1 relay. POST7[1]: CAM entry table 7 is referred to in port 1 reception. POST7[0]: CAM entry table 7 is referred to in port 1 to 0 relay.

23.3.51 CAM Entry Table POST2 Register (TSU_POST2)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST2 specifies the conditions for referring to TSU_ADRH8 to TSU_ADRH15 and TSU_ADRL8 to TSU_ADRL15. The settings of this register are valid when the POSTENU bit in TSU_FWSLC is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST8[3:0]				POST9[3:0]				POST10[3:0]				POST11[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST12[3:0]				POST13[3:0]				POST14[3:0]				POST15[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST8[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 8. By setting multiple bits to 1, multiple conditions can be selected. POST8[3]: CAM entry table 8 is referred to in port 0 reception. POST8[2]: CAM entry table 8 is referred to in port 0 to 1 relay. POST8[1]: CAM entry table 8 is referred to in port 1 reception. POST8[0]: CAM entry table 8 is referred to in port 1 to 0 relay.
27 to 24	POST9[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 9. By setting multiple bits to 1, multiple conditions can be selected. POST9[3]: CAM entry table 9 is referred to in port 0 reception. POST9[2]: CAM entry table 9 is referred to in port 0 to 1 relay. POST9[1]: CAM entry table 9 is referred to in port 1 reception. POST9[0]: CAM entry table 9 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST10[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 10. By setting multiple bits to 1, multiple conditions can be selected. POST10[3]: CAM entry table 10 is referred to in port 0 reception. POST10[2]: CAM entry table 10 is referred to in port 0 to 1 relay. POST10[1]: CAM entry table 10 is referred to in port 1 reception. POST10[0]: CAM entry table 10 is referred to in port 1 to 0 relay.
19 to 16	POST11[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 11. By setting multiple bits to 1, multiple conditions can be selected. POST11[3]: CAM entry table 11 is referred to in port 0 reception. POST11[2]: CAM entry table 11 is referred to in port 0 to 1 relay. POST11[1]: CAM entry table 11 is referred to in port 1 reception. POST11[0]: CAM entry table 11 is referred to in port 1 to 0 relay.
15 to 12	POST12[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 12. By setting multiple bits to 1, multiple conditions can be selected. POST12[3]: CAM entry table 12 is referred to in port 0 reception. POST12[2]: CAM entry table 12 is referred to in port 0 to 1 relay. POST12[1]: CAM entry table 12 is referred to in port 1 reception. POST12[0]: CAM entry table 12 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST13[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 13. By setting multiple bits to 1, multiple conditions can be selected. POST13[3]: CAM entry table 13 is referred to in port 0 reception. POST13[2]: CAM entry table 13 is referred to in port 0 to 1 relay. POST13[1]: CAM entry table 13 is referred to in port 1 reception. POST13[0]: CAM entry table 13 is referred to in port 1 to 0 relay.
7 to 4	POST14[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 14. By setting multiple bits to 1, multiple conditions can be selected. POST14[3]: CAM entry table 14 is referred to in port 0 reception. POST14[2]: CAM entry table 14 is referred to in port 0 to 1 relay. POST14[1]: CAM entry table 14 is referred to in port 1 reception. POST14[0]: CAM entry table 14 is referred to in port 1 to 0 relay.
3 to 0	POST15[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 15. By setting multiple bits to 1, multiple conditions can be selected. POST15[3]: CAM entry table 15 is referred to in port 0 reception. POST15[2]: CAM entry table 15 is referred to in port 0 to 1 relay. POST15[1]: CAM entry table 15 is referred to in port 1 reception. POST15[0]: CAM entry table 15 is referred to in port 1 to 0 relay.

23.3.52 CAM Entry Table POST3 Register (TSU_POST3)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST3 specifies the conditions for referring to TSU_ADRH16 to TSU_ADRH23 and TSU_ADRL16 to TSU_ADRL23. The settings of this register are valid when the POSTENL bit in TSU_FWSLC is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST16[3:0]				POST17[3:0]				POST18[3:0]				POST19[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST20[3:0]				POST21[3:0]				POST22[3:0]				POST23[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST16[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 16. By setting multiple bits to 1, multiple conditions can be selected. POST16[3]: CAM entry table 16 is referred to in port 0 reception. POST16[2]: CAM entry table 16 is referred to in port 0 to 1 relay. POST16[1]: CAM entry table 16 is referred to in port 1 reception. POST16[0]: CAM entry table 16 is referred to in port 1 to 0 relay.
27 to 24	POST17[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 17. By setting multiple bits to 1, multiple conditions can be selected. POST17[3]: CAM entry table 17 is referred to in port 0 reception. POST17[2]: CAM entry table 17 is referred to in port 0 to 1 relay. POST17[1]: CAM entry table 17 is referred to in port 1 reception. POST17[0]: CAM entry table 17 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST18[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 18. By setting multiple bits to 1, multiple conditions can be selected. POST18[3]: CAM entry table 18 is referred to in port 0 reception. POST18[2]: CAM entry table 18 is referred to in port 0 to 1 relay. POST18[1]: CAM entry table 18 is referred to in port 1 reception. POST18[0]: CAM entry table 18 is referred to in port 1 to 0 relay.
19 to 16	POST19[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 19. By setting multiple bits to 1, multiple conditions can be selected. POST19[3]: CAM entry table 19 is referred to in port 0 reception. POST19[2]: CAM entry table 19 is referred to in port 0 to 1 relay. POST19[1]: CAM entry table 19 is referred to in port 1 reception. POST19[0]: CAM entry table 19 is referred to in port 1 to 0 relay.
15 to 12	POST20[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 20. By setting multiple bits to 1, multiple conditions can be selected. POST20[3]: CAM entry table 20 is referred to in port 0 reception. POST20[2]: CAM entry table 20 is referred to in port 0 to 1 relay. POST20[1]: CAM entry table 20 is referred to in port 1 reception. POST20[0]: CAM entry table 20 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST21[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 21. By setting multiple bits to 1, multiple conditions can be selected. POST21[3]: CAM entry table 21 is referred to in port 0 reception. POST21[2]: CAM entry table 21 is referred to in port 0 to 1 relay. POST21[1]: CAM entry table 21 is referred to in port 1 reception. POST21[0]: CAM entry table 21 is referred to in port 1 to 0 relay.
7 to 4	POST22[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 22. By setting multiple bits to 1, multiple conditions can be selected. POST22[3]: CAM entry table 22 is referred to in port 0 reception. POST22[2]: CAM entry table 22 is referred to in port 0 to 1 relay. POST22[1]: CAM entry table 22 is referred to in port 1 reception. POST22[0]: CAM entry table 22 is referred to in port 1 to 0 relay.
3 to 0	POST23[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 23. By setting multiple bits to 1, multiple conditions can be selected. POST23[3]: CAM entry table 23 is referred to in port 0 reception. POST23[2]: CAM entry table 23 is referred to in port 0 to 1 relay. POST23[1]: CAM entry table 23 is referred to in port 1 reception. POST23[0]: CAM entry table 23 is referred to in port 1 to 0 relay.

23.3.53 CAM Entry Table POST4 Register (TSU_POST4)

When using the CAM, the conditions for referring to each CAM entry table can be specified by using the TSU_POST1 to TSU_POST4 registers. TSU_POST4 specifies the conditions for referring to TSU_ADRH24 to TSU_ADRH31 and TSU_ADRL24 to TSU_ADRL31. The settings of this register are valid when the POSTENL bit in TSU_FWSLC is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POST24[3:0]				POST25[3:0]				POST26[3:0]				POST27[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POST28[3:0]				POST29[3:0]				POST30[3:0]				POST31[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	POST24[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 24. By setting multiple bits to 1, multiple conditions can be selected. POST24[3]: CAM entry table 24 is referred to in port 0 reception. POST24[2]: CAM entry table 24 is referred to in port 0 to 1 relay. POST24[1]: CAM entry table 24 is referred to in port 1 reception. POST24[0]: CAM entry table 24 is referred to in port 1 to 0 relay.
27 to 24	POST25[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 25. By setting multiple bits to 1, multiple conditions can be selected. POST25[3]: CAM entry table 25 is referred to in port 0 reception. POST25[2]: CAM entry table 25 is referred to in port 0 to 1 relay. POST25[1]: CAM entry table 25 is referred to in port 1 reception. POST25[0]: CAM entry table 25 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
23 to 20	POST26[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 26. By setting multiple bits to 1, multiple conditions can be selected. POST26[3]: CAM entry table 26 is referred to in port 0 reception. POST26[2]: CAM entry table 26 is referred to in port 0 to 1 relay. POST26[1]: CAM entry table 26 is referred to in port 1 reception. POST26[0]: CAM entry table 26 is referred to in port 1 to 0 relay.
19 to 16	POST27[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 27. By setting multiple bits to 1, multiple conditions can be selected. POST27[3]: CAM entry table 27 is referred to in port 0 reception. POST27[2]: CAM entry table 27 is referred to in port 0 to 1 relay. POST27[1]: CAM entry table 27 is referred to in port 1 reception. POST27[0]: CAM entry table 27 is referred to in port 1 to 0 relay.
15 to 12	POST28[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 28. By setting multiple bits to 1, multiple conditions can be selected. POST28[3]: CAM entry table 28 is referred to in port 0 reception. POST28[2]: CAM entry table 28 is referred to in port 0 to 1 relay. POST28[1]: CAM entry table 28 is referred to in port 1 reception. POST28[0]: CAM entry table 28 is referred to in port 1 to 0 relay.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	POST29[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 29. By setting multiple bits to 1, multiple conditions can be selected. POST29[3]: CAM entry table 29 is referred to in port 0 reception. POST29[2]: CAM entry table 29 is referred to in port 0 to 1 relay. POST29[1]: CAM entry table 29 is referred to in port 1 reception. POST29[0]: CAM entry table 29 is referred to in port 1 to 0 relay.
7 to 4	POST30[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 30. By setting multiple bits to 1, multiple conditions can be selected. POST30[3]: CAM entry table 30 is referred to in port 0 reception. POST30[2]: CAM entry table 30 is referred to in port 0 to 1 relay. POST30 [1]: CAM entry table 30 is referred to in port 1 reception. POST30[0]: CAM entry table 30 is referred to in port 1 to 0 relay.
3 to 0	POST31[3:0]	All 0	R/W	These bits set the conditions for referring to CAM entry table 31. By setting multiple bits to 1, multiple conditions can be selected. POST31[3]: CAM entry table 31 is referred to in port 0 reception. POST31[2]: CAM entry table 31 is referred to in port 0 to 1 relay. POST31[1]: CAM entry table 31 is referred to in port 1 reception. POST31[0]: CAM entry table 31 is referred to in port 1 to 0 relay.

23.3.54 CAM Entry Table 0H to 31H Registers (TSU_ADRH0 to TSU_ADRH31)

TSU_ADRH0 to TSU_ADRH31 are entry tables referred to by the CAM in reception and relay. Each of these registers sets the upper 32 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADRHn[31:16] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRHn[15:0] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ADRHn[31:0] (n: 0 to 31)	All 0	R/W	MAC Address Bits These bits set the upper 32 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), set H'01234567 in this register.

Note: Set the CAM entry tables following the procedure below.

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC addresses by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC addresses by TSU_ADRL0 to TSU_ADRL31.

23.3.55 CAM Entry Table 0L to 31L Registers (TSU_ADRL0 to TSU_ADRL31)

TSU_ADRL0 to TSU_ADRL31 are entry tables referred to by the CAM in reception and relay. Each of these registers sets the lower 16 bits of the 48-bit MAC address. Maximum 32 entries of MAC addresses can be registered.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRLn[15:0] (n = 0 to 31)															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	ADRLn[15:0] (n: 0 to 31)	All 0	R/W	MAC Address Bits These bits set the lower 16 bits of the MAC address. When the MAC address is 01-23-45-67-89-AB (displayed in hexadecimal), set H'000089AB in this register.

Note: Set the CAM entry tables following the procedure below.

1. Check that the ADSBSY bit in TSU_ADSBSY is cleared to 0.
2. Set the upper 32 bits of the MAC addresses by TSU_ADRH0 to TSU_ADRH31.
3. Set the lower 16 bits of the MAC addresses by TSU_ADRL0 to TSU_ADRL31.

23.3.56 Transmit Frame Counter Register (Port 0) (Normal Transmission Only) (TXNLCR0)

TXNLCR0 is a 32-bit counter indicating the number of frames successfully transmitted in the E-MAC-0. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NTC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NTC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC0[31:0]	All 0	R	Port 0 Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted.

23.3.57 Transmit Frame Counter Register (Port 0) (Normal and Erroneous Transmission) (TXALCR0)

TXALCR0 is a 32-bit counter indicating the number of frames transmitted in the E-MAC-0, including the number of frames erroneously transmitted. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TC0[31:0]	All 0	R	Port 0 Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted and erroneously transmitted.

23.3.58 Receive Frame Counter Register (Port 0) (Normal Reception Only) (RXNLCR0)

RXNLCR0 is a 32-bit counter indicating the number of frames successfully received in the E-MAC-0. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NRC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NRC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC0[31:0]	All 0	R	Port 0 Receive Frame Counter Bits These bits indicate the number of frames successfully received.

23.3.59 Receive Frame Counter Register (Port 0) (Normal and Erroneous Reception) (RXALCR0)

RXALCR0 is a 32-bit counter indicating the number of frames received in the E-MAC-0, including the number of frames erroneously received. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RC0[31:0]	All 0	R	Port 0 Receive Frame Counter Bits These bits indicate the number of frames successfully received and erroneously received.

23.3.60 Relay Frame Counter Register (Port 1 to 0) (Normal Relay Only) (FWNLCR0)

FWNLCR0 is a 32-bit counter indicating the number of frames successfully relayed in port 1 to 0 relay operations. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NFC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NFC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC0[31:0]	All 0	R	Port 1 to 0 Relay Frame Counter Bits These bits indicate the number of frames successfully relayed.

23.3.61 Relay Frame Counter Register (Port 1 to 0) (Normal and Erroneous Transmission) (FWALCR0)

FWALCR0 is a 32-bit counter indicating the number of frames relayed in port 1 to 0 relay operations, including the number of frames erroneously relayed. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FC0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FC0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FC0[31:0]	All 0	R	Port 1 to 0 Relay Frame Counter Bits These bits indicate the number of frames successfully relayed and erroneously relayed.

23.3.62 Transmit Frame Counter Register (Port 1) (Normal Transmission Only) (TXNLCR1)

TXNLCR1 is a 32-bit counter indicating the number of frames successfully transmitted in the E-MAC-1. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NTC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NTC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NTC1[31:0]	All 0	R	Port 1 Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted.

23.3.63 Transmit Frame Counter Register (Port 1) (Normal and Erroneous Transmission) (TXALCR1)

TXALCR1 is a 32-bit counter indicating the number of frames transmitted in the E-MAC-1, including the number of frames erroneously transmitted. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TC1[31:0]	All 0	R	Port 1 Transmit Frame Counter Bits These bits indicate the number of frames successfully transmitted and erroneously transmitted.

23.3.64 Receive Frame Counter Register (Port 1) (Normal Reception Only) (RXNLCR1)

RXNLCR1 is a 32-bit counter indicating the number of frames successfully received in the E-MAC-1. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NRC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NRC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRC1[31:0]	All 0	R	Port 1 Receive Frame Counter Bits These bits indicate the number of frames successfully received.

23.3.65 Receive Frame Counter Register (Port 1) (Normal and Erroneous Reception) (RXALCR1)

RXALCR1 is a 32-bit counter indicating the number of frames received in the E-MAC-1, including the number of frames erroneously received. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RC1[31:0]	All 0	R	Port 1 Receive Frame Counter Bits These bits indicate the number of frames successfully received and erroneously received.

23.3.66 Relay Frame Counter Register (Port 0 to 1) (Normal Relay Only) (FWNLCR1)

FWNLCR1 is a 32-bit counter indicating the number of frames successfully relayed in port 0 to 1 relay operations. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NFC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NFC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NFC1[31:0]	All 0	R	Port 0 to 1 Relay Frame Counter Bits These bits indicate the number of frames successfully relayed.

23.3.67 Relay Frame Counter Register (Port 0 to 1) (Normal and Erroneous Transmission) (FWALCR1)

FWALCR1 is a 32-bit counter indicating the number of frames relayed in port 0 to 1 relay operations, including the number of frames erroneously relayed. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter is cleared to 0 by reading from this register. This register cannot be written to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FC1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FC1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FC1[31:0]	All 0	R	Port 0 to 1 Relay Frame Counter Bits These bits indicate the number of frames successfully relayed and erroneously relayed.

23.3.68 E-DMAC Start Register (EDSR)

EDSR specifies activation of the transmitting unit and receiving unit of the E-DMAC. This register can only be written to, and the read values are invalid.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	ENT	0	W	E-DMAC Transmitting Unit Start 0: Stops the E-DMAC transmitting unit 1: Starts the E-DMAC transmitting unit
0	ENR	0	W	E-DMAC Receiving Unit Start 0: Stops the E-DMAC receiving unit 1: Starts the E-DMAC receiving unit

23.3.69 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies E-DMAC resetting and the transmit/receive descriptor length. This register is to be set before the transmitting or receiving function is enabled (before the TR bit in EDTRR or the RR bit in EDRRR is set to 1). However, the SWRR and SWRT bits can be written to even after the transmitting or receiving function is enabled. If a software reset is executed with this register during data transmission, abnormal data may be transmitted on the line. Execute a software reset with this register before specifying the transmit/receive descriptor length or modifying the settings of TDLAR, RDLAR, and so forth, the setting of ECMR (E-MAC mode register), and the settings of registers related to the E-DMAC and E-MAC operation.

To execute a software reset with this register, 1 must be written to both the SWRT and SWRR bits simultaneously. Writing 1 to the SWRT and SWRR bits initializes the E-MAC registers and E-DMAC registers, except for TDLAR, RDLAR, and RMFCR of the E-DMAC. The TSU registers (registers whose names are prefixed with TSU_) are not initialized. Writing 1 to the SWRT and SWRR bits in EDMR0 initializes the registers related to the E-DMAC0 and E-MAC-0, whereas, writing 1 to the SWRT and SWRR bits in EDMR1 initializes the registers related to the E-DMAC1 and E-MAC-1. When relay operations are enabled in the TSU by specifying the relay enable register (port 0 to 1) (TSU_FWEN0) and relay enable register (port 1 to 0) (TSU_FWEN1), a software reset should not be performed using this register. Note that during the period a software reset is issued (for 64 cycles of the internal bus clock Bck), accesses to all Ethernet-related registers are prohibited.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]		—	—	SWRT	SWRR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W

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	Transmit/Receive Frame Endian Sets the endian mode for DMA transfer of frame data between the transmit/receive FIFO and transmit/receive buffer. 0: Big endian (longword access) 1: Little endian (longword access)
5, 4	DL[1:0]	00	R/W	Transmit/Receive Descriptor Length These bits specify the descriptor length. (See section 23.4.1, Descriptors and Descriptor List.) 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SWRT	0	R/W	Software Reset of Transmit FIFO Controller [Writing] 0: Disabled 1: Software reset started [Reading] 0: Software reset not executed (or completed) 1: Software reset being executed
0	SWRR	0	R/W	Software Reset of Receive FIFO Controller [Writing] 0: Disabled 1: Software reset started [Reading] 0: Software reset not executed (or completed) 1: Software reset being executed

23.3.70 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. After writing 11 to bits TR[1:0] in this register, the E-DMAC reads the transmit descriptor at the address specified by TDLAR. If the TACT bit of this transmit descriptor is set to 1 (valid), transmit DMA transfer by the E-DMAC starts. When DMA transfer based on the first transmit descriptor is completed, the E-DMAC reads the next transmit descriptor. If the TACT bit of that transmit descriptor is set to 1 (valid), the E-DMAC continues transmit DMA operation. If the TACT bit of a transmit descriptor is cleared to 0 (invalid), the E-DMAC clears bits TR[1:0] and stops transmit DMA operation.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	TR[1:0]	00	R/W	Transmit Request 00, 01, 10: Transmission-halted state If 00, 01, or 10 is written to these bits, the E-DMAC stops DMA transfer of the currently processed transmit descriptor, reads the next transmit descriptor, and then clears these bits. (Write-back is completed for the valid transmit descriptors that have been detected up till then.) The E-DMAC clears these bits when transmit descriptor empty occurs, or transmission of a transmit descriptor has completed. (Write-back is completed for the valid transmit descriptors that have been detected up till then.) 11: Transmit DMA operation by E-DMAC After writing 11 to these bits, the E-DMAC starts reading a transmit descriptor.

23.3.71 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. After writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor at the address specified by RDLAR. If the RACT bit of this receive descriptor is set to 1 (valid), and the receive FIFO holds a receive frame, the E-DMAC starts receive DMA transfer. When DMA transfer based on the first receive descriptor is completed, the E-DMAC reads the next receive descriptor. If the RACT bit of that receive descriptor is set to 1 (valid), the E-DMAC continues receive DMA operation. However, if the receive FIFO holds no receive data, the E-DMAC places receive DMA operation in the standby state. If the RACT bit of the receive descriptor is cleared to 0 (invalid), the E-DMAC clears the RR bit and stops receive DMAC operation.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RR	0	R/W	Receive Request 0: Receiving function is disabled* If 0 is written to this bit, the E-DMAC stops receive operation after DMA transfer of one frame has completed and then clears this bit. The E-DMAC clears this bit when receive descriptor empty occurs. 1: Receive descriptor is read, and the E-DMAC is ready to receive

Note: * If the receiving 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 E-DMAC reception enabled again, execute a software reset by the SWRT and SWRR bits in EDMR0 (EDMR1). To disable the E-DMAC receiving function without executing a software reset, specify the RE bit in ECMR0 (ECMR1). Next, after the E-DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using this register.

23.3.72 Transmit Descriptor List Start 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 bits in EDMR. This register must not be modified during transmission. Modifications to this register should only be made in the transmission-halted state specified by bits TR[1:0] (= 00) in the E-DMAC transmit request register (EDTRR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDLA[31:15]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA[31:0]	All 0	R/W	Transmit Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: TDLA[3:0] = 0000 32-byte boundary: TDLA[4:0] = 00000 64-byte boundary: TDLA[5:0] = 000000

23.3.73 Receive Descriptor List Start 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 bits in EDMR. This register must not be modified 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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDLA[31:15]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDLA[31:0]	All 0	R/W	Receive Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: RDLA[3:0] = 0000 32-byte boundary: RDLA[4:0] = 00000 64-byte boundary: RDLA[5:0] = 000000

23.3.74 E-MAC/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 E-MAC. The information in this register is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a read-only bit that is not 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 E-MAC/E-DMAC status interrupt permission register (EESIPR).

The interrupts generated by this status register are GEINT0 for port 0 and GEINT1 for port 1. For interrupt priorities, see section 9.4.6, Interrupt Exception Handling and Priority in section 9, Interrupt Controller (INTC). GEINT2 is an interrupt generated by TSU_FWSR in the TSU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TWB[1:0]		TC[1]	TUC	ROC	TABT	RABT	RFCOF	—	ECI	TC[0]	TDE	TFUF	FR	RDE	RFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DLC	CD	TRO	RMAF	CEEF	CELF	RRF	RTLF	RTSF	PRE	CERF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	TWB[1:0]	00	R/W	Write-Back Complete Indicates that write-back from the E-DMAC to the corresponding descriptor after frame transmission has completed. This operation is enabled only when the TWBI bit in the transmit descriptor that includes the end of the transmit frame is set to 1. 00: Write-back has not completed, or no transmission directive 11: Write-back has completed Others: Setting disabled

Bit	Bit Name	Initial Value	R/W	Description
29	TC[1]	0	R/W	Frame Transmission Complete Indicates, in combination with the TC[0] bit, that all the data specified by the transmit descriptor has been transmitted from the E-MAC. This bit is set to 1 on assuming the completion of transmission. This is when transmission of one frame is completed and the transmit descriptor valid bit (TACT) of the next transmit descriptor not being set in single-frame/single-descriptor operation or when the last data of a frame has been transmitted and the transmit descriptor valid bit (TACT) of the next descriptor not being set in multi-buffer frame processing based on single-frame/multi-descriptor operation. After frame transmission has completed, the E-DMAC writes the transmission status back to the relevant descriptor. TC[1:0] 00: Transmission has not completed, or no transmission directive 11: Transmission has completed Others: Setting disabled
28	TUC	0	R/W	Transmit Underflow Frame Write-Back Complete 0: Write-back has not completed for the frame causing transmit underflow 1: Write-back has completed for the frame causing transmit underflow
27	ROC	0	R/W	Receive Overflow Frame Write-Back Complete 0: Write-back has not completed for the frame causing receive overflow 1: Write-back has completed for the frame causing receive overflow
26	TABT	0	R/W	Transmit Abort Detect Indicates that the E-MAC aborts transmitting a frame because of failures during frame transmission. 0: Frame transmission has not been aborted or no transmission directive 1: Frame transmission has been aborted

Bit	Bit Name	Initial Value	R/W	Description
25	RABT	0	R/W	Receive Abort Detect Indicates that the E-MAC aborts receiving a frame because of failures during frame reception. 0: Frame reception has not been aborted or no reception directive 1: Frame reception has been aborted
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the frame counter in the receive FIFO has overflowed. 0: Receive frame counter has not overflowed 1: Receive frame counter has overflowed
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22	ECI	0	R	E-MAC Status Register Source This bit is a read-only bit. When the source of an ECSR interrupt is cleared, this bit is also cleared. 0: E-MAC status interrupt source has not been detected 1: E-MAC status interrupt source has been detected
21	TC[0]	0	R/W	Frame Transmission Complete Indicates, in combination with the TC[1] bit, that all the data specified by the transmit descriptor has been transmitted from the E-MAC. For details, see the description of the TC[1] bit.

Bit	Bit Name	Initial Value	R/W	Description
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmit descriptor valid bit (TACT) of a transmit descriptor read by the E-DMAC is not set if the previous descriptor does not represent the end of a frame in multi-buffer frame processing based on single-frame/multi-descriptor operation. As a result, an incomplete frame may be sent. 0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected When transmit descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, transmission starts from the address that is stored in the transmit descriptor list start address register (TDLAR).
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that an underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred
18	FR	0	R/W	Frame Reception Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received. 0: Frame has not been received 1: Frame has been received
17	RDE	0	R/W	Receive Descriptor Empty Indicates that the RACT bit of a receive descriptor read by the E-DMAC for receive DMA operation is cleared to 0 (invalid). When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RACT bit (cleared to 0) of the receive descriptor to 1 and then writing 1 to the RR bit in EDRRR. 0: Receive descriptor active bit RACT = 1 detected 1: Receive descriptor active bit RACT = 0 detected

Bit	Bit Name	Initial Value	R/W	Description
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 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
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 has not been detected 1: Loss of carrier has been detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been detected during frame transmission. 0: Delayed collision has not been detected 1: Delayed collision has been detected
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 have failed after the E-MAC 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	CEEF	0	R/W	Carrier Extension Error Indicates that a carrier extension error has occurred during frame reception in 1-Gigabit/half-duplex transfer. 0: Carrier extension error has not occurred 1: Carrier extension error has occurred

Bit	Bit Name	Initial Value	R/W	Description
5	CELF	0	R/W	Carrier Extension Loss Indicates that the carrier extension has been lost in 1-Gigabit/half-duplex transfer. This means that the sum of a frame and carrier extension is smaller than SLOT_TIME (4096 bits). 0: Carrier extension loss has not occurred 1: Carrier extension loss has occurred
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 a frame whose byte size exceeds the upper limit for the receive frame length set by RFLR has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error has not been detected 1: CRC error has been detected

23.3.75 E-MAC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the E-MAC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TWB1IP	TWB0IP	TC1IP	TUCIP	ROCIP	TABTIP	RABTIP	RFCOFIP	—	ECIIP	TC0IP	TDEIP	TFUFIP	FRIP	RDEIP	RFEIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DLCIP	CDIP	TROIP	RMAFIP	CEEFIP	CELFIP	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TWB1IP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
30	TWB0IP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29	TC1IP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
28	TUCIP	0	R/W	Transmit Underflow Frame Write-Back Complete Interrupt Enable 0: Transmit underflow frame write-back complete interrupt is disabled 1: Transmit underflow frame write-back complete interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
27	ROCIP	0	R/W	Receive Overflow Frame Write-Back Complete Interrupt Enable 0: Receive overflow frame write-back complete interrupt is disabled 1: Receive overflow frame write-back complete interrupt is enabled
26	TABTIP	0	R/W	Transmit Abort Detect Interrupt Enable 0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detect Interrupt Enable 0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Enable 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
22	ECIIP	0	R/W	E-MAC Status Register Source Interrupt Enable 0: E-MAC status interrupt is disabled 1: E-MAC status interrupt is enabled
21	TC0IP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Enable 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Reception Interrupt Enable 0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Overflow interrupt is disabled 1: Overflow interrupt is enabled
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Enable 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Enable 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Enable 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Enable 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6	CEEFIP	0	R/W	Carrier Extension Error Interrupt Enable 0: Carrier extension error interrupt is disabled 1: Carrier extension error interrupt is enabled
5	CELFIP	0	R/W	Carrier Extension Loss Interrupt Enable 0: Carrier extension loss interrupt is disabled 1: Carrier extension loss interrupt is enabled
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Enable 0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Enable 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Enable 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Enable 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame Interrupt Enable 0: CRC error interrupt is disabled 1: CRC error interrupt is enabled

23.3.76 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether the information for the transmit and receive state reported by bits 17, 16, and 10 to 0 in the E-MAC/E-DMAC status register (EESR) is to be reflected in the TFE or RFE bit of the corresponding descriptor. The bits in this register correspond to bits 17, 16, and 10 to 0 in EESR. When a bit is cleared to 0, the transmit status (bits 17 and 10 to 8 in EESR) is reflected in the TFE bit of the transmit descriptor, and the receive status (bits 16 and 7 to 0 in EESR) is reflected in the RFE bit of the receive descriptor. In this case, the state of a status bit set to 1 is reflected as the TFE or RFE bit set to 1. When a bit is set to 1, the occurrence of the corresponding source is not reflected in the descriptor. After this LSI is reset, all bits are cleared to 0.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DLCCE	CDCE	TROCE	RMAFCE	CEEFCE	CELFCE	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

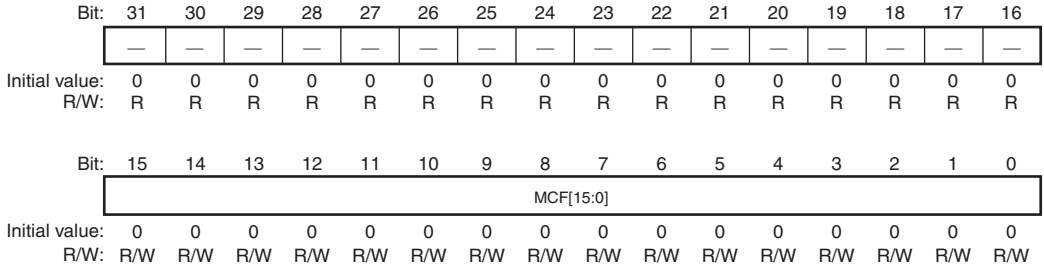
Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	TABTCE	0	R/W	TABT Bit Copy Directive 0: Reflects the TABT bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
16	RABTCE	0	R/W	RABT Bit Copy Directive 0: Reflects the RABT bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

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	DLCCE	0	R/W	DLC Bit Copy Directive 0: Reflects the DLC bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
9	CDCE	0	R/W	CD Bit Copy Directive 0: Reflects the CD bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Reflects the TRO bit status in the TFE bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFE bit of the transmit descriptor
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
6	CEEFCE	0	R/W	CEEF Bit Copy Directive 0: Reflects the CEEF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
5	CELFCE	0	R/W	CELF Bit Copy Directive 0: Reflects the CELF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

Bit	Bit Name	Initial Value	R/W	Description
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Reflects the RRF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Reflects the RTLF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Reflects the RTSF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Reflects the PRE bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Reflects the CERF bit status in the RFE bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFE bit of the receive descriptor

23.3.77 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames that could not be saved in the receive buffer and so were discarded 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, count-up is halted. Clear the counter by writing H'0000 in this register.



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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MFC[15:0]	All 0	R/W	Missed-Frame Counter These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.

23.3.78 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 E-MAC 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 one frame of data write is performed. This register must not be written to during transmission (bits TR[1:0] in EDTRR = 11).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TFT[10:0]	All 0	R/W	Transmit FIFO Threshold A value in 32-byte units and smaller than the FIFO size specified by FDR must be set as the transmit FIFO threshold. H'000: Store and forward modes H'008: 32 bytes H'010: 64 bytes H'018: 128 bytes : : H'07F: 508 bytes H'080: 512 bytes : : H'0FF: 1,020 bytes H'100: 1,024 bytes : : H'1FF: 2,044 bytes H'200: 2,048 bytes

Note: When starting transmission before one frame of data write has completed, take care no underflow occurs.

23.3.79 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the sizes of the transmit and receive FIFOs.

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

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

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	TFD[2:0]	All 1	R/W	Transmit FIFO Size Specifies 256 bytes to 2 Kbytes in 256-byte units as the size of the transmit FIFO whose maximum size is 2 Kbytes. The setting must not be changed after transmission/reception has started. H'00 : 256 bytes H'01 : 512 bytes : : H'07 : 2048 bytes
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 to 0	RFD[4:0]	All 1	R/W	Receive FIFO Size Specifies 256 bytes to 8 Kbytes in 256-byte units as the size of the receive FIFO whose maximum size is 8 Kbytes. The setting must not be changed after transmission/reception has started. H'00 : 256 bytes H'01 : 512 bytes : : H'1F : 8192 bytes

23.3.80 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RE bit in ECMR while a frame is received. This register must be set during the receiving-halted state.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	RNC	0	R/W	Receive Enable Control Sets whether to continue frame reception. 0: Upon completion of reception of one frame, the E-DMAC writes the receive status to the descriptor and clears the RR bit in EDRRR to 0. 1: Upon completion of reception of one frame, the E-DMAC writes (writes back) the receive status to the descriptor. In addition, the E-DMAC reads the next descriptor and prepares for reception of the next frame.

23.3.81 Receive Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the receive descriptor. Which receive 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. In the initial setting, set the address of the receive descriptor at which receive processing is to be started.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFA[31:0]	All 0	R/W	Receive Descriptor Fetch Address Writing to these bits during the reception is prohibited.

23.3.82 Receive Descriptor Finished Address Register (RDFXR)

RDFXR stores the start address of the receive descriptor for which the E-DMAC has just completed the write-back processing. Up to which receive descriptor has been processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. In the initial setting, set the address of the descriptor immediately before the descriptor that is pointed to by the address in RDFAR.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFX[31:0]	All 0	R/W	Receive Descriptor Finished Address Writing to these bits during the reception is prohibited.

23.3.83 Receive Descriptor Final Flag Register (RDFFR)

RDFFR indicates whether the receive descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in RDFXR is at the end of the receive descriptor queue (descriptor list).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RDLF	0	R/W	Receive Descriptor Queue Last Flag Indicates whether the receive descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in RDFXR is at the end of the receive descriptor queue (descriptor list). 0: Not the last descriptor in the receive descriptor queue 1: Last descriptor in the receive descriptor queue

23.3.84 Transmit Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmit descriptor. Which transmit 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. In the initial setting, set the address of the transmit descriptor at which transmit processing is to be started.

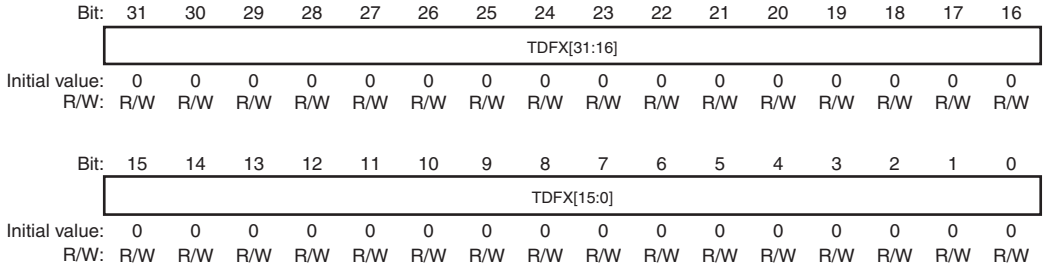
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFA[31:0]	All 0	R/W	Transmit Descriptor Fetch Address Writing to these bits during transmission is prohibited.

23.3.85 Transmit Descriptor Finished Address Register (TDFXR)

TDFXR stores the start address of the transmit descriptor for which the E-DMAC has just completed the write-back processing. Up to which transmit descriptor has been processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. In the initial setting, set the address of the transmit descriptor immediately before the descriptor that is pointed to by the address in TDFAR.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFX[31:0]	All 0	R/W	Transmit Descriptor Finished Address Writing to these bits during transmission is prohibited.

23.3.86 Transmit Descriptor Final Flag Register (TDFFR)

TDFFR indicates whether the transmit descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in TDFXR is at the end of the transmit descriptor queue (descriptor list).

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDLF	0	R/W	Transmit Descriptor Queue Last Flag Indicates whether the transmit descriptor for which the E-DMAC has just completed the write-back processing and whose start address is stored in TDFXR is at the end of the transmit descriptor queue (descriptor list). 0: Not the last descriptor in the transmit descriptor queue 1: Last descriptor in the transmit descriptor queue

23.3.87 Overflow Alert FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the E-MAC. The threshold can be set by the size of the receive FIFO data (bits RFD[7:0]) and the number of receive frames (bits RFF[4:0]).

If the same receive FIFO size as set by the FIFO depth register (FDR) is set when flow control is turned on according to the RFD setting condition, flow control is turned on with (FIFO data size • 64) bytes. For instance, when the RFD bits in FDR = 7 and the RFD bits in this register = 7, flow control is turned on when (2,048 • 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 that set in the RFD bits in FDR.

Flow control is turned on when either of the setting conditions of bits RFF[4:0] and bits RFD[7:0] is satisfied. Flow control is turned off when neither of the conditions is satisfied (release).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RFF[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	RFF[4:0]	H'1F	R/W	Receive FIFO Overflow Alert Signal Output Threshold H'00: When one receive frame has been stored in the receive FIFO H'01: When two receive frames have been stored in the receive FIFO : : H'16: When 23 receive frames have been stored in the receive FIFO H'17: When 24 receive frames have been stored in the receive FIFO

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 to 0	RFD[7:0]	H'FF	R/W	Receive FIFO Overflow Alert Signal Output Threshold H'00: When (256 – 32) bytes of data is stored in the receive FIFO H'01: When (512 – 32) bytes of data is stored in the receive FIFO : : H'06: When (1,792 – 32) bytes of data is stored in the receive FIFO H'07: When (2,048 – 64) bytes of data is stored in the receive FIFO

23.3.88 Receive Data Padding Insert Register (RPADIR)

RPADIR is a 32-bit readable/writable register that inserts padding in receive data. When changing the settings of this register, execute a software reset by means of the SWRT and SWRR bits in the E-DMAC mode register (EDMR) before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	PADS[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	PADS[4:0]	H'00	R/W	Padding Size H'00: No padding insertion H'01: 1-byte insertion : : H'1F: 31-byte insertion
15 to 0	PADR[15:0]	H'0000	R/W	Padding Slot H'0000: Inserts specified size of padding at the first byte H'0001: Inserts specified size of padding at the second byte : : H'FFFF: Inserts specified size of padding at the 64K byte

23.4 Operation

The GETHER consists of the following three function units:

- DMA transfer controller (E-DMAC): DMA transfer between the transmit/receive buffer in the memory and the transmit/receive FIFO
- MAC controller (E-MAC): Transmission/reception processing between the transmit/receive FIFO and the GMII/MII/RMII
- Transfer Switching Unit (TSU): Transfer processing between port 0 and port 1, and CAM processing

Using its direct memory access (DMA) function, the E-DMAC performs DMA transfer of frame data between a user-specified Ethernet frame transmission/reception data storage destination (accessible memory space: transmit buffer/receive buffer) and the transmit/receive FIFO in the E-DMAC. The user cannot read and write data from and to the transmit/receive FIFO directly via the CPU.

To enable the E-DMAC to perform DMA transfer, information (data) including a transmit/receive data storage address and so forth, referred to as a descriptor, is required. The E-DMAC reads transmit data from the transmit buffer or writes receive data to the receive buffer according to the descriptor information. By arranging multiple descriptors as a descriptor row (list) (to be placed in a readable/writable memory space), multiple Ethernet frames can be transmitted or received continuously.

The E-DMAC consists of two systems: one for port 0 and the other for port 1, and both operate independently for transmission and reception.

The E-MAC constructs an Ethernet frame using the data written to the transmit FIFO and transmits the frame to the GMII/MII/RMII. It also performs a CRC check of an Ethernet frame received from the GMII/MII/RMII and deconstructs the frame to write to the receive FIFO. The E-MAC supports three formats MII, GMII and RMII for interface to the PHI-LSI connected externally to this LSI.

The E-MAC consists of two controllers: E-MAC0 for port 0 and E-MAC1 for port 1, which correspond to E-DMAC0 and E-DMAC1 respectively.

The TSU performs Ethernet frame data transfer between the E-MAC0 and E-MAC1. The TSU, which is placed between the E-DMAC and E-MAC, references the CAM entry table to select one of the following tasks according to the Ethernet frame destination address (DA) input to the E-MAC.

- Receives data and writes to the receive FIFO.
- Transfers data and writes to the transfer FIFO.
- Receives data and writes to the receive FIFO and transfer FIFO.
- Discards data.

The TSU performs transfers from port 0 to port 1 and from port 1 to port 0 independently.

Figure 23.2 shows the frame data path and an overview of each setting.

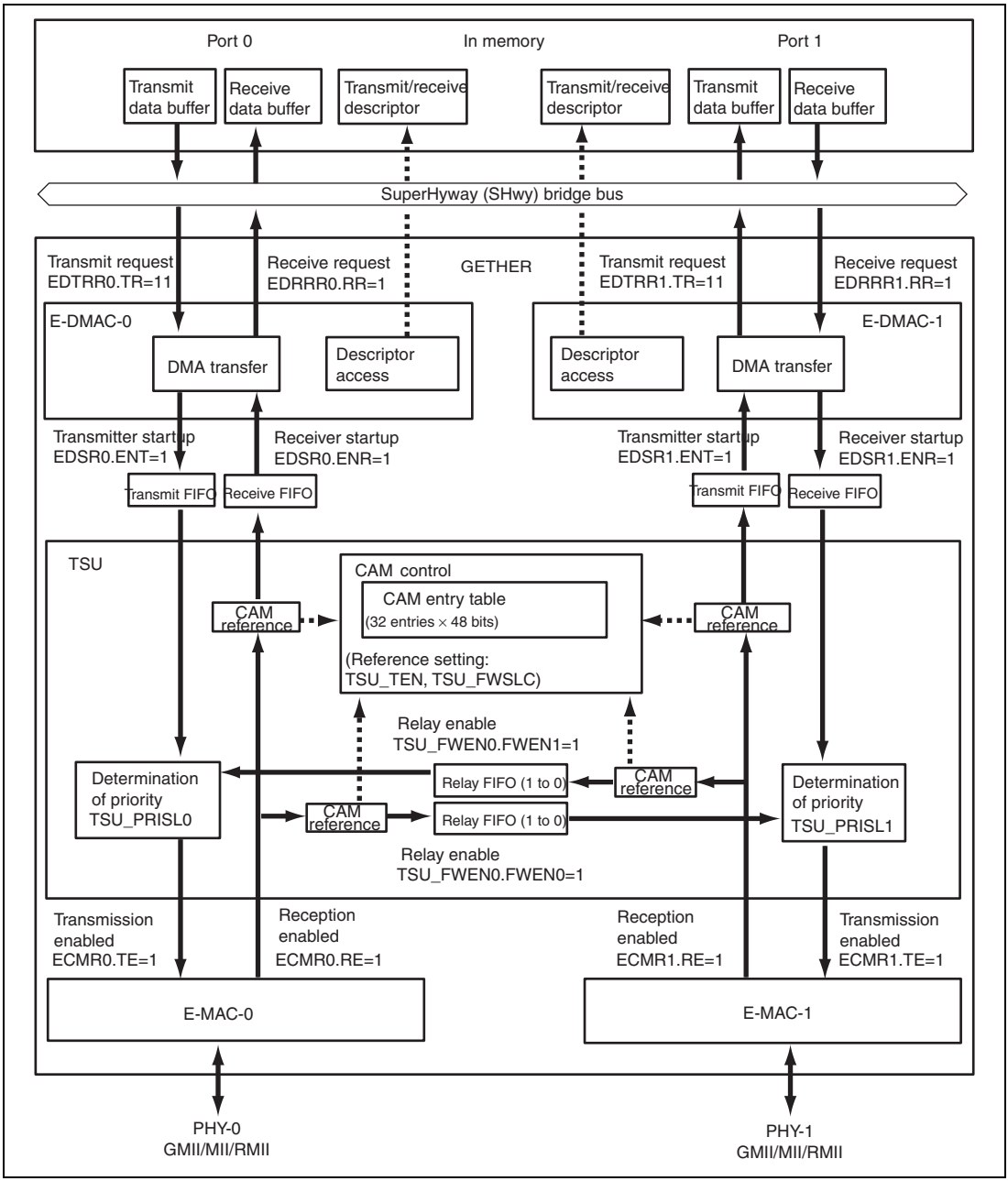


Figure 23.2 GETHER Data Path and Various Settings

23.4.1 Descriptors and Descriptor List

The E-DMAC performs DMA transfer according to the information (data), referred to as a descriptor, written in memory space. There are two types of descriptors: transmit descriptors and receive descriptors. Before a DMA transfer, DMA transfer information including a transmit/receive frame data storage address must be set by software.

The E-DMAC automatically starts reading a transmit/receive descriptor when the TR bits in EDTRR are set to 11 or the RR bit in EDRRR is set to 1, and performs DMA transfer of frame data between the transmit/receive buffer and transmit/receive FIFO according to the information stored in the descriptor. After completion of Ethernet frame transmission/reception, the E-DMAC disables the descriptor valid/invalid bit and reflects the result of transmission/reception in the status bits.

Descriptors are placed in a readable/writable memory space. The address of the start descriptor (descriptor to be read first by the E-DMAC) is set in TDLAR/RDLAR. When multiple descriptors are prepared as a descriptor row (descriptor list), the descriptors are placed in continuous addresses (memory) according to the descriptor length set in the DL0 and DL1 bits in EDMR.

The E-DMAC consists of two systems: one for port 0 and the other for port 1. The DMAC for transmission and the DMAC for reception operate independently, and the DMAC for port 0 and the DMAC for port 1 operate independently. Place descriptors for transmission and reception and descriptors for port 0 and port 1 in those address spaces that do not overlap. If addresses are overlapped, E-DMAC does not successfully operate.

(1) Transmit Descriptor

Figure 23.3 shows the configuration of a transmit descriptor and the relationship with a transmit buffer.

The data of a transmit descriptor consists of TD0, TD1, TD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR.

TD0 indicates whether the transmit descriptor is valid or invalid, and information about the descriptor configuration and status. TD1 indicates the length of data in a transmit buffer to be transferred (TDL) as specified by the descriptor. TD2 indicates the start address of a transmit buffer that holds data to be transferred (TBA).

Depending on the descriptor specification, one transmit descriptor can specify all transmit data of one frame (single-frame/single-buffer) or multiple descriptors can specify the transmit data of one

frame (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation, the data portion that is used in a fixed manner in each Ethernet frame transmission can be referenced by multiple descriptors. For example, multiple descriptors can share the destination address and transmit source address in an Ethernet frame, and the remaining data can be stored in each separate buffer.

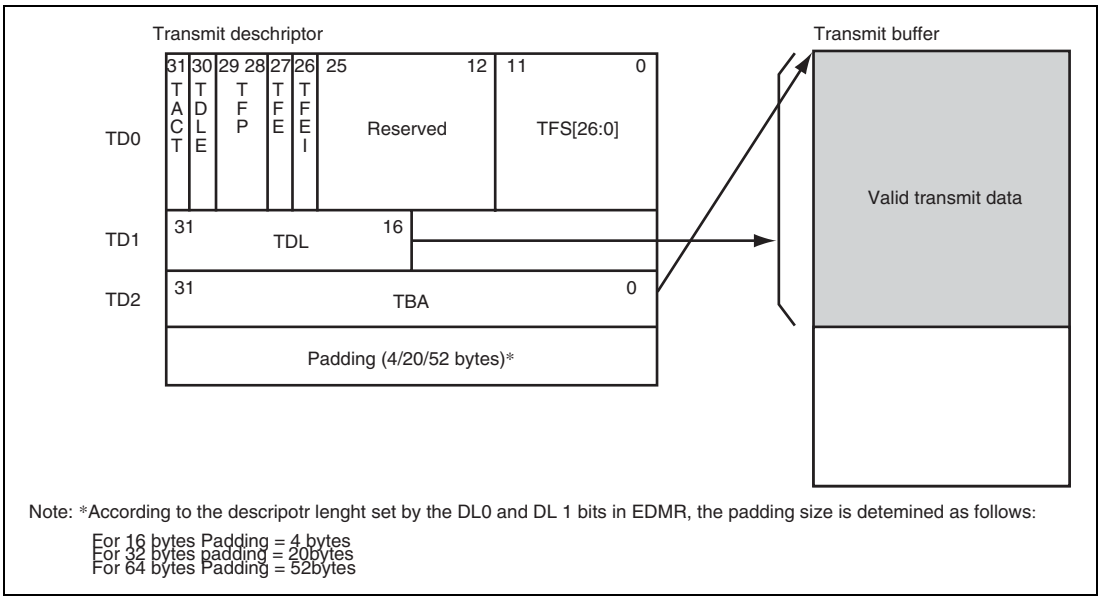


Figure 23.3 Relationship between Transmit Descriptor and Transmit Buffer

(a) Transmit Descriptor 0 (TD0)

Before the TR bits in EDTRR are set to 11, the user sets whether the bits of the descriptor are valid or invalid bit and sets other descriptor configuration. After Ethernet frame transmission, the E-DMAC disables the valid/invalid bits of the descriptor and writes status information. This operation is referred to as write-back.

When using TD0, the user should write desired values to bits 31 to 28 and 26 according to the descriptor configuration. Bits 27 and 25 to 0 should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Valid/Invalid Indicates whether the corresponding descriptor is valid or invalid. To make this bit valid, store transmit data in a transmit buffer (user-specified transmit data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 after data transfer. 0: Indicates that this transmit descriptor is invalid Indicates the initial setting state, the state after 0 is written, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because the E-DMAC data transfer processing is completed. If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the TR bit in EDTRR to 0, and halts transfer operation related to transmission by the E-DMAC. 1: Indicates that this transmit descriptor is valid After the user writes 1 to this bit, this bit indicates that data is not transferred yet or data is being transferred. When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid.
30	TDLE	0	R/W	Transmit Descriptor List End Indicates whether the corresponding descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor After transfer of the corresponding descriptor, the E-DMAC reads the next one in the list of continuous descriptors. 1: Last descriptor After transfer of the corresponding descriptor, the E-DMAC reads the descriptor placed at the address indicated by TDLAR.

Bit	Bit Name	Initial Value	R/W	Description
29, 28	TFP[1:0]	00	R/W	Transmit Frame Position These bits indicate whether information of this descriptor represents information about the start, middle, or end of the transmit frame. 00: The information of the descriptor represents information about the middle of the frame. 01: The information of the descriptor represents information about the end of the frame. 10: The information of the descriptor represents information about the start of the frame. 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)). Reference When one frame is divided for use, the method of specifying this bit for a descriptor row according to the number of divisions is described below. • For single-frame/single-descriptor operation First descriptor: TFP[1:0] = 11 • For single-frame/two-descriptor operation First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 01 • For single-frame/three-descriptor operation First descriptor: TFP[1:0] = 10 Second descriptor: TFP[1:0] = 00 Third descriptor: TFP[1:0] = 01 When the number of divisions is large, a descriptor row is configured by adding intermediate descriptors with TFP[1:0] = 00.

Bit	Bit Name	Initial Value	R/W	Description
27	TFE	0	R/W	Transmit Frame Error Occurrence Indicates that an error occurred in the transmit frame. 0: The TFS11 to TFS0 bits are all 0 1: One of the TFS11 to TFS0 bits is 1 The TFS8 to TFS0 bits can be masked for each factor by using TRSCER. TheTFS11 to TFS9 bits cannot be masked. This bit is set by the E-DMAC write-back operation.
26	TWBI	0	R/W	Write-Back Completion Interrupt Notification 0: Does not notify of a write-back completion interrupt 1: After a write-back operation to this descriptor is complete, this bit sets the TWB1 and TWB0 bits in EESR to 11 and notifies the CPU of a write-back completion interrupt. This bit is valid only for the descriptor including the end of transmit frame (TFP = 01 or 11). This bit is cleared to 0 by the E-DMAC write-back operation.
25 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	TFS[11:0]	All 0	R/W	Transmit Frame Status These bits indicate the status of the corresponding frame. A bit below, which is set by the E-DMAC write-back operation, indicates the occurrence of the corresponding event when set to 1. • TFS[11:10]: Reserved (The write value should always be 0.) • TFS[9]: Transmit FIFO underflow (Corresponding to the TUC bit in EESR) • TFS[8]: Detection of transmission abort (Corresponding to the TABT bit in EESR) • TFS[7:0]: Reserved (The write value should always be 0.)

(b) Transmit Descriptor 1 (TD1)

TD1 indicates the data length of the transmit buffer used by the corresponding descriptor.

The user should set TD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TDL	All 0	R/W	Transmit Buffer Data Length (in bytes) These bits indicate the data length of the corresponding transmit buffer in bytes. The maximum length is between 64 kbytes and 32 bytes (H'FFE0).
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(c) Transmit Descriptor 2 (TD2)

TD2 indicates the start address of the corresponding 32-bit width transmit buffer. An address value should be specified in a longword boundary.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBA	All 0	R/W	Transmit Buffer Start Address These bits set the start address of the corresponding transmit buffer in a 16-bit boundary.

If descriptors are set below, the E-DMAC does not return to normal operation until a system reset is performed.

- TFP (transmit frame position) is not logically correct
  Example: The TFP bits are set to 11 in a descriptor (descriptor A) and the TFP bits are set to 01 in the next descriptor (descriptor B). This specification means that there is no descriptor indicating the start of the transmit frame specified by descriptor B.
- TBL (transmit buffer length) is set to 0

When one transmit frame is divided into three parts or more with transmit descriptors, the E-DMAC performs the following write-back operation:

- A write-back operation is performed for a transmit descriptor including information for the start of the transmit frame (TFP = 10 or 11) and for a transmit descriptor including information for the end of the frame (TFP = 01 or 11).
- A write-back operation is not performed for a transmit descriptor for the middle of the frame (TFP = 00).

However, TFE (transmit frame error occurrence) or TFS (transmit frame status) is written only to a transmit descriptor including information for the end of the frame (TFP = 01 or 11) by a write-back operation.

Before changing a transmit descriptor with the software, make sure that a write-back operation has been performed (TACT = 0) for the transmit descriptor including information for the end of the frame (TFP = 01 or 11) to avoid overwriting (re-setting) an unprocessed transmit descriptor.

(2) Receive Descriptor

Figure 23.4 shows the relationship between a receive descriptor and receive buffer.

The data of a receive descriptor consists of RD0, RD1, RD2, and padding data in groups of 32 bits from top to end. The length of padding data is determined according to the descriptor length specified by the DL0 and DL1 bits in EDMR.

RD0 indicates whether the receive descriptor is valid or invalid, and information about descriptor configuration and status. RD1 indicates the length of data that can be received in the receive buffer specified by the descriptor (RBL) and the length of the received frame data (RDL). RD2 indicates the start address of the receive buffer for storing receive data (RBA).

Depending on the descriptor specification, one receive descriptor can specify the storing of all receive data of one frame in a receive buffer (single-frame/single-buffer) or multiple descriptors can specify the storing of the receive data of one frame in receive buffers (single-frame/multi-buffer). As an example of single-frame/multi-buffer operation, suppose that a row of multiple descriptors (descriptor list) is prepared, RBL of each descriptor is 500 bytes, and a 1514-byte Ethernet frame is received. In such a case, the received Ethernet frame is transferred sequentially to buffers, 500 bytes for each buffer, starting with the first descriptor. Only the last 14 bytes are transferred to the fourth buffer. When a frame longer than RBL of a descriptor is received, the E-DMAC transfers the remaining data to the receive buffer by using the subsequent descriptors. As an example of efficient single-frame/multi-buffer operation, information items on different processing layers in an Ethernet frame can be separated from each other by using different buffers. For example, the destination address, transmit source address, and type field data in an Ethernet

frame can be stored in buffer 1 (set RBL to 14 bytes) and the remaining data can be stored in buffer 2 (set RBL to 1500 bytes). All receive frames, of course, can be stored in a single buffer if multiple descriptors are prepared and RBL of each descriptor is set to more than 1514 bytes (maximum Ethernet frame length).

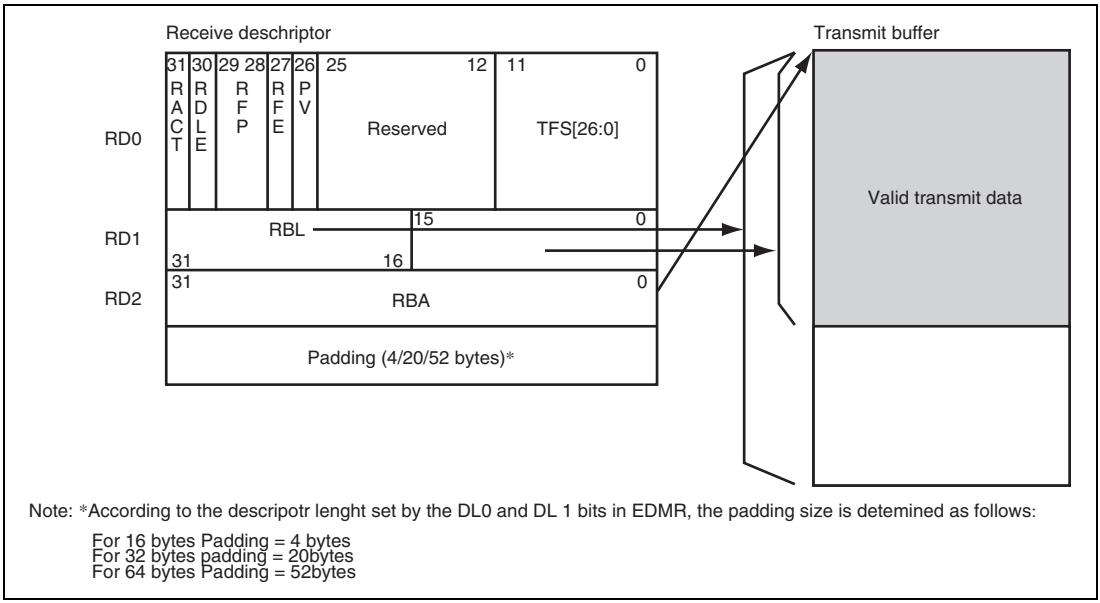


Figure 23.4 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

The user sets whether the bits of the descriptor are valid or invalid and whether the descriptor represents the end of the descriptor list in RD0 before the RR bit in EDRRR is set to 1 and the start of a read by the E-DMAC. After receive DMA transfer of an Ethernet frame by the E-DMAC, the E-DMAC disables the valid/invalid bits of the descriptor and writes status information. This operation is referred to as write-back.

When using RD0, the user should write desired values to bits 31 and 30 according to the descriptor configuration. Bits 29 to 0 should be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Valid/Invalid Indicates whether this descriptor is valid or invalid. To make this bit valid, prepare a receive buffer (user-specified receive data storage destination) beforehand, then write 1 to this bit. The E-DMAC clears this bit to 0 after data transfer. 0: Indicates that this receive descriptor is invalid Indicates the initial setting state, the state after 0 is written to, or (in case the user writes 1 to this bit) that this bit is cleared to 0 because the E-DMAC data transfer processing is completed If this state is recognized when the E-DMAC reads a descriptor, the E-DMAC clears the RR bit in EDRRR to 0, and halts transfer operation related to reception by the E-DMAC 1: Indicates that this receive descriptor is valid Indicates that data is not transferred yet after the user writes 1 to this bit, or that data is being transferred When there is a descriptor row (descriptor list) consisting of multiple continuous descriptors, the E-DMAC can continue operation when this bit of the next descriptor is valid
30	RDLE	0	R/W	Receive Descriptor List End Indicates whether this descriptor is the last descriptor of the descriptor row (descriptor list). 0: Not last descriptor After transfer of this descriptor, the E-DMAC reads the next one in the list of continuous descriptors 1: Last descriptor After transfer of this descriptor, the E-DMAC reads the descriptor placed at the address indicated by RDLAR

Bit	Bit Name	Initial Value	R/W	Description
29, 28	RFP[1:0]	00	R/W	Receive Frame Position 1, 0 The E-DMAC indicates by write-back operation whether information of the corresponding descriptor represents information about the start, middle, or end of the receive frame. 00: The information of the descriptor represents information about the middle of the frame 01: The information of the descriptor represents information about the end of the frame 10: The information of the descriptor represents information about the start of the frame 11: The information of the descriptor represents all information about the frame (single-frame/single-descriptor (single-buffer)) Reference The relationship between a frame after reception of one frame and a descriptor is described below. • For single-frame/single-descriptor operation First descriptor: RFP[1:0] = 11 • For single-frame/two-descriptor operation First descriptor: RFP[1:0] = 10 Second descriptor: RFP[1:0] = 01 • For single-frame/three-descriptor operation First descriptor: RFP[1:0] = 10 Second descriptor: RFP[1:0] = 00 Third descriptor: RFP[1:0] = 01 When the number of divisions is large, a descriptor row is configured by adding intermediate descriptors with RFP[1:0] = 00.
27	RFE	0	R/W	Receive Frame Error Occurrence Indicates that an error occurred in the receive frame. 0: RFS11 to RFS0 are all 0 1: One of RFS11 to RFS0 is 0 Each of RFS8 to RFS0 can be masked by using TRSCER. RFS11 to RFS9 cannot be masked. This bit is set by the E-DMAC write-back operation.

Bit	Bit Name	Initial Value	R/W	Description
26	PV	0	R/W	Padding Insertion Indicates whether the padding specified by RPADIR was inserted in the receive frame processed with this descriptor or not. 0: No padding inserted 1: Padding inserted This bit can be changed by the E-DMAC write-back processing.
25 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	RFS[11:0]	All 0	R/W	Receive Frame Status Indicate the status of the corresponding frame. A bit below, when set to 1, indicates the occurrence of the corresponding event. If an event indicated by any of RFS9 to RFS0 occurs, the frame is not received completely. RFS[11:10]: Reserved RFS[9]: Receive FIFO overflow (corresponding to the RFOF bit in EESR) RFS[8]: Detection of reception abort (Corresponding to the RABT bit in EESR) RFS[7]: Multicast address frame received (corresponding to the RMAF bit in EESR) RFS[6]: Carrier extension error (corresponding to the CEEF bit in EESR) RFS[5]: Carrier extension loss (corresponding to the CELF bit in EESR) RFS[4]: Residual-bit frame receive error (corresponding to the RRF bit in EESR) RFS[3]: Long frame receive error (corresponding to the RTLF bit in EESR) RFS[2]: Short frame receive error (corresponding to the RTSF bit in EESR) RFS[1]: PHY-LSI receive error (corresponding to the PRE bit in EESR) RFS[0]: CRC error on receive frame (corresponding to the CERF bit in EESR)

(b) Receive Descriptor 1 (RD1)

In RD1, the user specifies the data length of a receive buffer usable by the corresponding descriptor. After reception of a frame, RD1 indicates the length of a frame received by the E-DMAC.

The user should set RD1 before the start of a read by the E-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Data Length (in bytes, to be specified with a 32-byte boundary) These bits set the length of data that can be received by the corresponding receive buffer with an integral multiple of 32 bytes. The maximum receive buffer data length is between 64 kbytes and 32 bytes (H'FFE0).
15 to 0	RDL	All 0	R	Receive Data Length These bits indicate the data length of a receive frame stored in the receive buffer. Receive data transferred to the receive buffer does not include CRC data (4 bytes) placed at the end of a frame. As a receive frame length, the number of bytes (valid data bytes) not including CRC data are reported. In single-frame/multi-buffer (descriptor) operation, only the receive data length of the last descriptor is valid. The receive data length of an intermediate descriptor has no meaning. The maximum frame length that can be received is: When padding function is invalid: 64 kbytes between 1 byte (H'FFFF) When padding function is valid: 64 kbytes between 32 bytes (H'FFE0)

(c) Receive Descriptor 2 (RD2)

RD2 indicates the start address of the corresponding receive buffer. Set the start address of a receive buffer with a 32-byte boundary.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBA	All 0	R/W	Receive Buffer Start Address These bits set the start address of the corresponding receive buffer with a 32-byte boundary.

The E-DMAC performs DMAtransfer for a receive frame from the address specified by RBA (receive buffer address) to the receive buffer in 32-byte units. RBL (receive buffer length) must be set to be an integral multiple of 32 bytes.

If data to be transferred is less than 32 bytes, invalid data will be written to.

[Example]

When the receive frame length is 170 bytes and the required receive buffer capacity is 192 bytes (32 bytes × 6), the sixth DMA-transfer causes invalid data to be written to the receive buffer (In the 32-byte DMA data, the former 10 bytes are valid and the latter 22 bytes are invalid).

Padding of the value 0 can be inserted into only one position in the receive frame by setting RPADIR. The padding size can be selected from 1 byte to 31 bytes in byte units. When padding is inserted into a receive frame, a receive buffer area equal to the total of "receive frame length and padding size" is required. RPADIR setting is valid for all receive frames.

RFE (receive frame error occurrence), PV (padding insertion), RFS (receive frame status) and RFS (receive frame status) are only set in the receive descriptor including information for the end of the frame (TFP = 01 or 11) by a write-back operation.

Before re-setting a receive descriptor with the software, completion of a write-back operation for the receive descriptor (RACT = 0) must be confirmed to avoid rewriting to (and re-setting) an unprocessed receive descriptor.

(3) Descriptor and Transmit/Receive Buffer

(a) Transmission

Each transmit descriptor specifies one transmit buffer. The E-DMAC transfers a transmit frame stored in a transmit buffer specified by a transmit descriptor to the transmit FIFO. Multiple transmit frames stored in transmit buffers specified by multiple descriptors can be connected into one transmit frame and transferred to the transmit FIFO.

Figure 23.5 shows the relationship between the transmit descriptors and transmit buffers.

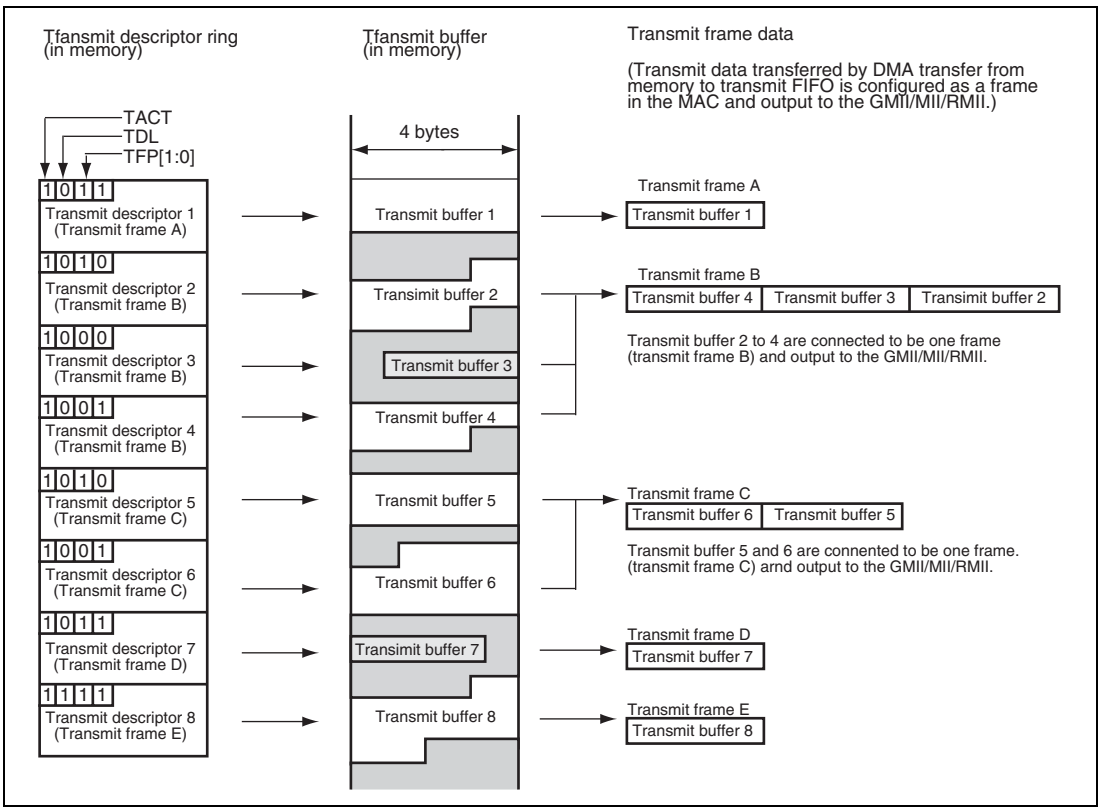


Figure 23.5 Relationship between Transmit Descriptor and Transmit Buffer

(b) Reception

Each receive descriptor specifies one receive buffer. The E-DMAC receives a receive frame from the receive FIFO and stores it in a receive buffer specified by a receive descriptor. If the receive frame size exceeds the receive buffer size, the remaining data of the receive frame can be stored in

a different receive buffer specified by a different receive descriptor. Thus, one receive frame can be stored in multiple receive buffers.

Figure 23.6 shows the relationship between the receive descriptors and receive buffers.

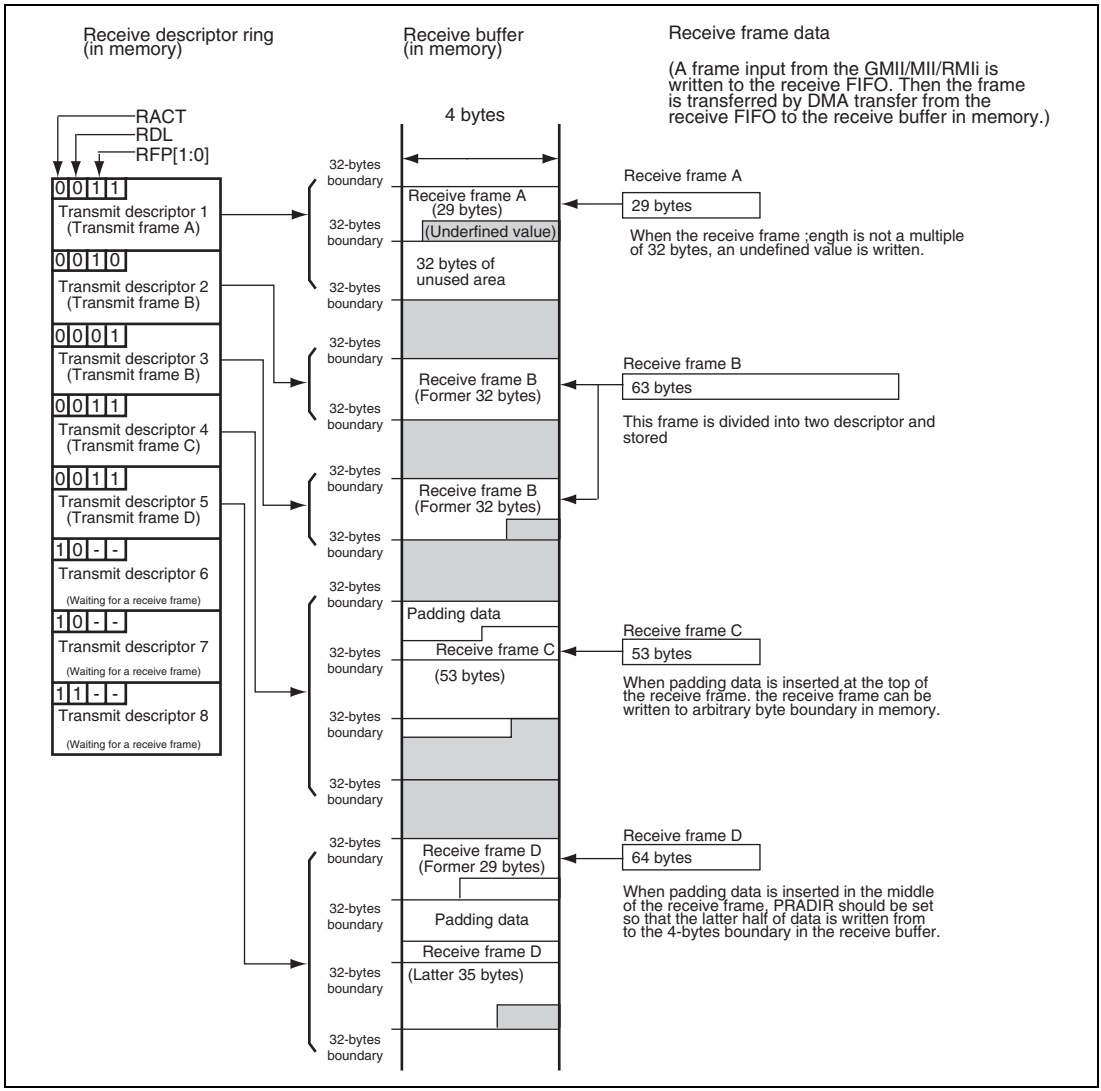


Figure 23.6 Relationship between Receive Descriptor and Receive Buffer

(4) Descriptor Pointer

The E-DMAC controls the transmit and receive descriptor addresses in memory and the processing priority by using the following registers.

1. Registers related to a transmit descriptor
- TDLAR: Address of the start descriptor in a list of transmit descriptors.
- TDFAR: Address of the transmit descriptor to be processed
- TDFXR: Address of the transmit descriptor that finished processing (set by a write-back operation) last
- TDFFR (DL bit): Indicates whether the TDLE value of the transmit descriptor specified by TDFXR is 1 or not.

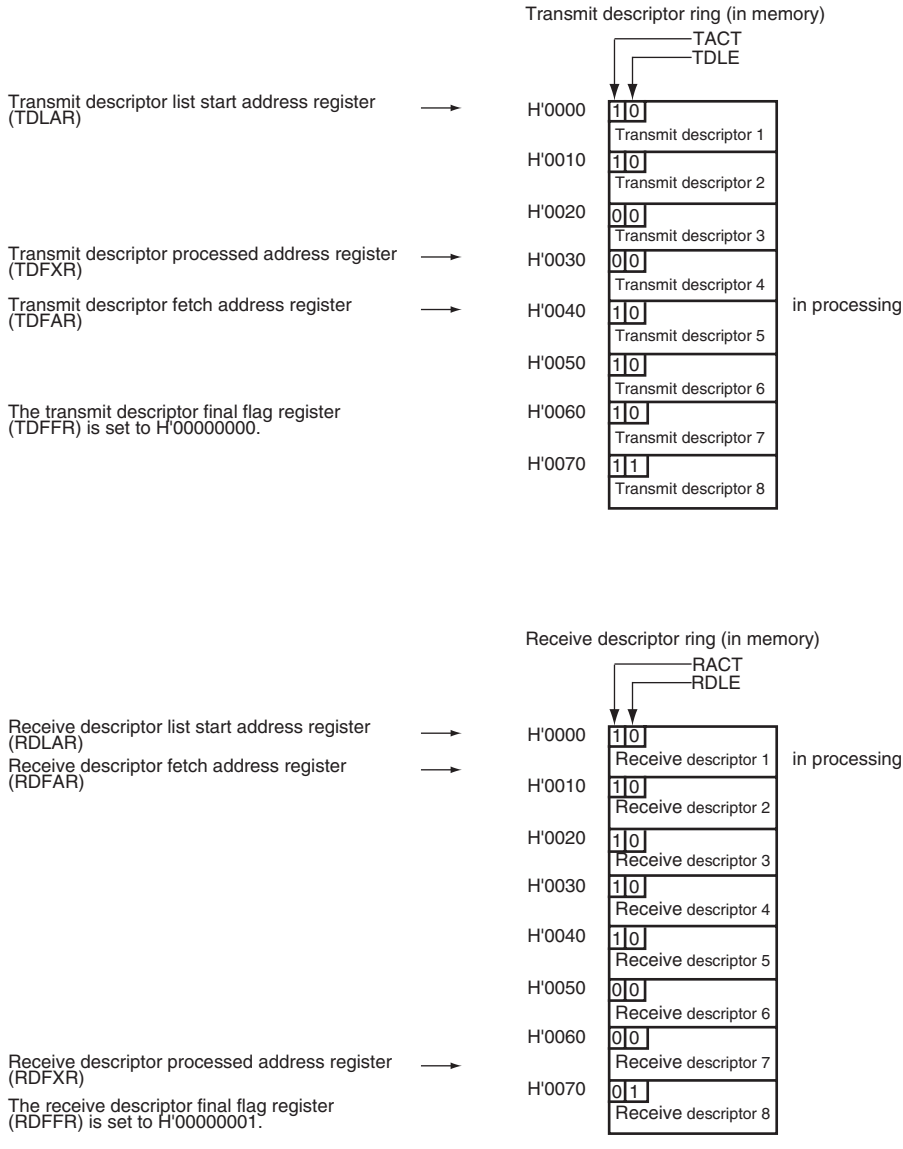
2. Registers related to receive descriptor:
- RDLAR: Address of the start descriptor in a list of receive descriptors.
- RDFAR: Address of the receive descriptor to be processed
- RDFXR: Address of the receive descriptor that finished processing (set by a write-back operation) last
- RDFFR (DL bit): Indicates whether the RDLE value of the receive descriptor specified by RDFXR is 1 or not.

Transmit descriptors and receive descriptors have a ring structure. When the TDLE (RDLE) value of the processed transmit (receive) descriptor is 0, the next descriptor will be processed. The next descriptor is the transmit (receive) descriptor at the address obtained by adding the processed transmit (receive) descriptor address to the descriptor length specified by the DL bit in EDMR. When the TDLE (RDLE) value of the processed transmit (receive) descriptor is 1, the transmit descriptor indicated by TDLAR (RDLAR) will be processed next. Figure 23.7 shows the relationship between the transmit/receive descriptor ring and read pointer.

The transmit descriptor list must be large enough to point to five or more transmit frames. If four or less transmit frames are pointed to in a list, E-DMAC operation is not guaranteed. Accordingly, do not set that all the transmit descriptors in a ring are used by four or less descriptors. The receive descriptor list does not have this restriction. For example, one receive frame can use all receive descriptors in a list.

In the initial setting, the start address of a descriptor list must be set to TDLAR (RDLAR) and TDFAR (RDFAR), and the end descriptor address of the descriptor list to TDFXR (RDFXR) by the software.

The E-DMAC updates TDFAR (RDFAR), TDFXR (RDFXR) and the DL bit in TDFFR (DL bit in RDFFR) each time a descriptor is processed.



Note: Addresses in the descriptor list are shown as an example when the descriptor length is 16 bytes

Figure 23.7 Relationship between Transmit/Receive Descriptor and Descriptor Pointing Registers

23.4.2 Transmission

(1) Transmission Procedure and Processing Flow

When 11 is written to the TR bits in EDTRR with the TE bit in ECMR set to 1 and there is empty space of 32 bytes or more in the transmit FIFO, the E-DMAC reads the descriptor following the previously used descriptor from the transmit descriptor list (or the descriptor indicated by TDLAR at the initial startup).

If the TACT bit of the read descriptor is set to 1 (valid), the E-DMAC sequentially reads transmit frame data from the transmit buffer start address specified by TD2 and transfers the data to the transmit FIFO. The E-DMAC configures a transmit frame and starts transmission to the GMII/MII/RMII. 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.

- TFP = 10 (start of a frame)
  Descriptor write-back (writing 0 to the TACT bit) is performed after completion of DMA transfer.
- TFP = 01 or 11 (end of a frame)
  Descriptor write-back (writing 0 to the TACT bit and writing status) is performed after completion of frame transmission.
- TFP = 00 (frame continued)
  Descriptor write-back is not performed. The TACT bit retains the value 1.

As long as the TACT bit of a read descriptor is set to 1 (valid), the reading of E-DMAC descriptors and the transmission of frames continue.

When a descriptor with the TACT bit cleared to 0 (invalid) is read, the E-DMAC performs the following processing and completes transmit processing.

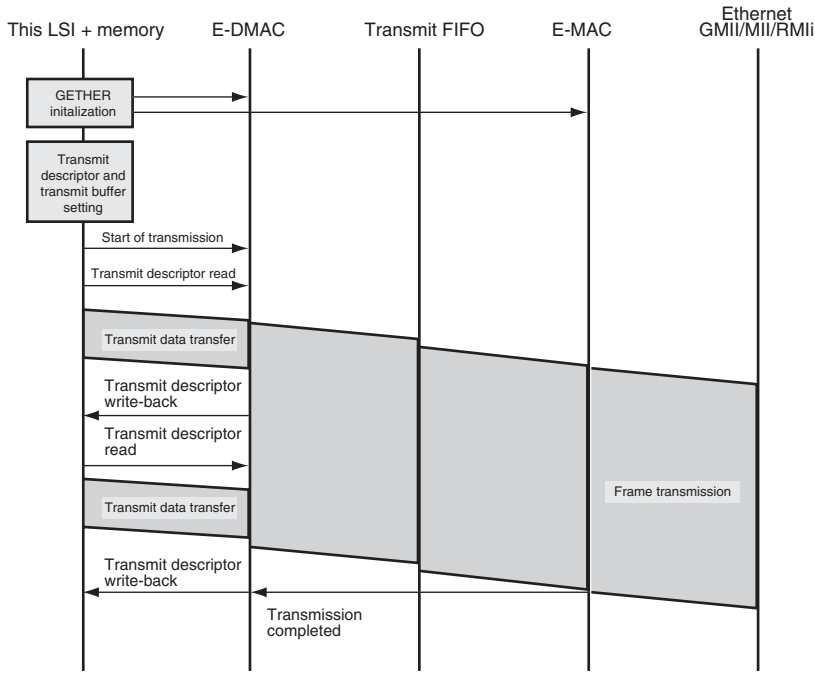
- Clears the TR bits in EDTRR to 00.
- Writes the TC bits in EESR to 11 and generates an interrupt to the CPU.

The E-DMAC can store up to four frames of data in the transmit FIFO.

When the following conditions are satisfied, the E-MAC transmit processing section reads transmit data from the transmit FIFO to configure a frame and transmits the frame to the GMII/MII/RMII

- The amount of data in the transmit FIFO exceeds the number of bytes specified by TFTR.
- One or more frame of data is stored in the transmit FIFO.
- The transmit FIFO has no space (full of transmit wait data for the GMII/MII/RMII).

Figure 23.8 shows an example of transmission flow.



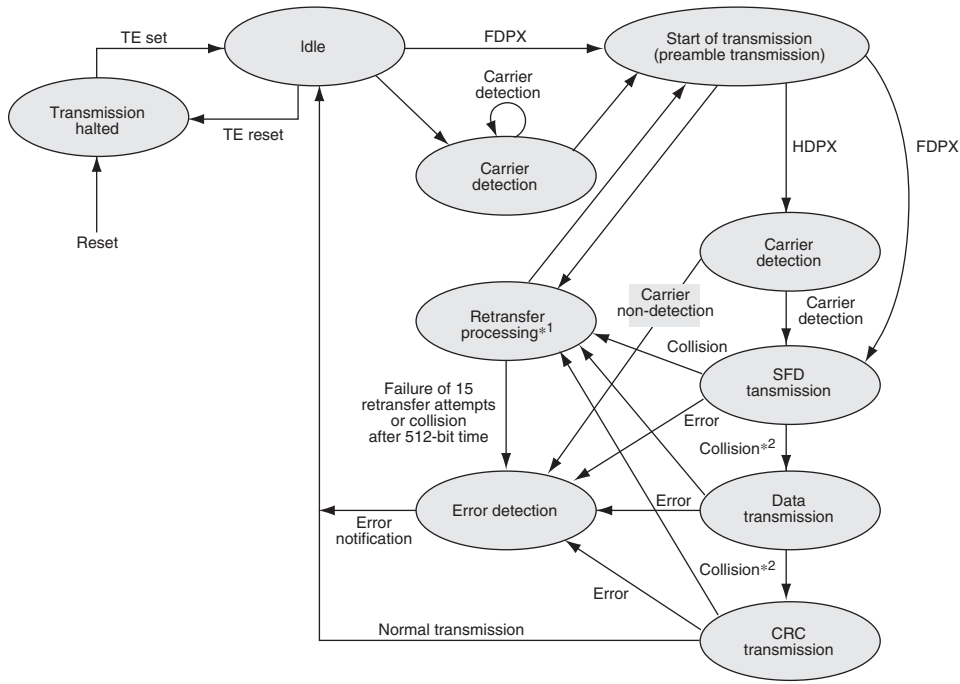
[Legend]
GETHER initialization: Executes a software reset with the SWR bit in EDMR set to 1.
Transmit descriptor and transmit setting: Setes transmit descriptors and transmit buffer, and sets E-MAC and E-DMAC registeers, then writes 11 to the TE bit in ECMR and the TR bit in EDTRR.
Start of transmission: Occurs when 1 is written to the TE bit in ECMR and 11 is written to the TR bit in EDTRR.
Transmit descriptor read: The E-DMAC reads a transmit descriptor.
Transmit data transfer : Writes transmit dara to the transmit FIFO by using DMA transfer by the E-DMAC
Transmit descriptor write-back: The E-DMAC writes 0 to the TACT bit and writes the transmit status to the transmit descriptor.

Figure 23.8 Sample Transmission Flowchart (Single-Frame/Two-Description)

Figure 23.9 shows the status change of the E-MAC transmitter. This operation is common among port 0 and port 1.

1. When the TE bit in ECMR is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit E-DMAC, the E-MAC sends the preamble to GMII/MII/RMII after a transmission delay caused by the carrier detection and 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.

Transmission flowchart



Legend
FDPX: Full-duplex
HDPX: Half-duplex

Notes:
1. Transmission retry processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm.
2. Transmission is retried only when data of 512 bits or less (including the premble and SFD) is transmitted. When a xollision is detected during the transmission of data oreater than 512 bits, only jam is transmitted and transmission based on the back-off algorithm is not retried.

Figure 23.9 E-MAC Transmitter State Transitions

(2) Transmission Error Processing

(a) Transmission Abort

If a transmission error is detected during frame transmission from the transmit FIFO to the GMII/MII/RMII, transmission of the frame data is aborted. At this time, if DMA transfer of the appropriate frame from the transmit buffer to the transmit FIFO has not been completed, the DMA transfer is also aborted.

Following a write-back operation to the transmit descriptor related to the transmit frame aborted by a transmission error, 1 is written to the TABT bit in EESR and an interrupt is issued to the CPU. The subsequent transmit descriptors will be processed normally.

(b) Transmit FIFO Underflow

If the transmit FIFO is empty (transmit FIFO underflow) during frame transmission from the transmit FIFO to the GMII/MII/RMII, the E-MAC forcibly aborts transmission of the frame to the GMII/MII/RMII. At this time, the frame that the E-MAC receives from the E-DMAC is cut off halfway. Then, the E-MAC performs the following operation:

- Writes the TFUF bit in EESR to 1 and generates an interrupt to the CPU.
- Performs a write-back operation to the transmit descriptor corresponding to the transmit frame.
- Following the write-back operation, writes the TUC bit in EESR and generates an interrupt to the CPU.

The subsequent transmit descriptors operate normally.

The E-MAC waits to start frame transmission from the transmit FIFO to the GMII/MII/RMII until the data that was stored in the transmit FIFO exceeds the number of the bytes specified by TFTR. Through the effective use of TFTR, the transmit FIFO underflow counts can be controlled.

(c) Transmit Descriptor Empty

When the TFP bits of the descriptor previously processed are set to 00 or 10 and the TACT bit of the read transmit descriptor is set to 0 (invalid), a transmit descriptor empty state is determined and 1 is written to the TDE bit in EESR, and then an interrupt is issued to the CPU.

When a transmit descriptor state is empty, start transmission processing after a software reset.

23.4.3 Reception

(1) Reception Procedure and Processing Flow

The E-MAC receiver separates the frame from the GMII/MII/RMII into preamble, SFD, data and CRC, and transfers the fields from DA (destination address) to the data to the receive FIFO. Up to 24 frames can be written in the receive FIFO. Figure 23.10 shows the status change of the E-MAC receiver. This operation is common among port 0 and port 1.

1. When the RE bit in ECMR is set to 1, 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. A frame with an invalid pattern is discarded.
3. In normal mode, if the destination of the frame address is this LSI, the receiver starts data reception when broadcast or multicast transmission is specified. In promiscuous mode, data reception starts regardless of the frame type.
4. Following data reception from the GMII/MII/RMII, 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 the receive FIFO. Reports an error status in the case of an abnormality.

After one frame has been received, if the RE bit in ECMR is set to 1, the receiver prepares to receive the next frame.

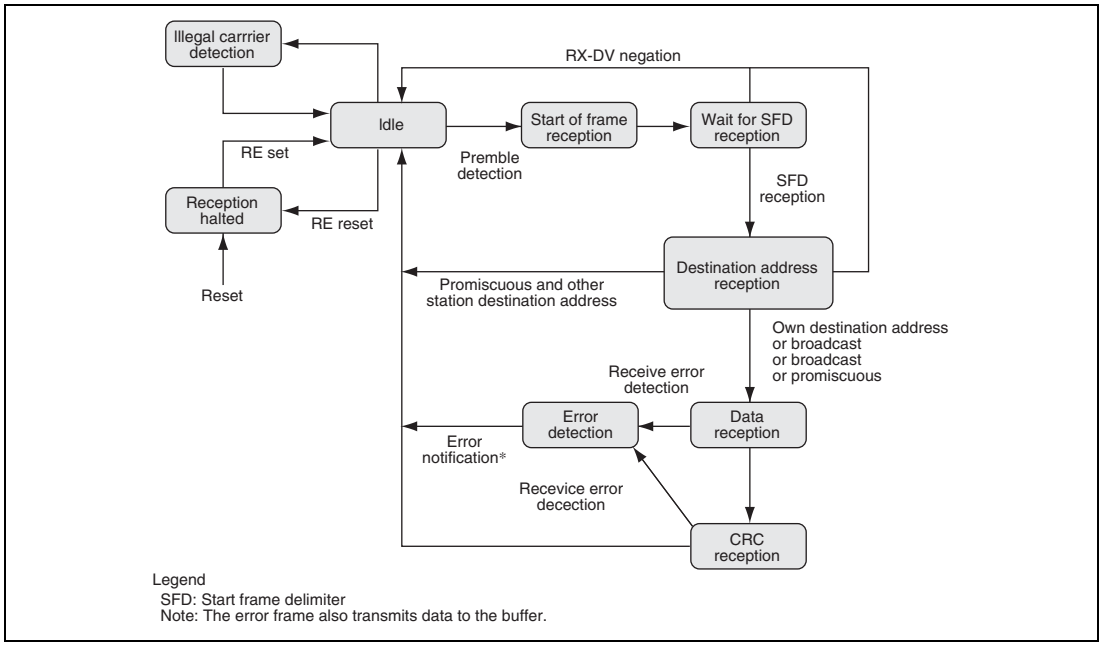


Figure 23.10 E-MAC Receiver State Transitions

CAM evaluation can be referenced during frame processing in reception (for details on the CAM function, refer to section 23.4.5, CAM Function).

When 1 is written to the RR bit in EDRRR while the RE bit in ECMR is set to 1, the E-DMAC reads the descriptor following the previously used descriptor from the receive descriptor list (or the descriptor indicated by RDLAR at the initial startup) then enters the receive wait state. If 32 bytes or more of data or the last byte of the receive frame is stored in the receive FIFO, the E-DMAC transfers receive FIFO data to the receive buffer specified by RD2 according to the receive descriptor with the RACT bit set to 1 (valid).

If the data length of a received frame is longer than the buffer length specified by RD1, the E-DMAC performs a write-back operation to the descriptor (set RFP to 10 or 00) when the buffer is full, then reads the next descriptor. The E-DMAC then continues to transfer data to the receive buffer specified by the new RD2.

When the following conditions are satisfied, a write-back operation is performed for the descriptor (RFP = 11 or 01), 11 is written to the FR bits in EESR, and an interrupt is issued to the CPU.

- The receive buffer has been full during DMA transfer.
- DMA transfer to the receive buffer of the last byte of the receive frame has been completed.

After the reception processing of the frame, the next descriptor reading standby state begins. At this time, if 32 bytes or more of data or the last byte of the receive frame is stored in the receive FIFO, the next receive descriptor process is performed continuously.

When the TACT bit of the read receive descriptor is 0 (invalid), the receive descriptor empty state is determined and the RDE bit in EESR is written to 1, and then an interrupt is issued to the CPU.

To receive frames continuously, set the RNC bit in RMCR to 1. The initial value is 0.

Figure 23.11 shows an example of reception flow.

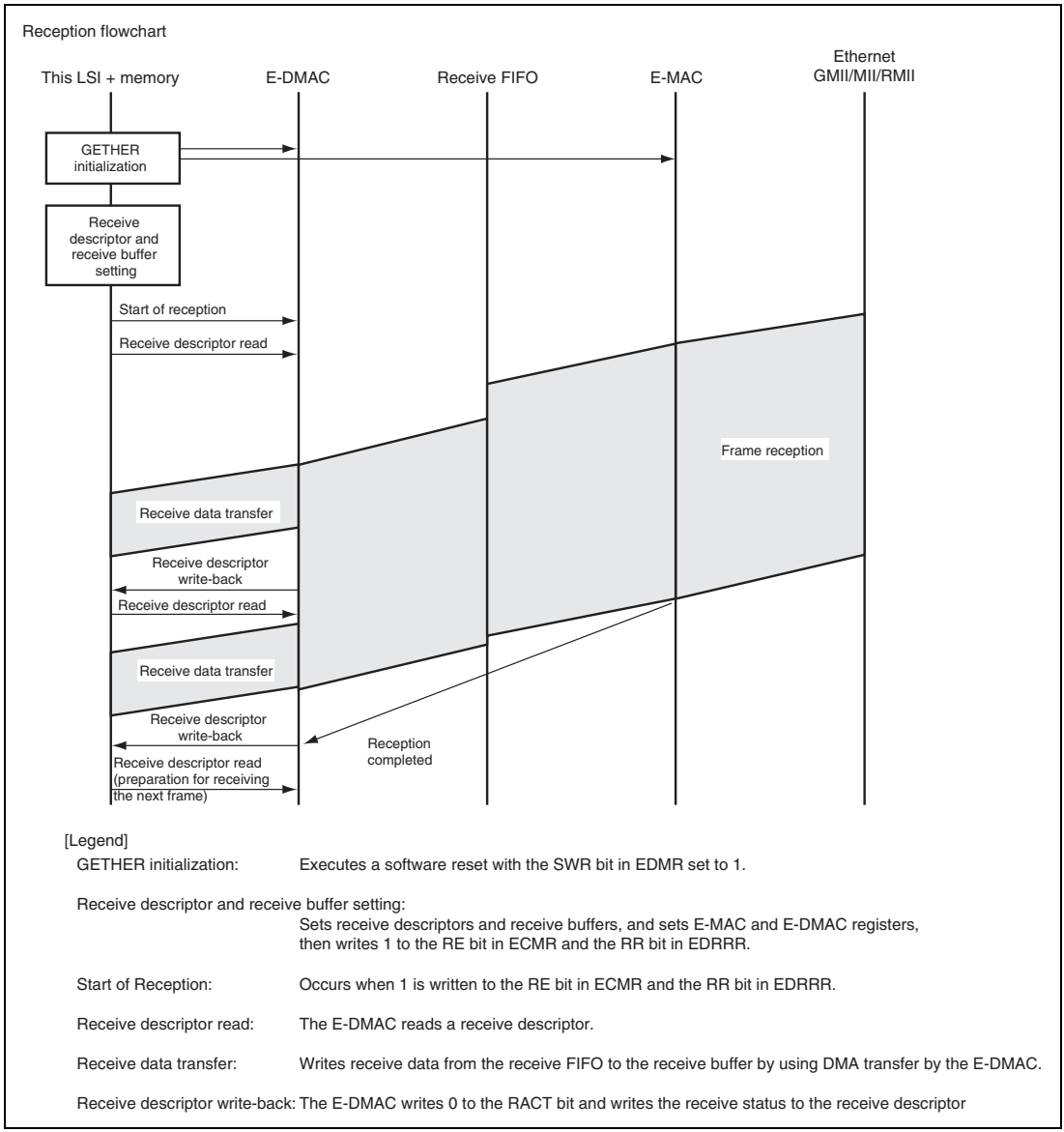


Figure 23.11 Sample Reception Flowchart (Single-Frame/Two-Descriptor)

(2) Reception Error Processing

(a) Reception Error

When a reception error occurs, the FR and RABT bits in EESR are set to 1 and an interrupt is issued to the CPU after a write-back operation for the receive descriptor related to the reception error frame.

If a reception error occurs when the length of the frame received from the GMII/MII/RMII is less than 32 bytes, DMA transfer to the receive buffer for the frame is not performed. At this time, the receive frame is discarded in the E-DMAC (flush function). However, if padding is inserted in the receive frame by RPADIR, the flush function is performed when the frame length including the padding bytes is less than 32 bytes.

(b) Receive FIFO Overflow

In any of the following cases, the E-MAC cannot receive frames from the GMII/MII/RMII because it has no space to store receive frames, and all the receive frames that have been transferred to the E-MAC will be discarded in the E-MAC (receive FIFO overflow).

- Receive FIFO is full of data waiting for DMA transfer (the receive FIFO has no space).
- The number of receive frames waiting for DMA transfer is 24 in total (the receive frame information managing area has no empty space; up to 24 frames can be managed).

If an overflow occurs due to the former case, the RFE bit in EESR is set to 1 and an interrupt is generated to the CPU. If an overflow occurs due to the latter case, the RFCOF bit in EESR is set to 1 and an interrupt is generated to the CPU. Each time a receive frame is discarded due to an overflow, RMFCR is incremented. However, RMFCR is not incremented for a receive frame that is cut off due to insufficient receive FIFO space. If a receive frame is cut off due to insufficient receive FIFO space (the frame is partially stored in the receive FIFO), the E-DMAC performs the following operation:

- Performs DMA transfers for the cut-off frame stored in the receive FIFO to the receive buffer.
- After the DMA transfer, performs a write-back operation on the receive descriptor.
- After the write-back operation, sets the ROC bit in EESR to 1 and generates an interrupt to the CPU.

When the receive FIFO is full of data waiting for DMA transfer, frame reception from the GMII/MII/RMII can be resumed if DMA transfer is performed from the receive FIFO to the receive buffer and 32 bytes or more of empty space is generated in the receive FIFO. When the number of receive frames waiting for DMA transfer is 24 in total, frame reception from the GMII/MII/RMII can be resumed if one or more frame has been DMA transferred from the receive

FIFO to the receive buffer. For restarting frame reception from the GMII/MII/RMII, when the E-DMAC resumes frame reception from the GMII/MII/RMII, it only accepts from the start of the frame.

(c) Flow Control

When the amount of receive data or the number of receive frames in the receive FIFO leads to one of the following conditions, the E-DMAC notifies the E-MAC to control E-MAC writing to the receive FIFO.

- When the space used in the receive FIFO exceeds the data amount specified by FCFTR
- When the number of receive frames in the receive FIFO exceeds the value specified by FCFTR

The threshold of the receive data amount can be set in a range from 256 to 65536 bytes in 256-byte units.

The threshold of receive frames can be set in a range from 1 to 24 frames (by the frame) in frame units.

(d) Receive Descriptor Empty

When the RACT bit of the read descriptor is 0 (invalid), the receive descriptor empty state is determined and DMA transfer is stopped. Then the following operation is performed.

- Writes the RR bit in EDRRR to 0
- Sets the RDE bit in EESR to 1 and generates an interrupt to the CPU.

To resume the DMA transfer to the receive buffer, the interrupt source needs to be cleared by software, the receive descriptor needs to be re-set and the RR bit in EDRRR should be set to 1.

Even if receive descriptor is empty, frame reception from the GMII/MII/RMII to the receive FIFO is continued if there is empty space left in the receive FIFO and receive frame information management area. Therefore, even if a receive descriptor empty state is determined, the DMA transfer can be performed without discarding the frames received from the GMII/MII/RMII if DMA transfer to the receive buffer can be resumed before an overflow occurs.

23.4.4 Relay

(1) Relay Procedure and Processing Flow

The GETHER has a function to relay frames received by either E-MAC0 or E-MAC1 to the other E-MAC. When relay is enabled, frames input from the E-MAC are sent to both relay FIFO and receive FIFO in the TSU, and determined independently whether to receive or not by the receive system and whether to relay or not by the relay system. To perform relay, both E-MAC controllers should be set as promiscuous mode, and the MAC address in both E-MAC controllers should be the same one (hereafter this MAC address is referred to as MAC address of this LSI).

The relay frame processing (relay/discard) is set by the TSU_FWSL0 and TSU_FWSL1. Frames passing the relay FIFO during relaying are sent to the GMII/MII/RMII from E-MAC-1 in a relay from E-MAC0 to E-MAC1, from E-MAC0 in a relay from E-MAC1 to E-MAC0. At this time, collision with the relay frames from the E-DMAC may occur. The priority of the process when collision occurs can be set by TSU_PRISL0/1. When the relay FIFO use exceeds the TSU_PRISL0/1 setting, frame transmission from the relay FIFO takes priority. By using this function, lost frames due to relay FIFO overflow can be prevented.

For multicast frames and frames their destinations are other than this LSI, the CAM evaluation in frame relay processing can be referenced (for details on the CAM function, refer to section 23.4.5, CAM Function). Table 23.4 shows the settings of the relay frame processing (without CAM).

Table 23.4 Relay Frame Process (Without CAM)

Frame Type	Relay Function Setting Register Bit	Frame Processing
Frame for this LSI	FW40/1 = 0	Discarded
	FW40/1 = 1	Relayed
Broadcast frame	FW30/1 = 0	Discarded
	FW30/1 = 1	Relayed
Multicast frame	FW20/1 = 0	Discarded
	FW20/1 = 1	Relayed
Frames having destinations other than this LSI	FW10/1 = 0	Discarded
	FW10/1 = 1	Relayed

23.4.5 CAM Function

Frames input to the E-MAC are grouped into the following four types; unicast for this LSI, broadcast, multicast, and unicast to other destinations. The MAC addresses of unicast for this LSI and broadcast are fixed, and determined only by register settings. Consequently, only multicast and unicast to other destinations determine whether to receive or not and whether to transfer or not by using the CAM (unicast frames whose destination MAC addresses match this LSI are called unicast frames to this LSI, and those that do not are called unicast frames to other destinations).

Furthermore, the evaluation of receive and relay of unicast to other destinations and multicast frames by using CAM are performed by referencing the registered MAC addresses of the CAM entry table in the TSU. By using this function, receive FIFO overflow can be prevented caused by accumulation of frame data not required for reception, and CPU processing for determining receive can be reduced.

The POST table is composed of 4 bits, and each bit corresponds to port 0 reception, port 1 reception, port 0 to port 1 relay, and port 1 to port 0 relay. When the corresponding bit is set to 1, the CAM evaluation results are used for determining receive and relay. In other words, when the corresponding bit of the POST table is cleared to 0, receive and relay evaluation will be the same as when CAM is not used shown in table 23.4.

The on-chip CAM has entry tables which can register the MAC address of 32 entries, the details of which can be set by TSU_ADRH0 to TSU_ADRH31 and TSU_ADRL0 to TSU_ADRL31. The setting to enable/disable referencing of the on-chip CAM entry table is performed by the CAM entry table enable setting register which sets whether to perform CAM evaluation or not, and the CAM entry table POST setting register for setting whether to use the CAM determination results for determining receive or relay. When on-chip CAM entry table referencing during receive is enabled, the destination address in the frame and MAC address registered in the CAM entry table are compared, and it is determined whether to transfer the frames input to the E-MAC to E-DMAC (have E-DMAC receive the frames) or discard the frames. When relaying and CAM entry table referencing during relay are both enabled, whether to transfer or discard multicast frames and frames for destinations other than this LSI can be determined by comparing the destination address in the frame and MAC address registered in the CAM entry table. Table 23.5 shows the processing method of frames (receive or discard) in reception from E-MAC0 to E-DMAC0 or that from E-MAC1 to E-DMAC1, while table 23.6 shows the processing for frames in relay from E-MAC0 to E-MAC1 or that from E-MAC1 to E-MAC0 (relay or discard).

Table 23.5 Receive Frame Processing

CAM Entry Table Referencing Results	Types of Frame	Normal Mode		Promiscuous Mode	
		MCT = 0	MCT = 1	MCT = 0	MCT = 1
CAM hit (when addresses match)	Frame to this LSI	Discarded		Discarded	
	Broadcast frame	Discarded		Discarded	
	Multicast frame	Discarded	Received	Discarded	Received
	Frames having destinations other than this LSI	Received		Discarded	
CAM mishit (when addresses do not match)	Frames to this LSI	Received		Received	
	Broadcast frame	Received		Received	
	Multicast frame	Received	Discarded	Received	Discarded
	Frames having destinations other than this LSI	Discarded		Received	

[Legend]

MCT (Bit 13 in ECMR): Multicast receive mode (0: Receive when CAM mishit/1: Receive when CAM hit)

Table 23.6 Relay Frame Process (With CAM)

Frame	Relay Function Setting Register Bit	CAM Hit	CAM Mishit
Multicast frame	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed
Frames having destinations other than this LSI	FW40/1 = 0	Relayed	Discarded
	FW40/1 = 1	Discarded	Relayed

Note: CAM can only be referenced by multicast frames and frames with destinations other than this LSI. The frames with destinations set to this LSI and broadcast frames are processed based on the relay function setting register values, regardless of CAM reference.

23.4.6 Transmit/Receive Processing of Multi-Buffer Frame (Single-Frame/Multi-Descriptor)

(1) Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in figure 23.12 is carried out by the E-DMAC.

In the figure where the transmit descriptor is shown as inactive (TACT bit = 0), buffer data has already been transmitted successfully, 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 is 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, 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 EESIPR, an interrupt is generated immediately after the final descriptor write-back.

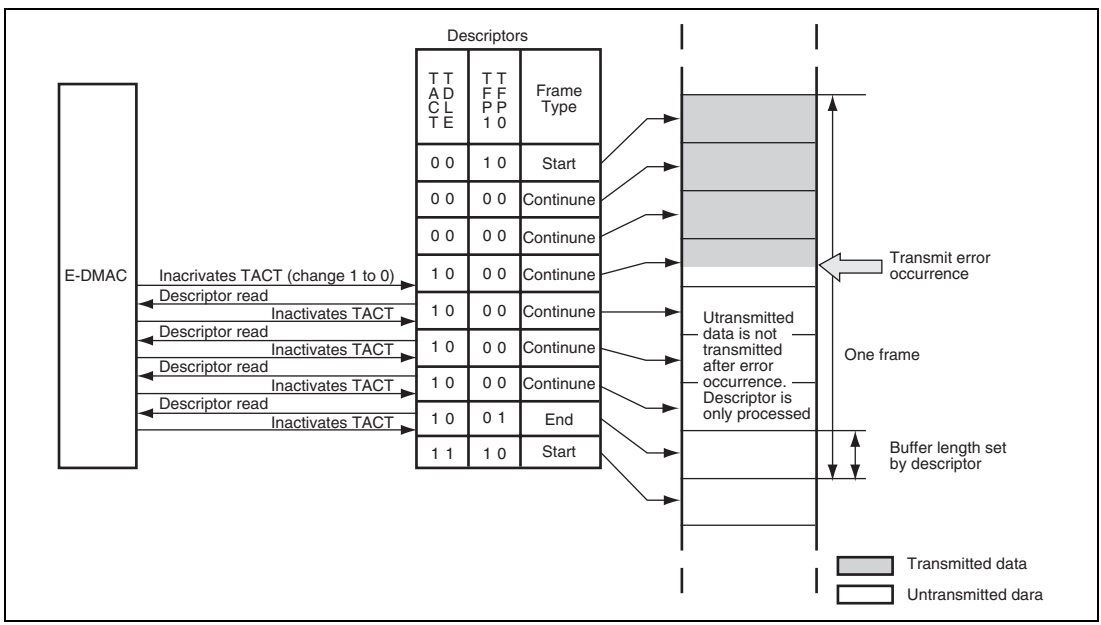


Figure 23.12 E-DMAC Operation after Transmit Error

(2) Receive Processing in the Case of Multi-Buffer Frame

If an error occurs during reception in the case of a multi-buffer frame where a receive frame is divided for storage in multiple buffers, the E-DMAC performs the processing shown in figure 23.13.

In the figure, the invalid receive descriptors (with the RACT bit cleared to 0) represent the successful reception of data to be stored in buffers, and the valid receive descriptors (with the RACT bit set to 1) represent unreceived buffers. If a frame receive error occurs with a descriptor shown in the figure, the status is written back to the corresponding descriptor.

If error interrupts are enabled in 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.

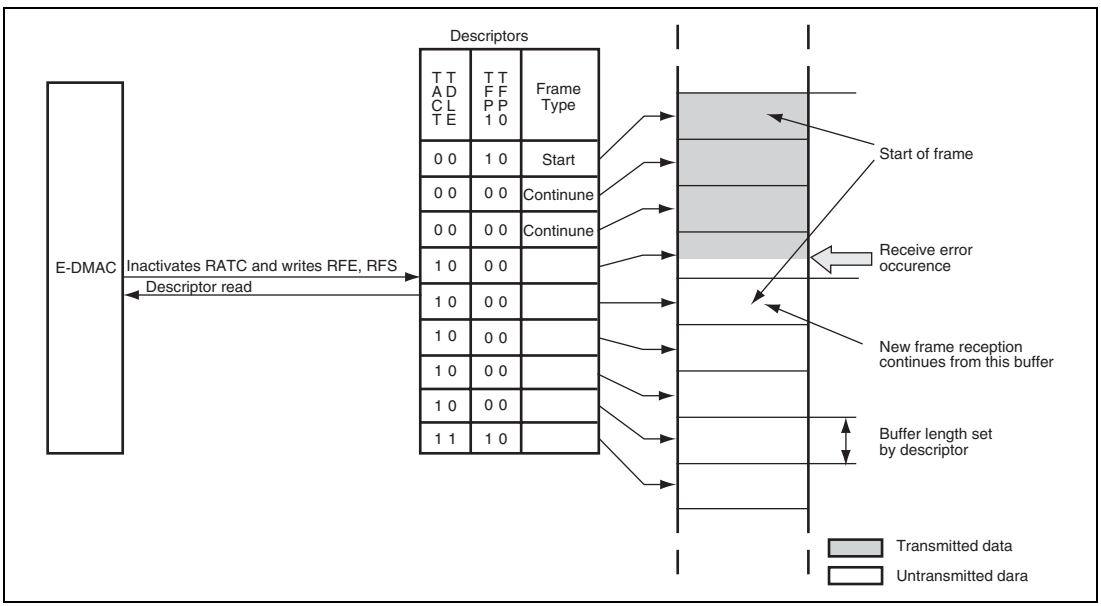


Figure 23.13 E-DMAC Operation after Receive Error

23.4.7 Padding Insertion in Receive Data

In the E-DMAC, one to three bytes of padding can be inserted in any byte position of receive data to improve software handling capability. By using this function, for instance, inserting 2-byte padding after the MAC header (14 bytes) of Ethernet frame enables data following the MAC header to set in 4-byte boundary.

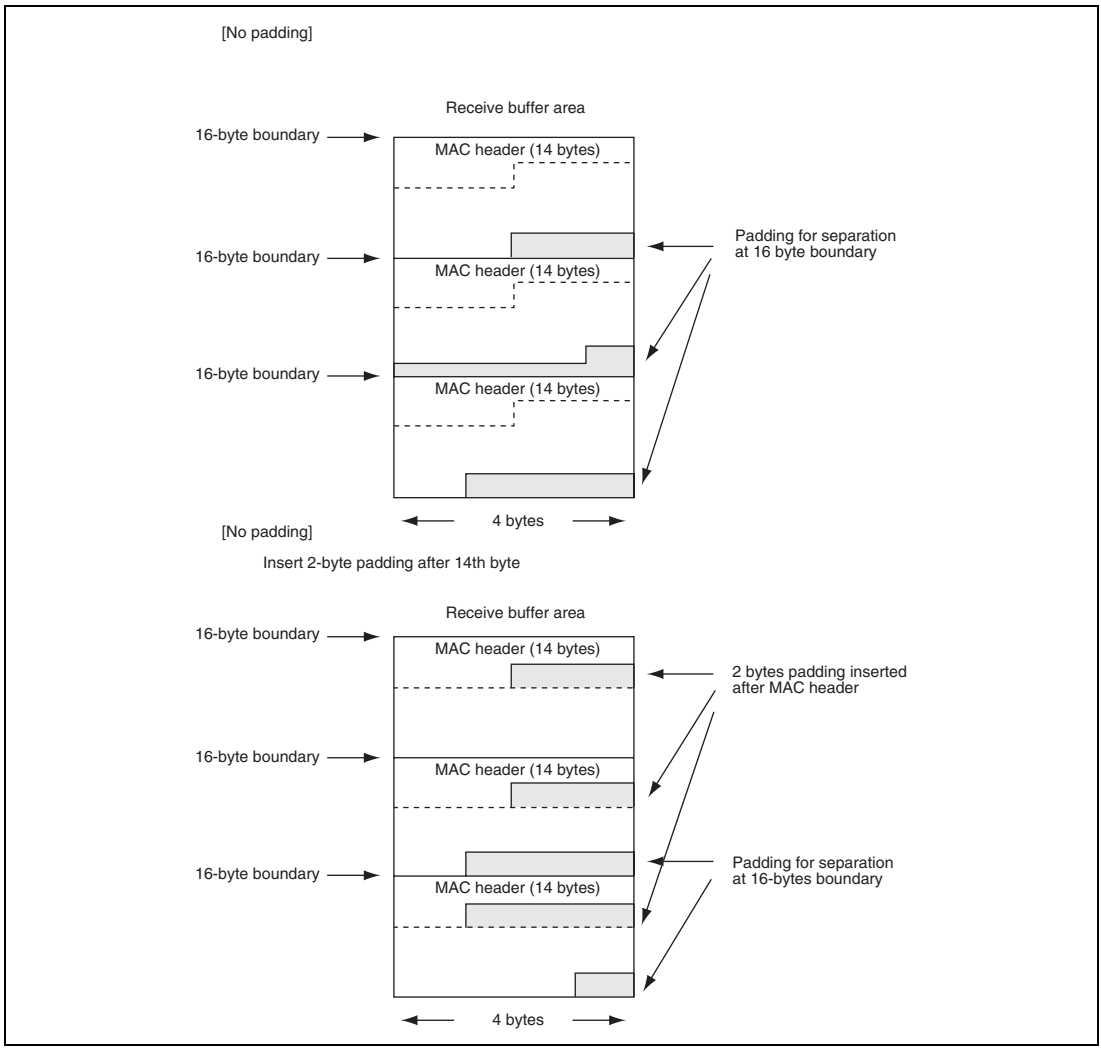


Figure 23.14 Padding Insertion in Receive Data

23.4.8 Interrupt Processing

(1) Interrupt Sources

The GETHER issues three types of interrupts to the CPU: receive/transmit interrupts for port 0 (GEINT0), receive/transmit interrupts for port 1 (GEINT1) and transfer interrupts between port 0 and port 1 (GEINT2). Table 23.7 shows these three interrupts, the interrupt sources, interrupt status registers/bits set at interrupt occurrence, and interrupt generation timing.

GEINT0 or GEINT1 interrupts are generated in correspondence with the port 0 or port 1 transmit/receive operation. When an interrupt source is generated, it is set in EESR0 or EESR1 and an interrupt is issued to the CPU. For some interrupt sources, the EESR0/EESR1 setting and an interrupt to the CPU are performed after a write-back operation to a descriptor is completed, not immediately after the interrupt source is detected. Interrupt sources other than the E-MAC status register source (ECI bit) are cleared by writing a 1 to the corresponding source bit. The E-MAC status register source (ECI bit) is cleared by writing a 1 to the corresponding source bit in ECSR. Interrupt source bits retain the values until they are cleared. GEINT0 or GEINT1 interrupt source is allowed to issue interrupts by setting the corresponding bit in EESIPR0 or EESIPR1. Each E-MAC state register source (ECI bit) is allowed to issue an interrupt by setting the corresponding bit in ECSIPR. In the initial value, interrupts are disabled.

GEINT2 interrupt is issued in correspondence with relay operation between port 1 and port 0. When an interrupt source is generated, it is set to the corresponding bit in TSU_FWSR and an interrupt is issued to the CPU. Each GEINT2 interrupt source is cleared by writing a 1 to the corresponding bit. The interrupt source bit retains the value until it is cleared. Each GEINT2 interrupt source is allowed to issue an interrupt by setting the corresponding bit in TSU_FWSR. In the initial state, interrupts are disabled.

Table 23.7 shows these three interrupts, interrupt sources, interrupt status registers and bits set at interrupt occurrence and interrupt generation timing.

Table 23.7 List of GETHER Interrupts

Interrupt	Interrupt Source	Register and Bit	Interrupt Generation Timing
Transmit/ receive interrupt for port 0 (GEINT0)	Write-back completed	EESR0.TWB	After write-back
	Transmit underflow frame write-back completed	EESR0.TUC	After write-back
	Receive underflow frame write-back completed	EESR0.ROC	After write-back
	Transmission abort detection	EESR0.TABT	After write-back
	Reception abort detection	EESR0.RABT	After write-back
	Receive frame counter overflow	EESR0.RFCOF	When the interrupt source is detected
	E-MAC status register source	EESR0.ECI	When the interrupt source is detected
	Frame transmission completed	EESR0.TUC	After write-back
	Transmit descriptor empty	EESR0.TDE	When the interrupt source is detected
	Transmit FIFO underflow	EESR0.TFUF	When the interrupt source is detected
	Frame reception	EESR0.FR	After write-back
	Receive descriptor empty	EESR0.RDE	When the interrupt source is detected
	Receive FIFO overflow	EESR0.RFOF	When the interrupt source is detected
	Detect Loss of Carrier	EESR0.DLC	When the interrupt source is detected
	Delayed Collision Detect	EESR0.CD	When the interrupt source is detected
	Transmit Retry Over	EESR0.TRO	When the interrupt source is detected

Interrupt	Interrupt Source	Register and Bit	Interrupt Generated Timing
Transmit/ receive interrupt for port 0 (GEINT0)	Receive Multicast Address Frame	EESR0.RMAF	After write-back
	Carrier Extension Error	EESR0.CEEF	After write-back
	Carrier Extension Loss	EESR0.CELF	After write-back
	Receive Residual-Bit Frame	EESR0.RRF	After write-back
	Receive Too-Long Frame	EESR0.RTLF	After write-back
	Receive Too-Short Frame	EESR0.RTSF	After write-back
	PHY-LSI Receive Error	EESR0.PRE	After write-back
	CRC Error on Received Frame	EESR0.CERF	After write-back
Transmit/ receive interrupt for port 1 (GEINT1)	Write-Back Completed	EESR1.TWB	After write-back
	Transmit Underflow Frame Write-Back Completed	EESR1.TUC	After write-back
	Receive Overflow Frame Write-Back Completed	EESR1.ROC	After write-back
	Transmit Abort Detect	EESR1.TABT	After write-back
	Receive Abort Detect	EESR1.RABT	After write-back
	Receive Frame Counter Overflow	EESR1.RFCOF	When the interrupt source is detected
	E-MAC Status Register Source	EESR1.ECI	When the interrupt source is detected
	Frame Transmission Completed	EESR1.TUC	After write-back
	Transmit Descriptor Empty	EESR1.TDE	When the interrupt source is detected
	Transmit FIFO Underflow	EESR1.TFUF	When the interrupt source is detected
	Frame Reception	EESR1.FR	After write-back
	Receive Descriptor Empty	EESR1.RDE	When the interrupt source is detected
	Receive FIFO Overflow	EESR1.RFOF	When the interrupt source is detected
	Carrier Loss Detection	EESR1.DLC	When the interrupt source is detected
	Delayed Collision Detect	EESR1.CD	When the interrupt source is detected

Interrupt	Interrupt Source	Register and Bit	Interrupt Generated Timing
Transmit/ receive interrupt for port 1 (GEINT1)	Transmit Retry Over	EESR1.TRO	When the interrupt source is detected
	Receive Multicast Address Frame	EESR1.RMAF	After write-back
	Carrier Extension Error	EESR1.CEEF	After write-back
	Carrier Extension Loss	EESR1.CELF	After write-back
	Receive Residual-Bit Frame	EESR1.RRF	After write-back
	Receive Too-Long Frame	EESR1.RTLF	After write-back
	Receive Too-Short Frame	EESR1.RTSF	After write-back
	PHY-LSI Receive Error	EESR1.PRE	After write-back
	CRC Error on Received Frame	EESR1.CERF	After write-back
Interrupt with transfer between port 0 and port 1 (GEINT2)	Port 0-to-1 Transfer FIFO Overflow Detect	TSU_FWSR.OVF0	When the interrupt source is detected
	E-MAC-0 Overflow Alert Signal Output	TSU_FWSR.RBSY0	When the interrupt source is detected
	E-MAC-0 Carrier Extension Loss Error Detect	TSU_FWSR.RINT60	When the interrupt source is detected
	E-MAC-0 Residual-Bit Frame Receive	TSU_FWSR.RINT50	When the interrupt source is detected
	E-MAC-0 Too-Long Frame Receive	TSU_FWSR.RINT40	When the interrupt source is detected
	E-MAC-0 Too-Short Frame Receive	TSU_FWSR.RINT30	When the interrupt source is detected
	E-MAC-0 Frame Receive Error	TSU_FWSR.RINT20	When the interrupt source is detected
	E-MAC-0 CRC Error Frame Receive	TSU_FWSR.RINT10	When the interrupt source is detected
	Port 1-to-0 Transfer FIFO Overflow Detect	TSU_FWSR.OVF1	When the interrupt source is detected
	E-MAC-1 Overflow Alert Signal Output	TSU_FWSR.RBSY1	When the interrupt source is detected
	E-MAC-1 Carrier Extension Loss Error Detect	TSU_FWSR.RINT61	When the interrupt source is detected
	E-MAC-1 Residual-Bit Frame Receive	TSU_FWSR.RINT51	When the interrupt source is detected

Interrupt	Interrupt Source	Register and Bit	Interrupt Generated Timing
Interrupt with transfer between port 0 and port 1 (GEINT2)	E-MAC-1 Too-Long Frame Receive	TSU_FWSR.RINT41	When the interrupt source is detected
	E-MAC-1 Too-Short Frame Receive	TSU_FWSR.RINT31	When the interrupt source is detected
	E-MAC-1 Frame Receive Error	TSU_FWSR.RINT21	When the interrupt source is detected
	E-MAC-1 CRC Error Frame Receive	TSU_FWSR.RINT11	When the interrupt source is detected

23.4.9 Activation Procedure

The GETHER should be activated by the following procedure:

(1) Reset

1. Perform a power-on reset.
2. Start the E-DMAC transmitter and receiver (activation of descriptor engine).

- Set ENT to 1 and ENR to 1 in EDSR.

3. Perform a software reset.

- Set SWRR to 1 and SWRT to 1 in EDMR simultaneously.

4. Initialize the descriptor entry table.
5. Confirm cancellation of the software reset.

- Check that the SWRR and SWRT bits in EDMR are cleared to 0.

(2) Registration of Descriptor Ring

The address of a descriptor ring configured in memory is registered in the descriptor entry table.

1. Transmit Descriptor Setting

- Set TDLAR.
- Set TDFAR.
- Set TDFXR.
- Set TDFFR. When the descriptor indicated by TDFXR is the last descriptor in the descriptor list, set H'00000001.

2. Receive Descriptor Setting
- Set RDLAR.
- Set RDFAR.
- Set RDFXR.
- Set RDFFR. When the descriptor indicated by RDFXR is the last descriptor in the descriptor list, set H'00000001.

(3) Register Settings

The following registers should be set as necessary.

1. E-DMAC-related registers
- Set EDMR: Operating mode, etc.
- Set EESIPR: Interrupt masks
- Set TRSCER: Error masks
- Set TFTR: Transmit FIFO threshold
- Set FDR: External FIFO size
- Set RMCR: Reset method for reception activation
- Set RPADIR: Padding insertion into receive data
- Set FCFTR: Receive BSY output threshold

2. E-MAC-related registers
- Set ECMR setting: Transmission/reception specifications
- Set ECSIPR setting: Interrupt masks
- Set MAHR: MAC address
- Set MALR: MAC address
- Set RFLR: Maximum receive frame length
- Set PIPR: ET_PHY_INT pin polarity
- Set APR: TIME parameter value of an automatic pause frame
- Set MPR: TIME parameter value of a manual PAUSE frame
- Set TPAUSER: Upper limit of automatic PAUSE frame retransmission
- Set GECMR: Transfer speed
- Set BCULR: Upper limit of burst cycles

(4) Activation

1. Start the E-DMAC transmission/reception function
- Set the TR bits in EDTRR to 11.
- Set the RR bit in EDRRR to 1.

2. Start the E-MAC transmission/reception function
- Set the TE and RE bits in ECMR to 1.

23.4.10 Flow Control

The GETHER supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control can be applied to both receive and transmit operations. When transmitting PAUSE frames, flow control can be performed by the following two procedures :

(1) Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the number of data written to the receive FIFO reaches the value set in FCFTR. The TIME parameter included in the PAUSE frame is set by APR. The automatic PAUSE frame transmission is repeated until the number of data in the receive FIFO becomes less than the value set in FCFTR as the receive data is read from the FIFO. Using TPAUSER, the upper limit of retransmission counts of the PAUSE frames can also be set in the range from 1 to 65535. In this case, PAUSE frame transmission is repeated until the number of receive FIFO data becomes less than the FCFTR value, or the number of transmits reaches the value set by TPAUSER. The transmission counter is cleared to 0 when the next PAUSE frame is transmitted after the number of data in the receive FIFO becomes less than the FCFTR value.

The automatic PAUSE frame transmission is enabled when the TXF bit in ECMR is 1.

(2) Manual PAUSE Frame Transmission

PAUSE frames are transmitted by directives from the software. When writing the Timer value to MPR, manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

(3) 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 ECMR is set to 1. The number of times of PAUSE frame receptions is counted.

(4) 0-Time PAUSE Frame Control

Flow control is performed using a PAUSE frame with the TIME parameter value set to 0. The PAUSE frame with the TIME parameter set to 0 can be enabled or disabled by the ZPF bit in ECMR.

- When PAUSE frame control with the TIME parameter value set to 0 is enabled
  A PAUSE frame with the TIME parameter 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 TIME parameter value has not elapsed. When a PAUSE frame with the time indicated by the TIME parameter value set to 0 is received, the transmit standby state is canceled.
- When PAUSE frame control with the TIME parameter value set to 0 is disabled
  A PAUSE frame with the TIME parameter value set to 0 is not transmitted. When a PAUSE frame with the TIME parameter value set to 0 is received, the PAUSE frame is discarded.

23.4.11 Magic Packet Detection

The GETHER has a Magic Packet detection function. This function provides a Wake-On-LAN (WOL) facility that starts each peripheral device connected to a LAN from the host device or other source. This enables to construct a system in which a peripheral device receives a Magic Packet sent from the host device or other source, and starts itself. When the Magic Packet is detected, data is stored in the FIFO by the broadcast packet that has received data previously and the E-MAC is notified of the receiving status. To return to normal operation from the interrupt processing, the E-MAC, TSU and E-DMAC must be initialized by using ARST bit in ARSTR.

With a Magic Packet, reception is performed regardless of the destination address. As a result, this function is valid, and the ET_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 MPDE bit in ECMR.
3. Set the MPDIP bit in ECSIPR to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set peripheral modules to module standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The ET_WOL pin notifies peripheral LSIs that the Magic Packet has been detected.

23.4.12 Direction for IEEE802.1Q Qtag

The GETHER supports IEEE802.1Q frame processing. It can add or delete Qtags to or from frames processed in relay. This function can also transmit and receive QoS frames. During relay, if the Ethernet device connected to one E-MAC controller cannot transmit or receive QoS frames, the frames can converted to the normal IEEE802.3 frames and relayed in this LSI. Whether to add or delete Qtags depends on TSU_QTAGM0/1. When the Qtag is added, the Qtag to be added can be set by TSU_ADQT0/1. Figure 23.15 shows the outlines of the Qtag add function. Figure 23.16 shows the comparison between the normal Ethernet frames and IEEE802.1Q frames (with Qtag). For details on Qtag setting, see the specifications on Qtag control specified in IEEE802.1Q

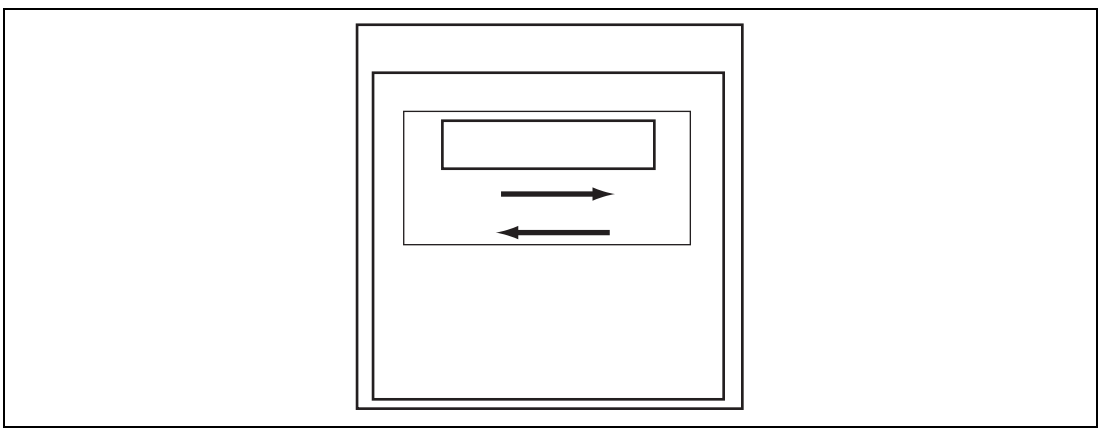


Figure 23.15 Outlines of Qtag Additional Functions

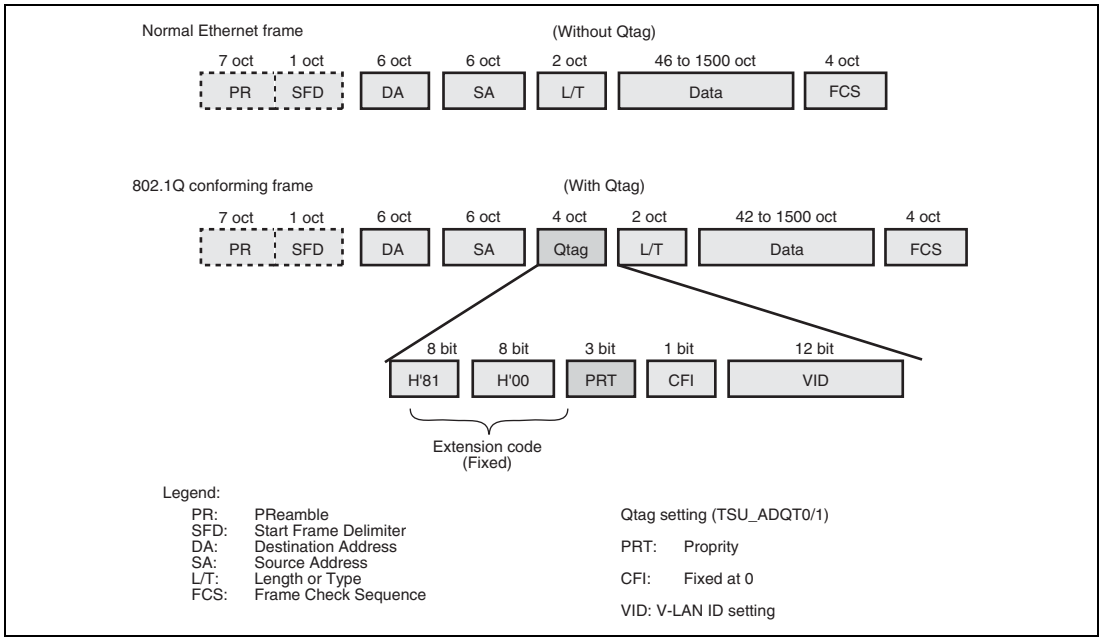


Figure 23.16 Comparison of Normal Ethernet Frame and IEEE802.1Q Frame (with Qtag)

23.5 Connection to PHY-LSI

23.5.1 MII Frame Transmission/Reception Timing

Each MII frame transmission/reception timing is shown in figures 23.17 to 23.22.

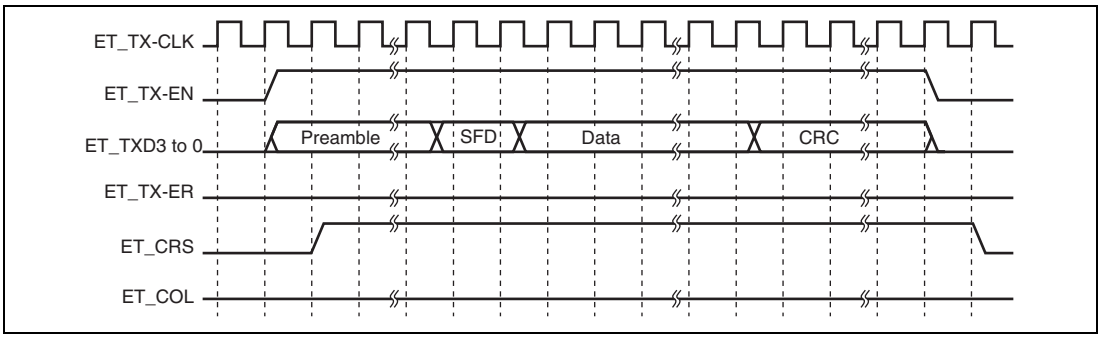


Figure 23.17 MII Frame Transmit Timing (Normal Transmission)

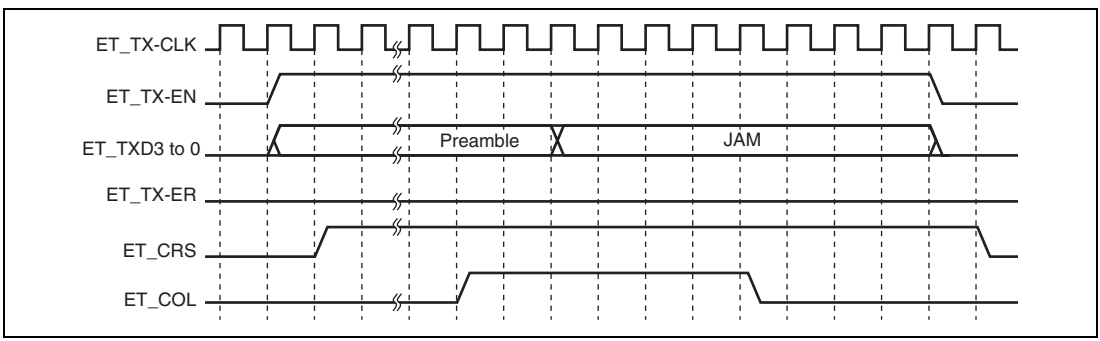


Figure 23.18 MII Frame Transmit Timing (Collision)

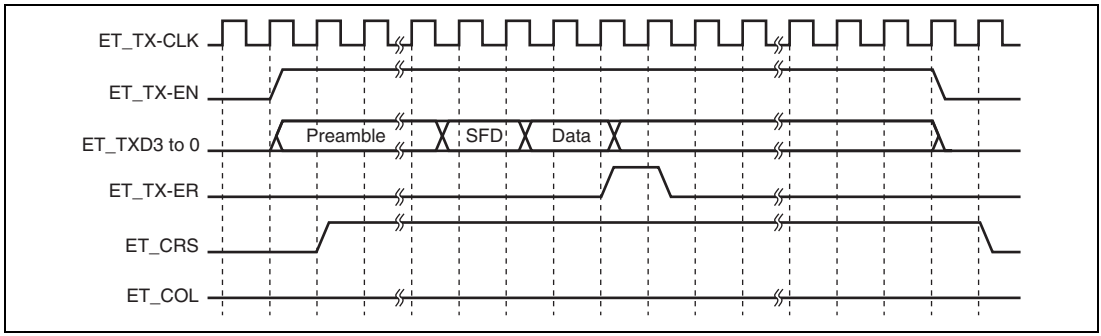


Figure 23.19 MII Frame Transmit Timing (Transmit Error)

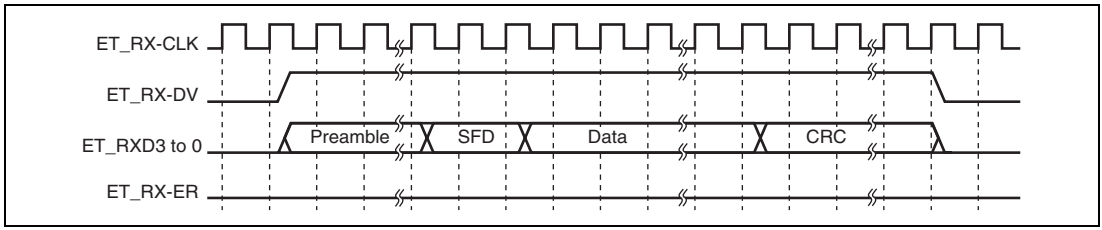


Figure 23.20 MII Frame Receive Timing (Normal Reception)

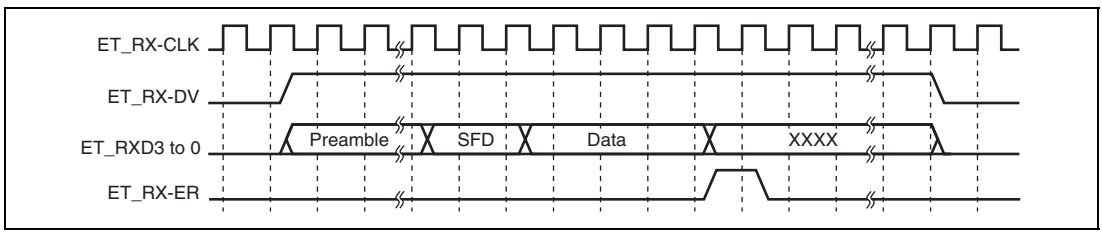


Figure 23.21 MII Frame Receive Timing (Reception Error (1))

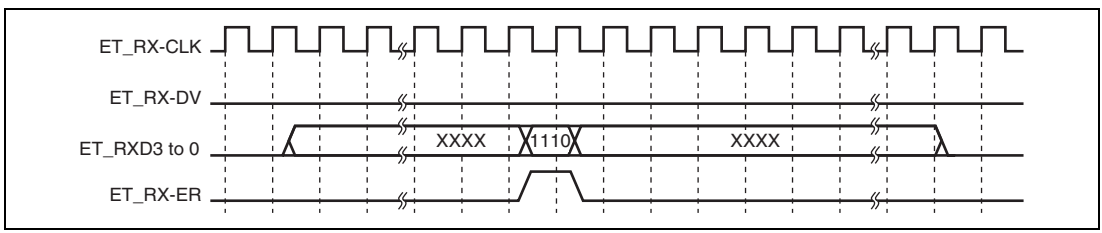


Figure 23.22 MII Fame Receive Timing (Reception Error (2))

23.5.2 GMII/MII Frame Reception Timing

Each GMII/MII frame reception timing is shown in figures 23.23 to 23.28.

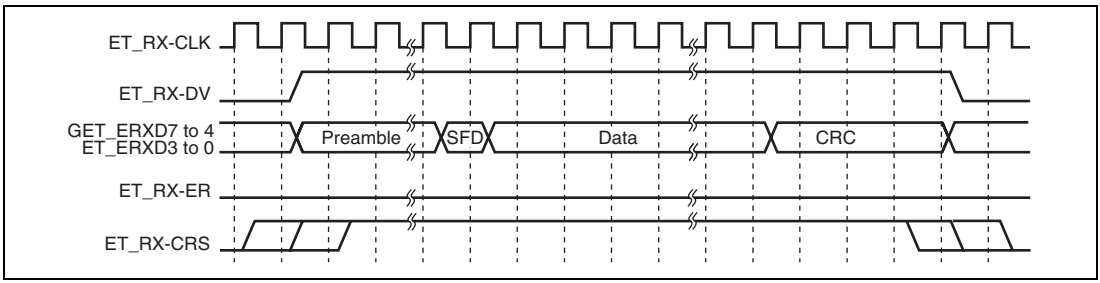


Figure 23.23 GMII/MII Fame Receive Timing (Normal Reception)

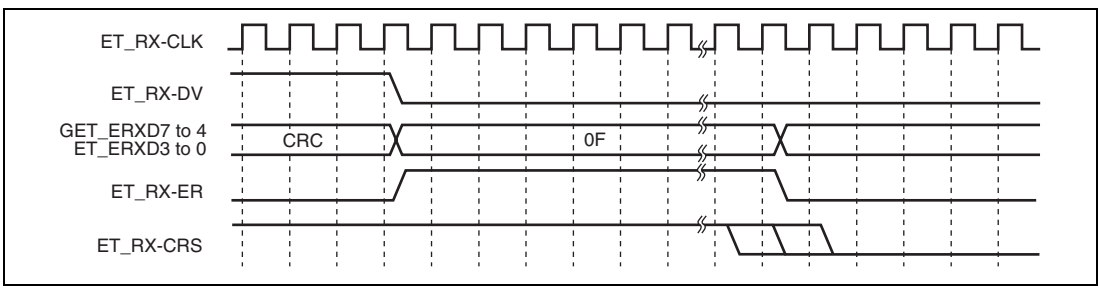


Figure 23.24 GMII/MII Fame Receive Timing (with Carrier Extension)

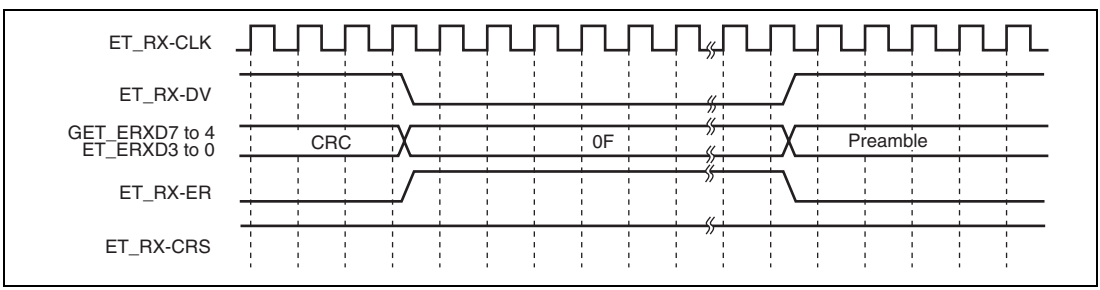


Figure 23.25 GMII/MII Fame Receive Timing (Burst Reception)

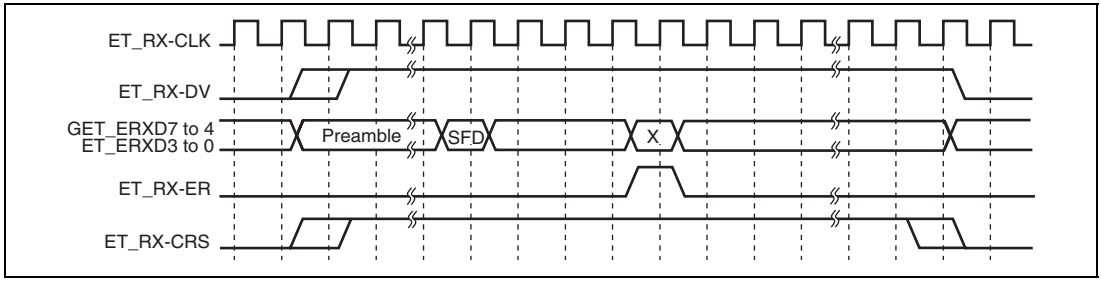


Figure 23.26 GMII/MII Fame Receive Timing (Reception Error)

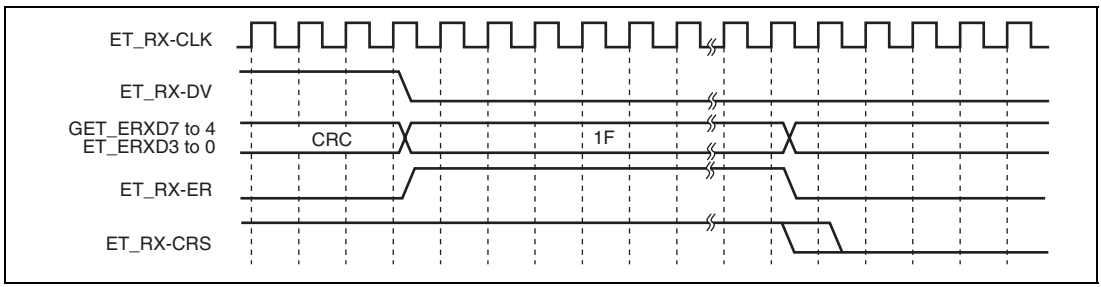


Figure 23.27 GMII/MII Fame Receive Timing (Error with Carrier Extension)

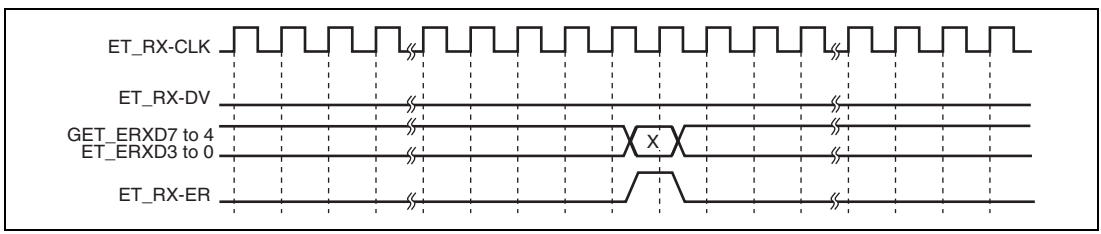


Figure 23.28 GMII/MII Fame Receive Timing (False Carrier Indication)

23.5.3 RMII Frame Transmission/Reception Timing

Each RMII frame transmission/reception timing is shown in figures 23.29 to 23.31.

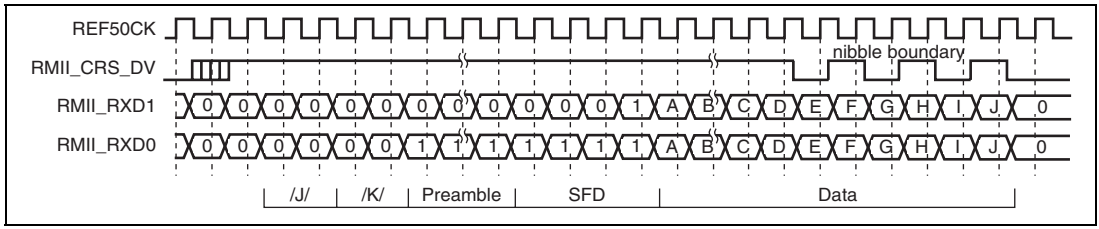


Figure 23.29 RMII Fame Receive Timing (Normal 100-Mbps Reception)

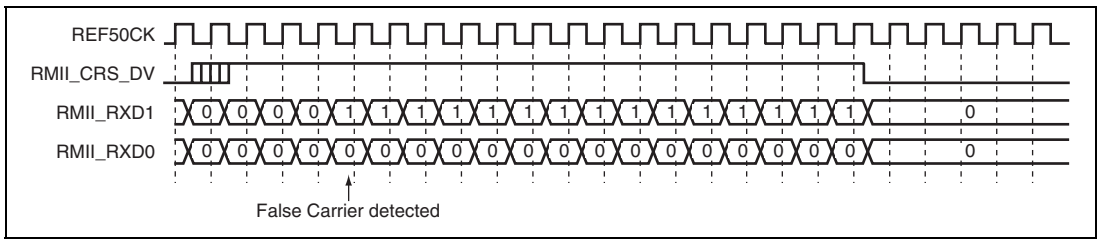


Figure 23.30 RMII Fame Receive Timing (100-Mbps Reception with Illegal Carrier Detected)

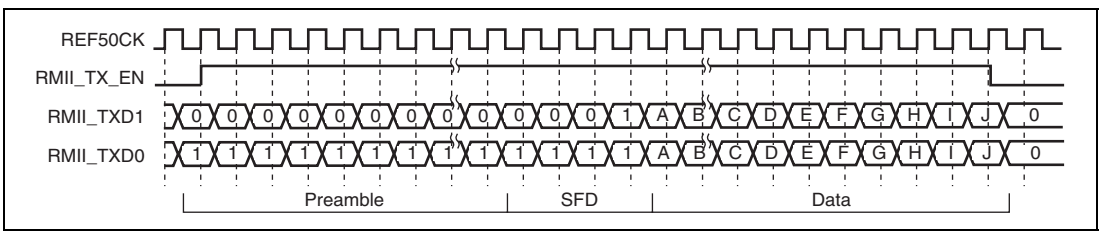


Figure 23.31 RMII Fame Transmit Timing (Normal 100-Mbps Transmission)

23.5.4 Accessing MII Registers

MII registers in the PHY-LSI are accessed via PIR in this LSI. PIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

(1) MII Management Frame Format

Figure 23.32 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI address.
REGAD: Write of 000q if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) perdormed
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) performad
(d) Read: Bus already released in TA: control unnecessary

Figure 23.32 MII Management Frame Format

(2) MII Register Access Procedure

The program accesses MII registers via 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 23.33 shows the MII register access timing. The timing will differ depending on the PHY-LSI type.

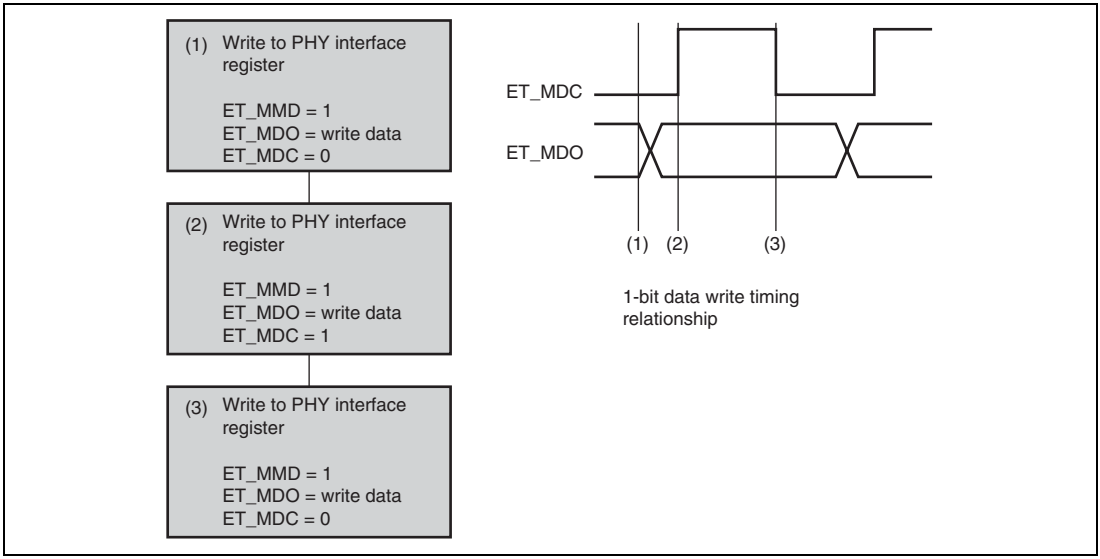


Figure 23.33 1-Bit Data Write Flowchart

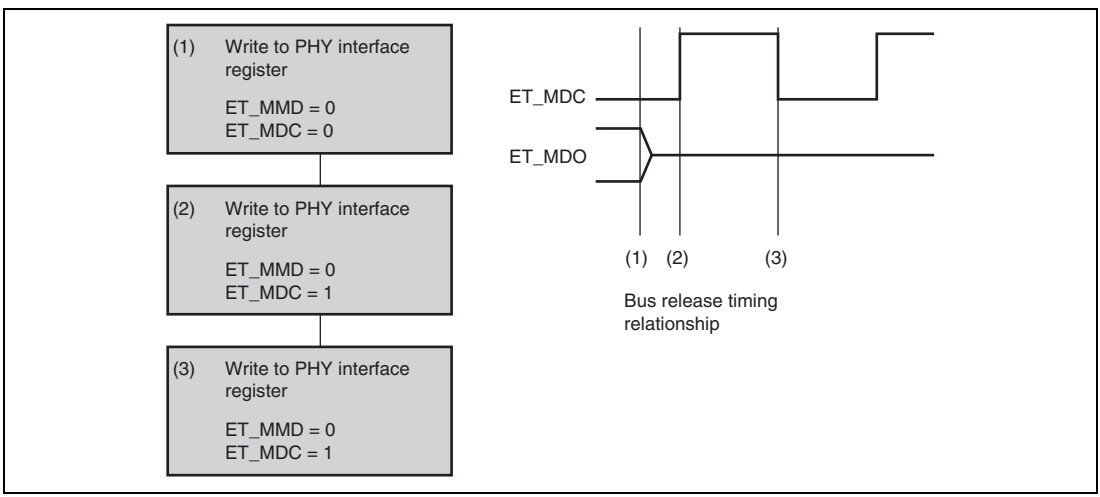


Figure 23.34 Bus Release Flowchart (TA in Read in Figure 23.33)

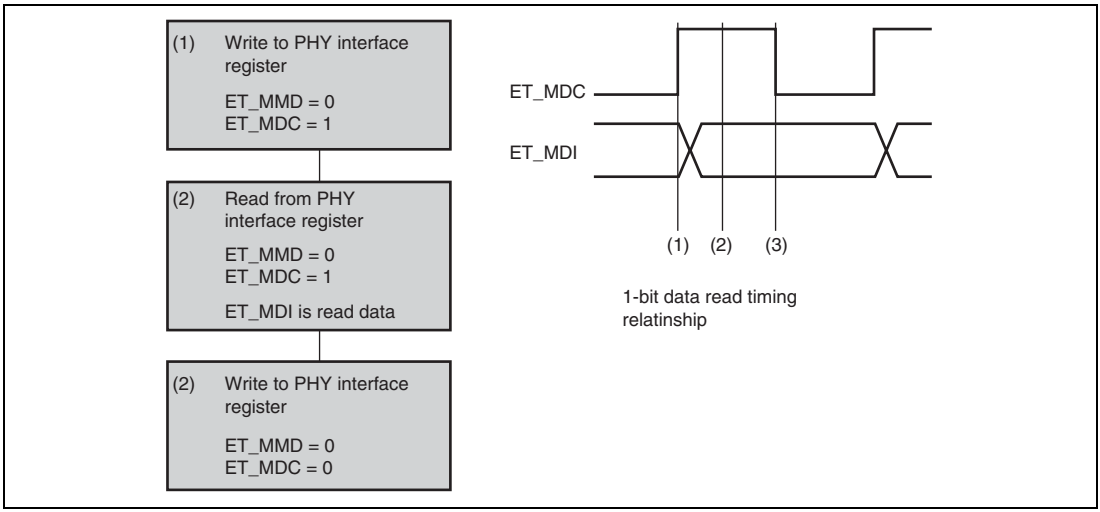


Figure 23.35 1-Bit Data Read Flowchart

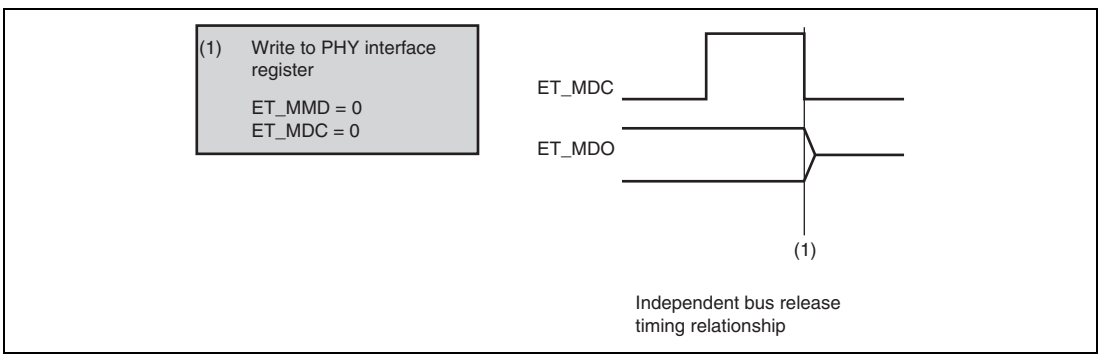


Figure 23.36 Independent Bus Release Flowchart (IDLE in Write in Figure 23.33)

23.5.5 Mll-RMII Interface Conversion

This LSI supports an RMII interface. The RMII signals are generated by converting the MII signals in the MII-RMII conversion circuit.

(1) Clock

REF50CK (50 MHz) from the RMII interface is divided and ET_TX-CLK/ET_RX-CLK (25 MHz or 2.5 MHz) is output.

(2) Reception

Waveforms received from the RMII interface are converted to MII waveforms and output (10 Mbps or 100 Mbps). Illegal carrier detection signal received from the RMII interface is converted to MII signal and output. RMII_RX-ER signal received from the RMII interface is converted to MII interface signal and output.

Note: Illegal carrier detection is not generated from preamble detection to reception completion (ET_RX_DV negation).

(3) Transmission

Transmit waveforms from the MII interface is converted to the RMII interface waveforms and output (10 Mbps or 100 Mbps). The collision signal, ET_COL, is generated by AND operation of the ET_CRS and ET_TX-EN signals.

(4) Full-Duplex/Half-Duplex Selection

In full-duplex transfer mode, the assertion of the COL is suppressed. Figure 23.37 shows a schematic of the conversion cicuit.

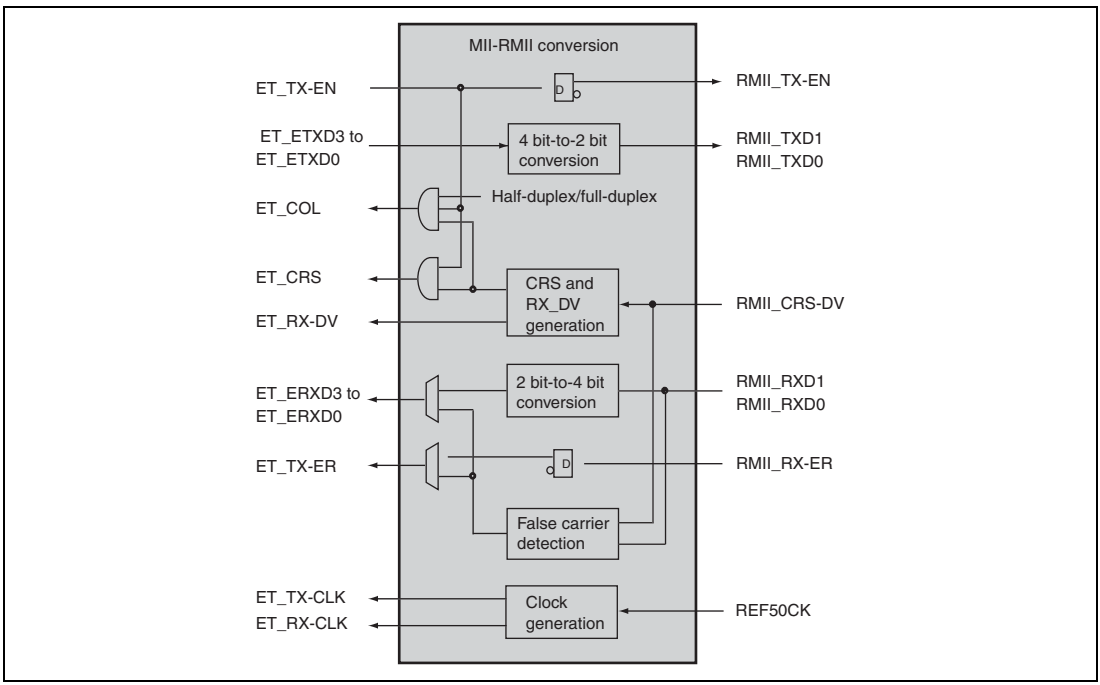


Figure 23.37 MII-RMll Conversion Circuit

23.6 Usage Notes

23.6.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. Calculation involves 16-bit addition only; it does not involve bit reversal.

Note: Also for the frames with VLANTag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

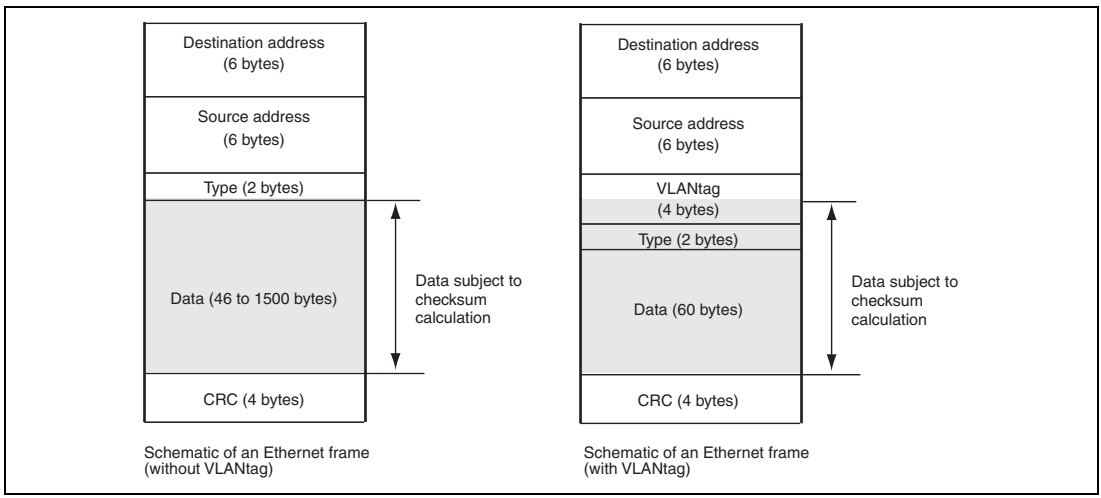


Figure 23.38 Data Subject to Checksum Calculation

23.6.2 Notes on TSU Use

The TSU of this LSI supports up to 100BASE-T data transfers. Therefore, even when The TSU of this LSI is used with 1000BASE-T, the transfer performance is equal to that with 100BASE-T.

Section 24 IP Security Accelerator (SECURITY)

This section will be made available on conclusion of a nondisclosure agreement.

For details, contact your Renesas Technology sales agency.

Section 25 Stream Interface (STIF)

The stream interface (STIF) transfers stream data between an 8-bit parallel bus and external memory using general DMAC peripheral module requests (transfer size is fixed at 16 bytes).

25.1 Features

- Number of parallel stream data transfer channels: 2 channels
- Stream data transfer interface
  Clock valid reception
  Strobe reception
  Clock valid transmission
  Strobe transmission
- Input/output packet length: 188 or 192 bytes is selectable
  The external pin input or peripheral clock 0 (Pck0) can be selected as the stream data transfer clock source.
- Transmit/receive FIFO size: 768 bytes
- Time stamp adding function
  Includes a free-running timer for time stamp.
  (The free-running timer input clock can be selected from among 1/2, 1/4, and 1/8 of peripheral clock 0.)
  At reception: The free-running timer value is added to the receive packet as the time stamp value and stored in memory.
  At transmission: A packet is transmitted with the free-running timer value added as the time stamp value.
- DMA transfer
  Data transfer with external memory by means of DMA transfer is supported.

Figure 25.1 shows a block diagram of the STIF.

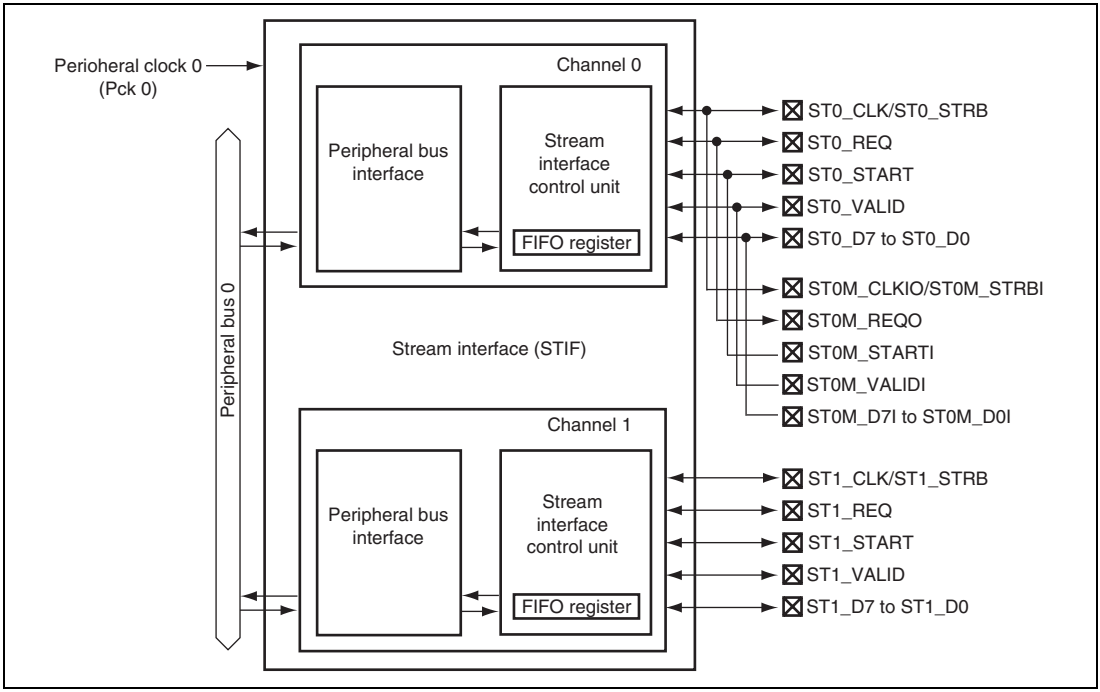


Figure 25.1 Block Diagram of STIF

25.2 Input/Output Pins

Table 25.1 shows the pin configuration of this module. Channel 0 has two pin groups: normal I/O pins and mirror input pins. Note that the mirror input pin group can only be used for input. The pin select register of the PFC is used to select normal I/O pins or mirror input pins. The normal I/O pins and mirror input pins cannot be used simultaneously or mixed together.

Table 25.1 Pin Configuration

Channel		Pin Name	I/O	Function	Description
0	Normal I/O pins	ST0_CLK/ST0_STRB	I/O	Stream data clock/strobe	
		ST0_REQ	I/O	Stream data receive ready request	
		ST0_START	I/O	Stream data synchronization	
		ST0_VALID	I/O	Stream data valid	
		ST0_D7 to ST0_D0	I/O	Stream data input/output	
	Mirror input pins*	ST0M_CLKIO/ ST0M_STRBI	I/O	Stream data clock/strobe	
		ST0M_REQO	Output	Stream data receive ready request	
		ST0M_STARTI	Input	Stream data synchronization input	
		ST0M_VALIDI	Input	Stream data valid input	
		ST0M_D7I to ST0M_D0I	Input	Stream data input	
1		ST1_CLK/ST1_STRB	I/O	Stream data clock/strobe	
		ST1_REQ	I/O	Stream data receive ready request	
		ST1_START	I/O	Stream data synchronization	
		ST1_VALID	I/O	Stream data valid	
		ST1_D7 to ST1_D0	I/O	Stream data input	

Note: * Mirror pins are only for input.

25.3 Register Descriptions

Table 25.2 shows the STIF register configuration. Table 25.3 shows the register states in each operating mode.

Table 25.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
Mode register 0	STIMDR0	R/W	H'FFEE 0000	H'1FEE 0000	32
Control register 0	STICR0	R/W	H'FFEE 0004	H'1FEE 0004	32
Interrupt status register 0	STIISR0	R/W	H'FFEE 0008	H'1FEE 0008	32
Interrupt enable register 0	STIIER0	R/W	H'FFEE 000C	H'1FEE 000C	32
Time stamp counter register 0	STITSC0	R/W	H'FFEE 0010	H'1FEE 0010	32
Transmit/receive packet count register 0	STIPNR0	R/W	H'FFEE 0018	H'1FEE 0018	32
Transmit/receive packet counter register 0	STIPCR0	R/W	H'FFEE 0014	H'1FEE 0014	32
Transmit/receive FIFO data register 0	STIFIFO0	R/W	H'FFEE 0400	H'1FEE 0400	32
Mode register 1	STIMDR1	R/W	H'FFEE 8000	H'1FEE 8000	32
Control register 1	STICR1	R/W	H'FFEE 8004	H'1FEE 8004	32
Interrupt status register 1	STIISR1	R/W	H'FFEE 8008	H'1FEE 8008	32
Interrupt enable register 1	STIIER1	R/W	H'FFEE 800C	H'1FEE 800C	32
Time stamp counter register 1	STITSC1	R/W	H'FFEE 8010	H'1FEE 8010	32
Transmit/receive packet count register 1	STIPNR1	R/W	H'FFEE 8018	H'1FEE 8018	32
Transmit/receive packet counter register 1	STIPCR1	R/W	H'FFEE 8014	H'1FEE 8014	32
Transmit/receive FIFO data register 1	STIFIFO1	R/W	H'FFEE 8400	H'1FEE 8400	32

Table 25.3 Register States in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Mode register 0	STIMDR0	H'00000000	H'00000000	Retained	Retained
Control register 0	STICR0	H'00000000	H'00000000	Retained	Retained
Interrupt status register 0	STIISR0	H'00000000	H'00000000	Retained	Retained
Interrupt enable register 0	STIIER0	H'00000000	H'00000000	Retained	Retained
Time stamp counter register 0	STITSC0	H'00000000	H'00000000	Retained	Retained
Transmit/receive packet count register 0	STIPNR0	H'00000000	H'00000000	Retained	Retained
Transmit/receive packet counter register 0	STIPCR0	H'00000000	H'00000000	Retained	Retained
Transmit/receive FIFO data register 0	STIFIFO0	H'00000000	H'00000000	Retained	Retained
Mode register 1	STIMDR1	H'00000000	H'00000000	Retained	Retained
Control register 1	STICR1	H'00000000	H'00000000	Retained	Retained
Interrupt status register 1	STIISR1	H'00000000	H'00000000	Retained	Retained
Interrupt enable register 1	STIIER1	H'00000000	H'00000000	Retained	Retained
Time stamp counter register 1	STITSC1	H'00000000	H'00000000	Retained	Retained
Transmit/receive packet count register 1	STIPNR1	H'00000000	H'00000000	Retained	Retained
Transmit/receive packet counter register 1	STIPCR1	H'00000000	H'00000000	Retained	Retained
Transmit/receive FIFO data register 1	STIFIFO1	H'00000000	H'00000000	Retained	Retained

25.3.1 Mode Registers 0, 1 (STIMDR0, STIMDR1)

STIMDR sets the STIF operating mode and clock definition for stream data transmission/reception.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MD[2:0]			—	—	—	PLEN	—	—	STMP[1:0]		—	—	WORK[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CKSL	—	CKDV[1:0]		—	—	—	REQ EN	—	—	FRC[1:0]		STRB	REQ	VLD	STAT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	MD[2:0]	000	R/W	Stream Data Transfer Interface 000: Clock valid reception 010: Strobe reception 100: Clock valid transmission 101: Strobe transmission Other than above: Setting prohibited
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	PLEN	0	R/W	Transmit/Receive Packet Length Sets the packet length of the stream data to be transmitted or received. 0: Packet length is 188 bytes 1: Packet length is 192 bytes
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
21, 20	STMP[1:0]	00	R/W	Time Stamp At reception: These bits select whether to add a fixed value, add the time stamp, or do not add any value when transferring the receive packet to external memory. 00: Adds a fixed value to the receive packet and transfers it to external memory 01: Adds the time stamp to the receive packet and transfers it to external memory 10: Transfers the receive packet to external memory without any changes (only when the packet size is 192 bytes) 11: Setting prohibited At transmission: These bits select the packet interval for transmitting the transmit packet. 00: Packet interval is in accordance with the ICYC bits in STICR 01: Packet interval is in accordance with the time stamp 10, 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	WORK[1:0]	00	R/W	Work Area Size These bits specify the work area size allocated at the front of the packet in external memory. 00: Work area is 0 bytes 01: Work area is 16 bytes 10: Work area is 32 bytes 11: Work area is 48 bytes

Bit	Bit Name	Initial Value	R/W	Description
15	CKSL	0	R/W	Operating Clock Selects the source clock for the stream data transfer clock 0: Peripheral clock 0 is used as the stream data transfer clock (stream data transfer clock is output from the ST_CLK pin) 1: External input clock is used as the stream data transfer clock (stream data transfer clock is input from the ST_CLK pin)
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13, 12	CKDV[1:0]	00	R/W	Operating Clock Division Ratio These bits specify the division ratio when peripheral clock 0 is selected as the stream data transfer clock. 00: Stream data transfer clock is 1/2 of peripheral clock 0 01: Stream data transfer clock is 1/4 of peripheral clock 0 10: Stream data transfer clock is 1/8 of peripheral clock 0 11: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	REQEN	0	R/W	ST_REQ Pin Enable Selects whether or not to use the ST_REQ pin. 0: ST_REQ pin is not used 1: ST_REQ pin is used (1) At reception: ST_REQ is output when the free space in FIFO is 8 bytes or less (2) At transmission: Transmission is stopped when ST_REQ is input

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	FRC[1:0]	00	R/W	Input Clock of Free-Running Timer 00: Timer input clock is 1/2 of peripheral clock 0 01: Timer input clock is 1/4 of peripheral clock 0 10: Timer input clock is 1/8 of peripheral clock 0 11: Setting prohibited
3	STRB	0	R/W	ST_STRB Pin Polarity 0: Data is transferred/received on the rising edge of ST_STRB 1: Data is transferred/ received on the falling edge of ST_STRB
2	REQ	0	R/W	ST_REQ Pin Polarity 0: ST_REQ is active-high 1: ST_REQ is active-low
1	VLD	0	R/W	ST_VALID Pin Polarity 0: ST_VALID is active- high 1: ST_VALID is active- low
0	STAT	0	R/W	ST_START Pin Polarity 0: ST_START is active- high 1: ST_START is active- low

25.3.2 Control Registers 0, 1 (STICR0, STICR1)

STICR enables or disables the STIF module and sets the packet interval for stream data transmission.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ICYC[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 16	ICYC[11:0]	All 0	R/W	Number of Cycles between Transmit Packets These bits set the fixed value when a fixed value is used as the number of cycles between packets during transmission. 1 to 4096 cycles of peripheral clock 0 can be inserted as idle cycles between packets.
15	RST	0	R/W	STIF Module Reset Writing 1 to this bit resets the STIF module. This bit is always read as 0.
14 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EN	0	R/W	STIF Module Enable 0: STIF module is disabled 1: STIF module is enabled

25.3.3 Interrupt Status Registers 0, 1 (STIISR0, STIISR1)

STIISR shows the states of STIF interrupts.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RPN	—	—	LONG	SHORT	—	—	—	ROVF	—	—	—	TSTO
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	TPN	0	R/W*	Transmit Packet Count Interrupt 0: Transmit packet count register value > Transmit packet counter value 1: Transmit packet count register value = Transmit packet counter value After an interrupt is issued, the transmit packet counter is cleared to 0 and continues counting.
27 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	RPN	0	R/W*	Receive Packet Count Interrupt 0: Receive packet count register value > Receive packet counter value 1: Receive packet count register value = Receive packet counter value After an interrupt is issued, the receive packet counter is cleared to 0 and continues counting.

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	LONG	0	R/W*	Long Packet Reception Interrupt 0: Packet exceeding 188 or 192 bytes has not been received 1: Packet exceeding 188 or 192 bytes has been received When a packet exceeding 188 or 192 bytes is received, the long packet counter and packet counter are both incremented by one. Data of 188 or 192 bytes is transferred to memory and the excess data is discarded.
8	SHORT	0	R/W*	Short Packet Reception Interrupt 0: Packet less than 188 or 192 bytes has not been not received 1: Packet less than 188 or 192 bytes has been received When a packet less than 188 or 192 bytes is received, the short packet counter is incremented by one and the packet is discarded.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	ROVF	0	R/W*	Receive FIFO Overflow Interrupt 0: Receive FIFO has not overflowed 1: Receive FIFO has overflowed The packets already received are retained, but the packet that caused overflow is discarded.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TSTO	0	R/W*	Time Stamp Counter Overflow Interrupt 0: Time stamp counter has not cycled once after receiving the last packet. 1: Time stamp counter has cycled once after receiving the last packet.

Note: * Write 1 to clear the bit.

25.3.4 Interrupt Enable Registers 0, 1 (STIIER0, STIIER1)

STIIER enables or disables the STIF interrupts.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RPNE	—	—	LONGE	SHORTE	—	—	—	ROVFE	—	—	—	TSTOE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	TPNE	0	R/W	Transmit Packet Count Interrupt Enable 0: Transmit packet count interrupt is disabled 1: Transmit packet count interrupt is enabled
27 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	RPNE	0	R/W	Receive Packet Count Interrupt Enable 0: Receive packet count interrupt is disabled 1: Receive packet count interrupt is enabled
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	LONGE	0	R/W	Long Packet Reception Interrupt Enable 0: Long packet reception interrupt is disabled 1: Long packet reception interrupt is enabled
8	SHORTE	0	R/W	Short Packet Reception Interrupt Enable 0: Short packet reception interrupt is disabled 1: Short packet reception interrupt is enabled

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	ROVFE	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Receive FIFO overflow interrupt is disabled 1: Receive FIFO overflow interrupt is enabled
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TSTOE	0	R/W	Time Stamp Counter Overflow Interrupt Enable 0: Time stamp counter overflow interrupt is disabled 1: Time stamp counter overflow interrupt is enabled

25.3.5 Time Stamp Counter Registers 0, 1 (STITSC0, STITSC1)

STITSC is used to count the time stamp.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TS[31:0]	All 0	R/W	Time Stamp Counter • When time stamp is used At reception: Starts counting from reception of the first packet. Counting can be started from any desired value by setting the value before reception. However, this register cannot be written to during reception of a packet. At transmission: Starts counting from transmission of the first packet. Counting can be started from any desired value by setting the value before transmission. However, this register cannot be written to during transmission of a packet. • When fixed value is used At reception: This register value is added to the front of a packet as a fixed value.

25.3.6 Transmit/Receive Packet Count Registers 0, 1 (STIPNR0, STIPNR1)

STIPNR sets the number of packets of the stream data to be transmitted or received.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 0	PN[20:0]	All 0	R/W	Number of Transmit/Receive Packets These bits set the number of packets for transmission or reception. An interrupt occurs when the number of packets actually transmitted or received has reached the value set in these bits. An interrupt does not occur when 0 is set in these bits.

25.3.7 Transmit/Receive Packet Counter Registers 0, 1 (STIPCR0, STIPCR1)

The number of packets of stream data that have been transmitted or received is set in STIPCR. At reception, the number of packets in this register is incremented after the last byte in the packet has been transferred to memory. At transmission, the number of packets is incremented after the last data in the packet has been sent from the ST_D7 to ST_D0 pins. In addition, the numbers of short packets and long packets that have been received are also set in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC[3:0]				LC[3:0]				—	—	—	PC[20:16]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	SC[3:0]	All 0	R	Number of Received Short Packets Cleared to 0 at a reset.
27 to 24	LC[3:0]	All 0	R	Number of Received Long Packets Cleared to 0 at a reset.
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 0	PC[20:0]	All 0	R	Number of Transmitted/Received Packets Cleared to 0 when a transmit packet count interrupt or receive packet count interrupt occurs, or at a reset.

25.3.8 Transmit/Receive FIFO Data Registers 0, 1 (STIFIFO0, STIFIFO1)

STIFIFO is an FIFO register that relays the stream data to be transmitted or received.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	STD[31:0]	All 0	R/W	Transmit/Receive Stream Data At transmission, transmit data should be written to this register. At reception, received data is read from this register.

25.3.9 Operation

25.3.10 External Memory Configuration for Stream Data Transmission/Reception

Figure 25.2 shows the external memory data configuration when transmitting or receiving stream data (in the case of a 16-byte work area). The work area size can be selected from among 0, 16, 32, and 48 bytes by the WORK bits in STIMDR. If addition of time stamp or fixed value is selected and the data length is 192 bytes, the first four bytes of the received data will be overwritten.

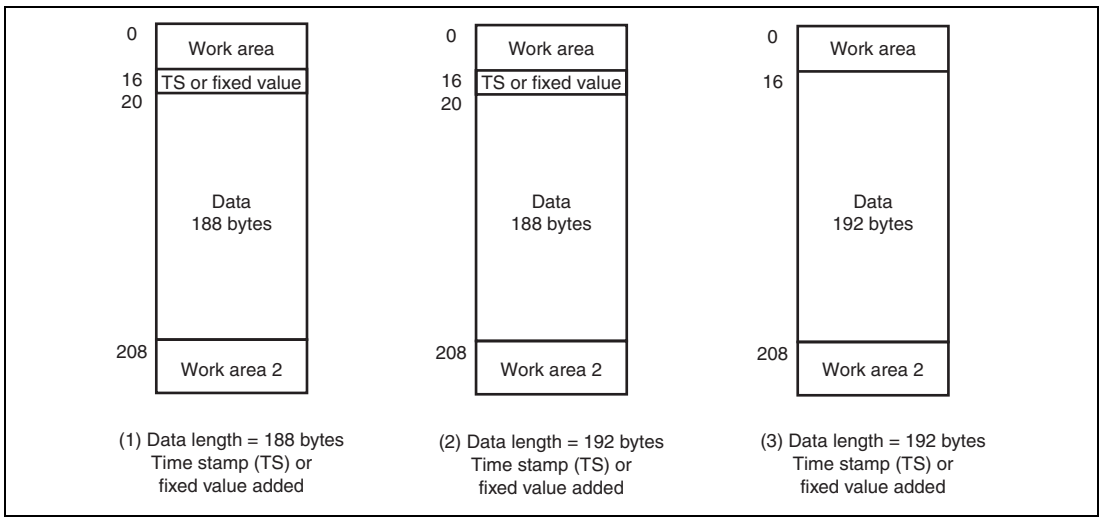


Figure 25.2 Transmit/Receive Data Structure in External Memory (with 16-Byte Work Area)

25.3.11 Stream Data Receive Operation

(1) DMAC Register Setting

When starting the stream data receive processing, set the following DMAC registers.

- Set the P4 area address for the data register of the transmit/receive FIFO of the STIF in SAR.
- Set the external memory address in DAR.
- Set the DMA transfer count in TCR according to the following equation. Only the value calculated below should be set.

  Transfer count =

  (192 bytes + work area byte count)/16 bytes × transmit/receive packet count

- Set H'0001 0001 in TCRB. The upper bits indicate the transfer count until reloading is performed, and the lower bits indicate the transfer counter value.
- Set H'0E20 5819 in CHCR.*
- Set the module ID and register ID (H'D3 when STIF channel 0 is used and H'D7 when STIF channel 1 is used) of the transfer request source in the DMARS bits corresponding to the DMAC channel used.

Note: * When STIF is not used, do not set CHCR.DVMD bit to 1.

(2) Clock Valid Reception (Input Data Rate: Max. 30 Mbps)

(a) Clock Valid Reception Interface

- Timing chart

  Figure 25.3 shows the timing of the clock valid reception interface.

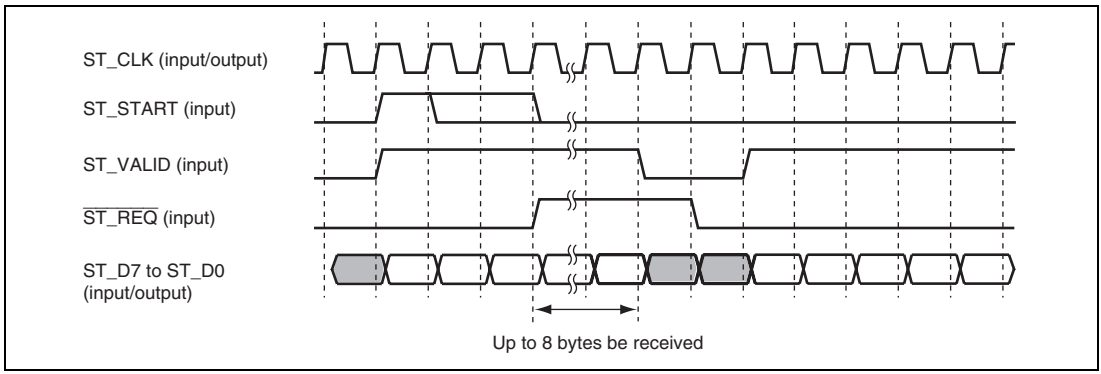


Figure 25.3 Clock Valid Reception Timing

- I/O selection for ST_CLK pin

  For the ST_CLK pin, input of an external clock or output of an internally generated clock can be selected by the CKSL bit in STIMDR (maximum frequency is 33 MHz).

- Active level setting for ST_START, ST_VALID, and ST_REQ pins

  The active levels of the ST_START, ST_VALID, and ST_REQ pins can be set by the STAT, VLD, and REQ bits in STIMDR, respectively.

- Selection of ST_REQ pin usage

  Whether or not to use the ST_REQ pin can be selected by the REQEN bit in STIMDR.

  When usage of the ST_REQ pin is enabled, the ST_REQ pin is asserted when the free space in the transmit/receive FIFO for stream data becomes eight bytes or less. After assertion, up to eight bytes of data can be received. The ST_REQ pin is negated when the free space in the FIFO has become 192 bytes or more.

  When usage of the ST_REQ pin is disabled, the ST_REQ pin output is fixed at low or high depending on the REQ bit value.

(b) Receive Packet Length

The receive packet length can be selected from 188 and 192 bytes.

(c) Work Area

The size of the work area in external memory can be selected from among 0, 16, 32, and 48 bytes.

(d) Time Stamp Setting at Reception

The time stamp setting at reception can be selected from among reception with a fixed value added, reception with a time stamp added, or reception without changes, according to the STMP[1:0] bits in STIMDR.

- Reception with a fixed value added

  The value set in the time stamp counter is used as the fixed value (counting is not performed).

  When the packet length is 188 bytes, a 4-byte fixed value is added to the front of the packet.

  When the packet length is 192 bytes, the first four bytes of the packet are overwritten with the fixed value.
- Reception with a time stamp added

  When the packet length is 188 bytes, the time stamp counter value is added to the front of the packet.

  When the packet length is 192 bytes, data at the beginning of the packet is overwritten with the time stamp counter value.

  The time stamp counter starts counting from reception of the first packet.

  Counting can be started from any desired value by setting the value before reception.

  Writing to the counter during reception of a packet is prohibited.
- Reception without changes

  This selection is possible only when the packet length is 192 bytes. In this case, the received packet is transferred to memory without any changes.

(e) Interrupt Sources during Reception

During clock valid reception, the following interrupt sources are available.

- Receive packet count interrupt
- Short packet reception interrupt
- Long packet reception interrupt
- Receive FIFO overflow interrupt
- Time stamp counter overflow interrupt (only at reception with a time stamp added)

(3) Strobe Reception

(a) Strobe Reception Interface

- Timing chart

  Figure 25.4 shows the timing of the strobe reception interface.

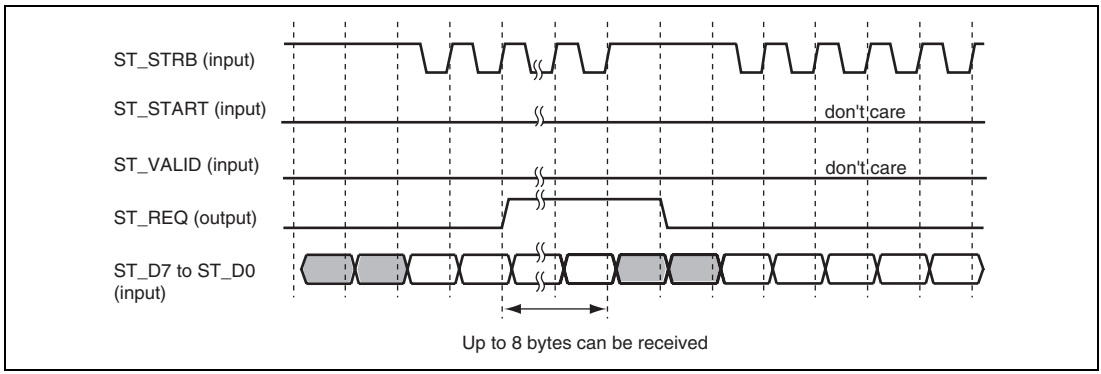


Figure 25.4 Strobe Reception Timing

- Active level setting for ST_STRB, ST_START, ST_VALID, and ST_REQ pins

  The active levels of the ST_STRB, ST_START, ST_VALID, and ST_REQ pins can be set by the STRB, STAT, VLD, and REQ bits in STIMDR, respectively.
- Selection of ST_REQ pin usage

  Whether or not to use the ST_REQ pin can be selected by the REQEN bit in STIMDR.

  When usage of the ST_REQ pin is enabled, the ST_REQ pin is asserted when the free space in the transmit/receive FIFO for stream data becomes eight bytes or less. After assertion, up to eight bytes of data can be received. The ST_REQ pin is negated when the free space in the FIFO has become 192 bytes or more.

  When usage of the ST_REQ pin is disabled, the ST_REQ pin output is fixed at low or high depending on the REQ bit value.

(b) Receive Packet Length

The receive packet length can be selected from 188 and 192 bytes.

(c) Work Area

The size of the work area in external memory can be selected from among 0, 16, 32, and 48 bytes.

(d) Time Stamp Setting at Reception

For the time stamp setting at strobe reception, only reception with a fixed value added can be selected. Set the STMP[1:0] bits in STIMDR to 00.

- Reception with a fixed value added

  The value set in the time stamp counter is used as the fixed value. Counting is not performed.

  When the packet length is 188 bytes, a 4-byte fixed value is added to the front of the packet.

  When the packet length is 192 bytes, the first four bytes of the packet are overwritten with the fixed value.

(e) Interrupt Sources during Reception

During strobe reception, the following interrupt sources are available.

- Receive packet count interrupt
- Receive FIFO overflow interrupt

25.3.12 Stream Data Transmit Operation

(1) DMAC Register Setting

When starting the stream data transmit processing, set the following DMAC registers.

- Set the external memory address in SAR.
- Set the P4 area address for the data register of the transmit/receive FIFO of the STIF in DAR.
- Set the DMA transfer count in TCR according to the following equation. Only the value calculated below should be set.

  Transfer count =

  (192 bytes + work area byte count)/16 bytes × transmit/receive packet count

- Set H'0001 0001 in TCRB. The upper bits indicate the transfer count until reloading is performed, and the lower bits indicate the transfer counter value.
- Set H'0E20 5819 in CHCR.*
- Set the module ID and register ID (H'D3 when STIF channel 0 is used and H'D7 when STIF channel 1 is used) of the transfer request source in the DMARS bits corresponding to the used DMAC channel.

Note: * Besides a purpose to use STIF, do not set CHCR.DVMD bit to 1.

(2) Clock Valid Transmission

(a) Clock Valid Transmission Interface

- Timing chart

  Figure 25.5 shows the timing of the clock valid transmission interface.

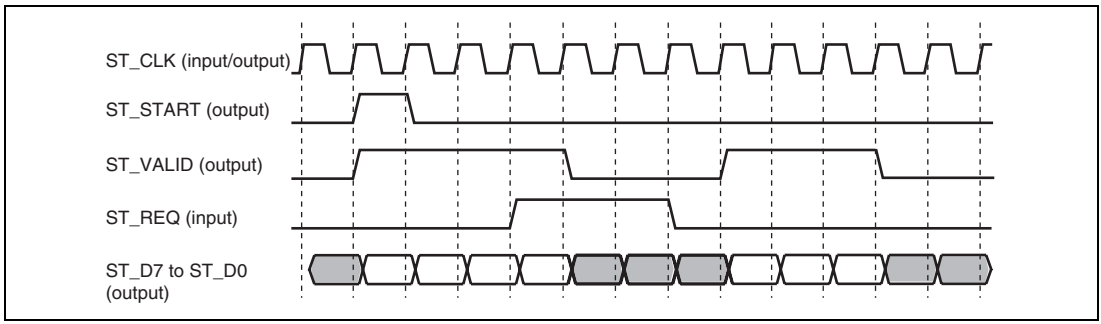


Figure 25.5 Clock Valid Transmission Timing

- I/O selection for ST_CLK pin

  For the ST_CLK pin, input of an external clock or output of an internally generated clock can be selected by the CKSL bit in STIMDR (maximum frequency is 33 MHz).
- Active level setting for ST_START, ST_VALID, and ST_REQ pins

  The active levels of the ST_START, ST_VALID, and ST_REQ pins can be set by the STAT, VLD, and REQ bits in STIMDR, respectively.
- Selection of ST_REQ pin usage

  Whether or not to use the ST_REQ pin can be selected by the REQEN bit in STIMDR.

  When usage of the ST_REQ pin is enabled, the ST_VALID pin is negated within four bytes after assertion of the ST_REQ pin.

  When usage of the ST_REQ pin is disabled, the ST_VALID pin is not negated until 188 or 192 bytes have been transferred.

(b) Transmit Packet Length

The transmit packet length can be selected from 188 and 192 bytes.

Since the packet length is handled as 192 bytes in external memory, the first four bytes of a packet are removed before transmission when the transmit packet length is set to 188 bytes. When the transmit packet length is set to 192 bytes, external memory data is transmitted without changes.

(c) Work Area

The size of the work area in external memory can be selected from among 0, 16, 32, and 48 bytes.

(d) Transmit Packet Interval Setting at Transmission

The transmit packet interval setting at transmission can be selected from fixed-interval transmission or time-stamp transmission, according to the STMP[1:0] bits in STIMDR.

- Fixed-interval transmission

  Transmission is performed using the value set in the ICYC[11:0] bits in STICR as the packet interval.

  1 to 4096 cycles of peripheral clock 0 can be set.
- Time-stamp transmission

  Transmission is performed with time stamp-based packet intervals. The time stamp counter starts counting from transmission of the first packet. Counting can be started from any desired value by setting the value before transmission.

  Note that writing to the counter during transmission of a packet is prohibited.

(e) Interrupt Source during Transmission

During clock valid transmission, the following interrupt source is available.

- Transmit packet count interrupt

(3) Strobe Transmission

(a) Strobe Transmission Interface

- Timing chart

  Figure 25.6 shows the timing of the strobe transmission interface. Data is updated simultaneously with the falling edge of the ST_STRB pin.

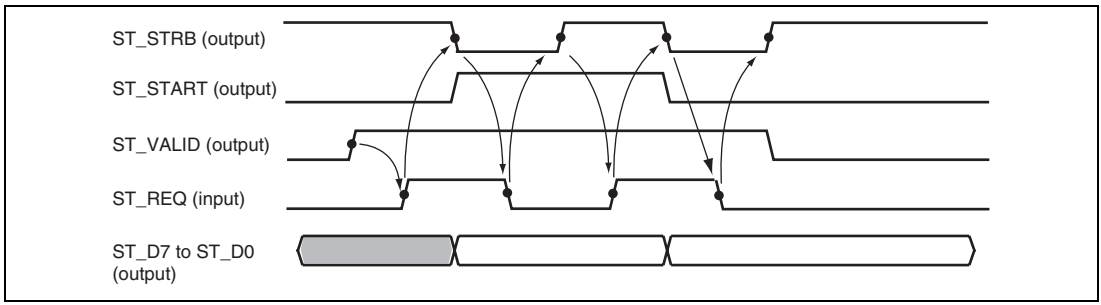


Figure 25.6 Strobe Transmission Timing

- Active level setting for ST_STRB, ST_START, ST_VALID, and ST_REQ pins

  The active levels of the ST_STRB, ST_START, ST_VALID, and ST_REQ pins can be set by the STRB, STAT, VLD, and REQ bits in STIMDR, respectively.
- Selection of ST_REQ pin usage

  When strobe transmission is selected, the ST_REQ pin always functions as an input pin regardless of the REQEN bit setting in STIMDR.

(b) Transmit Packet Length

The transmit packet length can be selected from 188 and 192 bytes.

Since the packet length is handled as 192 bytes in external memory, the first four bytes of a packet are removed before transmission when the transmit packet length is set to 188 bytes. When the transmit packet length is set to 192 bytes, external memory data is transmitted without changes.

(c) Work Area

The size of the work area in external memory can be selected from among 0, 16, 32, and 48 bytes.

(d) Transmit Packet Interval Setting at Transmission

For the transmit packet interval setting at strobe transmission, only fixed-interval transmission can be selected. Set the STMP[1:0] bits in STIMDR to 00.

- Fixed-interval transmission

  Transmission is performed using the value set in the ICYC[11:0] bits in STICR as the packet interval.

  Cycles of half the frequency of peripheral clock 0 are counted, and 1 to 4096 cycles can be set.

(e) Interrupt Source during Transmission

During strobe transmission, the following interrupt source is available.

- Transmit packet count interrupt

Section 26 I²C Bus Interface (IIC)

26.1 Features

The I²C bus interface has the following features:

- Supports the Philips I²C bus interface
- Multi-master compatible
- Seven- or ten-bit address compatible master
- Seven-bit slave address
- Fast mode compatible
- Variable clock frequencies

Figure 26.1 shows a block diagram for the I²C bus interface.

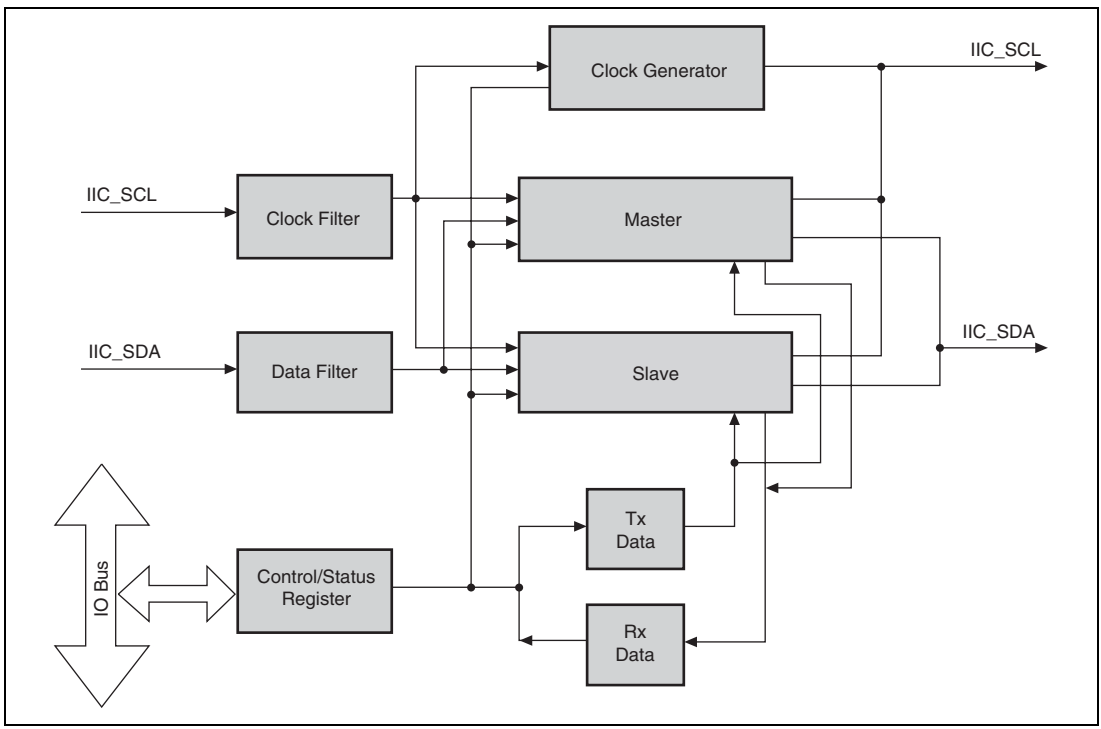


Figure 26.1 Block Diagram for I²C Bus Interface

26.2 Input/Output Pins

Table 26.1 lists the pins used in the I²C bus interface.

Table 26.1 Pin Configuration

Channel	Pin Name	I/O	Description
0	IIC0_SCL	I/O	I²C serial clock input/output pin*
	IIC0_SDA	I/O	I²C serial data input/output pin*
1	IIC1_SCL	I/O	I²C serial clock input/output pin*
	IIC1_SDA	I/O	I²C serial data input/output pin*

Note: * The SCL and SDA pins are open drain pins (3.3 V).

26.3 Register Descriptions

Table 26.2 shows the IIC register configuration. Table 26.3 shows the register state in each operating mode.

Table 26.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Area P4 Address*[1]	Area 7 Address*[1]	Access Size
0	Slave control register 0	ICSCR0	R/W	H'FFE7 0000	H'1FF7 0000	8
	Master control register 0	ICMCR0	R/W	H'FFE7 0004	H'1FF7 0004	8
	Slave status register 0	ICSSR0	R/(W)*[2]	H'FFE7 0008	H'1FF7 0008	8
	Master status register 0	ICMSR0	R/(W)*[3]	H'FFE7 000C	H'1FF7 000C	8
	Slave interrupt enable register 0	ICSIER0	R/W	H'FFE7 0010	H'1FF7 0010	8
	Master interrupt enable register 0	ICMIER0	R/W	H'FFE7 0014	H'1FF7 0014	8
	Clock control register 0	ICCCR0	R/W	H'FFE7 0018	H'1FF7 0018	8
	Slave address register 0	ICSAR0	R/W	H'FFE7 001C	H'1FF7 001C	8
	Master address register 0	ICMAR0	R/W	H'FFE7 0020	H'1FF7 0020	8
	Receive data register 0	ICRXD0	R/W	H'FFE7 0024	H'1FF7 0024	8
	Transmit data register 0	ICTXD0	R/W	H'FFE7 0024	H'1FF7 0024	8

Channel	Register Name	Abbreviation	R/W	Area P4 Address*[1]	Area 7 Address*[1]	Access Size
1	Slave control register 1	ICSCR1	R/W	H'FFE7 8000	H'1FF7 8000	8
	Master control register 1	ICMCR1	R/W	H'FFE7 8004	H'1FF7 8004	8
	Slave status register 1	ICSSR1	R/(W)*[2]	H'FFE7 8008	H'1FF7 8008	8
	Master status register 1	ICMSR1	R/(W)*[3]	H'FFE7 800C	H'1FF7 800C	8
	Slave interrupt enable register 1	ICSIER1	R/W	H'FFE7 8010	H'1FF7 8010	8
	Master interrupt enable register 1	ICMIER1	R/W	H'FFE7 8014	H'1FF7 8014	8
	Clock control register 1	ICCCR1	R/W	H'FFE7 8018	H'1FF7 8018	8
	Slave address register 1	ICSAR1	R/W	H'FFE7 801C	H'1FF7 801C	8
	Master address register 1	ICMAR1	R/W	H'FFE7 8020	H'1FF7 8020	8
	Receive data register 1	ICRXD1	R/W	H'FFE7 8024	H'1FF7 8024	8
	Transmit data register 1	ICTXD1	R/W	H'FFE7 8024	H'1FF7 8024	8

Notes: 1. P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

2. Only 0 can be written to bits 4 to 0 to clear the flags.

3. Only 0 can be written to bits 6 to 0 to clear the flags.

Table 26.3 Register State in Each Operating Mode

Channel	Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
0	Slave control register 0	ICSCR0	H'00	H'00	Retained	Retained
	Master control register 0	ICMCR0	H'x0	H'x0	Retained	Retained
	Slave status register 0	ICSSR0	H'00	H'00	Retained	Retained
	Master status register 0	ICMSR0	H'00	H'00	Retained	Retained
	Slave interrupt enable register 0	ICSIER0	H'00	H'00	Retained	Retained
	Master interrupt enable register 0	ICMIER0	H'00	H'00	Retained	Retained
	Clock control register 0	ICCCR0	H'00	H'00	Retained	Retained
	Slave address register 0	ICSAR0	H'00	H'00	Retained	Retained
	Master address register 0	ICMAR0	H'00	H'00	Retained	Retained
	Receive data register 0	ICRXD0	H'00	H'00	Retained	Retained
	Transmit data register 0	ICTXD0	H'00	H'00	Retained	Retained
1	Slave control register 1	ICSCR1	H'00	H'00	Retained	Retained
	Master control register 1	ICMCR1	H'x0	H'x0	Retained	Retained
	Slave status register 1	ICSSR1	H'00	H'00	Retained	Retained
	Master status register 1	ICMSR1	H'00	H'00	Retained	Retained
	Slave interrupt enable register 1	ICSIER1	H'00	H'00	Retained	Retained
	Master interrupt enable register 1	ICMIER1	H'00	H'00	Retained	Retained
	Clock control register 1	ICCCR1	H'00	H'00	Retained	Retained
	Slave address register 1	ICSAR1	H'00	H'00	Retained	Retained
	Master address register 1	ICMAR1	H'00	H'00	Retained	Retained
	Receive data register 1	ICRXD1	H'00	H'00	Retained	Retained
	Transmit data register 1	ICTXD1	H'00	H'00	Retained	Retained

26.3.1 Slave Control Register (ICSCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	SDBS	SIE	GCAE	FNA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved The write value should always be 0.
3	SDBS	0	R/W	Slave Data Buffer Select This bit is used to select the data buffer. The double-buffer mode and single-buffer mode are available. When this bit is set to 0, the double-buffer mode is selected. During a reception, as long as both buffers are full and the SDR flag has not been cleared, SCL is held low. When the SDR flag is cleared, the low level state of SCL is released. When this bit is set to 1, the single-buffer mode is selected. SCL will be held low from the timing when the receive data register acquires the data packet until the SDR flag is cleared. 0: Double-buffer mode 1: Single-buffer mode
2	SIE	0	R/W	Slave Interface Enable This bit must be set for the slave operation. If this bit is low, the slave interface is reset. This bit is cleared by setting the MIE bit to 1.
1	GCAE	0	R/W	General Call Acknowledgement Enable When a master requires a slave to issue an acknowledgement, this bit must be set to 1

Bit	Bit Name	Initial Value	R/W	Description
0	FNA	0	R/W	Forced Non Acknowledgement In the slave receive mode, the level of this bit is sent to the transmitting device as the acknowledge signal. This bit is set to 0 during the period that the data packet is being received, and set to 1 on completion of data reception. Forced non acknowledgement is returned to the master during slave reception. When the slave has received the last byte of data in a data packet, the slave communicates with the master by sending a nack, meaning that the acknowledgement is not driven. The master issues a stop on the bus after receiving a nack. The setting of this bit does not affect the acknowledgement of the slave address.

26.3.2 Slave Status Register (ICSSR)

The status bits (bits 0 to 4) in the slave status register are cleared by writing 0 to the respective status bit positions. The individual bits are held 1 until 0 is written to (other than the GCAR and STM bits).

Bit:	7	6	5	4	3	2	1	0
	—	GCAR	STM	SSR	SDE	SDT	SDR	SAR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0.
6	GCAR	0	R	General Call Address Received Indicates that the address received from the bus is a general call address (00H). This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the slave control register) is set to 0 or when the SSR bit (bit 4 in this register) is set to 1.
5	STM	0	R	Slave Transmit Mode Indicates whether the current slave transmit mode is read or write. When this bit is set to 1, the mode is read. When this bit is set to 0, the mode is write. This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the slave control register) is set to 0 or when the SSR bit (bit 4 in the slave status register) is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	SSR	0	R/W*	Slave Stop Received A stop condition has been output on the bus. This status bit becomes active after the rising edge of SDA during the stop bit.
3	SDE	0	R/W*	Slave Data Empty Indicates that data to be transmitted has been loaded into the shift register. At the start of byte data transmission, the contents of the ICTXD register are loaded into a shift register ready for outputting data on the bus. This status bit indicates that data has been loaded and the ICTXD register is again ready for further data. This status bit becomes active on the falling edge of SCL before the first data bit. During the single-buffer mode, this bit must be reset every time new data has been written to the ICTXD register. This is because the slave holds SCL low to stop the bus while this bit is set to 1 even if a slave transmission cycle is started.
2	SDT	0	R/W*	Slave Data Transmitted A byte of data has been transmitted to the bus. This bit becomes active after the falling edge of SCL during the last data bit.
1	SDR	0	R/W*	Slave Data Received A byte of data has been received from the bus and is ready for read in the receive data register. This bit becomes active after the falling edge of SCL during the last data bit. During the single-buffer mode, this bit must be reset after data has been read from the ICRXD register. When SDBS is set to 1, SCL will be held low from the timing when the receive data register acquires the data packet until the SDR flag is cleared.

Bit	Bit Name	Initial Value	R/W	Description
0	SAR	0	R/W*	Slave Address Received Indicates that the slave has recognized its own address on the bus (defined by the contents of the slave address register). If the general call acknowledgement enable bit is enabled in the slave control register, then this status bit is also set to 1 even if the address on the bus is a general call address. In this case, the GCAR bit in this register is used to determine whether or not the address is a general call address. The STM bit indicates whether the access is read (high) or write (low). This status becomes active after the falling edge of SCL during the last address bit. The slave holds SCL low during the start of the ACK phase until the software resets this status bit.

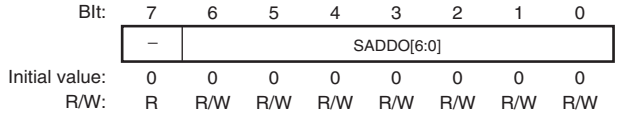
Note: * This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

26.3.3 Slave Interrupt Enable Register (ICSIER)

Bit:	7	6	5	4	3	2	1	0
	–	–	–	SSRE	SDEE	SDTE	SDRE	SARE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0.
4	SSRE	0	R/W	Slave Stop Received Interrupt Enable 0: Disables the SSR interrupt. 1: Enables the SSR interrupt.
3	SDEE	0	R/W	Slave Data Empty Interrupt Enable 0: Disables the SDE interrupt. 1: Enables the SDE interrupt.
2	SDTE	0	R/W	Slave Data Transmitted Interrupt Enable 0: Disables the SDT interrupt. 1: Enables the SDT interrupt.
1	SDRE	0	R/W	Slave Data Received Interrupt Enable 0: Disables the SDR interrupt. 1: Enables the SDR interrupt.
0	SARE	0	R/W	Slave Address Received Interrupt Enable 0: Disables the SAR interrupt. 1: Enables the SAR interrupt.

26.3.4 Slave Address Register (ICSAR)



Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0.
6 to 0	SADD0[6:0]	All 0	R/W	Slave Address This is the unique 7-bit address allocated to the slave on the I²C bus. The slave interface compares this address with the first seven bits transmitted as the slave address, at the beginning of a data packet transmission.

26.3.5 Master Control Register (ICMCR)

Bit:	7	6	5	4	3	2	1	0
	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
Initial value:	0	–	–	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MDBS	0	R/W	Master Data Buffer Select This bit is used to select the data buffer. The double-buffer mode and singe-buffer mode are available. When this bit is set to 0, the double-buffer mode is selected. During a reception, as long as both buffers are full and the MDR flag has not been cleared, SCL is held low. When the MDR flag is cleared, the low level state of SCL is released. When this bit is set to 1, the single-buffer mode is selected. SCL will be held low from the timing when the receive data register acquires the data packet until the MDR flag is cleared. 0: Double-buffer mode 1: Single-buffer mode
6	FSCL	—	R/W	Forced SCL This bit controls the status of the I2C_SCL pin (reading reflects the current level on the I²C bus). When the OBPC bit is set, this bit directly controls the SCL line on the bus. During a read cycle, the level on this bit (which includes the reset level) will change depending on the level on I2C_SCL since it reflects the level on the I2C_SCL.
5	FSDA	—	R/W	Forced SDA This bit controls the status of the I2C_SDA pin (reading reflects the busy status level on the I2C_SDA). When the OBPC bit is set then this bit directly controls the SDA line on the bus. During a read cycle, the level of this bit (which includes the reset level) will show the busy status of the I²C bus (1 for busy; 0 for not busy).

Bit	Bit Name	Initial Value	R/W	Description
4	OBPC	0	R/W	Override Bus Pin Control When this bit is set to 1, the FSDA and FSCL bits in this register control SDA and SCL directly. This mode is used for testing purposes only.
3	MIE	0	R/W	Master Interface Enable When this bit is set to 1, the master interface is enabled.
2	TSBE	0	R/W	Start Byte Transmission Enable When this bit is set to 1, the master transmit is issuing a start byte (01H) on the bus after. The start byte is used for interfacing to slower microcontroller compatible with I²C bus interfaces.
1	FSB	0	R/W	Forced Stop onto the Bus When this bit is set to 1, the master transmits a STOP condition on the bus at the end of the current transfer. If ESG is also set, the master immediately transmits a START condition and begins transmitting a new data packet. If ESG is not set, state the master enters the idle state.
0	ESG	0	R/W	Enable Start Generation When this bit is set to 1, the master starts transmission of a data packet. If the bus is idle when ESG is set, the master transmits a START condition on the bus and then transmits the slave address. If the master is transferring data when ESG is set, at the end of that data byte transfer, the master transmits a repeated START condition before transmitting the slave address. When transmitting a data packet, the software must reset this bit when the slave address has been transmitted, otherwise a repeated START condition is transmitted after every transmission is completed.

26.3.6 Master Status Register (ICMSR)

The status bits (bits 0 to 6) in the master status register are cleared by writing 0 to the respective status bit positions. The individual status bits are held 1 until a reset by writing 0 to the appropriate bit position.

Bit:	7	6	5	4	3	2	1	0
	–	MNR	MAL	MST	MDE	MDT	MDR	MAT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0.
6	MNR	0	R/W*	Master Nack Received When this bit is set to 1, this bit indicates that the master has received a nack response (the SDA line is high during the acknowledge cycle on the bus) to either an address or data transmission.
5	MAL	0	R/W*	Master Arbitration Lost In a multi-master system, when this bit is set to 1, it indicates that the master has lost arbitration to one of other masters on the bus. At this point, MIE is reset and the master interface is disabled.
4	MST	0	R/W*	Master Stop Transmitted When this bit is set to 1, it indicates that the master has sent a STOP condition on the bus. A STOP condition can be sent either as a result of the setting of the forced stop bit in the control register, or from a nack being received from a slave during a slave receive data packet.

Bit	Bit Name	Initial Value	R/W	Description
3	MDE	0	R/W*	Master Data Empty At the start of a byte data byte transmission, the contents of the transmit data register are loaded into a shift register ready for transmitting on the bus. When this bit is set to 1, it indicates that the transmit data register is available for further data by setting this register. During master transmit mode, the MDE bit is set at the same timing as the MAT bit is also set after transmission of the slave address. In this case, you need to set the MDT and MAT bits after the ICMCR's ESG bit is cleared. The clearing will restart the data transmission.
2	MDT	0	R/W*	Master Data Transmitted Byte data has been sent to the slave on the bus. This status bit becomes active after the falling edge of SCL during the last data bit.
1	MDR	0	R/W*	Master Data Received Byte data has been received from the bus and is in the receive data register. This status bit becomes active after the falling edge of SCL during the last data bit. During single-buffer mode, this status bit must be reset after data has been read from the receive data register. When MDBS is set to 1, SCL will be held low from the timing when the receive data register acquires the data packet until the MDR flag is cleared. During master reception mode, the MDR bit is set at the same timing as the MAT bit set after transmission of the salve address. In this case, you must clear the MDR and MAT bits after the ICMCR's ESG bit is cleared. Clearing will start the data reception
0	MAT	0	R/W*	Master Address Transmitted The master has been transmitted the slave address byte of a data packet. This bit becomes active after the falling edge of SCL during the ack bit of after the address.

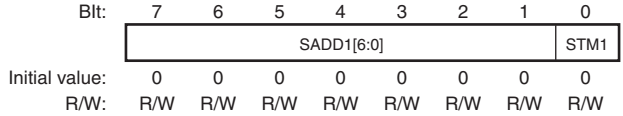
Note: * This bit can be read from or written to. Writing 0 clears this bit to 0 and writing 1 is ignored.

26.3.7 Master Interrupt Enable Register (ICMIER)

Bit:	7	6	5	4	3	2	1	0
	–	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

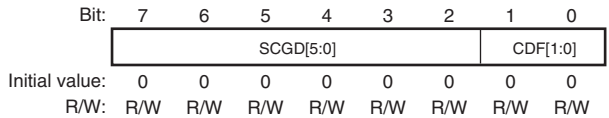
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0.
6	MNRE	0	R/W	Master Nack Received Interrupt Enable 0: Disables the MNR interrupt. 1: Enables the MNR interrupt.
5	MALE	0	R/W	Master Arbitration Lost Interrupt Enable 0: Disables the MAL interrupt. 1: Enables the MAL interrupt.
4	MSTE	0	R/W	Master Stop Transmitted Interrupt Enable 0: Disables the MST interrupt. 1: Enables the MST interrupt.
3	MDEE	0	R/W	Master Data Empty Interrupt Enable 0: Disables the MDE interrupt. 1: Enables the MDE interrupt.
2	MDTE	0	R/W	Master Data Transmitted Interrupt Enable 0: Disables the MDT interrupt. 1: Enables the MDT interrupt.
1	MDRE	0	R/W	Master Data Received Interrupt Enable 0: Disables the MDR interrupt. 1: Enables the MDR interrupt.
0	MATE	0	R/W	Master Address Transmitted Interrupt Enable 0: Disables the MAT interrupt. 1: Enables the MAT interrupt.

26.3.8 Master Address Register (ICMAR)



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SADD1[6:0]	All 0	R/W	Slave Address These bits are the address of the slave which the master communicates with.
0	STM1	0	R/W	Slave Transfer Mode This bit specifies the mode in which the slave operates. Bit STM1 sets the operating mode (transmit or receive mode) of the slave, which is an external slave device whose address matches the slave address (SADD1) sent from the master. The slave device is automatically set to transmit/receive mode by hardware on reception of the STM1 signal. When this bit is set to 1, it indicates a read operation, when this bit is cleared to 0, it indicates a write operation.

26.3.9 Clock Control Register (ICCCR)



Bit	Bit Name	Initial Value	R/W	Description
7 to 2	SCGD	All 0	R/W	SCL Clock Generation Divider When operating in master mode, the SCL clock is generated from the internal clock using SCGD as the ratio. The slave will also operate on the clock generated from the internal clock when SCL is held low to hold the bus up when an overflow occurs. SCGD must be specified in both master and slave modes. The formula expressing the relationship is: Equation 2 SCL rate calculation SCLfreq = IICck / (20 + (SCGD * 8)) IICck: I²C internal clock frequency Suggested settings for CDF and SCGD for various CPU speeds and the two I²C bus speeds are given in table 26.4.
1, 0	CDF	All 0	R/W	Clock Division Factor The internal clock used in most blocks in the I²C module is a divided peripheral clock. The internal I²C clock is generated from the peripheral clock using the CDF as the division ratio: Equation 1 I²C internal clock frequency calculation IICck = Pck0 / (1 + CDF) Pck0: Peripheral clock The minimum time to ensure adequate setup and hold times on the SDA line relative to the SCL line on the bus. The clock frequency is to ensure that the glitch filtering will operate with glitches of up to 50 ns as described in the fast mode I²C specification.

Note: CDF must be set so that the clock frequency (IICck) is lower than 20 MHz.

Table 26.4 Suggested Settings for CDF and SCGD*

Peripheral Clock Frequency	100 kHz		400 kHz	
	CDF	SCGD	CDF	SCGD
66.7 MHz	3	19	3	3
Error	- 3.05 %		- 5.26 %	

Note: * These are suggested values for the SCL rate.

26.3.10 Receive and Transmit Data Registers (ICRXD and ICTXD)

Reading from or writing to these registers access different physical internal registers. When data is to be transmitted, the contents of the shift register are loaded via TXD. After data has been received into the shift register from the I²C bus, it is then loaded into RXD.

- Receive Data Register (ICRXD)

Bit:	7	6	5	4	3	2	1	0
	RXD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RXD[7:0]	All 0	R	Read—Receive Data Data received by master or slave.

- Transmit Data Register (ICTXD)

Bit:	7	6	5	4	3	2	1	0
	TXD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TXD[7:0]	All 0	W	Write—Transmit Data Data transmitted by master or slave.

26.4 Operations

26.4.1 Data and Clock Filters

These blocks filter out glitches on signals coming from the I²C bus. Glitches up to one internal clock period in width are rejected (For details on the internal clock frequency see section 26.3.9, Clock Control Register (ICCCR)). This is for the faster I²C bit rate (400 KHz) but does not violate the slower I²C bus rate specification.

These blocks also resynchronizes bus signals with the internal clock.

26.4.2 Clock Generator

The clock generator has two functions. Firstly, it generates the SCL I²C bus clock according to commands from of the master or slave interface. Secondly, it controls the internal clock rate, used by filtering blocks and the master and slave interfaces. This clock functions as a clock enable signal of the registers in these blocks.

26.4.3 Master/Slave Interfaces

These two interfaces run independently and in parallel. The master interface controls the transmission of address and data on the I²C bus. The slave interface monitors the I²C bus and takes part in transmissions if its programmed address is seen on the bus. The interfaces communicate with the control/status registers independently. There is only one interrupt line output from the I²C module. The interrupt source is either the master or the slave.

26.4.4 Software Status Interlocking

In order that the software interface to the I²C module be as robust as possible, various status interlocks are built into the operation of the master and slave interfaces. The status bits involved are:

(1) MDR and SDR

MDR and SDR are set to 1 when data is received. Clear the status after reading the receive data register. If data is received while MDR and SDR are set, hardware recognizes that unread data remains in the receive data register and automatically holds SCL at low level and suspends data transmission. In this case, transmission can be resumed by clearing the status after reading the receive data.

Consequently, when receiving data continuously, be sure to clear the status of MDR and SDR after reading the receive data register.

(2) MDE and SDE

If the MDE or SDE status bits are still set data in the transmit data register is to be transmitted on the I²C bus by the slave or master, the SCL line must be held low until the MDE and SDE status bits are reset. The MDE or SDE status bit being set indicates that the data currently held in the Transmit Data Register has already been transmitted on the I²C bus.

The software must clear this status bit when it writes to the transmit data register which is ready to transmit subsequent data bytes. This is not required for the first byte of data to be transmitted on the bus.

(3) MAL

When the master loses arbitration, the MAL bit (of the master status register) is set and the MIE bit (of the master control register) is reset. At this point, master mode is invalid and the I²C bus interface enters the slave mode. When master operation is restarted, data transfer from the master begins after the MAL bit has been cleared.

(4) SAR

The SAR status bit is set when the slave identifies its address on the I²C bus. At this point the slave interface forces the SCL line low until the SAR status bit is reset.

This is particularly important when a slave transmit is about to take place on the bus, and the slave will transmit the data from the transmit data register. The software responds to the SAR status by writing the required data into the transmit data register and then resetting the SAR status bit. This allows the slave interface to continue the access.

When the slave is about to receive data, the software may be reading data loaded in a previous access from the receive data register. In this case the valid data still held in the receive data register is overwritten. However, this is avoided using the SAR status bit. After the software has read data in the receive data register, reset the SAR bit (if it is set). Then overwriting the receive data register is avoided.

26.4.5 I²C Bus Data Format

Figure 26.2 shows a timing chart for the I²C bus interface. Table 26.5 describes the meaning of each symbol in figure 26.2.

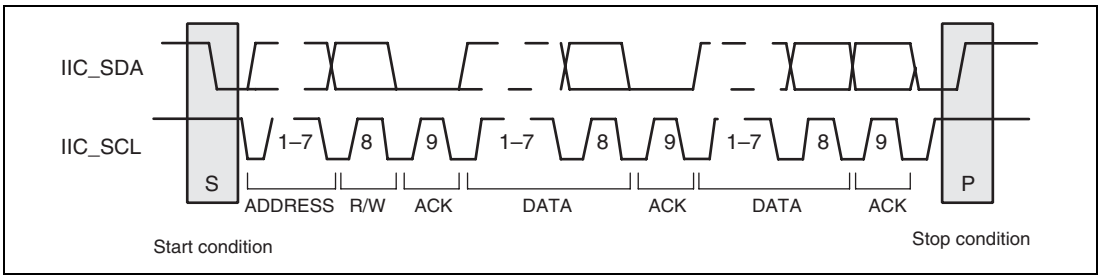


Figure 26.2 I²C Bus Timing

Table 26.5 Description on Symbols of I²C Bus Data Format

Symbol	Description
S	Indicates a start condition. A master device changes SDA from high to low while SCL is high level.
SLA	Indicates a slave address. A slave address is used when a master device selects a slave device.
R/W	Indicates the direction of data transmission. If the R/W bit is 1, the data flows from the slave to the master device. If the bit is 0, the data flows from the master to the slave device.
A	Indicates data acknowledge. Data receiving device makes SDA low level (the slave device returns a data acknowledge signal in master transmission mode, and vice versa).
DATA	Indicates transmit or receive data. The data length is eight bits, which are transferred in the MSB first.
P	Indicates a stop condition. A master device changes SDA from low to high while SCL is high.

26.4.6 7-Bit Address Format

Figure 26.3 shows the format of data transfer from a master to a slave device (master data transmit format). Figure 26.4 shows the data transfer format (master data receive format) when a master device reads the second and the following byte data from a slave device.

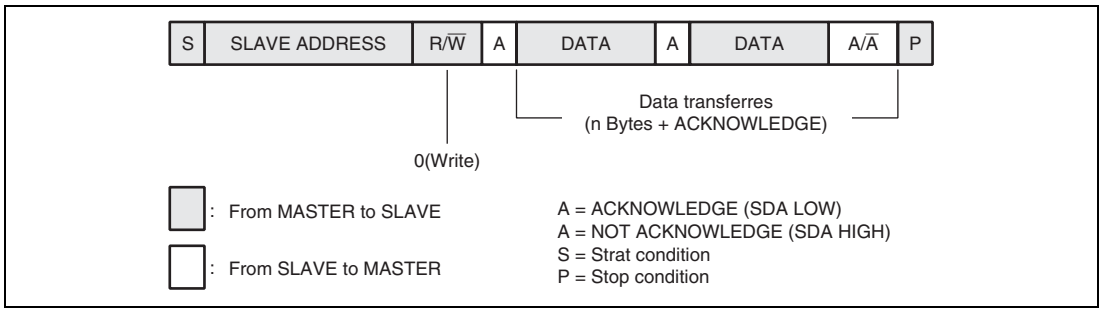


Figure 26.3 Master Data Transmit Format

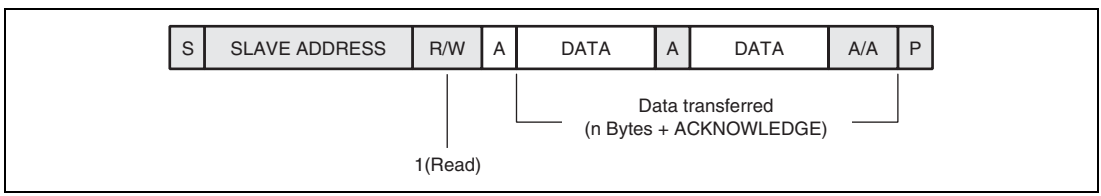


Figure 26.4 Master Data Receive Format

Figure 26.5 shows the combined format when the data transfer direction changes during one transfer. When changing the direction after the first transfer, the repeated START condition (Sr), slave address and R/W bits are transmitted. In this case, the R/W bit is set to the direction opposite to the first transfer direction. The repeated START condition is issued by the master at the end of a transmit or receive cycle if the enable start generation bit in the master control register has been set.

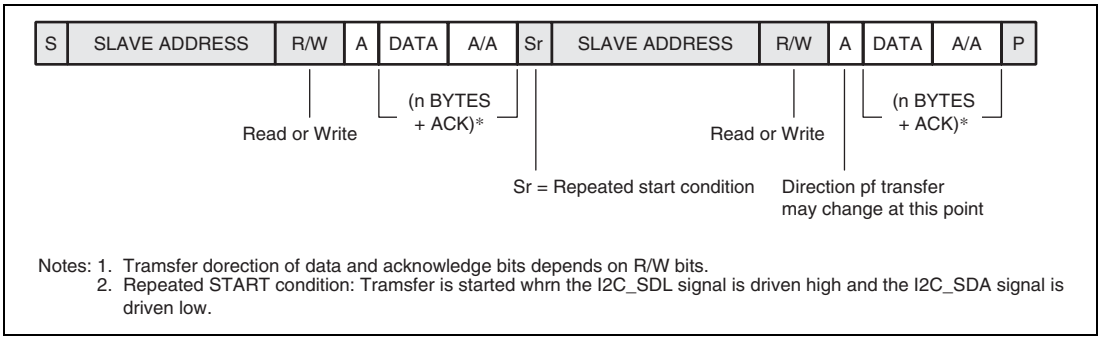


Figure 26.5 Combination Transfer Format of Master Transfer

26.4.7 10-Bit Address Format

Description is given below on the 10-bit address transfer format supported in master mode.

This format has three transfer methods as the 7-bit address transfer format.

Figure 26.6 shows the data transmit format. The set value in the master address register is output in one byte following the first START condition (S). The value set in the transmit data register (TXD) is transmitted as a slave address in the second byte. Data on and after the third byte is transferred in the same way as the 7-bit address data.

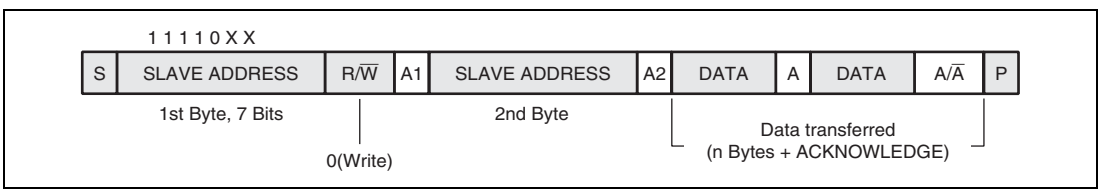


Figure 26.6 10-Bit Address Data Transmit Format

Figure 26.7 shows the data receive format. Two bytes of an address is transmitted a repeated START in the same way as in the data transmit format. Then, repeated START condition (Sr) is transmitted and the value set in the address register is output. At this time, STM1 must be set to 1 (receive mode). Data is transferred in the same way as in the 7-bit address data receive format.

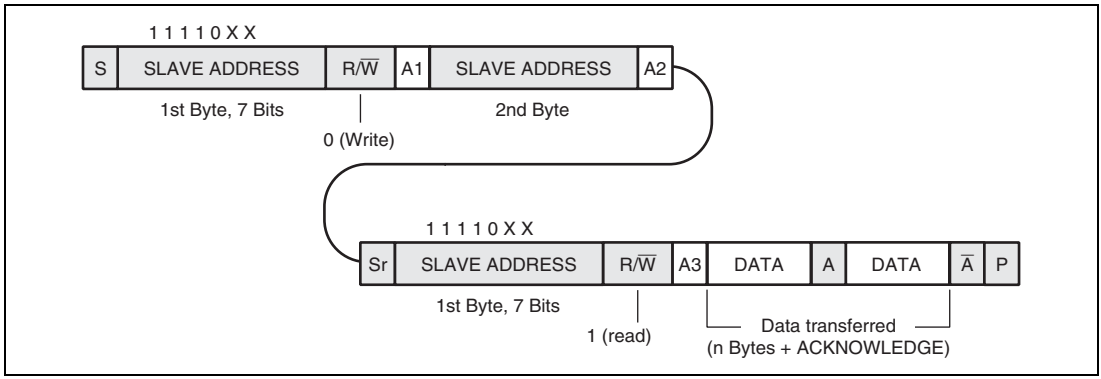


Figure 26.7 10-Bit Address Data Receive Format

Figure 26.8 shows the data transmit/receive combined format.

In the data transmit/receive combined format, data is transmitted after an address is transmitted with the first two bytes. Then, the repeated START condition (Sr) is transmitted instead of STOP condition (P). After Sr is transmitted, the procedure is the same as that in the data receive format.

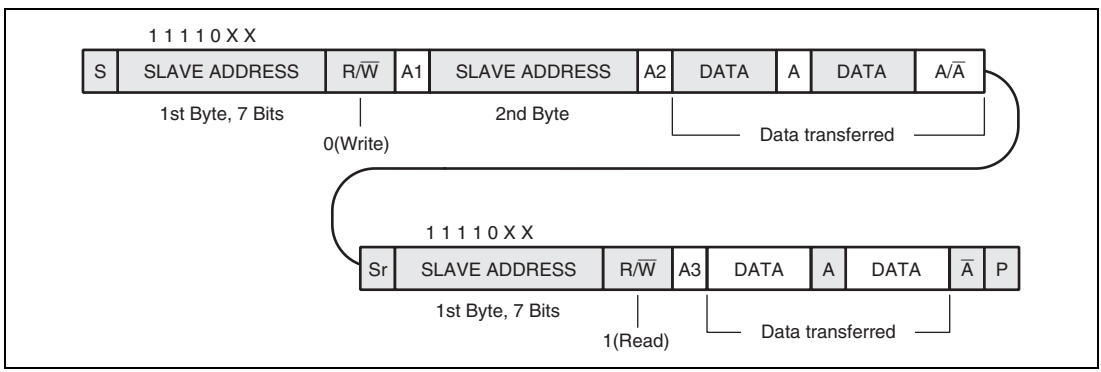


Figure 26.8 10-Bit Address Transmit/Receive Combined Format

26.4.8 Master Transmit Operation

The transmit procedure and operation in master transmit mode are described below. Figure 26.9 shows the timing chart in master transmit mode. Setting the MDBS bit in the master control register allows the IIC to operate in single-buffer mode.

1. For initial setting, set the clock control register and the master interrupt enable register according to the slave address, transmit data, and the transmit speed. Since slave mode is also required even when master mode is used, set the device address in the slave address register.
2. Monitor the FSDA bit in the master control register. Confirm that this bit is low, meaning that other I²C devices are not using the bus. After confirmation, set the MIE (bit 3) and ESG (bit 0) bits in the master control register to 1 to start master transmission.
3. After the transmit START condition, slave address, and data transfer direction bits are transmitted, an interrupt due to the MAT and MDE bits in the master status register is generated at the timing of (1) in figure 26.9. At this time, clear the ESG bit to 0. To suspend the data transmission, the master device will hold SCL low until the MDE bit is cleared.
4. An interrupt due to the SAR bit is generated at the timing of (3) shown in figure 26.9. If the IRQ handling in the slave device is delayed, the slave device extends the IIC_SCL period to suspend data transmission (at the timing of (7) in figure 26.9). The slave device drives IIC_SDA low at the ninth clock and returns ACK.
5. Data is transmitted in units of nine bits: 8-bit data and 1-bit ACK. An interrupt of MDE (bit 3) is generated at the ninth clock before data transfer (at the timing of (2) in figure 26.9). An interrupt of MDT (bit 2) is generated at the eighth clock after 1-byte data transfer (at the timing of (4) in figure 26.9). Clear MDE to 0 after setting transmit data. An interrupt of SDR (slave data receive) of the slave device is generated at the eighth clock (at the timing of (6) in figure 26.9). Clear SDR after the slave device reads the receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmit (at the timing of (8) in figure 26.9).
6. To end data transfer, an interrupt of MNR (bit 6) in the master status register is generated at the ninth clock (at the timing of (5) in figure 26.9) when ACK from the slave device is 1 (Nack). The master device receives this Nack and outputs data transfer end condition. When data transmission ends on the master device side, set FSB (bit 1) in the master control register to 1 to output the suspend condition. After the IIC module fetches FSB on completion of transmission or reception of the last of byte data, it enters the stop state. Therefore in order to stop the communication after the predetermined number of byte data is transferred, the FSB bit needs to be set before the last byte data transfer is started.
7. The FSB bit needs to be set before the last byte data is transferred. In master transmit mode, after the last byte data is set, the MST (master stop transmitted) bit is checked by either

interrupt or polling. At the same time MNR (master NACK received) bit must be checked. If NACK is returned, an error routine is executed to retransmit the last byte data.

Signal level changes of (1) to (6) in figure 26.9 are generated after the falling edge of the clock.

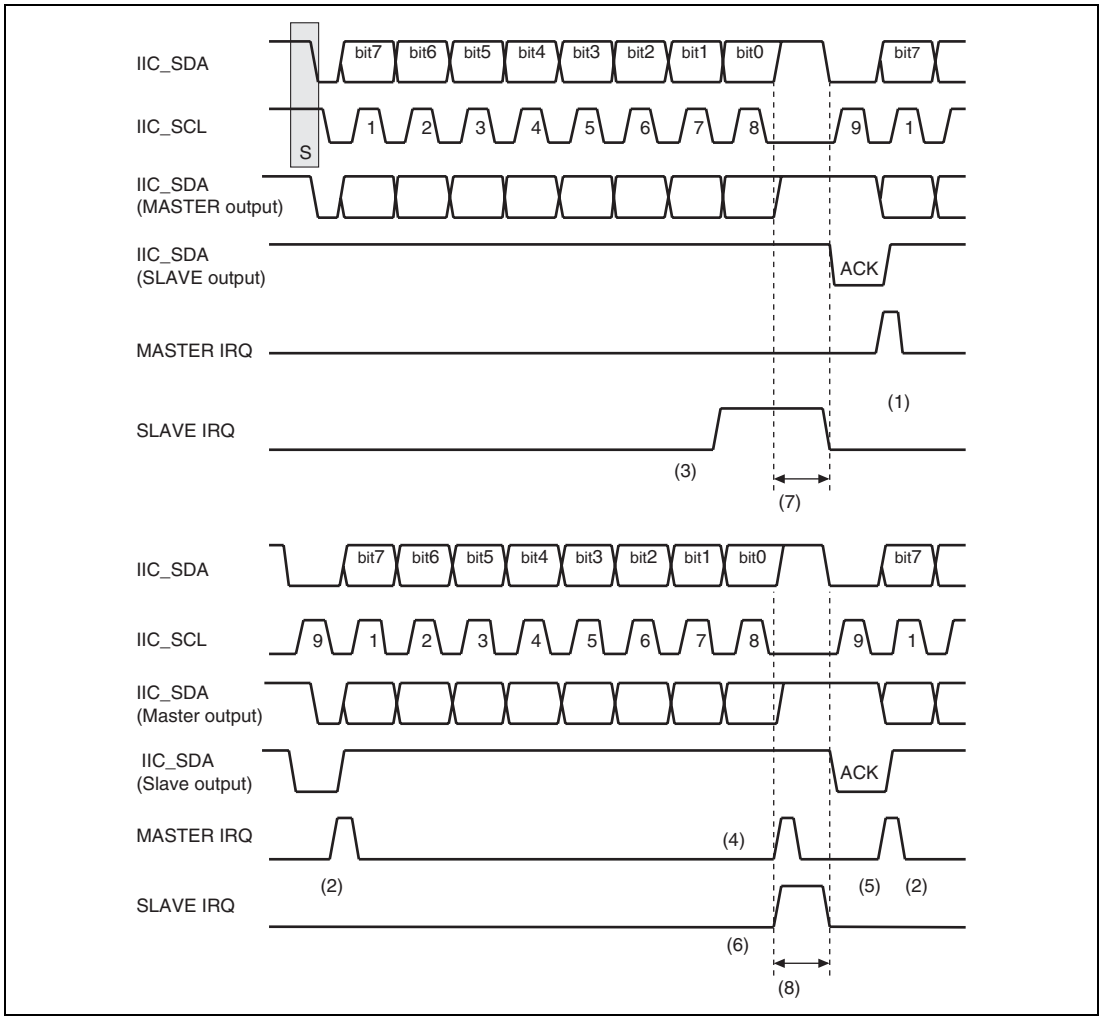


Figure 26.9 Data Transmit Mode Operation Timing

26.4.9 Master Receive Operation

The data receive procedure and operation in master receive mode are described below. Figure 26.10 shows the timing chart in master receive mode. Setting the MDBS bit in the master control register allows the IIC to operate in single-buffer mode.

1. In master receive mode, as to transmit of a slave address and a 1-bit signal indicating the data transfer direction, operation is the same as that in master transmit mode. At this time, set the data transfer direction to 1 (reception).
2. The slave device automatically enters the data transmit mode according to the signal that indicates the data transfer direction, and transmits 1-byte data in synchronization with the SCL clock output from the master device. The master device generates an interrupt of MDR (bit 1) at the eighth clock (at the timing of (2) in figure 11). Clear the MDR bit after the master device reads receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmission, as shown at the timing of (3) in figure 26.10.
3. The slave device generates an interrupt of the status SDT (bit 2) indicating 1-byte data transfer end at the eighth clock (at the timing of (2) in figure 26.10) and an interrupt of the status SDE (bit 3) indicating data empty at the ninth clock (at the timing of (1) in figure 26.10). Clear SDE after writing slave transmit data to TXD.
4. To end data transfer, set FSB (bit 1) in the master control register of the master device and output suspend condition. After the IIC module fetches FSB on completion of transmission or reception of the last of byte data, it enters the stop state. . Therefore in order to stop the communication after predetermined number of byte data is transferred, FSB bit needs to be set before the last byte data transfer is started. After confirmation of the last byte data reception, though the master receiver finishes the receive transaction, the protocol layer will inform the slave transmitter or retransmission if the last byte is incorrect.

Signal level changes of (1) to (3) in figure 26.10 are generated after the falling edge of the clock.

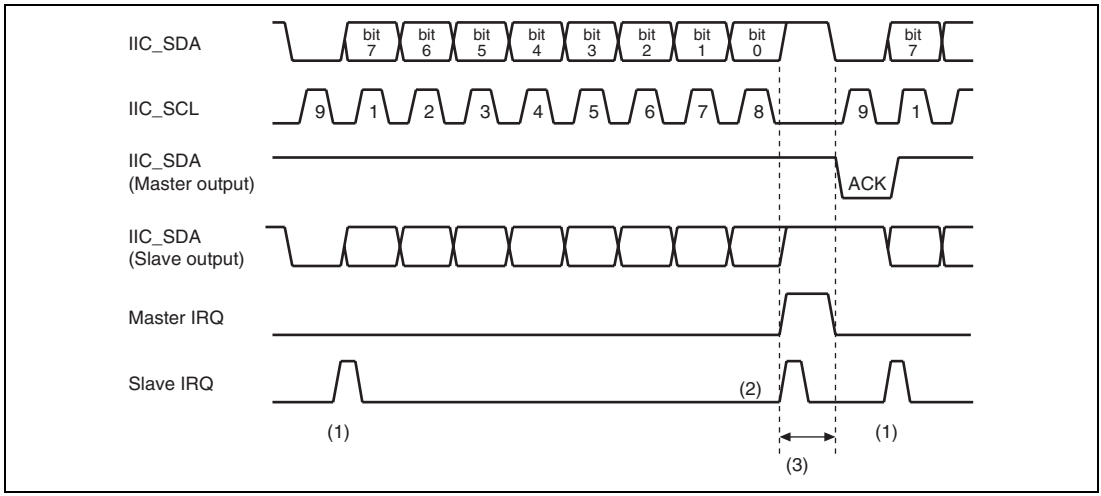


Figure 26.10 Data Receive Mode Operation Timing

26.5 Programming Examples

26.5.1 Master Transmitter

In order to set up the master interface to transmit a data packet on the I²C bus, follow the following procedure:

(1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = H'03
   (SCL frequency of 400 kHz)
2. Clock division ratio (CDF) = H'3
   (The peripheral clock is 66.7 MHz and the IIC's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register (First Data Byte and Address)

1. Master address register = address of slave being accessed and STM1 bit (write mode: 0)
2. Transmit data register = first data byte to be transmitted
3. Master control register = H'89
   (MDBS = 1, MIE = 1, ESG = 1)

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDE bits in the master status register).
2. Set the master control register to H'88 (To suspend the data transmission, the master device will hold the SCL low until the MDE bit is cleared.)
   If only one byte of data is transmitted, set the master control register to H'8A, meaning that the stop generation is enabled. This generates a stop on the bus as soon as one byte has been transmitted.
3. Reset the MAT bit.

(4) Monitor Transmission of Data

1. Wait for master event, MDE in the master status register.
2. Transmit data register = subsequent data.

3. Reset the MDE bit.

   Clear MDE after setting the last byte to be transmitted. After the last byte data is transmitted, MDE is generated. To clear the MDE, you must set the master control register to H'8A.

   (Set the force stop control bit).

(5) Wait for End of Transmission

1. Wait for the master event, MST in the master status register.
2. Reset the MST bit after confirming MNR (Master NACK Received).

26.5.2 Master Receiver

To set up the master interface to receive a data packet on the I²C bus, follow the following procedure:

(1) Load Clock Control Register

1. SCL clock generation divider (SCGD) = H'03

   (SCL frequency of 400 kHz).
2. Clock division ratio (CDF) = H'3

   (The peripheral clock is 66.7 MHz and the IIC's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register and Address

1. Set master address register to address of slave being accessed and STM1 bit (read mode: 1).
2. Set master control register to H'89

   (MDBS = 1, MIE = 1, ESG = 1).

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDR bits in the master status register).
2. Set the master control register to H'88

   (To suspend the data transmission, the master device will hold the SCL low until the MDR bit is cleared).

   If only one byte of data is received, set the master control register to H'8A, meaning that the stop generation is enabled. This generates a stop on the bus as soon as one byte has been received.
3. Reset the MAT bit.

(4) Monitor Reception of Data

1. Wait for master event, bit MDR in the master status register.
2. Read data from the received data register.
   If the next byte of data is the second to last byte to be transmitted by the slave device, the following applies to the receive interrupt (that is, MDR interrupt) in the second to last byte.
3. Set the master control register to H'8A
   (Set the force stop control bit).
4. Reset the MDR bit.

(5) Wait for End of Reception

1. Handle the receive interrupt (MDR) in the last byte: that is, read the data and clear the MDR.
2. Wait for master event, MST in the master status register.
3. Reset the MST bit.

26.5.3 Master Transmitter—Restart—Master Receiver

In order to set up the master interface to transmit a data packet on the I²C bus, issue a restart, then read byte data back from the slave, follow the following procedure:

(1) Load Clock Control Register

1. Set the SCL clock generation divider (SCGD) to H'03
   (SCL frequency of 400 kHz).
2. Set the clock division (CDF) to H'2

(The peripheral clock is 66.7 MHz and the IIC's internal clock IICck is 16.7 MHz.)

(2) Load Master Control Register and Address

1. Set the master address register to address of slave being accessed and STM1 bit (writes mode: 0).
2. Set the master control register to H'89
   (MDBS = 1, MIE = 1, ESG = 1).

(3) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDE bits in the master status register).

2. Set the master address register to address of slave being accessed and STM1 bit (read mode: 1).

   When the enable start generation bit in the master control register is still set, at the end of the byte transmission the master will issue a restart. Since the new address has been loaded above the bus direction will be changed.
3. Reset the MAT bit.

(4) Wait for Outputting Address

1. Wait for master event (an interrupt of the MAT and MDR bits in the master status register).
2. Set the master control register to H'88 (To suspend stop the data transmission, the master device will hold the IIC_SCL low until the MDR bit is cleared.)
3. Reset the MAT bit.

(5) Monitor of Data

1. Wait for master event, the MDR bit in the master status register.

   Read data from the received data register.

   If the next byte of data is the second to last byte but one to be transmitted by the slave device, the following applies to a receive interrupt (that is, MDR interrupt) in the second to last byte
2. Set the master control register to H'8A

   (set the force stop control bit).
3. Reset the MDR bit.

(6) Wait for End of Reception

1. Handle the receive interrupt (MDR) in the last byte: that is, read the data and clear the MDR.
2. Wait for the master event MST in the master status register.
3. Reset the MST bit.

Section 27 Serial Communication Interface with FIFO (SCIF)

This LSI is equipped with a 2-channel serial communication interface with built-in FIFO buffers (Serial Communication Interface with FIFO: SCIF). The SCIF can perform both asynchronous and clocked synchronous serial communications.

64-stage FIFO buffers are provided for both transmission and reception, enabling fast, efficient, and continuous communication.

Channels 0 and 1 of the SCIF have modem control functions ($\overline{\text{RTS}}$, $\overline{\text{CTS}}$).

27.1 Features

The SCIF has the following features.

- Asynchronous serial communication mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).

  There is a choice of 8 serial data transfer formats.

  — Data length: 7 or 8 bits
  — Stop bit length: 1 or 2 bits
  — Parity: Even/odd/none
  — Receive error detection: Parity, framing, and overrun errors
  — Break detection: A break is detected when a framing error lasts for more than 1 frame length at Space 0 (low level). When a framing error occurs, a break can also be detected by reading the SCIFn_RXD (n = 0, 1) pin level directly from the serial port register (SCSPTR).
- Clocked synchronous serial communication mode

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other LSIs that have a synchronous communication function.

  There is a single serial data communication format.

  — Data length: 8 bits
  — Receive error detection: Overrun errors

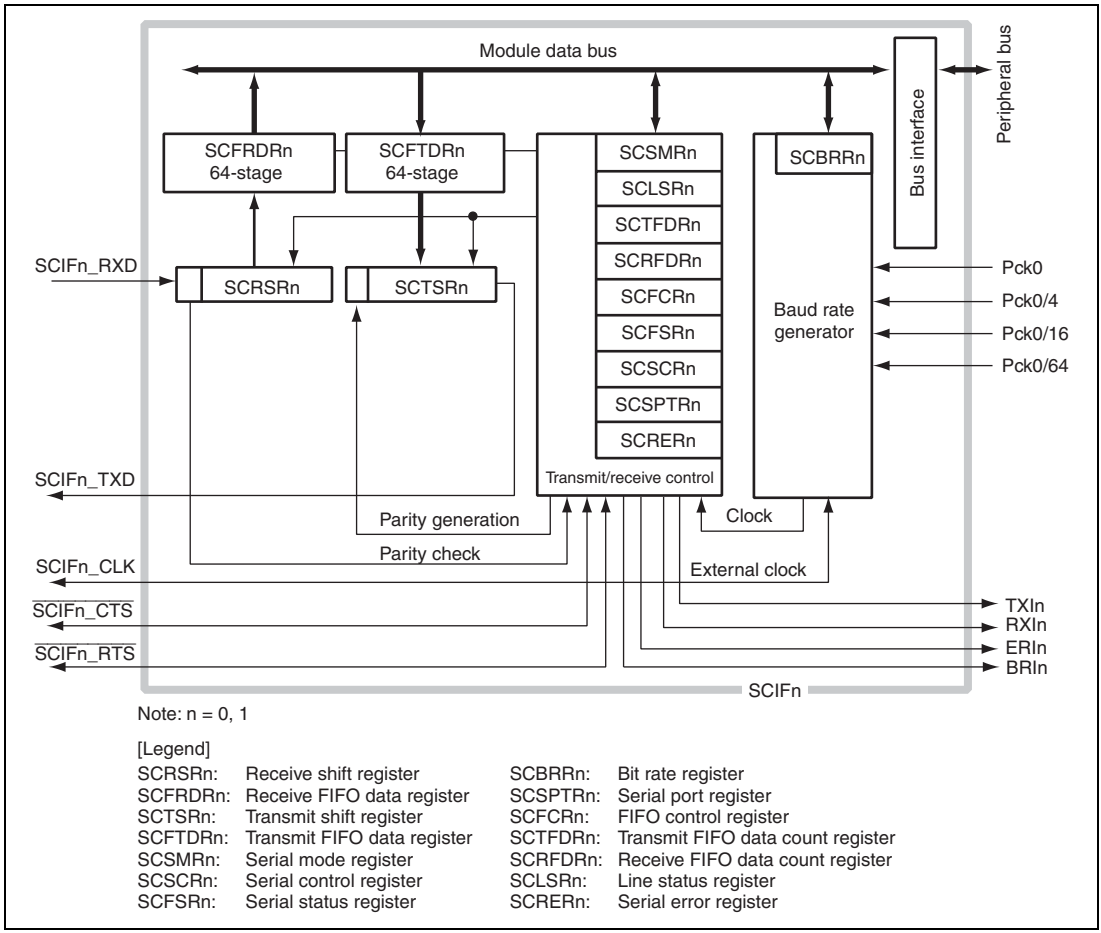
- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and reception to be performed simultaneously.

  The transmitter and receiver both have a 64-stage FIFO buffer structure, enabling continuous serial data transmission and reception.
- On-chip baud rate generator allows any bit rate to be selected.
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCIF_SCK0 or SCIF_SCK1 pin
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error—that can issue requests independently.
- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When not in use, the SCIF can be stopped by halting its clock supply to reduce power consumption.
- In asynchronous mode, modem control functions ($\overline{\text{SCIF0_RTS}}$, $\overline{\text{SCIF1_RTS}}$, $\overline{\text{SCIF0_CTS}}$, and $\overline{\text{SCIF1_CTS}}$) are provided.
- The amount of data in the transmit/receive FIFO registers, and the number of receive errors in the receive data in the receive FIFO register, can be ascertained.
- In asynchronous mode, a timeout error (DR) can be detected during reception.

Figure 27.1 shows a block diagram of the SCIF. Figures 27.2 to 27.6 show block diagrams of the I/O ports in SCIF. There are two channels in this LSI. In figures 27.1 to 27.6, the channels are omitted and explained.



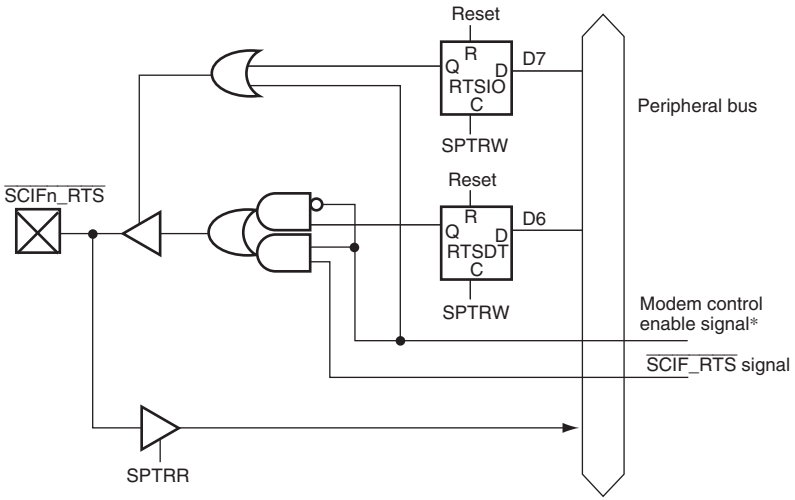
Note: n = 0, 1

[Legend]

SCRSRn:	Receive shift register	SCBRRn:	Bit rate register
SCFRDRn:	Receive FIFO data register	SCSPTRn:	Serial port register
SCTSRn:	Transmit shift register	SCFCRn:	FIFO control register
SCFTDRn:	Transmit FIFO data register	SCTFDRn:	Transmit FIFO data count register
SCSMRn:	Serial mode register	SCRFDRn:	Receive FIFO data count register
SCSCRn:	Serial control register	SCLSRn:	Line status register
SCFSRn:	Serial status register	SCRERn:	Serial error register

Figure 27.1 Block Diagram of SCIF

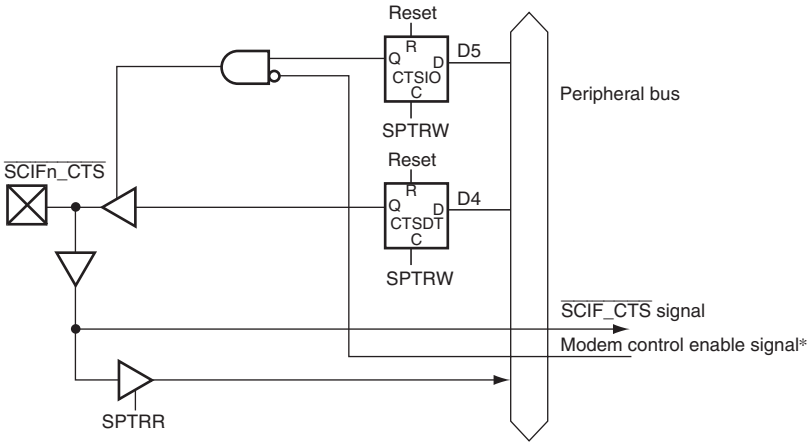
Figures 27.2 to 27.6 show block diagrams of the I/O ports in SCIF.



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Notes: n = 0, 1
∗ The SCIFn_RTS pin function is designated as modem control by the MCE bit in SCFCR.

Figure 27.2 SCIFn_RTS Pin (n = 0, 1)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Notes: n = 0, 1
* The SCIFn_CTS pin function is designated as modem control by the MCE bit in SCFCR.

Figure 27.3 SCIFn_CTS Pin (n = 0, 1)

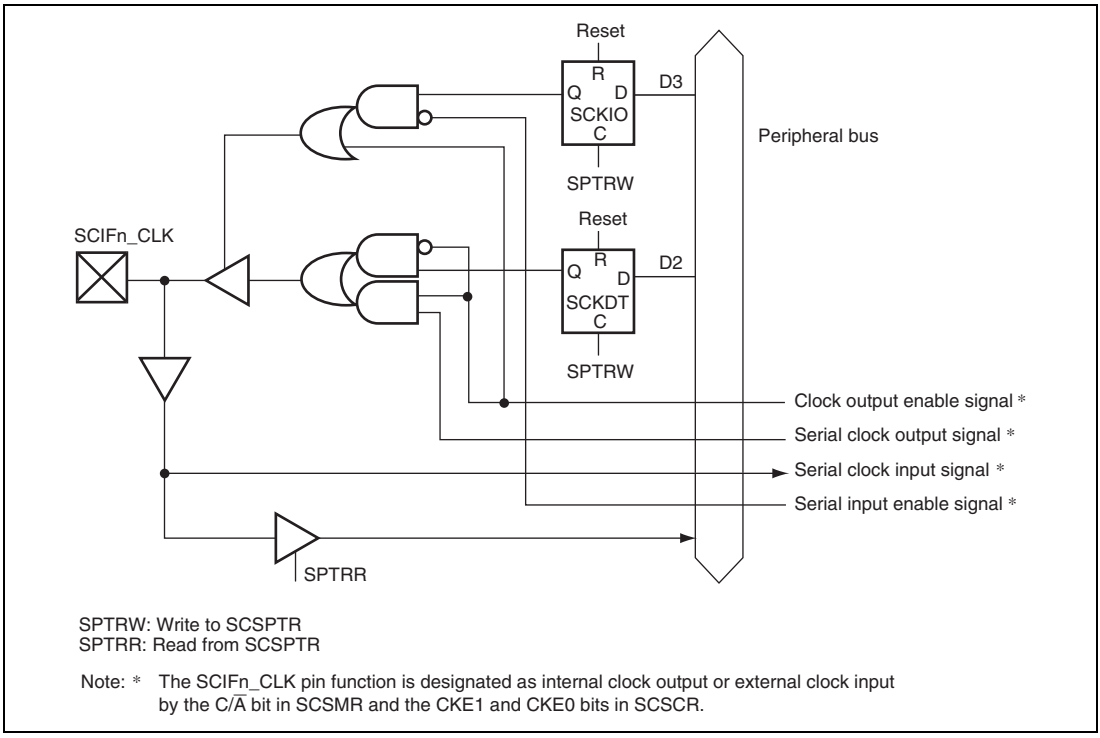


Figure 27.4 SCIFn_SCK Pin (n = 0, 1)

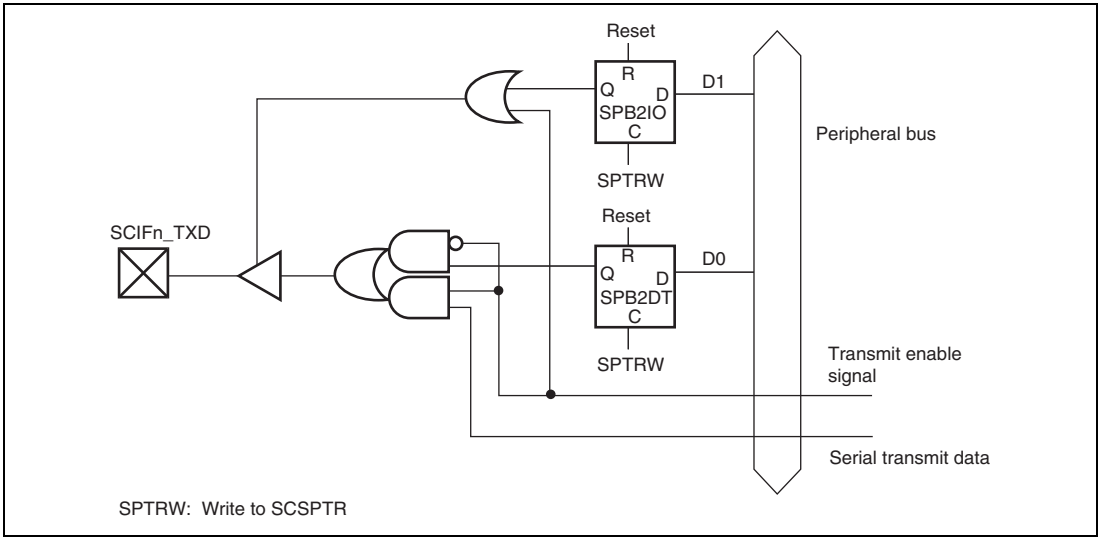


Figure 27.5 SCIFn_TXD Pin (n = 0, 1)

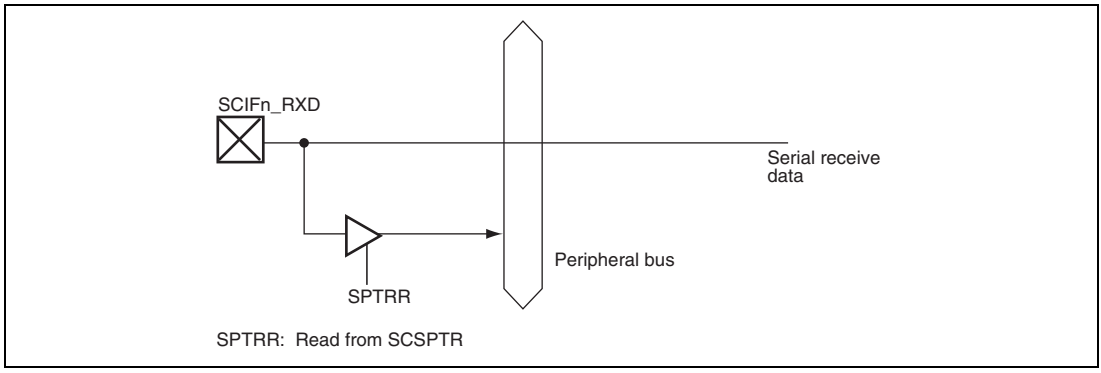


Figure 27.6 SCIFn_RXD Pin (n = 0, 1)

27.2 Input/Output Pins

Table 27.1 shows the SCIF pin configuration. Since the pin functions are the same in each channel, the channel number is omitted in the description below.

Table 27.1 Pin Configuration

Pin Name	Function	I/O	Description
SCIFn_SCK (n = 0, 1)	Serial clock pin	I/O	Clock input/output
SCIFn_RXD (n = 0, 1)	Receive data pin	Input	Receive data input
SCIFn_TXD (n = 0, 1)	Transmit data pin	Output	Transmit data output
$\overline{\text{SCIFn_CTS}}$ (n = 0, 1)	Modem control pin	I/O	Transmission enabled
$\overline{\text{SCIFn_RTS}}$ (n = 0, 1)	Modem control pin	I/O	Transmission request

Note: These pins are made to function as serial pins by performing SCIF operation settings with the C/$\overline{\text{A}}$ bit in SCSMR, the TE, RE, CKE1, and CKE0 bits in SCSCR, and the MCE bit in SCFCR. Break state transmission and detection can be set in SCSPTR of the SCIF.

27.3 Register Descriptions

The SCIF has the following registers. Since the register functions are the same in each channel, the channel number is omitted in the description below.

Table 27.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size
0	Serial mode register 0	SCSMR0	R/W	H'FFE0 0000	H'1FE0 0000	16
	Bit rate register 0	SCBRR0	R/W	H'FFE0 0004	H'1FE0 0004	8
	Serial control register 0	SCSCR0	R/W	H'FFE0 0008	H'1FE0 0008	16
	Transmit FIFO data register 0	SCFTDR0	W	H'FFE0 000C	H'1FE0 000C	8
	Serial status register 0	SCFSR0	R/W*[1]	H'FFE0 0010	H'1FE0 0010	16
	Receive FIFO data register 0	SCFRDR0	R	H'FFE0 0014	H'1FE0 0014	8
	FIFO control register 0	SCFCR0	R/W	H'FFE0 0018	H'1FE0 0018	16
	Transmit FIFO data count register 0	SCTFDR0	R	H'FFE0 001C	H'1FE0 001C	16
	Receive FIFO data count register 0	SCRFDR0	R	H'FFE0 0020	H'1FE0 0020	16
	Serial port register 0	SCSPTR0	R/W	H'FFE0 0024	H'1FE0 0024	16
	Line status register 0	SCLSR0	R/W*[2]	H'FFE0 0028	H'1FE0 0028	16
	Serial error register 0	SCRER0	R	H'FFE0 002C	H'1FE0 002C	16
1	Serial mode register 1	SCSMR1	R/W	H'FFE0 8000	H'1FE0 8000	16
	Bit rate register 1	SCBRR1	R/W	H'FFE0 8004	H'1FE0 8004	8
	Serial control register 1	SCSCR1	R/W	H'FFE0 8008	H'1FE0 8008	16
	Transmit FIFO data register 1	SCFTDR1	W	H'FFE0 800C	H'1FE0 800C	8
	Serial status register 1	SCFSR1	R/W*[1]	H'FFE0 8010	H'1FE0 8010	16
	Receive FIFO data register 1	SCFRDR1	R	H'FFE0 8014	H'1FE0 8014	8
	FIFO control register 1	SCFCR1	R/W	H'FFE0 8018	H'1FE0 8018	16
	Transmit FIFO data count register 1	SCTFDR1	R	H'FFE0 801C	H'1FE0 801C	16
	Receive FIFO data count register 1	SCRFDR1	R	H'FFE0 8020	H'1FE0 8020	16
	Serial port register 1	SCSPTR1	R/W	H'FFE0 8024	H'1FE0 8024	16
	Line status register 1	SCLSR1	R/W*[2]	H'FFE0 8028	H'1FE0 8028	16
	Serial error register 1	SCRER1	R	H'FFE0 802C	H'1FE0 802C	16

Notes: 1. To clear the flags, 0s can only be written to bits 7 to 4, 1, and 0.
2. To clear the flag, 0 can only be written to bit 0.

Table 27.3 Register State in Each Operating Mode

Ch.	Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Standby
0	Serial mode register 0	SCSMR0	H'0000	H'0000	Retained	Retained
	Bit rate register 0	SCBRR0	H'FF	H'FF	Retained	Retained
	Serial control register 0	SCSCR0	H'0000	H'0000	Retained	Retained
	Transmit FIFO data register 0	SCFTDR0	Undefined	Undefined	Retained	Retained
	Serial status register 0	SCFSR0	H'0060	H'0060	Retained	Retained
	Receive FIFO data register 0	SCFRDR0	Undefined	Undefined	Retained	Retained
	FIFO control register 0	SCFCR0	H'0000	H'0000	Retained	Retained
	Transmit FIFO data count register 0	SCTFDR0	H'0000	H'0000	Retained	Retained
	Receive FIFO data count register 0	SCRFDR0	H'0000	H'0000	Retained	Retained
	Serial port register 0	SCSPTR0	H'0000*[1]	H'0000*[1]	Retained	Retained
	Line status register 0	SCLSR0	H'0000	H'0000	Retained	Retained
	Serial error register 0	SCRER0	H'0000	H'0000	Retained	Retained
1	Serial mode register 1	SCSMR1	H'0000	H'0000	Retained	Retained
	Bit rate register 1	SCBRR1	H'FF	H'FF	Retained	Retained
	Serial control register 1	SCSCR1	H'0000	H'0000	Retained	Retained
	Transmit FIFO data register 1	SCFTDR1	Undefined	Undefined	Retained	Retained
	Serial status register 1	SCFSR1	H'0060	H'0060	Retained	Retained
	Receive FIFO data register 1	SCFRDR1	Undefined	Undefined	Retained	Retained
	FIFO control register 1	SCFCR1	H'0000	H'0000	Retained	Retained
	Transmit FIFO data count register 1	SCTFDR1	H'0000	H'0000	Retained	Retained
	Receive FIFO data count register 1	SCRFDR1	H'0000	H'0000	Retained	Retained
	Serial port register 1	SCSPTR1	H'0000*[2]	H'0000*[2]	Retained	Retained
	Line status register 1	SCLSR1	H'0000	H'0000	Retained	Retained
	Serial error register 1	SCRER1	H'0000	H'0000	Retained	Retained

Notes: 1. Bits 2 and 0 are undefined.
2. Bits 6, 4, 2,and 0 are undefined.

27.3.1 Receive Shift Register (SCRSR)

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the SCIF_RXD pin in SCRSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to SCFRDR, automatically.

SCRSR cannot be directly read from and written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

27.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is an 8-bit FIFO register of 64 stages that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SCRSR to SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled for reception, and consecutive receive operations can be performed until SCFRDR is full (64 data bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in SCFRDR, an undefined value will be returned. When SCFRDR is full of receive data, subsequent serial data is lost.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

27.3.3 Transmit Shift Register (SCTSR)

SCTSR is the register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTDR to SCTSR, then sends the data to the SCIF_TXD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from SCFTDR to SCTSR, and transmission started, automatically.

SCTSR cannot be directly read from and written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

27.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit FIFO register of 64 stages that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers the transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read by the CPU.

The next data cannot be written when SCFTDR is filled with 64 bytes of transmit data. Data written in this case is ignored.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

27.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial transfer format and select the baud rate generator clock source.

SCSMR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or clocked synchronous mode as the SCIF operating mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode data length. In clocked synchronous mode, the data length is fixed at 8 bits regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of SCFTDR is not transmitted. 0: 8-bit data 1: 7-bit data

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking is performed in reception. In clocked synchronous mode, parity bit addition and checking is disabled regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled* Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. In asynchronous mode, the O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking. In clocked synchronous mode or when parity addition and checking is disabled in asynchronous mode, the O/$\overline{E}$ bit setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is even. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is odd.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length In asynchronous mode, selects 1 or 2 bits as the stop bit length. The stop bit setting is valid only in asynchronous mode. Since the stop bit is not added in clocked synchronous mode, the STOP bit setting is invalid. 0: 1 stop bit*1 1: 2 stop bits*2 In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character. Note: 1. In transmission, a single 1-bit (stop bit) is added to the end of a transmit character before it is sent. 2. In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. The clock source can be selected from Pck0, Pck0/4, Pck0/16, and Pck0/64, according to the setting of bits CKS1 and CKS0. For details of the relationship between clock sources, bit rate register settings, and baud rate, see section 27.3.8, Bit Rate Register (SCBRR). 00: Pck0 clock 01: Pck0/4 clock 10: Pck0/16 clock 11: Pck0/64 clock

Note: Pck0 = Peripheral Clock0

27.3.6 Serial Control Register (SCSCR)

SCSCR is a register used to enable/disable transmission/reception by SCIF, serial clock output, interrupt requests, and to select transmission/reception clock source for the SCIF.

SCSCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when serial transmit data is transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR falls to or below the transmit trigger set number, and the TDFE flag in SCFSR is set to 1. TXI interrupt requests can be cleared using the following methods: Either by reading 1 from the TDFE flag, writing transmit data exceeding the transmit trigger set number to SCFTDR and then clearing the TDFE flag to 0, or by clearing the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag or DR flag in SCFSR is set to 1, a receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, and a break interrupt (BRI) request when the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1. 0: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request disabled 1: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request enabled Note: An RXI interrupt request can be cleared by reading 1 from the RDF or DR flag, then clearing the flag to 0, or by clearing the RIE bit to 0. ERI and BRI interrupt requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmission by the SCIF. Serial transmission is started when transmit data is written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled* Note: SCSMR and SCFCR settings must be made, the transmission format decided, and the transmit FIFO reset, before the TE bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the start of serial reception by the SCIF. Serial reception is started when a start bit is detected in this state in asynchronous mode or a synchronization clock is input while the RE bit is set to 1. It should be noted that clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER, and ORER flags, which retain their states. Serial reception begins once the start bit is detected in these states. 0: Reception disabled 1: Reception enabled* Note: * SCSMR and SCFCR settings must be made, the reception format decided, and the receive FIFO reset, before the RE bit is set to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-error interrupt (ERI) and break interrupt (BRI) requests. The REIE bit setting is valid only when the RIE bit is 0. Receive-error interrupt (ERI) and break interrupt (BRI) requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0. When REIE is set to 1, ERI and BRI interrupt requests will be generated even if RIE is cleared to 0. In DMAC transfer, this setting is made if the interrupt controller is to be notified of ERI and BRI interrupt requests. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	These bits select the SCIF clock source and whether to enable or disable the clock output from the SCIF_SCK pin. The CKE1 and CKE0 bits are used together to specify whether the SCIF_SCK pin functions as a serial clock output pin or a serial clock input pin. Note however that the CKE0 bit setting is valid only when an internal clock is selected as the SCIF clock source (CKE1 = 0). When an external clock is selected (CKE1 = 1), the CKE0 bit setting is invalid. The CKE1 and CKE0 bits must be set before determining the SCIF's operating mode with SCSMR. • Asynchronous mode 00: Internal clock/SCIF_SCK pin functions as port 01: Internal clock/SCIF_SCK pin functions as clock output*1 1x: External clock/SCIF_SCK pin functions as clock input*2 • Clocked synchronous mode 0x: Internal clock/SCIF_SCK pin functions as synchronization clock output 1x: External clock/SCIF_SCK pin functions as synchronization clock input

Notes: x: Don't care

1. Outputs a clock with a frequency 16 times the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

27.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register that consists of status flags that indicate the operating status of the SCIF.

SCFSR can be read from or written to by the CPU at all times. However, 1 cannot be written to flags ER, TEND, TDFE, BRK, RDF, and DR. Also note that in order to clear these flags they must be read as 1 beforehand. The FER flag and PER flag are read-only flags and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R	R	R/W*	R/W*

Note: * Only 0 can be written, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/W*[1]	Receive Error Indicates that a framing error or parity error occurred during reception. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a receive error occurs, the receive data is still transferred to SCFRDR, and reception continues. The FER and PER bits in SCFSR can be used to determine whether there is a receive error in the readout data from SCFRDR. 0: No framing error or parity error occurred during reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during reception [Setting conditions] • When the SCIF checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0*[2] • When, in reception, the number of 1-bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SCSMR
6	TEND	1	R/W*[1]	Transmit End Indicates that transmission has been ended without valid data in SCFTDR after transmission of the last bit of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, and 0 is written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DMAC 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR after transmission of the last bit of a 1-byte serial transmit character

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/W*[1]	Transmit FIFO Data Empty Indicates that data has been transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR has fallen to or below the transmit trigger data number set by bits TTRG1 and TTRG0 in SCFCR, and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the transmit trigger set number have been written to SCFTDR [Clearing conditions] • When transmit data exceeding the transmit trigger set number is written to SCFTDR after reading TDFE = 1, and 0 is written to TDFE • When transmit data exceeding the transmit trigger set number is written to SCFTDR by the DMAC 1: The number of transmit data bytes in SCFTDR does not exceed the transmit trigger set number (Initial value) [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger set number as the result of a transmit operation*[3]
4	BRK	0	R/W*[1]	Break Detect Indicates that a receive data break signal has been detected. 0: A break signal has not been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after reading BRK = 1 1: A break signal has been received*[4] [Setting condition] • When data with a framing error is received, followed by the space "0" level (low level) for at least one frame length

Bit	Bit Name	Initial Value	R/W	Description
3	FER	0	R	Framing Error In asynchronous mode, indicates whether or not a framing error has been found in the data that is to be read next from SCFRDR. 0: There is no framing error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data that is to be read next from SCFRDR 1: There is a framing error that is to be read from SCFRDR [Setting condition] • When there is a framing error in the data that is to be read next from SCFRDR
2	PER	0	R	Parity Error In asynchronous mode, indicates whether or not a parity error has been found in the data that is to be read next from SCFRDR. 0: There is no parity error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in the data that is to be read next from SCFRDR 1: There is a parity error in the receive data that is to be read from SCFRDR [Setting condition] • When there is a parity error in the data that is to be read next from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/W*[1]	Receive FIFO Data Full Indicates that the received data has been transferred from SCRSR to SCFRDR, and the number of receive data bytes in SCFRDR is equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. 0: The number of receive data bytes in SCFRDR is less than the receive trigger set number [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger set number after reading RDF = 1, and 0 is written to RDF • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR falls below the receive trigger set number 1: The number of receive data bytes in SCFRDR is equal to or greater than the receive trigger set number [Setting condition] • When SCFRDR contains at least the receive trigger set number of receive data bytes*[5]

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/W*[1]	Receive Data Ready In asynchronous mode, indicates that there are fewer than the receive trigger set number of data bytes in SCFRDR, and no further data has arrived for at least 15 etu after the stop bit of the last data received. This is not set when using clocked synchronous mode. 0: Reception is in progress or has ended normally and there is no receive data left in SCFRDR [Clearing conditions] • Power-on reset or manual reset • When all the receive data in SCFRDR has been read after reading DR = 1, and 0 is written to DR • When all the receive data in SCFRDR has been read by the DMAC 1: No further receive data has arrived [Setting condition] • When SCFRDR contains fewer than the receive trigger set number of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received*[6]

[Legend] etu: Elementary time unit (time for transfer of 1 bit)

Notes: 1. Only 0 can be written, to clear the flag.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the second stop bit is not checked.

3. As SCFTDR is a 64-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 64 − (transmit trigger set number). Data written in excess of this will be ignored.

   SCTFDR indicates the number of data bytes transmitted to SCFTDR.

4. When a break is detected, the receive data (H'00) following detection is not transferred to SCFRDR. When the break ends and the receive signal returns to mark "1", receive data transfer is resumed.

5. SCFRDR is a 64-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If all the data in SCFRDR is read and another read is performed, the data value will be undefined. The number of receive data bytes in SCFRDR is indicated by SCRFDR.

6. Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.

27.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that set the serial transmission/reception bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

SCBRR can always be read from and written to by the CPU.

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{Pck0}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{Pck0}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pck0: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See table 27.4 for the relation between n and the clock.)

Table 27.4 SCSMR Settings

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pck0	0	0
1	Pck0/4	0	1
2	Pck0/16	1	0
3	Pck0/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{Pck0 \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

27.3.9 FIFO Control Register (SCFCR)

SCFCR performs data count resetting and trigger data number setting for transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFCL	RFCL	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	$\overline{\text{SCIF_RTS}}$ Output Active Trigger
9	RSTRG1	0	R/W	The $\overline{\text{SCIF_RTS}}$ signal becomes high when the number of receive data stored in SCFRDR exceeds the trigger number shown below.
8	RSTRG0	0	R/W	000:63 001:1 010:8 011:16 100:32 101:48 110:54 111:60
7	RTRG1	0	R/W	Receive FIFO Data Number Trigger
6	RTRG0	0	R/W	These bits are used to set the number of receive data bytes that sets the RDF flag in SCFSR. The RDF flag is set when the number of receive data bytes in SCFRDR is equal to or greater than the trigger set number shown below. 00:1 01:16 10:32 11:48

Bit	Bit Name	Initial Value	R/W	Description
5	TTRG1	0	R/W	Transmit FIFO Data Number Trigger
4	TTRG0	0	R/W	These bits are used to set the number of remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when the number of transmit data bytes in SCFTDR is equal to or less than the trigger set number shown below. 00: 32 (32)*1 01:16 (48) 10: 2 (62) 11: 0 (64)
3	MCE	0	R/W	Modem Control Enable Enables the $\overline{\text{SCIF_CTS}}$ and $\overline{\text{SCIF_RTS}}$ modem control signals. Always set the MCE bit to 0 in clocked synchronous mode. 0: Modem signals disabled*2 1: Modem signals enabled
2	TFCL	0	R/W	Transmit FIFO Data Register Reset Invalidates the transmit data in the transmit FIFO data register and resets it to the empty state. 0: Reset operation disabled*3 1: Reset operation enabled
1	RFCL	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in the receive FIFO data register and resets it to the empty state. 0: Reset operation disabled*3 1: Reset operation enabled
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SCIF_TXD) and receive input pin (SCIF_RXD), and the $\overline{\text{SCIF_RTS}}$ pin and $\overline{\text{SCIF_CTS}}$ pin, enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Notes: 1. Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.
2. $\overline{\text{SCIF_CTS}}$ is fixed at active-0 regardless of the input value, and $\overline{\text{SCIF_RTS}}$ output is also fixed at 0.
3. A reset operation is performed in the event of a power-on reset or manual reset.

27.3.10 Transmit FIFO Data Count Register (SCTFDR)

SCTFDR is a 16-bit register that indicates the number of transmit data bytes stored in SCFTDR.

SCTFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	T6	T5	T4	T3	T2	T1	T0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	T6 to T0	All 0	R	These bits show the number of untransmitted data bytes in SCFTDR. A value of H'0000 indicates that there is no transmit data, and a value of H'0040 indicates that SCFTDR is full of transmit data.

27.3.11 Receive FIFO Data Count Register (SCRFDR)

SCRFDR is a 16-bit register that indicates the number of receive data bytes stored in SCFRDR.

SCRFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	R6	R5	R4	R3	R2	R1	R0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	R6 to R0	All 0	R	These bits show the number of receive data bytes in SCFRDR. A value of H'0000 indicates that there is no receive data, and a value of H'0040 indicates that SCFRDR is full of receive data.

27.3.12 Serial Port Register (SCSPTR)

SCSPTR is a 16-bit readable/writable register that controls input/output and data for the port pins multiplexed with the serial communication interface (SCIF) pins at all times. Input data can be read from the SCIF_RXD pin, output data written to the SCIF_TXD pin, and breaks in serial transmission/reception controlled, by means of bits 1 and 0.

All SCSPTR bits except bits 6, 4, 2, and 0 are initialized to 0 by a power-on reset or manual reset; the value of bits 6, 4, 2, and 0 is undefined. SCSPTR is not initialized in the module standby state.

Note that when reading data via a serial port pin in the SCIF, the peripheral clock value from 2 cycles before is read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RTS IO	RTS DT	CTS IO	CTS DT	SCK IO	SCK DT	SPB2 IO	SPB2 DT
Initial value:	0	0	0	0	0	0	0	0	0	—	0	—	0	—	0	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO	0	R/W	Serial Port $\overline{\text{SCIF_RTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF_RTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF_RTS}}$ pin as a port output pin to output the value set by the RTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value is not output to $\overline{\text{SCIF_RTS}}$ pin 1: RTSDT bit value is output to $\overline{\text{SCIF_RTS}}$ pin
6	RTSDT	—	R/W	Serial Port $\overline{\text{SCIF_RTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF_RTS}}$ pin input/output data. Input or output is specified by the RTSIO bit. In output mode, the RTSDT bit value is output to the $\overline{\text{SCIF_RTS}}$ pin. The $\overline{\text{SCIF_RTS}}$ pin value is read from the RTSDT bit regardless of the value of the RTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Bit	Bit Name	Initial Value	R/W	Description
5	CTSIO	0	R/W	Serial Port $\overline{\text{SCIF_CTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF_CTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF_CTS}}$ pin as a port output pin to output the value set by the CTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value is not output to $\overline{\text{SCIF_CTS}}$ pin 1: CTSDT bit value is output to $\overline{\text{SCIF_CTS}}$ pin
4	CTSDT	—	R/W	Serial Port $\overline{\text{SCIF_CTS}}$ Port Data Specifies the serial port $\overline{\text{SCIF_CTS}}$ pin input/output data. Input or output is specified by the CTSIO bit. In output mode, the CTSDT bit value is output to the $\overline{\text{SCIF_CTS}}$ pin. The $\overline{\text{SCIF_CTS}}$ pin value is read from the CTSDT bit regardless of the value of the CTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
3	SCKIO	0	R/W	Serial Port Clock Port Input/Output Specifies the serial port SCIF_SCK pin input/output condition. When actually setting the SCIF_SCK pin as a port output pin to output the value set by the SCKDT bit, the CKE1 and CKE0 bits in SCSCR should be cleared to 0. 0: SCKDT bit value is not output to SCIF_SCK pin 1: SCKDT bit value is output to SCIF_SCK pin
2	SCKDT	—	R/W	Serial Port Clock Port Data Specifies the serial port SCIF_SCK pin input/output data. Input or output is specified by the SCKIO bit. In output mode, the SCKDT bit value is output to the SCIF_SCK pin. The SCIF_SCK pin value is read from the SCKDT bit regardless of the value of the SCKIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	Serial Port Break Input/Output Specifies the serial port SCIF_TXD pin output condition. When actually setting the SCIF_TXD pin as a port output pin to output the value set by the SPB2DT bit, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_TXD pin 1: SPB2DT bit value is output to the SCIF_TXD pin
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input data and SCIF_TXD pin output data. The SCIF_TXD pin output condition is specified by the SPB2IO bit. When the SCIF_TXD pin is designated as an output, the value of the SPB2DT bit is output to the SCIF_TXD pin. The SCIF_RXD pin value is read from the SPB2DT bit regardless of the value of the SPB2IO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

27.3.13 Line Status Register (SCLSR)

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

Note: * Only 0 can be written, to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/W*[1]	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Reception in progress, or reception has ended normally*[2] [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1 1: An overrun error occurred during reception*[3] [Setting condition] • When the next serial reception is completed while SCFRDR receives 64-byte data (SCFRDR is full)

Notes: 1. Only 0 can be written, to clear the flag.
2. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0.
3. The receive data prior to the overrun error is retained in SCFRDR, and the data received subsequently is lost. Serial reception cannot be continued while the ORER flag is set to 1.

27.3.14 Serial Error Register (SCRER)

SCRER is a 16-bit register that indicates the number of receive errors in the data in SCFRDR. SCRER can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PER5	PER4	PER3	PER2	PER1	PER0	—	—	FER5	FER4	FER3	FER2	FER1	FER0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PER5	0	R	Number of Parity Errors These bits indicate the number of data bytes in which a parity error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits PER5 to PER0 is the number of data bytes in which a parity error occurred. If all 64 bytes of receive data in SCFRDR have parity errors, the value indicated by bits PER5 to PER0 will be 0.
12	PER4	0	R	
11	PER3	0	R	
10	PER2	0	R	
9	PER1	0	R	
8	PER0	0	R	
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	FER5	0	R	Number of Framing Errors These bits indicate the number of data bytes in which a framing error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits FER5 to FER0 is the number of data bytes in which a framing error occurred. If all 64 bytes of receive data in SCFRDR have framing errors, the value indicated by bits FER5 to FER0 will be 0.
4	FER4	0	R	
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

27.4 Operation

27.4.1 Overview

The SCIF can carry out serial communication in asynchronous mode, in which synchronization is achieved character by character and in synchronous mode, in which synchronization is achieved with clock pulses. For details on asynchronous mode, see section 27.4.2, Operation in Asynchronous Mode.

64-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead, and enabling fast and continuous communication to be performed.

$\overline{\text{SCIF_RTS}}$ and $\overline{\text{SCIF_CTS}}$ signals are also provided as modem control signals (channel 0 only).

The serial transfer format is selected using SCSMR, as shown in table 27.5. The SCIF clock source is determined by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as shown in table 27.6.

Asynchronous Mode:

- Data length: Choice of 7 or 8 bits
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun errors, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Choice of internal or external clock as SCIF clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock and can output a clock with frequency of 16 times the bit rate.

  When external clock is selected: A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used).

Clocked Synchronous Mode:

- Data length: Fixed at 8 bits
- Detection of overrun errors during reception
- Choice of internal or external clock as SCIF clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock and a serial clock is output to external devices.

When external clock is selected: The on-chip baud rate generator is not used and the SCIF operates on the input serial clock.

Table 27.5 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					SCIF Transfer Format		
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous mode	8-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
	1	0	0		7-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous mode	8-bit data	No	No

Note: x: Don't care

Table 27.6 SCSMR and SCSCR Settings for SCIF Clock Source Selection

SCSMR	SCSCR Settings				
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCK Pin Function**
0	0	0	Asynchronous mode	Internal	SCIF does not use SCIF_SCK pin
		1			Outputs clock with frequency of 16 times the bit rate
	1	0		External	Inputs clock with frequency of 16
		1			times the bit rate
1	0	0	Clocked synchronous mode	Internal	Outputs synchronization clock
		1			
	1	0		External	Inputs synchronization clock
		1			

27.4.2 Operation in Asynchronous Mode

In asynchronous mode, a character that consists of data with a start bit indicating the start of communication and a stop bit indicating the end of communication is transmitted or received. In this mode, serial communication is performed with synchronization achieved character by character.

Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

Figure 27.7 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCIF monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One character in serial communication consists of a start bit (low level), followed by transmit/receive data (LSB-first; from the lowest bit), a parity bit (high or low level), and finally stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. Receive data can be latched at the middle of each bit because the SCIF samples data at the eighth clock which has a frequency of 16 times the bit rate.

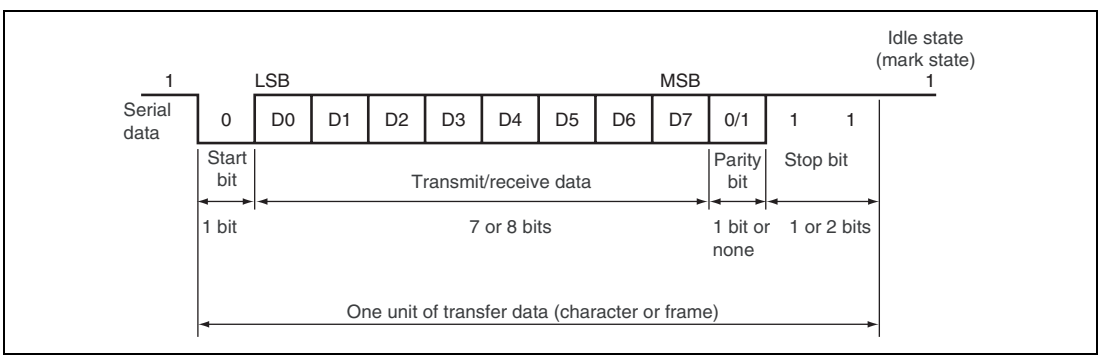


Figure 27.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

(1) Data Transfer Format

Table 27.7 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 27.7 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
1	0	1	S	7-bit data							STOP	STOP		
1	1	0	S	7-bit data							P	STOP		
1	1	1	S	7-bit data							P	STOP	STOP	

[Legend]
S : Start bit
STOP : Stop bit
P : Parity bit

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 27.6.

When an external clock is input at the SCIF_SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCIF is operated on an internal clock, a clock whose frequency is 16 times the bit rate is output from the SCIF_SCK pin.

(3) SCIF Initialization (Asynchronous Mode)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When the operating mode or transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure.

1. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR.
2. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND flag in SCFSR has been set. TEND can also be cleared to 0 during transmission, but the data being transmitted will go to the mark state after the clearance. Before setting TE again to start transmission, the TFCL bit in SCFCR should first be set to 1 to reset SCFTDR.
3. When an external clock is used the clock should not be stopped during operation, including initialization, since operation will be unreliable in this case.

Figure 27.8 shows a sample SCIF initialization flowchart.

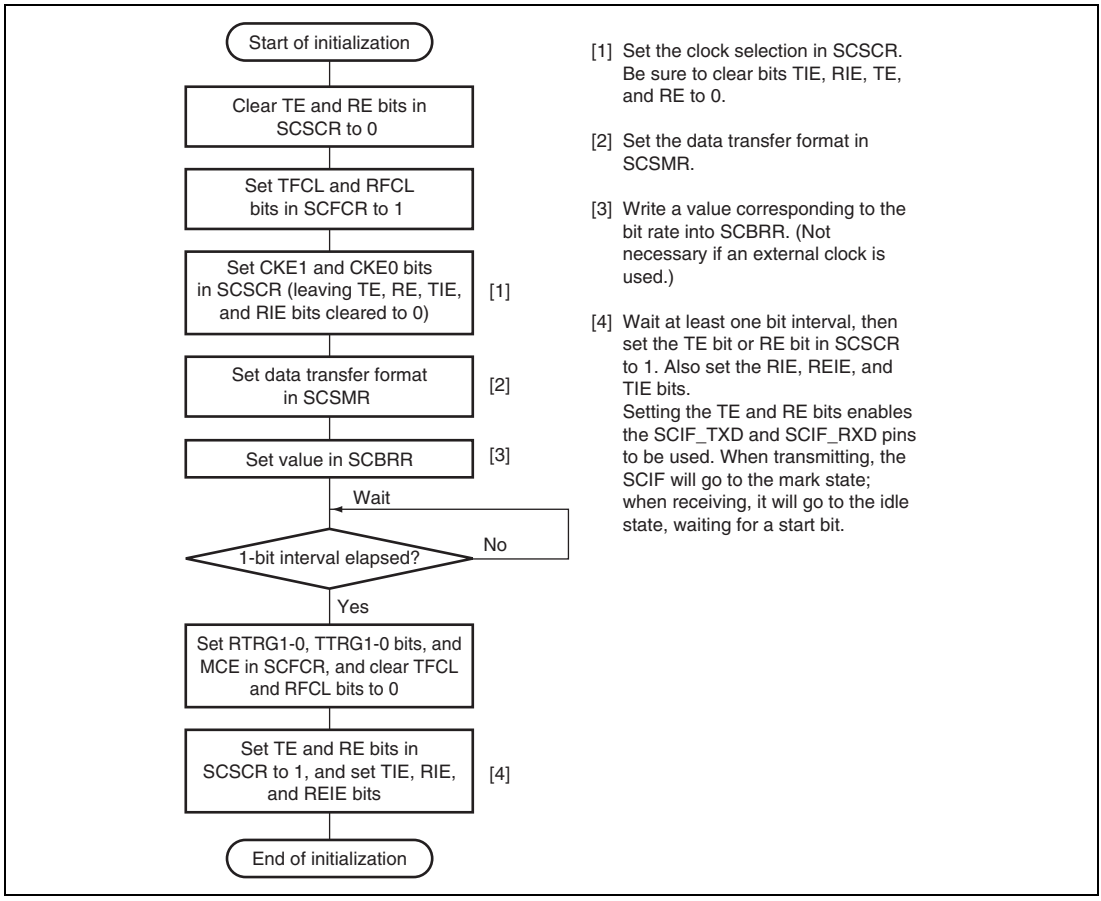


Figure 27.8 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Asynchronous Mode):

Figure 27.9 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

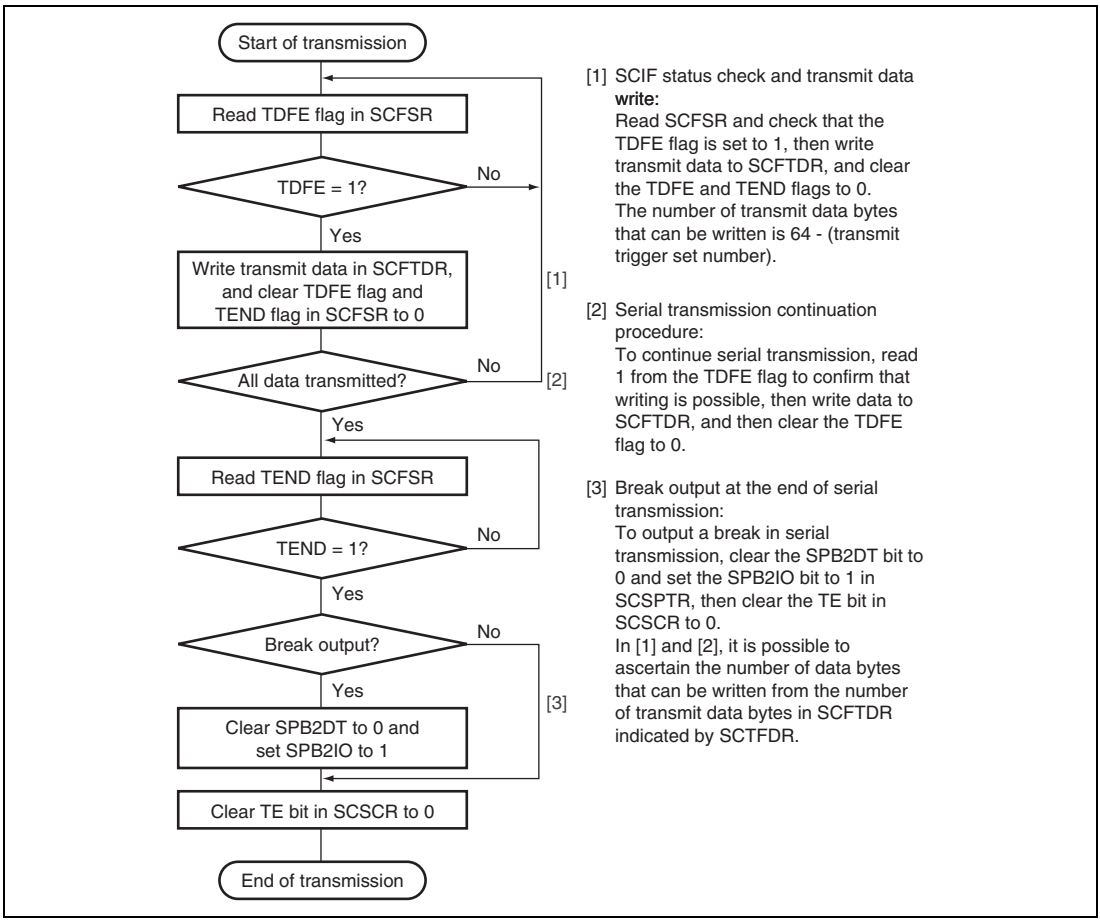


Figure 27.9 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 − (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the SCIF_TXD pin in the following order.

   (a) Start bit: One 0-bit is output.

   (b) Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   (c) Parity bit: One parity bit (even or odd parity) is output. A format in which a parity bit is not output can also be selected.

   (d) Stop bit(s): One or two 1-bits (stop bits) are output.

   (e) Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output from the SCIF_TXD pin.

Figure 27.10 shows an example of the operation for transmission in asynchronous mode.

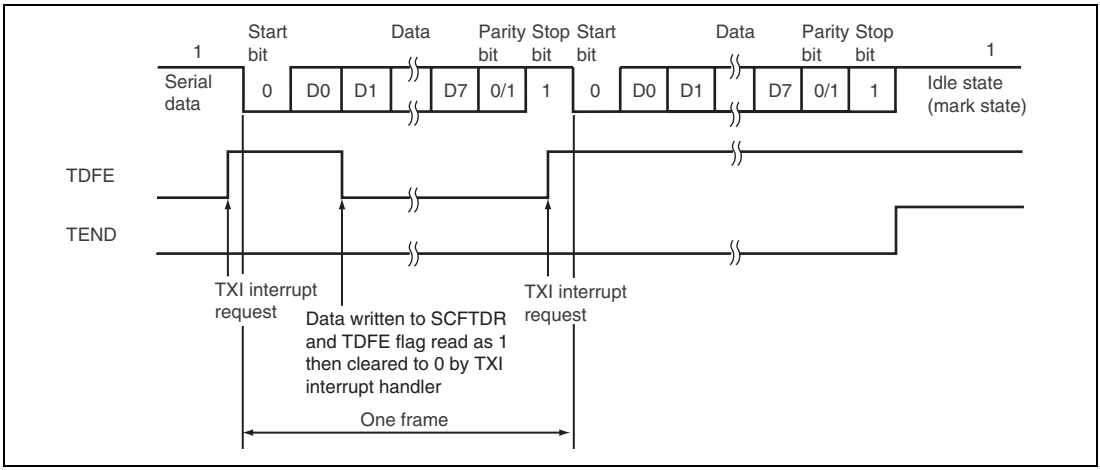


Figure 27.10 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{SCIF_CTS}}$ input value. When $\overline{\text{SCIF_CTS}}$ is set to 1 during transmission , the line goes to the mark state after transmission of one frame. When $\overline{\text{SCIF_CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 27.11 shows an example of the operation when modem control is used.

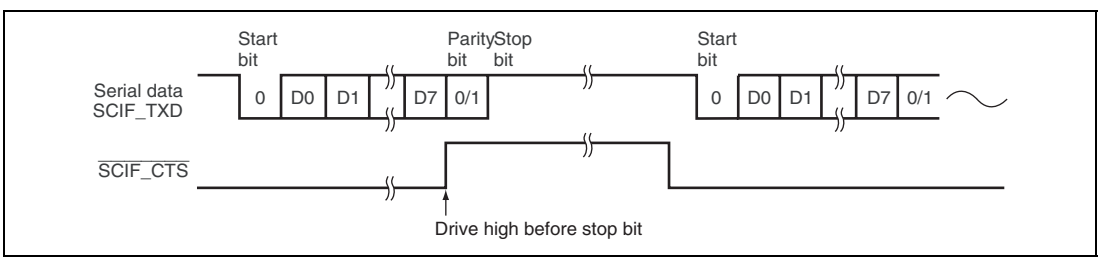


Figure 27.11 Sample Operation Using Modem Control ($\overline{\text{SCIF_CTS}}$)

(5) Serial Data Reception (Asynchronous Mode)

Figure 27.12 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

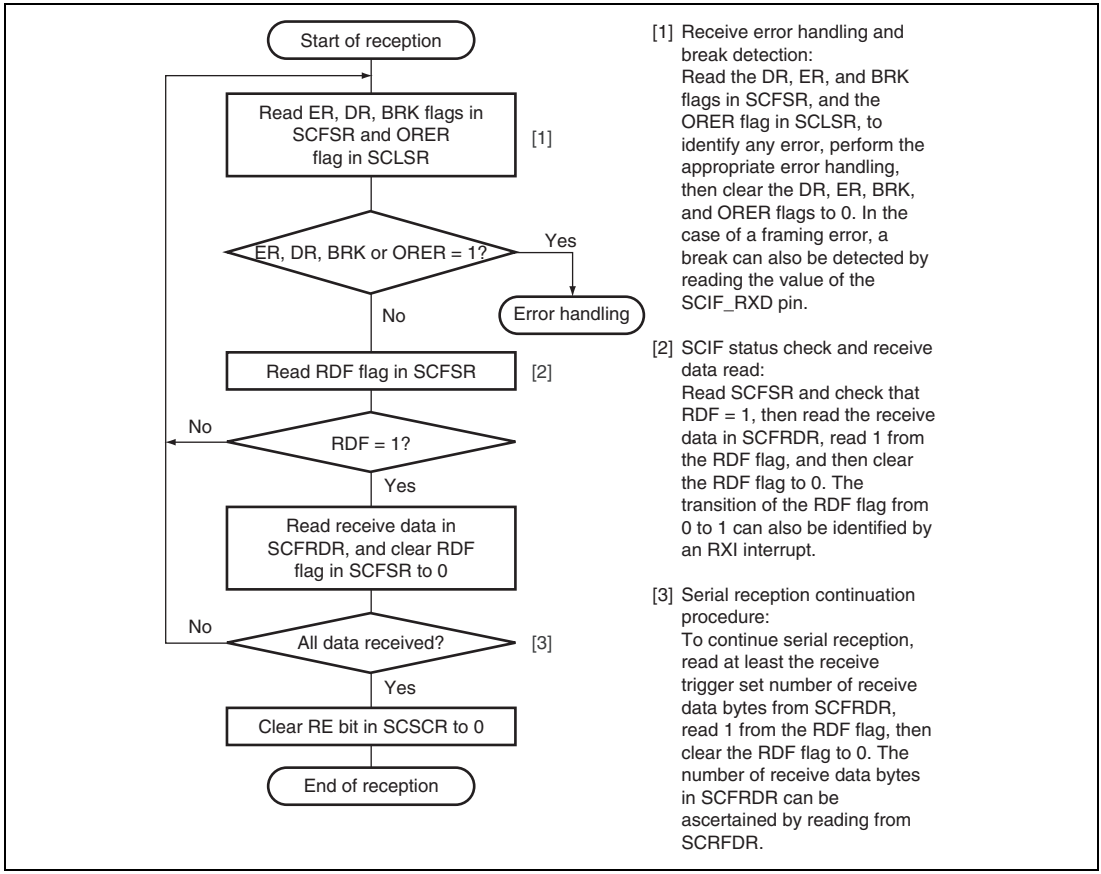
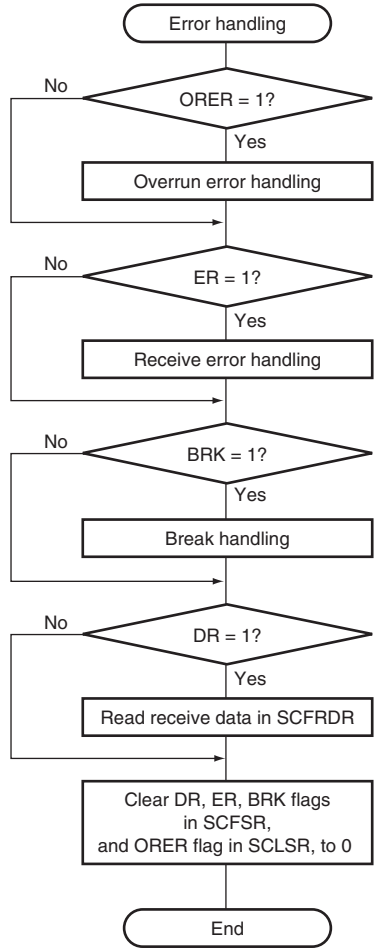


Figure 27.12 Sample Serial Reception Flowchart (1)



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 27.12 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0-start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCIF carries out the following checks.

   (a) Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   (b) The SCIF checks whether receive data can be transferred from SCRSR to SCFRDR.*

   (c) Overrun error check: The SCIF checks that the ORER flag is 0, indicating that no overrun error has occurred.*

   (d) Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.*

   If (b), (c), and (d) checks are passed, the receive data is stored in SCFRDR.

   Note: * Reception continues even when a parity error or framing error occurs.
4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 27.13 shows an example of the operation for reception in asynchronous mode.

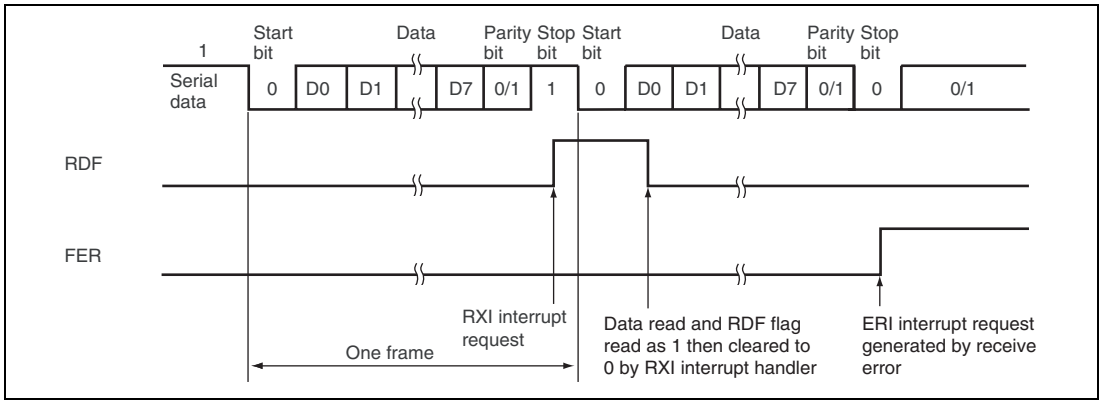


Figure 27.13 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{SCIF_RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{SCIF0_RTS}}$ is 0, reception is possible. When $\overline{\text{SCIF_RTS}}$ is 1, this indicates that SCFRDR contains bytes of data equal to or more than the $\overline{\text{SCIF_RTS}}$ output active trigger number. The $\overline{\text{SCIF_RTS}}$ output active trigger value is specified by bits 10 to 8 in the FIFO control register (SCFCR). For details, see section 27.3.9, FIFO control register (SCFCR). In addition, $\overline{\text{SCIF_RTS}}$ is also 1 when the RE bit in SCSCR is cleared to 0.

Figure 27.14 shows an example of the operation when modem control is used.

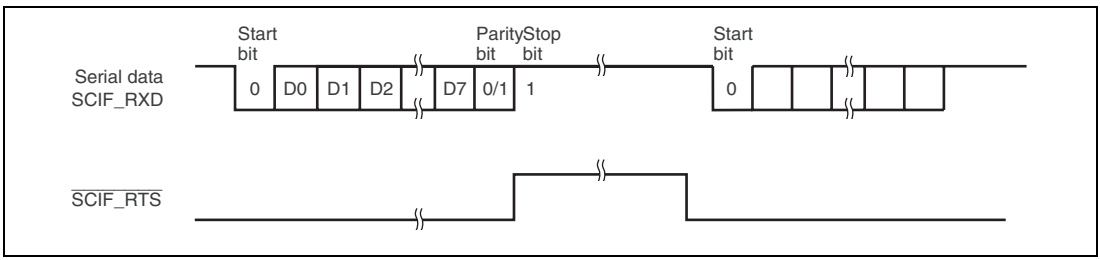


Figure 27.14 Sample Operation Using Modem Control ($\overline{\text{SCIF0_RTS}}$) (Only in Channel 0)

27.4.3 Operation in Clocked Synchronous Mode

Clocked synchronous mode, in which data is transmitted or received in synchronization with clock pulses, is suitable for fast serial communication.

Since the transmitter and receiver are independent units in the SCIF, full-duplex communication can be achieved by sharing the clock. Both the transmitter and receiver have a 64-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

Figure 27.15 shows the general format for clocked synchronous communication.

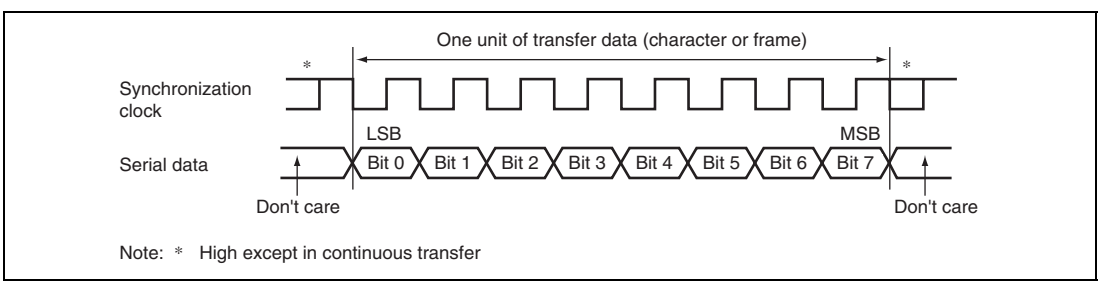


Figure 27.15 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, data on the communication line is output from one fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the start of the synchronization clock.

In serial communication, each character is output starting with the LSB and ending with the MSB. After the MSB is output, the communication line remains in the state of the last data.

In clocked synchronous mode, the SCIF receives data in synchronization with the rise of the synchronization clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity bit can be added.

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 27.6.

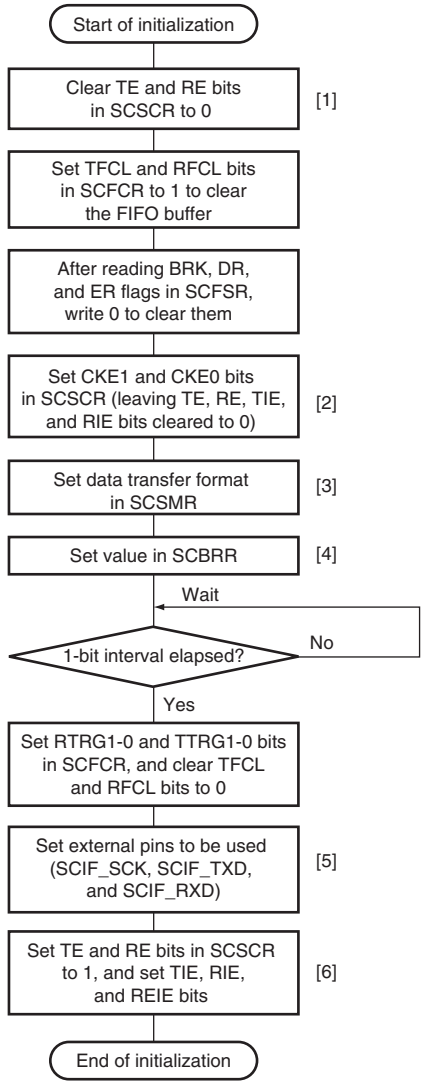
When the SCIF is operated on an internal clock, the synchronization clock is output from the SCIF_SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When an internal clock is selected in a receive operation only, as long as the RE bit in SCSCR is set to 1, clock pulses are output until the number of receive data bytes in the receive FIFO data register reaches the receive trigger number.

(3) SCIF Initialization (Clocked Synchronous Mode):

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When changing the operating mode or transfer format, etc., the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, FER, or ORER flag state or change the contents of SCFRDR.

Figure 27.16 shows a sample SCIF initialization flowchart.



[1] Leave the TE and RE bits cleared to 0 until the initialization almost ends. Be sure to clear the TIE, RIE, TE, and RE bits to 0.

[2] Set the CKE1 and CKE0 bits.

[3] Set the data transfer format in SCSMR.

[4] Write a value corresponding to the bit rate into SCBRR. This is not necessary if an external clock is used. Wait at least one bit interval after this write before moving to the next step.

[5] Set the external pins to be used. Set SCIF_RXD input for reception and SCIF_TXD output for transmission. The input/output of the SCIF_SCK pin must match the setting of the CKE1 and CKE0 bits.

[6] Set the TE or RE bit in SCSCR to 1. Also set the TIE, RIE, and REIE bits to enable the SCIF_TXD, SCIF_RXD, and SCIF_SCK pins to be used. When transmitting, the SCIF_TXD pin will go to the mark state. When receiving in clocked synchronous mode with the synchronization clock output (clock master) selected, a clock starts to be output from the SCIF_SCK pin at this point.

Figure 27.16 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Clocked Synchronous Mode)

Figure 27.17 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

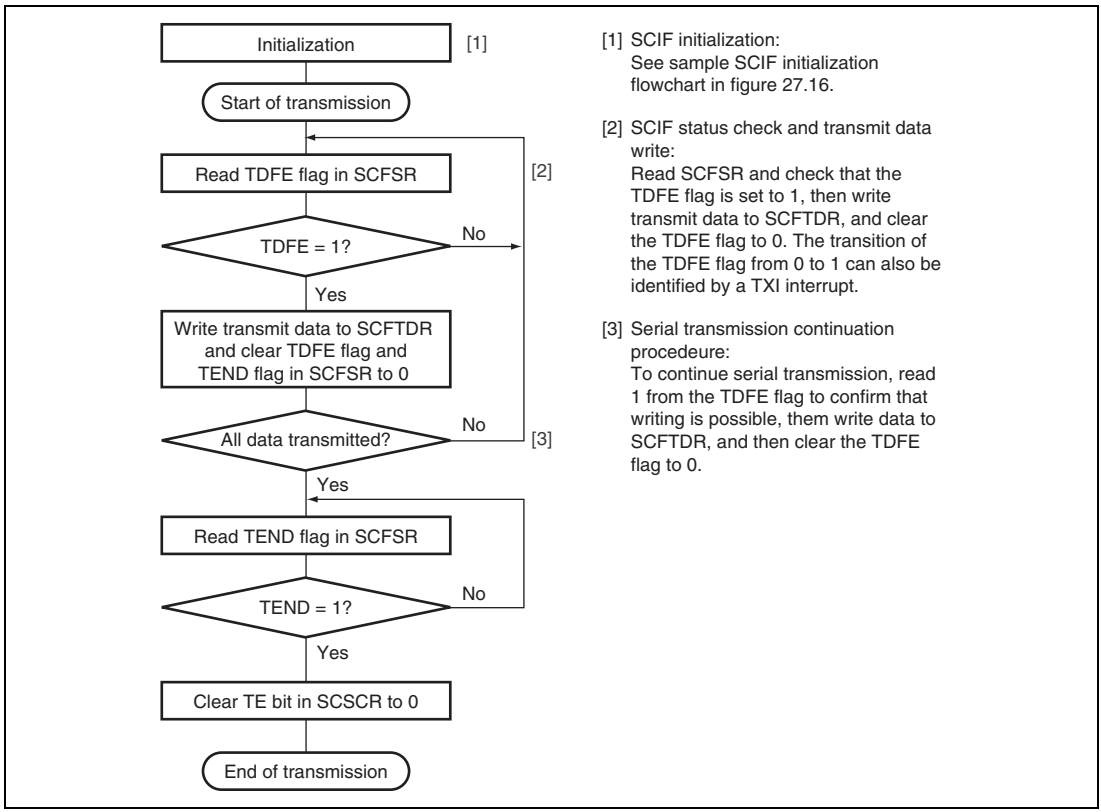


Figure 27.17 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 64 (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in

SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

If clock output mode is selected, the SCIF outputs eight synchronization clock pulses for each data.

When the external clock is selected, data is output in synchronization with the input clock.

The serial transmit data is sent from the SCIF_TXD pin in the LSB-first order.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the last bit. If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of the last bit.
4. After serial transmission ends, the CLK pin is fixed high.

Figure 27.18 shows an example of the operation for transmission in clocked synchronous mode.

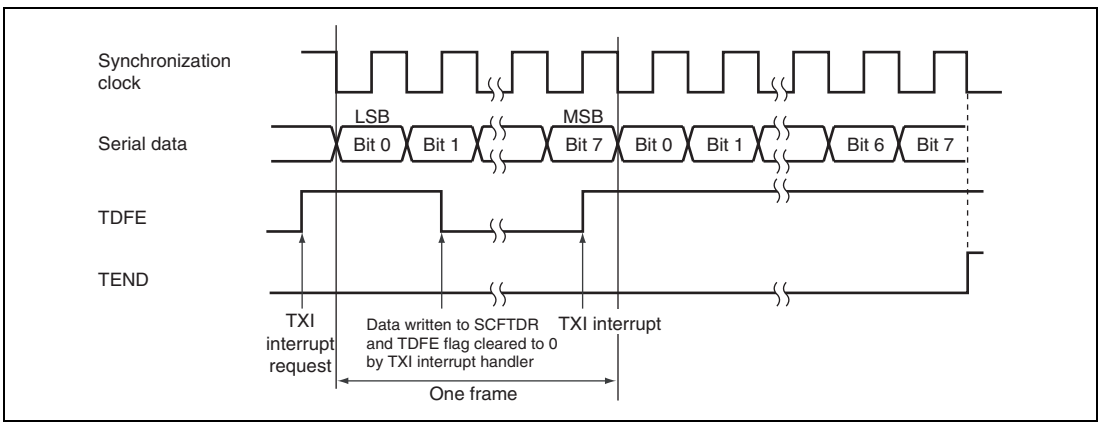


Figure 27.18 Sample SCIF Transmission Operation in Clocked Synchronous Mode

(5) Serial Data Reception (Clocked Synchronous Mode)

Figure 27.19 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching the operating mode from asynchronous mode to clocked synchronous mode without initializing the SCIF, make sure that the ORER, PER7 to PER0, and FER7 to FER0 flags are cleared to 0.

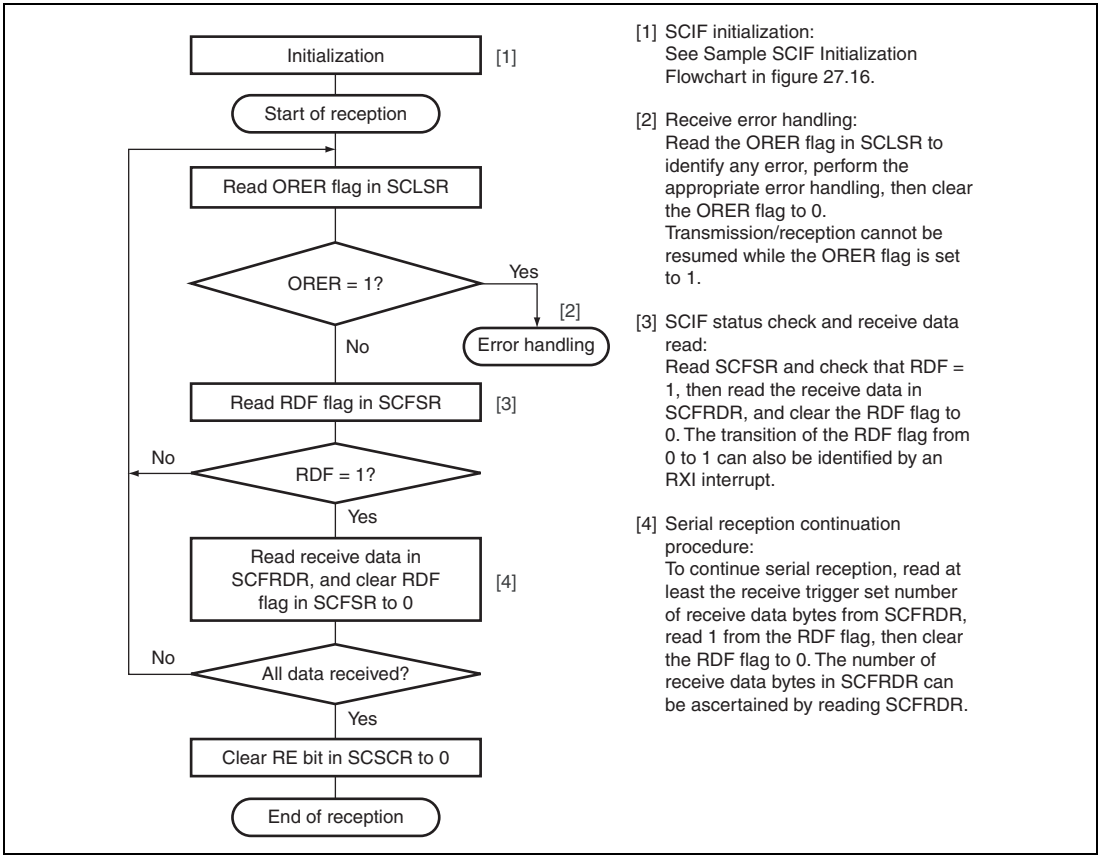


Figure 27.19 Sample Serial Reception Flowchart (1)

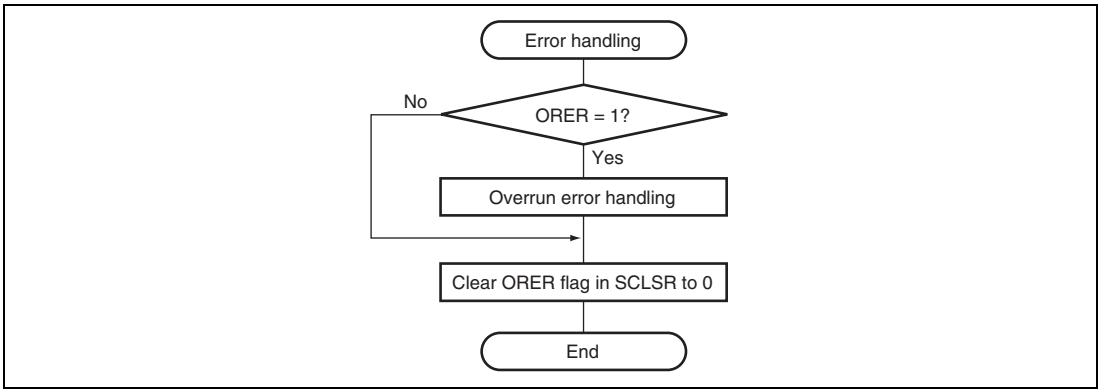


Figure 27.19 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF is initialized internally in synchronization with the input or output of the synchronization clock.
2. The received data is stored in SCRSR in LSB-to-MSB order.

   After receiving the data, the SCIF checks whether the receive data can be transferred from SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. If an overrun error is detected in the error check, reception cannot continue.
3. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interrupt (BRI) request is generated.

Figure 27.20 shows an example of the operation for reception in clocked synchronous mode.

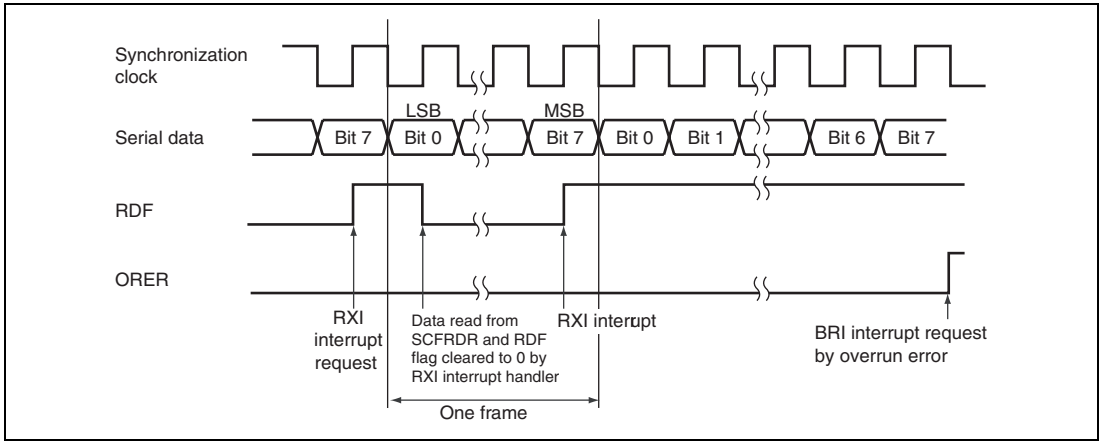
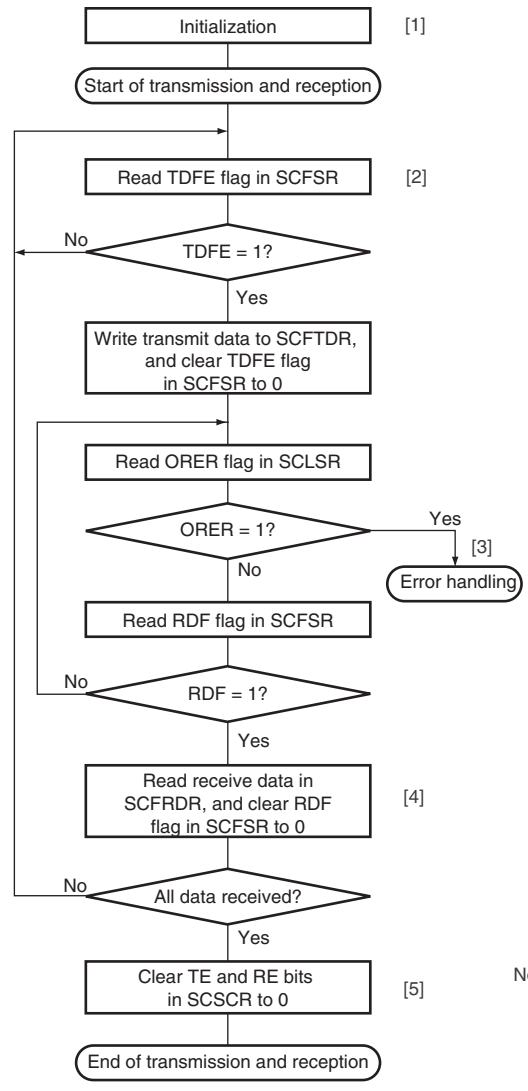


Figure 27.20 Sample SCIF Reception Operation in Clocked Synchronous Mode

(6) Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 27.21 shows a sample flowchart for simultaneous serial data transmission and reception.

Use the following procedure for simultaneous serial transmission and reception after enabling the SCIF for both transmission and reception.



[1] SCIF initialization:
See Sample SCIF Initialization Flowchart in figure 27.16.

[2] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] 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.

[4] 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.

[5] 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 27.21 Sample Simultaneous Serial Transmission and Reception Flowchart

27.5 SCIF Interrupt Sources and the DMAC

The SCIF has four interrupt sources: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and break interrupt (BRI) request.

Table 27.8 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

If the TDFE flag in SCFSR is set to 1 when a TXI interrupt is enabled by the TIE bit, a TXI interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated. If the TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIFO-data-empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can activate the DMAC to perform data transfer.

If the RDF or DR flag in SCFSR is set to 1 when an RXI interrupt is enabled by the RIE bit, an RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generated. If the RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receive-FIFO-data-full request for DMA transfer is generated. A receive-FIFO-data-full request can activate the DMAC to perform data transfer. Note that generation of an RXI interrupt request or a receive-FIFO-data-full request by setting the DR flag to 1 occurs only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated. If transmission/reception is carried out using the DMAC, set and enable the DMAC before making the SCIF settings. Also make settings to inhibit output of RXI and TXI interrupt requests to the interrupt controller. If output of interrupt requests is enabled, these interrupt requests to the interrupt controller can be cleared by the DMAC regardless of the interrupt handler.

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to output ERI interrupt requests, but not RXI interrupt requests.

Table 27.8 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready flag (DR)*	Possible	
BRI	Interrupt initiated by break flag (BRK) or overrun error flag (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous mode.

27.6 Usage Notes

Note the following when using the SCIF.

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in SCFCR. After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again, even after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from SCTFDR.

(2) SCFRDR Reading and the RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive data is read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFRDR to less than the trigger number.

The number of receive data bytes in SCFRDR can be found from SCRFDR.

(3) Break Detection and Processing

If a framing error (FER) is detected**,** break signals can also be detected by reading the SCIF_RXD pin value directly. In the break state the input from the SCIF_RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, the receive operation continues.

(4) Sending a Break Signal

The input/output condition and level of the SCIF_TXD pin are determined by bits SPB2IO and SPB2DT in SCSPTR. This feature can be used to send a break signal.

After the serial transmitter is initialized and until the TE bit is set to 1 (enabling transmission), the SCIF_TXD pin function is not selected and the value of the SPB2DT bit substitutes for the mark state. The SPB2IO and SPB2DT bits should therefore be set to 1 (designating output and high level) in the beginning.

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), and then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized, regardless of the current transmission state, and 0 is output from the SCIF_TXD pin.

(5) Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCIF operates on a base clock with a frequency of 16 times the bit rate.

In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

The timing is shown in figure 27.22.

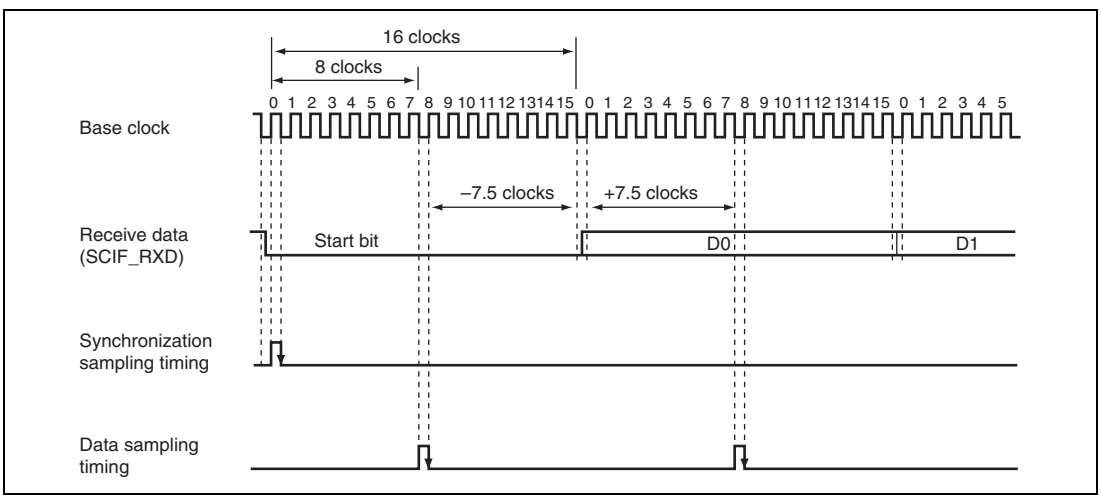


Figure 27.22 Receive Data Sampling Timing in Asynchronous Mode

Thus, the reception margin in asynchronous mode is given by formula (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

M: Receive margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

From equation (1), if F = 0 and D = 0.5, the reception margin is 46.875%, as given by formula (2).

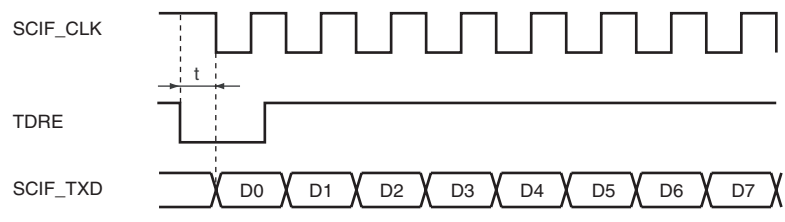
When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \text{..... (2)}$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

(6) When Using the DMAC

When using an external clock as the synchronization clock, after SCFTDR is updated by the DMAC, an external clock should be input after at least five peripheral clock (Pck) cycles. A malfunction may occur when the transfer clock is input within four cycles after updating SCFTDR (see figure 27.23).



Note: When the SCIF is operated on an external clock, set t > 4.

Figure 27.23 Example of Synchronization Clock Transfer by DMAC

Section 28 Serial Communication Interface with FIFO/IrDA Interface (SCIF/IrDA)

This LSI is equipped with a serial communication interface with FIFO/IrDA interface (SCIF/IrDA) that supports the infrared data communication function. The SCIF/IrDA consists of a serial communication interface with built-in FIFO buffers (SCIF) and infrared communication data modulation/demodulation units. By specifying the IrDA input/output options, infrared data communication can be performed with the infrared sensor/emitter in the same way as the synchronous serial data communication.

The SCIF/IrDA uses channel 2 (SCIF2) as a serial communication channel. Note that some part of the operating specification differs from that of channel 0 and 1 (SCIF0, SCIF1) in the serial communication interface with FIFO (SCIF).

28.1 Features

The SCIF/IrDA has the following features.

- Infrared data communication function

  Infrared data communication compliant with the IrDA standard 1.0 can be performed.

  This function modulates and demodulates the data format supported by serial communication to the data format supported by infrared data communication.
- Asynchronous serial communication mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).

  There is a choice of 8 serial data transfer formats.

  — Data length: 7 or 8 bits

  — Stop bit length: 1 or 2 bits

  — Parity: Even/odd/none

  — Receive error detection: Parity, framing, and overrun errors

  — Break detection: A break is detected when a framing error lasts for more than 1 frame length at Space 0 (low level). When a framing error occurs, a break can also be detected by reading the SCIF2_RXD pin level directly from the serial port register (SCSPTR).

- Clocked synchronous serial communication mode

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other LSIs that have a synchronous communication function.

  There is a single serial data communication format.

  — Data length: 8 bits

  — Receive error detection: Overrun errors
- Full-duplex communication capability

  The transmitter and receiver are independent units, enabling transmission and reception to be performed simultaneously.

  The transmitter and receiver both have a 16-stage FIFO buffer structure, enabling continuous serial data transmission and reception.
- The LSB is transmitted and received first (LSB first).
- On-chip baud rate generator allows any bit rate to be selected.
- Choice of clock source: internal clock from the baud rate generator based on the peripheral clock (Pck0) or external clock from the SCIF2_SCK pin
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error—that can issue requests independently.
- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- The amount of data in the transmit/receive FIFO registers, and the number of receive errors in the receive data in the receive FIFO register, can be ascertained.
- In asynchronous mode, a timeout error (DR) can be detected during reception.

Figure 28.1 shows a block diagram of the SCIF/IrDA. Figures 28.2 to 28.4 show block diagrams of the I/O ports in SCIF/IrDA.

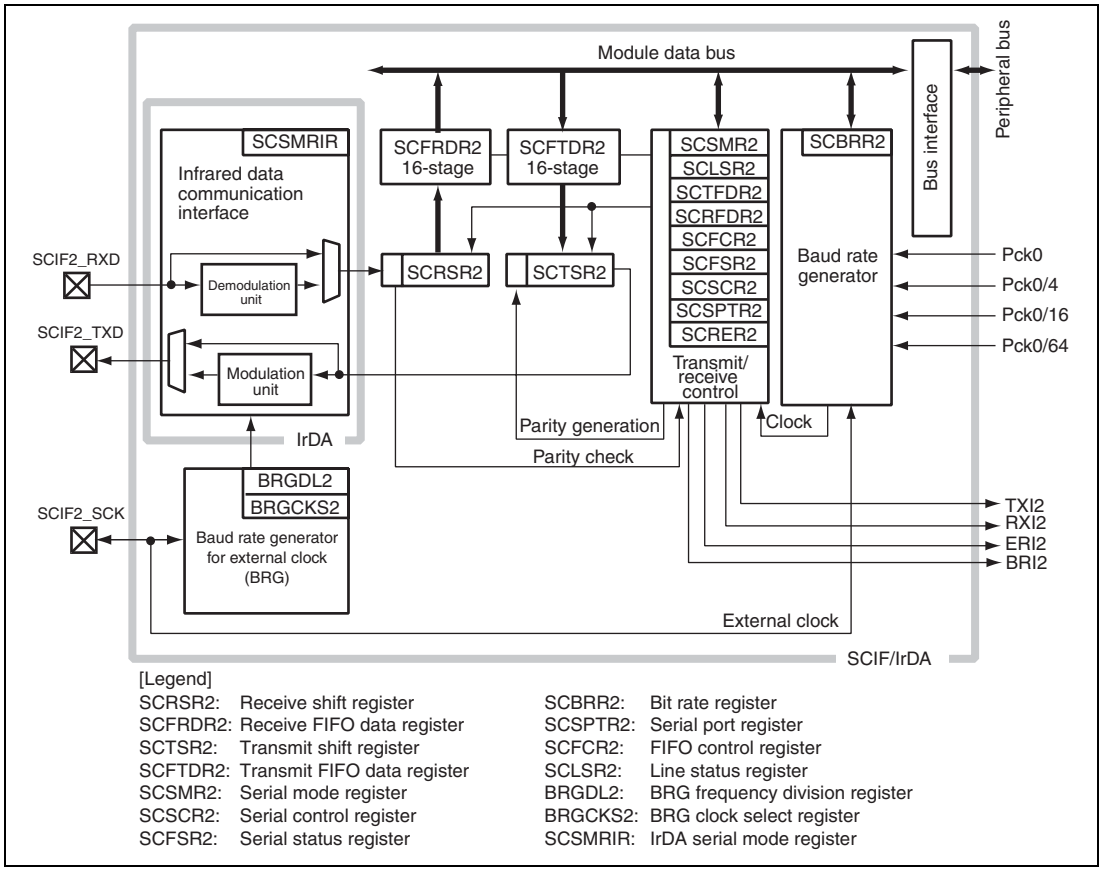


Figure 28.1 Block Diagram of SCIF/IrDA

Figures 28.2 to 28.4 show block diagrams of the I/O ports in SCIF/IrDA.

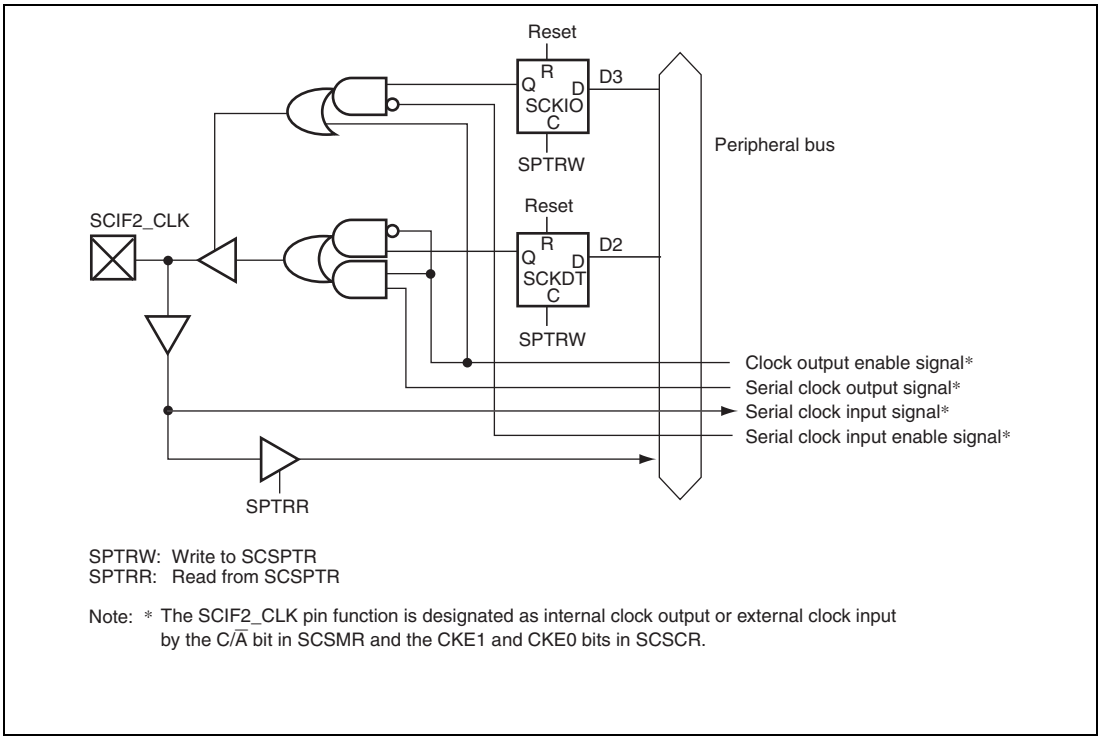


Figure 28.2 SCIF2_SCK Pin

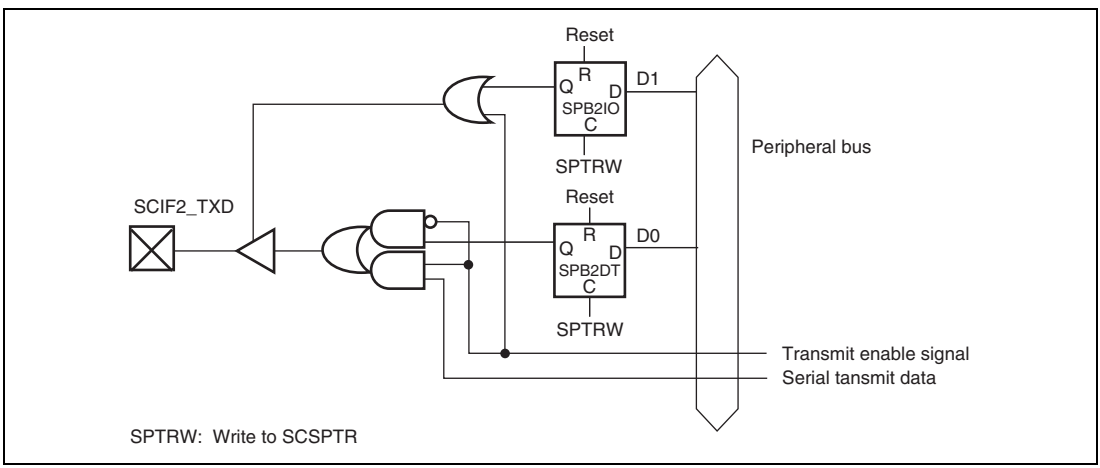


Figure 28.3 SCIF2_TXD Pin

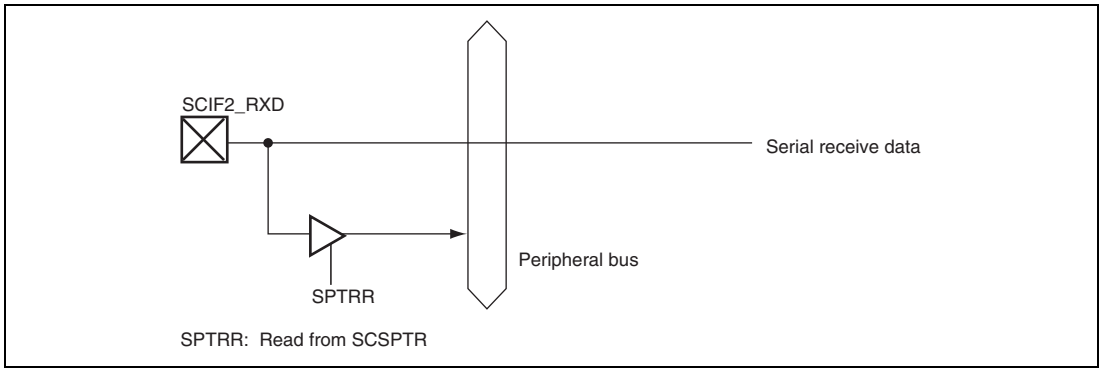


Figure 28.4 SCIF2_RXD Pin

28.2 Input/Output Pins

Table 28.1 shows the SCIF/IrDA pin configuration. The channel number is omitted in the description below.

Table 28.1 Pin Configuration

Pin Name	Function	I/O	Description
SCIF2_SCK	Serial clock pin	I/O*	Clock input/output
SCIF2_RXD	Receive data pin	Input	Receive data input
SCIF2_TXD	Transmit data pin	Output	Transmit data output

Notes: These pins are made to function as serial pins by performing SCIF operation settings with the C/$\overline{A}$ bit in SCSMR, and the TE, RE, CKE1, and CKE0 bits in SCSCR. Break state transmission and detection can be set in SCSPTR of the SCIF.

* When the infrared data communication function is selected, a serial clock should be input at the SCIF2_SCK pin

28.3 Register Descriptions

Table 28.2 shows the SCIF/IrDA register configuration. Table 28.3 shows the register states in each operating mode.

Table 28.2 Register Configuration

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size
2	Serial mode register 2	SCSMR2	R/W	H'FFE1 0000	H'1FE1 0000	16
	Bit rate register 2	SCBRR2	R/W	H'FFE1 0004	H'1FE1 0004	8
	Serial control register 2	SCSCR2	R/W	H'FFE1 0008	H'1FE1 0008	16
	Transmit FIFO data register 2	SCFTDR2	W	H'FFE1 000C	H'1FE1 000C	8
	Serial status register 2	SCFSR2	R/W*[1]	H'FFE1 0010	H'1FE1 0010	16
	Receive FIFO data register 2	SCFRDR2	R	H'FFE1 0014	H'1FE1 0014	8
	FIFO control register 2	SCFCR2	R/W	H'FFE1 0018	H'1FE10018	16
	FIFO data count register 2	SCFDR2	R	H'FFE1 001C	H'1FE1 001C	16
	Serial port register 2	SCSPTR2	R/W	H'FFE1 0020	H'1FE1 0020	16
	Line status register 2	SCLSR2	R/W*[2]	H'FFE1 0024	H'1FE1 0024	16
	BRG frequency division register	BRGDL2	R/W	H'FFE1 0030	H'FFE1 0030	16
	BRG clock select register	BRGCKS2	R/W	H'FFE1 0034	H'FFE1 0034	16
	IrDA serial mode register	SCSMRIR	R/W	H'FFE1 0040	H'FFE1 0040	16

Notes: 1. To clear the flags, 0s can only be written to bits 7 to 4, 1, and 0.
2. To clear the flag, 0 can only be written to bit 0.

Table 28.3 Register States in each Operation Mode

Ch.	Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Standby
2	Serial mode register 2	SCSMR2	H'0000	H'0000	Retained	Retained
	Bit rate register 2	SCBRR2	H'FF	H'FF	Retained	Retained
	Serial control register 2	SCSCR2	H'0000	H'0000	Retained	Retained
	Transmit FIFO data register 2	SCFTDR2	Undefined	Undefined	Retained	Retained
	Serial status register 2	SCFSR2	H'0060	H'0060	Retained	Retained
	Receive FIFO data register 2	SCFRDR2	Undefined	Undefined	Retained	Retained
	FIFO control register 2	SCFCR2	H'0000	H'0000	Retained	Retained
	FIFO data count register 2	SCFDR2	H'0000	H'0000	Retained	Retained
	Serial port register 2	SCSPTR2	H'0000*	H'0000*	Retained	Retained
	Line status register 2	SCLSR2	H'0000	H'0000	Retained	Retained
	BRG frequency division register	BRGDL2	H'0000	H'0000	Retained	Retained
	BRG clock select register	BRGCKS2	H'0000	H'0000	Retained	Retained
	IrDA serial mode register	SCSMRIR	H'0000	H'0000	Retained	Retained

Note: * Bits 2 and 0 are undefined.

28.3.1 Receive Shift Register (SCRSR)

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the SCIF_RXD pin in SCRSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to SCFRDR, automatically.

SCRSR cannot be directly read from and written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

28.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is an 8-bit FIFO register of 16 stages that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SCRSR to SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled for reception, and consecutive receive operations can be performed until SCFRDR is full (16 data bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in SCFRDR, an undefined value will be returned. When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is undefined at a power-on reset or a manual reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

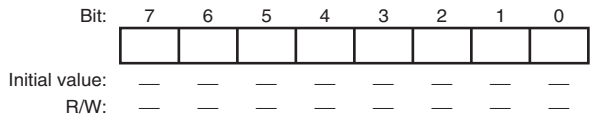
28.3.3 Transmit Shift Register (SCTSR)

SCTSR is the register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTDR to SCTSR, then sends the data to the SCIF_TXD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from SCFTDR to SCTSR, and transmission started, automatically.

SCTSR cannot be directly read from and written to by the CPU.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

28.3.4 Transmit FIFO Data Register (SCFTDR)

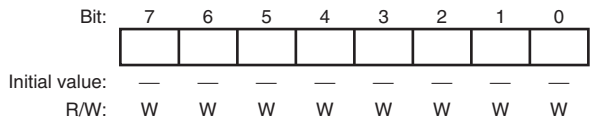
SCFTDR is an 8-bit FIFO register of 16 stages that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers the transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read by the CPU.

The next data cannot be written when SCFTDR is filled with 16 bytes of transmit data. Data written in this case is ignored.

SCFTDR is undefined at a power-on reset or a manual reset.



Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

28.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register used to set the SCIF's serial transfer format and select the baud rate generator clock source.

SCSMR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or clocked synchronous mode as the SCIF operating mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode data length. In clocked synchronous mode, the data length is fixed at 8 bits regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of SCFTDR is not transmitted. 0: 8-bit data 1: 7-bit data

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking is performed in reception. In clocked synchronous mode, parity bit addition and checking is disabled regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled* Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. In asynchronous mode, the O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking. In clocked synchronous mode or when parity addition and checking is disabled in asynchronous mode, the O/$\overline{E}$ bit setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is even. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is odd.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length In asynchronous mode, selects 1 or 2 bits as the stop bit length. The stop bit setting is valid only in asynchronous mode. Since the stop bit is not added in clocked synchronous mode, the STOP bit setting is invalid. 0: 1 stop bit*1 1: 2 stop bits*2 In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character. Note: 1. In transmission, a single 1-bit (stop bit) is added to the end of a transmit character before it is sent. 2. In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. The clock source can be selected from Pck0, Pck0/4, Pck0/16, and Pck0/64, according to the setting of bits CKS1 and CKS0. For details of the relationship between clock sources, bit rate register settings, and baud rate, see section 28.3.8, Bit Rate Register (SCBRR). 00: Pck0 clock 01: Pck0/4 clock 10: Pck0/16 clock 11: Pck0/64 clock

Note: Pck0 = Peripheral Clock 0

28.3.6 Serial Control Register (SCSCR)

SCSCR is a register used to enable/disable transmission/reception by SCIF, serial clock output, interrupt requests, and to select transmission/reception clock source for the SCIF.

SCSCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when serial transmit data is transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR falls to or below the transmit trigger set number, and the TDFE flag in SCFSR is set to 1. TXI interrupt requests can be cleared using the following methods: Either by reading 1 from the TDFE flag, writing transmit data exceeding the transmit trigger set number to SCFTDR and then clearing the TDFE flag to 0, or by clearing the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag or DR flag in SCFSR is set to 1, a receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, and a break interrupt (BRI) request when the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1. 0: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request disabled 1: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request enabled Note: An RXI interrupt request can be cleared by reading 1 from the RDF or DR flag, then clearing the flag to 0, or by clearing the RIE bit to 0. ERI and BRI interrupt requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmission by the SCIF. Serial transmission is started when transmit data is written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled* Note: SCSMR and SCFCR settings must be made, the transmission format decided, and the transmit FIFO reset, before the TE bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the start of serial reception by the SCIF. Serial reception is started when a start bit is detected in this state in asynchronous mode or a synchronization clock is input while the RE bit is set to 1. It should be noted that clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER, and ORER flags, which retain their states. Serial reception begins once the start bit is detected in these states. 0: Reception disabled 1: Reception enabled* Note: * SCSMR and SCFCR settings must be made, the reception format decided, and the receive FIFO reset, before the RE bit is set to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-error interrupt (ERI) and break interrupt (BRI) requests. The REIE bit setting is valid only when the RIE bit is 0. Receive-error interrupt (ERI) and break interrupt (BRI) requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0. When REIE is set to 1, ERI and BRI interrupt requests will be generated even if RIE is cleared to 0. In DMAC transfer, this setting is made if the interrupt controller is to be notified of ERI and BRI interrupt requests. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	These bits select the SCIF clock source and whether to enable or disable the clock output from the SCIF_SCK pin. The CKE1 and CKE0 bits are used together to specify whether the SCIF_SCK pin functions as a serial clock output pin or a serial clock input pin. Note however that the CKE0 bit setting is valid only when an internal clock is selected as the SCIF clock source (CKE1 = 0). When an external clock is selected (CKE1 = 1), the CKE0 bit setting is invalid. In clock synchronous mode, to select synchronization clock output, set the C/A bit in SCSMR to 1, then set the CKE1 and CKE0 bits. • Asynchronous mode 00: The SCIF_SCK pin is not used. The SCIF_SCK pin functions as an input pin (Input signals are ignored) 01: The SCIF_SCK pin functions as clock output*[1] 10: External clock/SCIF_SCK pin functions as clock input*[2] 11: Setting prohibited • Clocked synchronous mode 00,01: The SCIF_SCK pin functions as synchronization clock output 10: The SCIF_SCK pin functions as synchronization clock input 11: Setting prohibited

Notes: X: Don't care

1. Outputs a clock with a frequency 16 times the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

28.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. Its lower 8 bits are a status flag that indicates the operating status of the SCIF and its upper 8 bits indicate the number of errors for data received by the receive FIFO register.

SCFSR can be read from or written to by the CPU at all times. However, 1 cannot be written to flags ER, TEND, TDFE, BRK, RDF, and DR. Also note that in order to clear these flags they must be read as 1 beforehand. The FER flag and PER flag are read-only flags and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PERN[3:0]				FERN[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R/W*1	R/W*1

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PERN[3:0]	All 0	R	Number of Parity Errors These bits indicate the number of parity errors in data received and stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits 15 to 12 is the number of parity errors. If a parity error occurs in all 16-byte data received by SCFRDR, PERN indicates 0.
11 to 8	FERN[3:0]	All 0	R	Number of Framing Errors These bits indicate the number of framing errors in data received and stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits 11 to 8 is the number of framing errors. If a framing error occurs in all 16-byte data received by SCFRDR, FERN indicates 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/W*[1]	Receive Error Indicates that a framing error or parity error occurred during reception. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a receive error occurs, the receive data is still transferred to SCFRDR, and reception continues. The FER and PER bits in SCFSR can be used to determine whether there is a receive error in the readout data from SCFRDR. 0: No framing error or parity error occurred during reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during reception [Setting conditions] • When the SCIF checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0*[2] • When, in reception, the number of 1-bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SCSMR
6	TEND	1	R/W*[1]	Transmit End Indicates that transmission has been ended without valid data in SCFTDR after transmission of the last bit of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, and 0 is written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DMAC 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR after transmission of the last bit of a 1-byte serial transmit character

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/W*[1]	Transmit FIFO Data Empty Indicates that data has been transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR has fallen to or below the transmit trigger data number set by bits TTRG1 and TTRG0 in SCFCR, and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the transmit trigger set number have been written to SCFTDR [Clearing conditions] • When transmit data exceeding the transmit trigger set number is written to SCFTDR after reading TDFE = 1, and 0 is written to TDFE • When transmit data exceeding the transmit trigger set number is written to SCFTDR by the DMAC 1: The number of transmit data bytes in SCFTDR does not exceed the transmit trigger set number (Initial value) [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger set number as the result of a transmit operation*[3]
4	BRK	0	R/W*[1]	Break Detect Indicates that a receive data break signal has been detected. 0: A break signal has not been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after reading BRK = 1 1: A break signal has been received*[4] [Setting condition] • When data with a framing error is received, followed by the space "0" level (low level) for at least one frame length

Bit	Bit Name	Initial Value	R/W	Description
3	FER	0	R	Framing Error In asynchronous mode, indicates whether or not a framing error has been found in the data that is to be read next from SCFRDR. 0: There is no framing error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data that is to be read next from SCFRDR 1: There is a framing error that is to be read from SCFRDR [Setting condition] • When there is a framing error in the data that is to be read next from SCFRDR
2	PER	0	R	Parity Error In asynchronous mode, indicates whether or not a parity error has been found in the data that is to be read next from SCFRDR. 0: There is no parity error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in the data that is to be read next from SCFRDR 1: There is a parity error in the receive data that is to be read from SCFRDR [Setting condition] • When there is a parity error in the data that is to be read next from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/W*[1]	Receive FIFO Data Full Indicates that the received data has been transferred from SCRSR to SCFRDR, and the number of receive data bytes in SCFRDR is equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. 0: The number of receive data bytes in SCFRDR is less than the receive trigger set number [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger set number after reading RDF = 1, and 0 is written to RDF • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR falls below the receive trigger set number 1: The number of receive data bytes in SCFRDR is equal to or greater than the receive trigger set number [Setting condition] • When SCFRDR contains at least the receive trigger set number of receive data bytes*[5]

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/W*[1]	Receive Data Ready In asynchronous mode, indicates that there are fewer than the receive trigger set number of data bytes in SCFRDR, and no further data has arrived for at least 15 etu after the stop bit of the last data received. This is not set when using clocked synchronous mode. 0: Reception is in progress or has ended normally and there is no receive data left in SCFRDR [Clearing conditions] • Power-on reset or manual reset • When all the receive data in SCFRDR has been read after reading DR = 1, and 0 is written to DR • When all the receive data in SCFRDR has been read by the DMAC 1: No further receive data has arrived [Setting condition] • When SCFRDR contains fewer than the receive trigger set number of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received*[6]

[Legend] etu: Elementary time unit (time for transfer of 1 bit)

Notes: 1. Only 0 can be written, to clear the flag.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the second stop bit is not checked.

3. As SCFTDR is a 16-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 16 − (transmit trigger set number). Data written in excess of this will be ignored.

   SCFDR indicates the number of data bytes transmitted to SCFTDR.

4. When a break is detected, the receive data (H'00) following detection is not transferred to SCFRDR. When the break ends and the receive signal returns to mark "1", receive data transfer is resumed.

5. SCFRDR is a 16-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If all the data in SCFRDR is read and another read is performed, the data value will be undefined. The number of receive data bytes in SCFRDR is indicated by SCFDR.

6. Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.

28.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that set the serial transmission/reception bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

SCBRR can always be read from and written to by the CPU.

This baud rate generator is intended for Pck0, Pck0/4, Pck0/16, and Pck0/64. For details on the baud rate generator for external clock, see section 28.6, Baud Rate Generator for External Clock (BRG).

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{Pck0}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{Pck0}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pck0: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See Table 28.4 for the relation between n and the clock.)

Table 28.4 SCSMR Settings

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pck0	0	0
1	Pck0/4	0	1
2	Pck0/16	1	0
3	Pck0/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{\text{Pck0} \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n - 1}} - 1 \right\} \times 100$$

28.3.9 FIFO Control Register (SCFCR)

SCFCR performs data count resetting and trigger data number setting for transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RTRG[1:0]		TTRG[1:0]		—	TFCL	RFCL	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	RTRG[1:0]	All 0	R/W	Receive FIFO Data Number Trigger These bits are used to set the number of receive data bytes that sets the RDF flag in SCFSR. The RDF flag is set when the number of receive data bytes in SCFRDR is equal to or greater than the trigger set number shown below.

	Asynchronous mode	Clock Synchronous mode
00:	1	1
01:	4	2
10:	8	8
11:	14	14

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	All 0	R/W	Transmit FIFO Data Number Trigger These bits are used to set the number of remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when the number of transmit data bytes in SCFTDR is equal to or less than the trigger set number shown below. 00: 8 (8)* 01:4 (12) 10: 2 (14) 11: 0 (16) Note: * Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the transmit data in the transmit FIFO data register and resets it to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * A reset operation is performed in the event of a power-on reset or manual reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in the receive FIFO data register and resets it to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * A reset operation is performed in the event of a power-on reset or manual reset.
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SCIF_TXD) and receive input pin (SCIF_RXD) enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

28.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register that indicates the number of transmit data bytes stored in SCFTDR.

SCFDR can always be read from the CPU.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	TDN[4:0]	All 0	R	These bits show the number of untransmitted data bytes in SCFTDR. A value of H'00 indicates that there is no transmit data, and a value of H'10 indicates that SCFTDR is full of transmit data (16-bytes).
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	RDN[4:0]	All 0	R	These bits show the number of receive data bytes in SCFRDR. A value of H'00 indicates that there is no receive data, and a value of H'10 indicates that SCFRDR is full of receive data (16-bytes).

28.3.11 Serial Port Register (SCSPTR)

SCSPTR is a 16-bit readable/writable register that controls input/output and data for the port pins multiplexed with the serial communication interface (SCIF) pins at all times. Input data can be read from the SCIF_RXD pin, output data written to the SCIF_TXD pin, and breaks in serial transmission/reception controlled, by means of bits 1 and 0.

All SCSPTR bits except bits 6, 4, 2, and 0 are initialized to 0 by a power-on reset or manual reset; the value of bits 6, 4, 2, and 0 is undefined. SCSPTR is not initialized in the module standby state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	—	0	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

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	SCKIO	0	R/W	Serial Port Clock Port Input/Output Specifies the serial port SCIF_SCK pin input/output condition. When actually setting the SCIF_SCK pin as a port output pin to output the value set by the SCKDT bit, the CKE1 and CKE0 bits in SCSCR should be cleared to 0. 0: SCKDT bit value is not output to SCIF_SCK pin 1: SCKDT bit value is output to SCIF_SCK pin
2	SCKDT	—	R/W	Serial Port Clock Port Data Specifies the serial port SCIF_SCK pin input/output data. Input or output is specified by the SCKIO bit. In output mode, the SCKDT bit value is output to the SCIF_SCK pin. The SCIF_SCK pin value is read from the SCKDT bit regardless of the value of the SCKIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Bit	Bit Name	Initial Value	R/W	Description
1	SPB2IO	0	R/W	Serial Port Break Input/Output Specifies the serial port SCIF_TXD pin output condition. When actually setting the SCIF_TXD pin as a port output pin to output the value set by the SPB2DT bit, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_TXD pin 1: SPB2DT bit value is output to the SCIF_TXD pin
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input data and SCIF_TXD pin output data. The SCIF_TXD pin output condition is specified by the SPB2IO bit. When the SCIF_TXD pin is designated as an output, the value of the SPB2DT bit is output to the SCIF_TXD pin. The SCIF_RXD pin value is read from the SPB2DT bit regardless of the value of the SPB2IO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

28.3.12 Line Status Register (SCLSR)

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

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*1	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Reception in progress, or reception has ended normally*2 [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1 1: An overrun error occurred during reception*3 [Setting condition] • When the next serial reception is completed while SCFRDR receives 16-byte data (SCFRDR is full)

Notes: 1. Only 0 can be written, to clear the flag.
2. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0.
3. The receive data prior to the overrun error is retained in SCFRDR, and the data received subsequently is lost. Serial reception cannot be continued while the ORER flag is set to 1.
To resume data reception after clearing the ORER flag, be sure to first read (or clear) data in the receive FIFO and handle the error, then clear the ORER flag.

28.3.13 BRG Frequency Division Register (BRGDL2)

BRGDR2 specifies the division ratio of the division clocks generated by the BRG. The clock division ratio set in this register can be determined with the following equation.

Clock division ratio = clock input frequency/(required baud rate × 16)

Table 28.5 shows the clock division ratios when a 3.686 MHz crystal resonator is used.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	BRGDL [15:0]	All 0	R/W	Division Ratio of BRG Generated Clock These bits specify the division ratio of the division clocks generated by the BRG. A division ratio from 1 to 65535 can be specified.

Table 28.5 Baud Rate (3.6864 MHz Clock)

Baud Rate	Division Ratio	Error Rate*
50	4608	—
75	3072	—
110	2095	–0.022
134.5	1713	0.001
150	1536	—
300	768	—
600	384	—
1200	192	—
1800	128	—
2000	115	0.174
2400	96	—
3600	64	—

Baud Rate	Division Ratio	Error Rate*
4800	48	—
7200	32	—
9600	24	—
14400	16	—
19200	12	—
38400	6	—
76800	3	—
115200	2	—

Note: * —: Error rate = 0

28.3.14 BRG Clock Select Register (BRGCKS2)

BRGCKS2 switches output clock between the division clock generated by the BRG and the external clock (SCIF2_SCK).

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

Bit	Bit Name	Initial Value	R/W	Description
15	BRGCKS	0	R/W	Switches output clock between the division clock and external clock (SCIF2_SCK). 0: Selects division clock 1: Selects external clock
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

28.3.15 IrDA Serial Mode Register (SCSMRIR)

SCSMRIR is used for control when channel 2 (SCIF2) of the SCIF/IrDA is used as the IrDA infrared communication port.

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

BIt:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EDGEN	LOOP	IRMOD	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	EDGEN	0	R/W	SCIF2_RXD Pin Sampling Mode 0: The SCIF2_RXD pin is sample by an edge 1: The SCIF2_RXD pin is sample by both an edge and level
8	LOOP	0	R/W	IrDA Loop back Test 0: Normal operation 1: Loop back operation from the SCIF2_TXD pin to the SCIF2_RXD pin
7	IRMOD	0	R/W	IrDA Mode 0: SCIF2 of the SCIF/IrDA is used as the SCIF serial communication port 1: SCIF2 of the SCIF/IrDA is used as the IrDA infrared communication port
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

28.4 Operation

28.4.1 Overview

The SCIF can carry out serial communication in asynchronous mode, in which synchronization is achieved character by character and in synchronous mode, in which synchronization is achieved with clock pulses. For details on asynchronous mode, see section 28.4.2, Operation in Asynchronous Mode.

16-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead, and enabling fast and continuous communication to be performed.

The serial transfer format is selected using SCSMR, as shown in Table 28.6. The SCIF clock source is determined by the combination of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as shown in Table 28.7.

Asynchronous Mode:

- Data length: Choice of 7 or 8 bits
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun errors, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Choice of peripheral clock 0 (Pck0) or external clock (SCIF_SCK) as SCIF clock source

  When peripheral clock 0 (Pck0) is selected: The SCIF operates on the baud rate generator clock and can output a clock with frequency of 16 times the bit rate.

  When external clock (SCIF_SCK) is selected: A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used).

Clocked Synchronous Mode:

- Data length: Fixed at 8 bits
- Detection of overrun errors during reception
- Choice of peripheral clock 0 (Pck0) or external clock (SCIF_SCK) as SCIF clock source

  When peripheral clock 0 (Pck0) is selected: The SCIF operates on the baud rate generator clock and a serial clock is output to external devices.

  When external clock (SCIF_SCK) is selected: The on-chip baud rate generator is not used and the SCIF operates on the input serial clock.

Table 28.6 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					SCIF Transfer Format		
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous mode	8-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
	1	0	0		7-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous mode	8-bit data	No	No

Note: x: Don't care

Table 28.7 SCSMR and SCSCR Settings for SCIF Clock Source Selection

SCSMR	SCSCR Setting				
Bit 7	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	Mode	Clock Source	Description of SCK Pin
0	0	0	Asynchronous	Internal clock	The SCK pin is not used.
				Pck0, Pck0/4, Pck0/16, Pck0/64	The SCK pin functions as an input pin (Input signals are ignored). (Initial value)
		1			The SCK pin outputs the clock (with a frequency 16 times the bit rate).
	1	0		Input SC_CLK to baud rate generator for external clock (SCIF_CLK, Pck0) or SCK (switched by baud rate generator's CKS register)	The SCK pin inputs the clock (with a frequency 16 times the bit rate).
					When SC_CLK is selected, the SCK pin inputs the clock (Input signals are ignored).
					Set the SCK input or SC_CLK so that the frequency of BRGCLK is 16 times the bit rate.
					When selecting SC_CLK, set the frequency of SC_CLK and DL, or when selecting SCK, adjust the frequency of the SCK input to make the frequency of BRGCLK 16 times the bit rate.
	1	1	Prohibited	—	—
1	0	0	Clock synchronous	Internal clock	The SCK pin outputs the synchronization clock.
		1		Pck0, Pck0/4, Pck0/16, Pck0/64	The SCK pin outputs the synchronization clock.

SCSMR	SCSCR Setting				
Bit 7	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	Mode	Clock Source	Description of SCK Pin
1	1	0	Clock synchronous	SCK input (switched by baud rate generator's CKS register)	The SCK pin outputs the synchronization clock.
				Input SC_CLK to baud rate generator for external clock (SCIF_CLK, Pck0)	SC_CLK cannot be used as an input clock for synchronous communication.
	1	1	Prohibited	—	—

28.4.2 Operation in Asynchronous Mode

In asynchronous mode, a character that consists of data with a start bit indicating the start of communication and a stop bit indicating the end of communication is transmitted or received. In this mode, serial communication is performed with synchronization achieved character by character.

Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and receiver have a 16-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

Figure 28.5 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCIF monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One character in serial communication consists of a start bit (low level), followed by transmit/receive data (LSB-first; from the lowest bit), a parity bit (high or low level), and finally stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. Receive data can be latched at the middle of each bit because the SCIF samples data at the eighth clock which has a frequency of 16 times the bit rate.

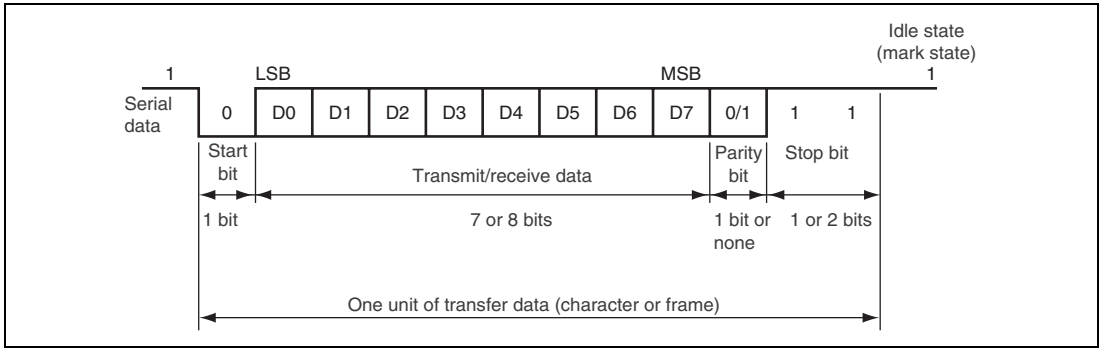


Figure 28.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

(1) Data Transfer Format

Table 28.8 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 28.8 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
1	0	1	S	7-bit data							STOP	STOP		
1	1	0	S	7-bit data							P	STOP		
1	1	1	S	7-bit data							P	STOP	STOP	

[Legend]
S : Start bit
STOP : Stop bit
P : Parity bit

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see Table 28.5.

When an external clock is input at the SCIF_SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCIF is operated on an internal clock, a clock whose frequency is 16 times the bit rate is output from the SCIF_SCK pin.

(3) SCIF Initialization (Asynchronous Mode)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When the operating mode or transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure.

1. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR.
2. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND flag in SCFSR has been set. TEND can also be cleared to 0 during transmission, but the data being transmitted will go to the mark state after the clearance. Before setting TE again to start transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCFTDR.
3. When an external clock is used the clock should not be stopped during operation, including initialization, since operation will be unreliable in this case.

Figure 28.6 shows a sample SCIF initialization flowchart.

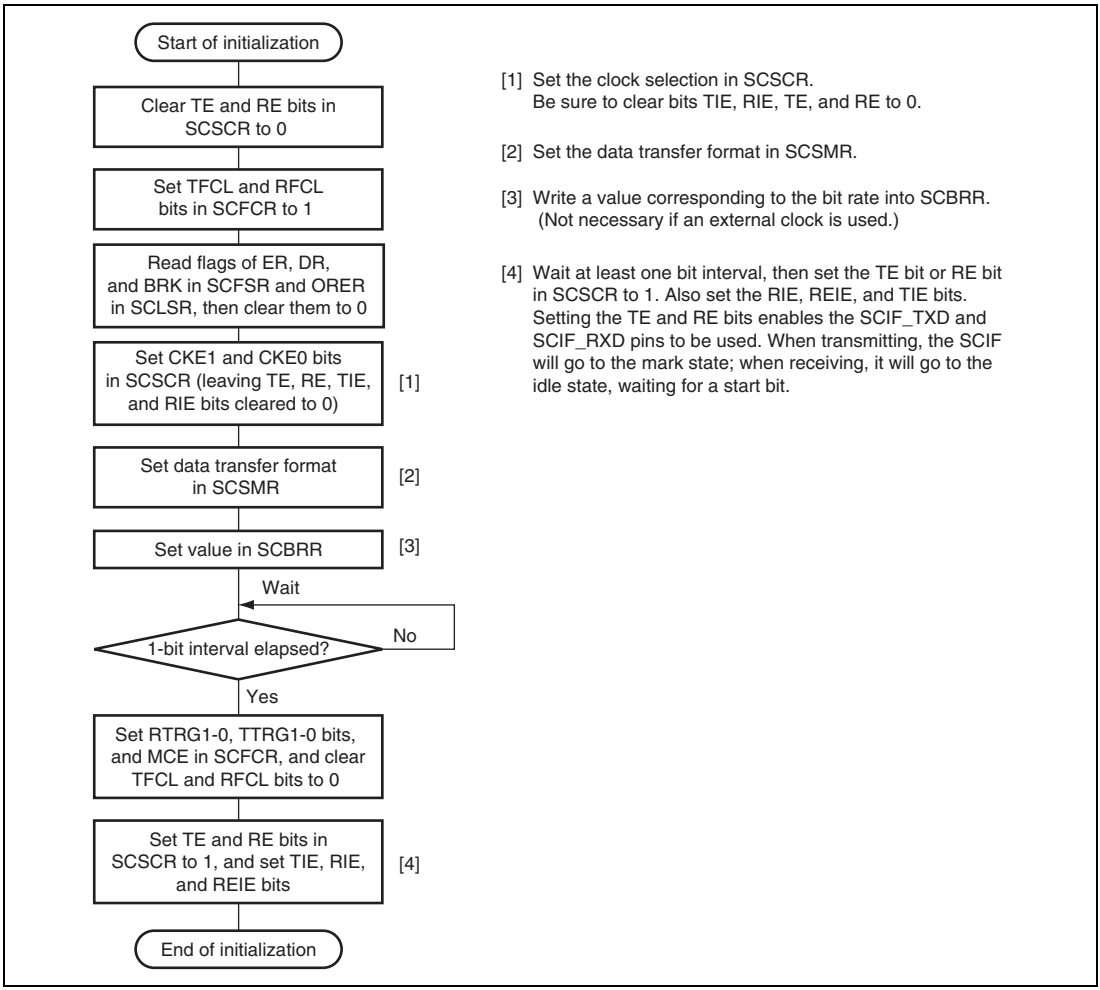


Figure 28.6 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Asynchronous Mode):

Figure 28.7 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

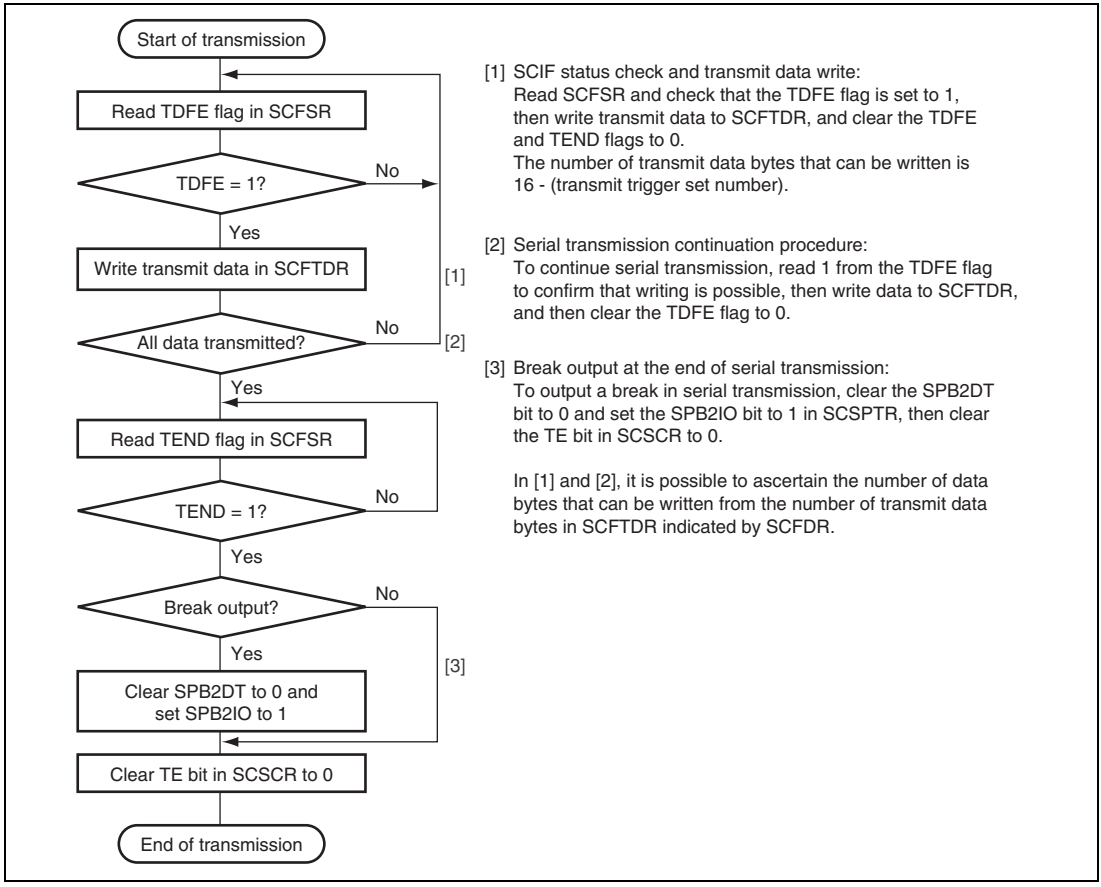


Figure 28.7 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 16 − (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the SCIF_TXD pin in the following order.

   (a) Start bit: One 0-bit is output.

   (b) Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   (c) Parity bit: One parity bit (even or odd parity) is output. A format in which a parity bit is not output can also be selected.

   (d) Stop bit(s): One or two 1-bits (stop bits) are output.

   (e) Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output from the SCIF_TXD pin.

Figure 28.8 shows an example of the operation for transmission in asynchronous mode.

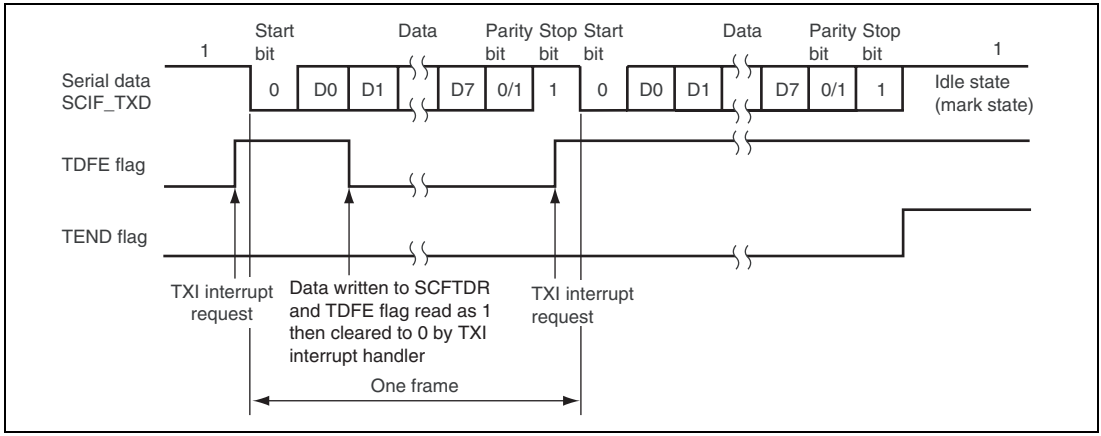


Figure 28.8 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

(5) Serial Data Reception (Asynchronous Mode)

Figures 28.9 and 28.10 shows a sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

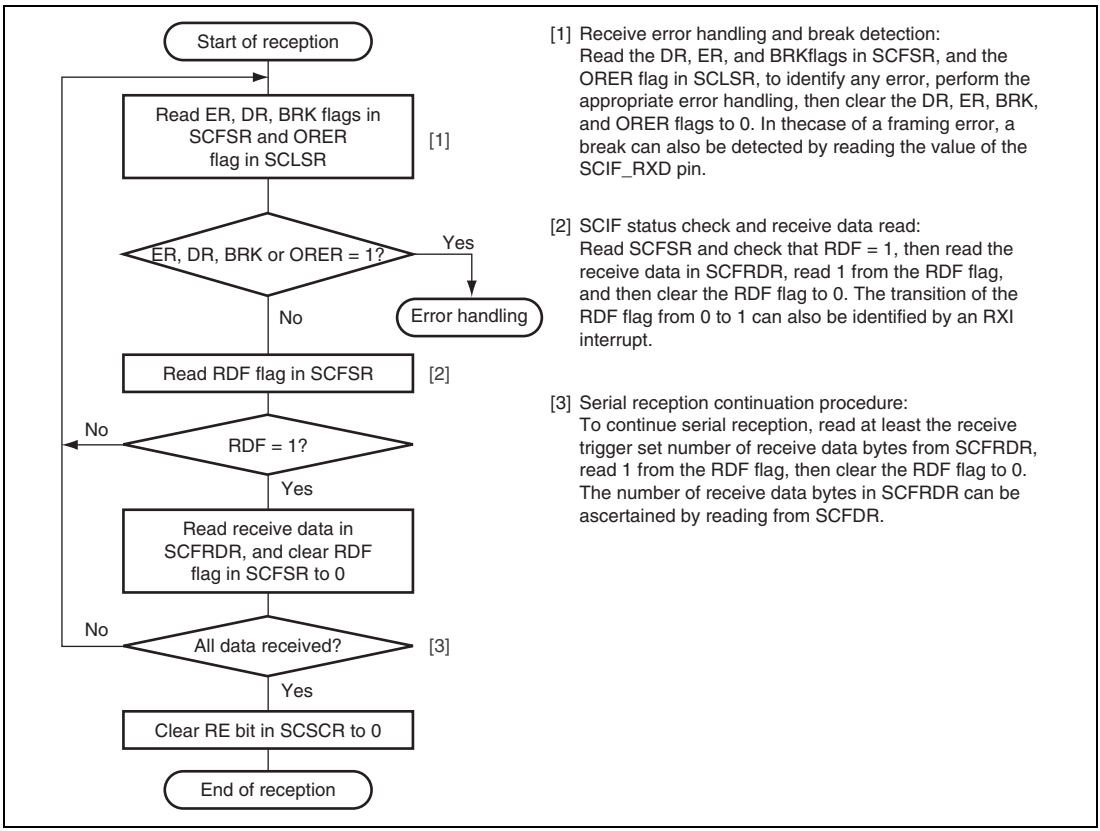
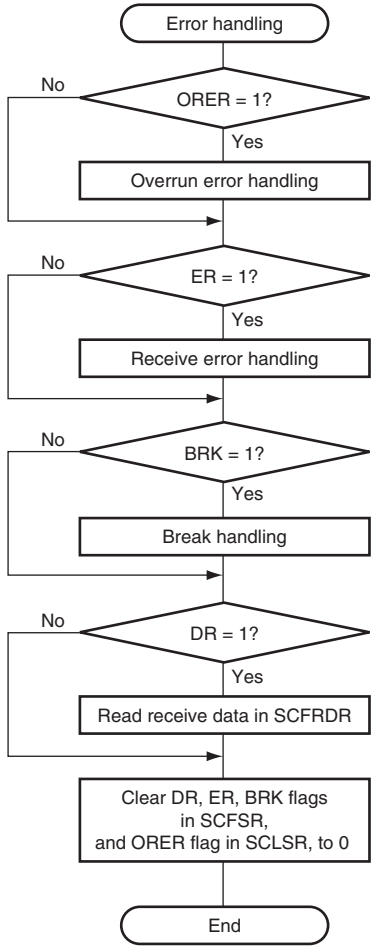


Figure 28.9 Sample Serial Reception Flowchart (1)



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

Figure 28.10 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0-start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCIF carries out the following checks.

   (a) Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   (b) The SCIF checks whether receive data can be transferred from SCRSR to SCFRDR.*

   (c) Overrun error check: The SCIF checks that the ORER flag is 0, indicating that no overrun error has occurred.*

   (d) Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.*

   If (b), (c), and (d) checks are passed, the receive data is stored in SCFRDR.

   Note: * Reception continues even when a parity error or framing error occurs.
4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 28.11 shows an example of the operation for reception in asynchronous mode.

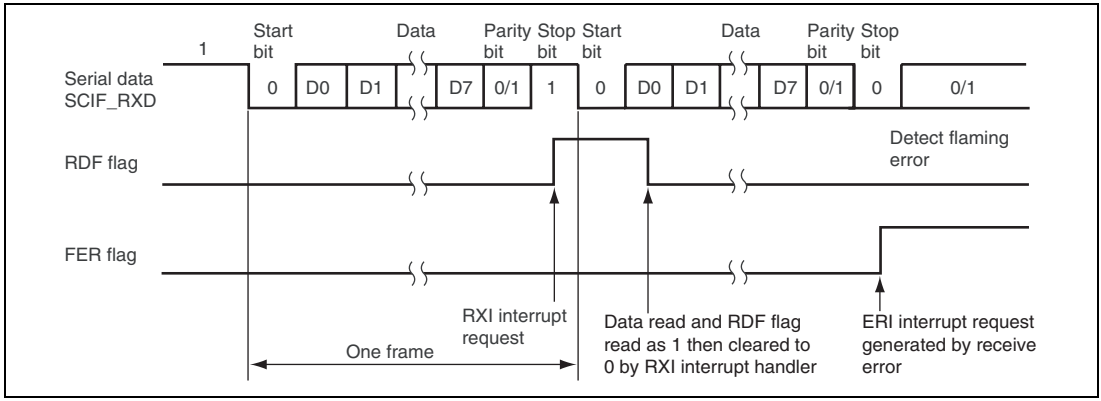


Figure 28.11 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

28.4.3 Operation in Clocked Synchronous Mode

Clocked synchronous mode, in which data is transmitted or received in synchronization with clock pulses, is suitable for fast serial communication.

Since the transmitter and receiver are independent units in the SCIF, full-duplex communication can be achieved by sharing the clock. Both the transmitter and receiver have a 16-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

Figure 28.12 shows the general format for clocked synchronous communication.

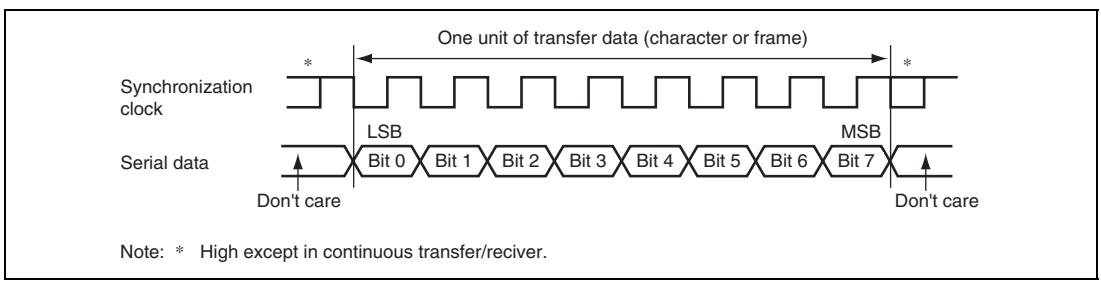


Figure 28.12 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, data on the communication line is output from one fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the start of the synchronization clock.

In serial communication, each character is output starting with the LSB and ending with the MSB. After the MSB is output, the communication line remains in the state of the last data.

In clocked synchronous mode, the SCIF receives data in synchronization with the rise of the synchronization clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity bit can be added.

(2) Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCIF_SCK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 28.7.

When the SCIF is operated on an internal clock, the synchronization clock is output from the SCIF_SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When an internal clock is selected in a receive operation only, as long as the RE bit in SCSCR is set to 1, clock pulses are output until the number of receive data bytes in the receive FIFO data register reaches the receive trigger number.

(3) SCIF Initialization (Clocked Synchronous Mode):

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When changing the operating mode or transfer format, etc., the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, FER, or ORER flag state or change the contents of SCFRDR.

Figure 28.13 shows a sample SCIF initialization flowchart.

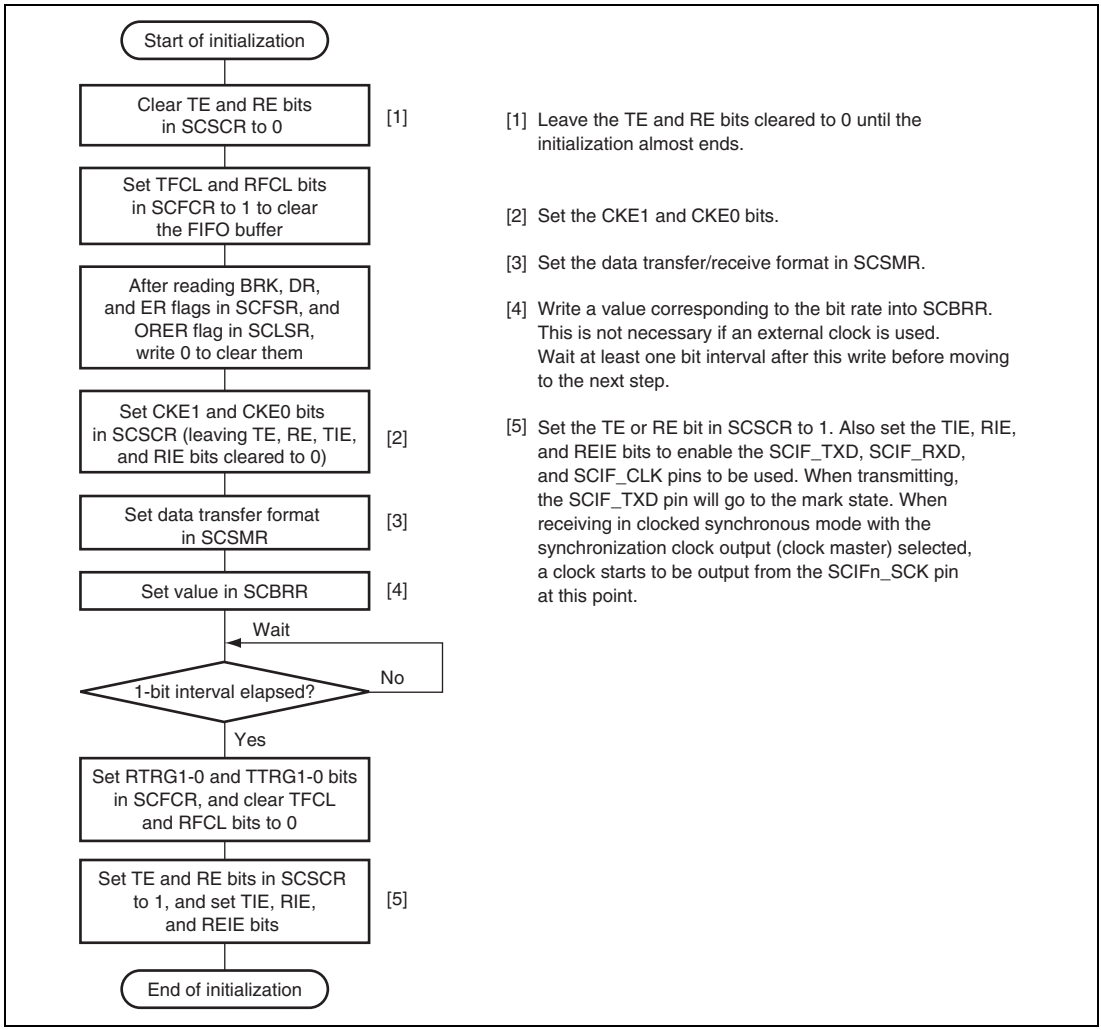
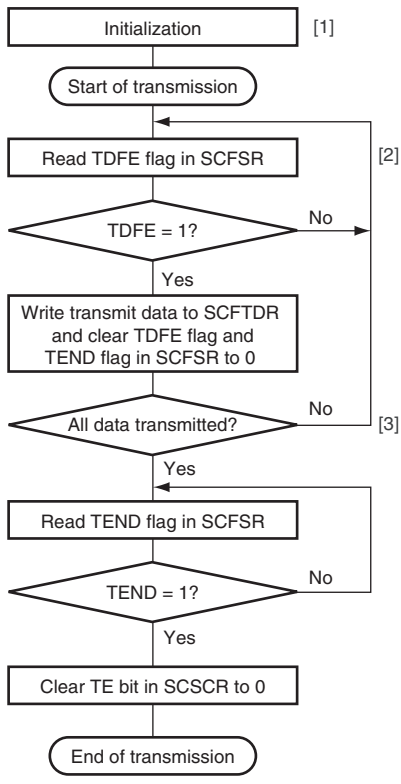


Figure 28.13 Sample SCIF Initialization Flowchart

(4) Serial Data Transmission (Clocked Synchronous Mode)

Figure 28.14 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



[1] SCIF initialization:
See sample SCIF initialization flowchart in figure 28.13.

[2] SCIF status check and transmit data write:
Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0.

[3] Serial transmission continuation procedeure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, them write data to SCFTDR, and then clear the TDFE flag to 0.

Note: In clock synchronous mode, when transmit data is written to the SCFTDR by the DMAC, the TEND flag may no be cleared. Therefore, if the DMAC is used for transmission in clock synchronous mode, read the TEND flag in the following method.
1. Confirm data transfer completion on the DMAC side.
2. Read the TEND flag.
3. Clear the TEND flag to 0 when TEND = 1.
4. Read the TEND flag again.
5. Use the TEND flag read for the second time.

Figure 28.14 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 16 (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   When the external clock is selected, data is output in synchronization with the input clock.

   The serial transmit data is sent from the SCIF_TXD pin in the LSB-first order.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the last bit. If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of the last bit.
4. After serial transmission ends, the CLK pin is fixed high.

Figure 28.15 shows an example of the operation for transmission in clocked synchronous mode.

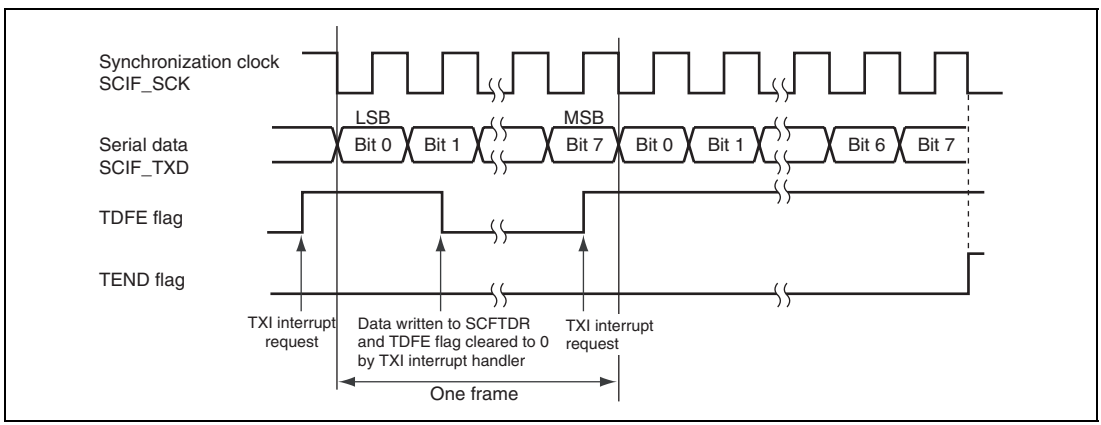


Figure 28.15 Sample SCIF Transmission Operation in Clocked Synchronous Mode

(5) Serial Data Reception (Clocked Synchronous Mode)

Figures 28.16 and 28.17 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching the operating mode from asynchronous mode to clocked synchronous mode without initializing the SCIF, make sure that the ORER flag is cleared to 0.

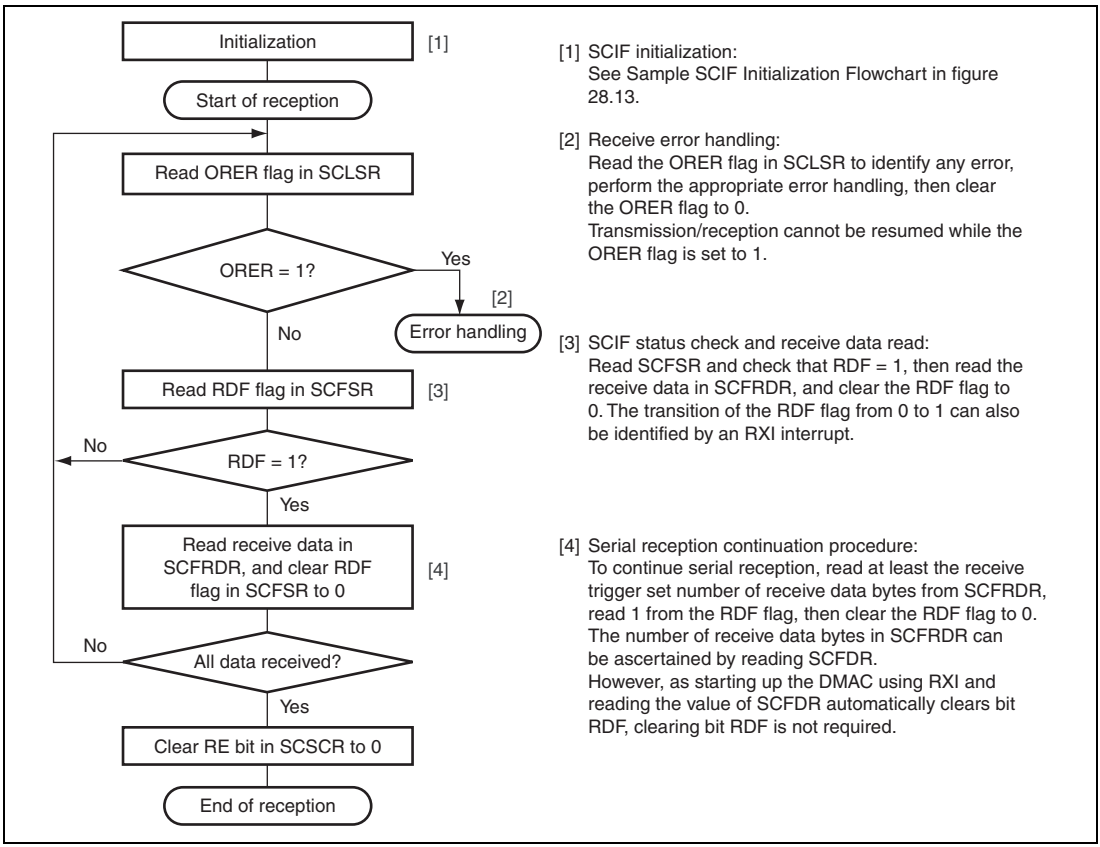


Figure 28.16 Sample Serial Reception Flowchart (1)

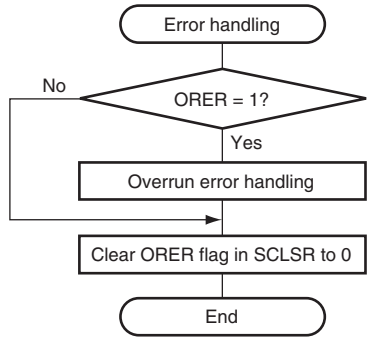


Figure 28.17 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF is initialized internally in synchronization with the input or output of the synchronization clock.
2. The received data is stored in SCRSR in LSB-to-MSB order.

   After receiving the data, the SCIF checks whether the receive data can be transferred from SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. If an overrun error is detected in the error check, reception cannot continue.
3. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interrupt (BRI) request is generated.

Figure 28.18 shows an example of the operation for reception in clocked synchronous mode.

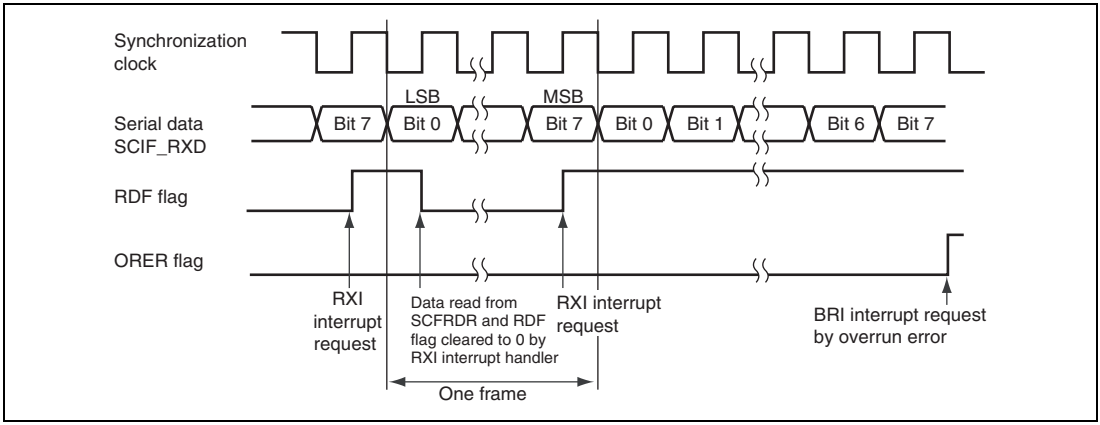


Figure 28.18 Sample SCIF Reception Operation in Clocked Synchronous Mode

(6) Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 28.19 shows a sample flowchart for simultaneous serial data transmission and reception.

Use the following procedure for simultaneous serial transmission and reception after enabling the SCIF for both transmission and reception.

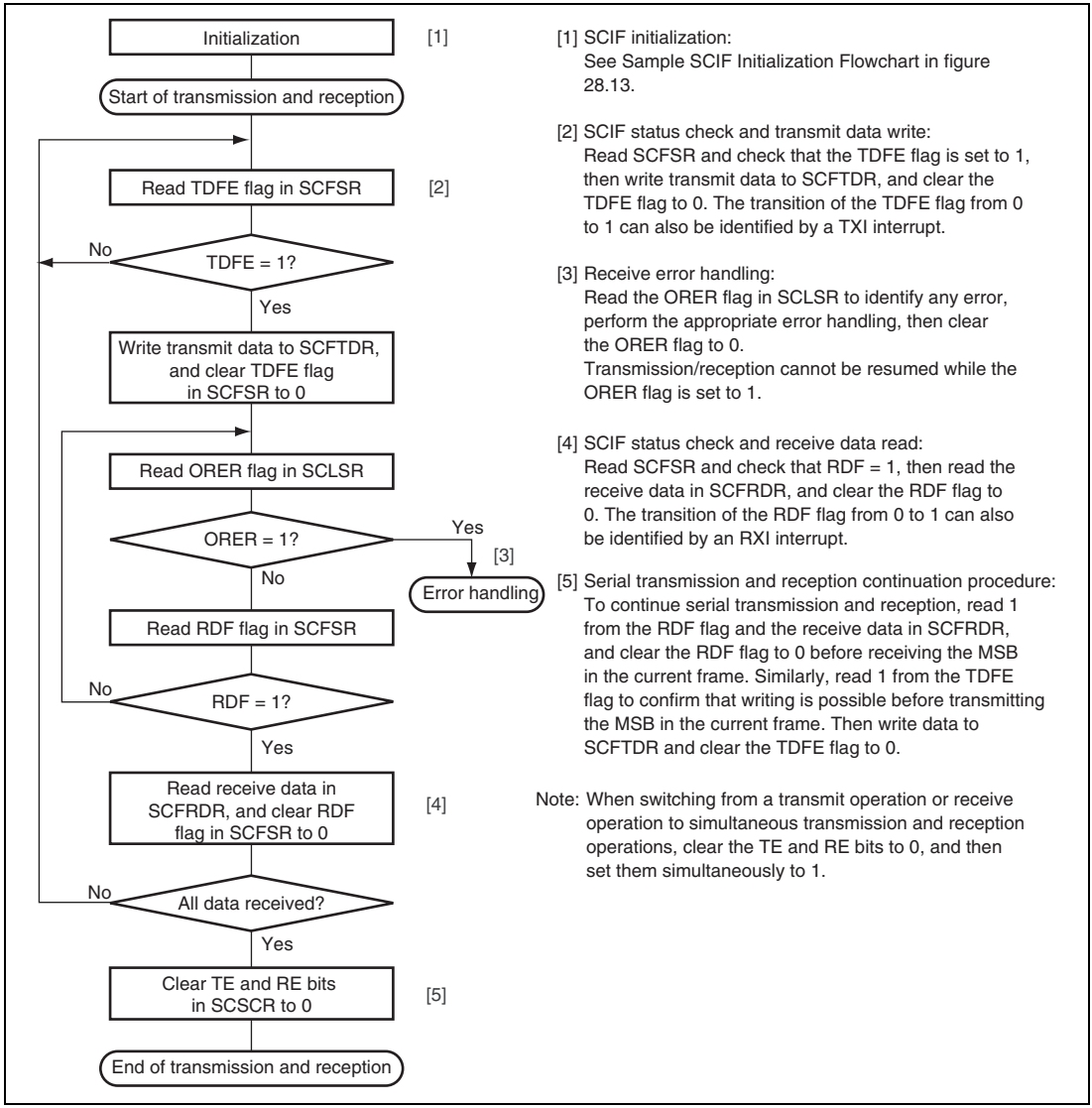


Figure 28.19 Sample Simultaneous Serial Transmission and Reception Flowchart

28.4.4 SCIF Interrupt Sources and the DMAC

The SCIF has four interrupt sources: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and break interrupt (BRI) request.

Table 28.9 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

If the TDFE flag in SCFSR is set to 1 when a TXI interrupt is enabled by the TIE bit, a TXI interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated. If the TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIFO-data-empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can activate the DMAC to perform data transfer.

If the RDF or DR flag in SCFSR is set to 1 when an RXI interrupt is enabled by the RIE bit, an RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generated. If the RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receive-FIFO-data-full request for DMA transfer is generated. A receive-FIFO-data-full request can activate the DMAC to perform data transfer. Note that generation of an RXI interrupt request or a receive-FIFO-data-full request by setting the DR flag to 1 occurs only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated. If transmission/reception is carried out using the DMAC, set and enable the DMAC before making the SCIF settings. Also make settings to inhibit output of RXI and TXI interrupt requests to the interrupt controller. If output of interrupt requests is enabled, these interrupt requests to the interrupt controller can be cleared by the DMAC regardless of the interrupt handler.

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to output ERI interrupt requests, but not RXI interrupt requests.

Table 28.9 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready flag (DR)*	Possible	
BRI	Interrupt initiated by break flag (BRK) or overrun error flag (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous mode.

28.4.5 Usage Notes

Note the following when using the SCIF.

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in SCFCR. After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again, even after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from SCFDR.

(2) SCFRDR Reading and the RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive data is read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFRDR to less than the trigger number.

The number of receive data bytes in SCFRDR can be found from SCRFDR.

(3) Break Detection and Processing

If a framing error (FER) is detected**,** break signals can also be detected by reading the SCIF_RXD pin value directly. In the break state the input from the SCIF_RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, the receive operation continues.

(4) Sending a Break Signal

The input/output condition and level of the SCIF_TXD pin are determined by bits SPB2IO and SPB2DT in SCSPTR. This feature can be used to send a break signal.

After the serial transmitter is initialized and until the TE bit is set to 1 (enabling transmission), the SCIF_TXD pin function is not selected and the value of the SPB2DT bit substitutes for the mark state. The SPB2IO and SPB2DT bits should therefore be set to 1 (designating output and high level) in the beginning.

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), and then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized, regardless of the current transmission state, and 0 is output from the SCIF_TXD pin.

(5) Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCIF operates on a base clock with a frequency of 16 times the bit rate.

In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

The timing is shown in figure 28.20.

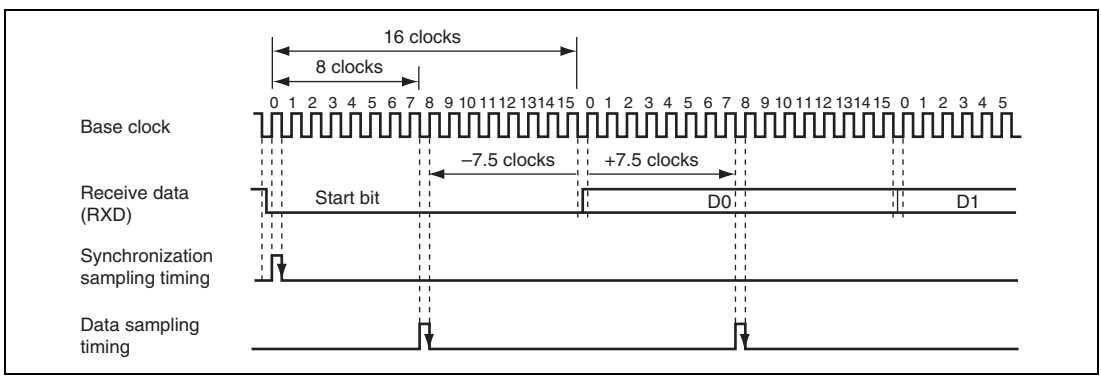


Figure 28.20 Receive Data Sampling Timing in Asynchronous Mode

Thus, the reception margin in asynchronous mode is given by formula (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{| D - 0.5 |}{N} (1 + F) \right| \times 100\ \% \quad \ldots\ldots\ldots\ldots\ (1)$$

M: Receive margin (%)

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

From equation (1), if F = 0 and D = 0.5, the reception margin is 46.875%, as given by formula (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \ldots\ldots\ldots\ldots\ (2)$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

(6) Reception margin and baud rate error

The 46.875% in formula (2) is the reception margin when the baud rate error is 0 (F = 0). In other words, reception is possible even with a misalignment of approximately half a bit if there is no reception or transmission baud rate error. If baud rate error is present in reception or transmission, the error accumulates until the stop bit is received, reducing the reception margin.

To calculate the allowable baud rate error, formula (1) can be modified to determine the value of F.

If D = 0.5, the result is formula (3).

$F = \{ (15/32 - M) / (L - 0.5) \} \times 100\ (\%)$ …… Expression (3)

Based on formula (3), the relationship between the allowable baud rate error and the reception margin when the frame length L = 12 are as follows.

Allowable baud rate error (%)	4.07	3.64	3.20	2.33	1.46
Reception margin (%)	0	5	10	20	30

28.5 Infrared Data Communication Interface

The IrDA infrared data communication interface converts (modulates) the serial communication data received from the SCIF2 channel into the data format used by the infrared data communication, and transmits the data from the SCIF2_TXD pin to the infrared sensor/emitter. Also, this module converts (demodulates) the infrared communication data transmitted from the infrared sensor/emitter to the SCIF2_RXD pin into the data format used by the serial data communication, and transmits the data to the SCIF2 channel.

28.5.1 Infrared Data Communication Format

Figure 28.21 shows the data format used by the infrared data communication.

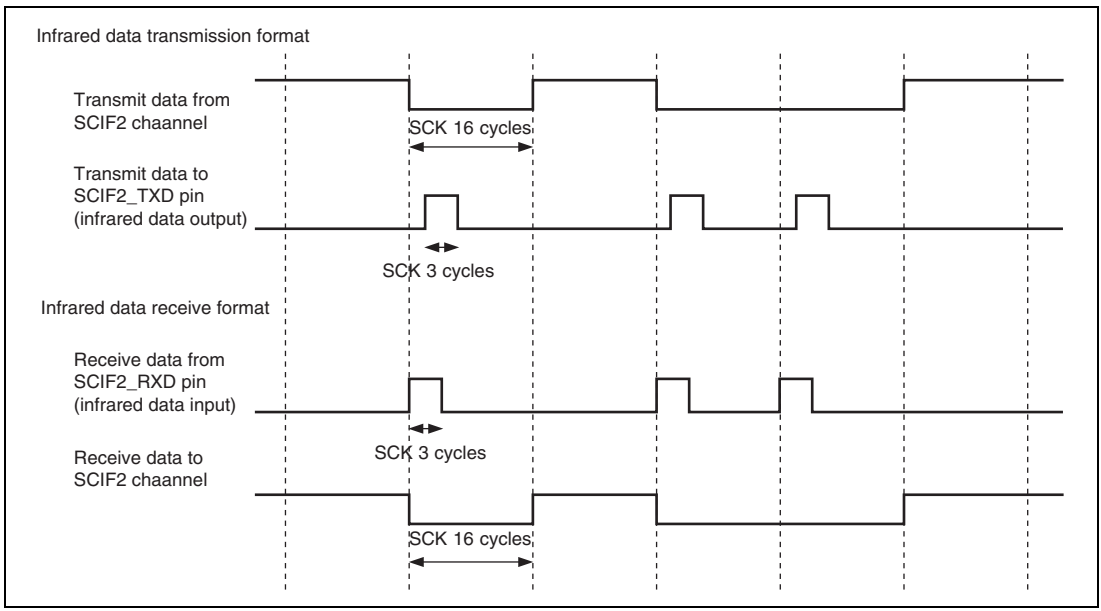


Figure 28.21 Infrared Communication Data Format

28.5.2 Operation of Infrared Data Communication Interface

Figure 28.22 shows a block diagram of the infrared data communication interface.

The infrared data communication interface selects whether serial communication data for the SCIF2 channel is modulated/demodulated to the infrared data for transmission or reception.

When the IRMOD bit in SCSMRIR is 1, the modulated/demodulated IrDA data is input/output from the SCIF2_TXD/SCIF2_RXD pin. When the IRMOD bit in SCSMRIR is 0, the data for the SCIF2 channel is bypassed to the SCIF2_TXD/SCIF2_RXD pin.

When the LOOP bit in SCSMRIR is set to active, output from the infrared data modulation unit is directly input to the demodulation unit (for loopback testing).

The selector does not use clocks for its logic. Therefore, the status of the selector after a reset or in standby depends on only the value of SCSMRIR.

Note: This module can detect short pulses such as chattering in reception, be sure to secure the shortest pulse width specified by the IrDA standard.

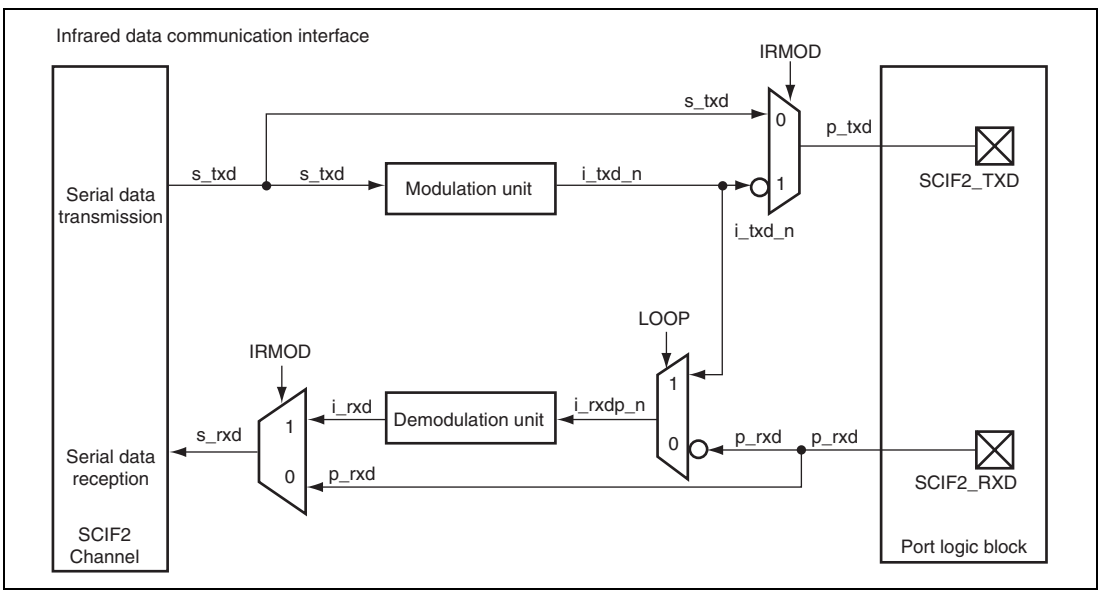


Figure 28.22 Block Diagram of Infrared Data Communication Interface

28.6 Baud Rate Generator for External Clock (BRG)

The baud rate generator for external clock (hereafter abbreviated as BRG) is included in the SCIF/IrDA and supplies the IrDA block with a sampling clock (BRGCLK), which is derived from the external clock (SCIF_CLK) or internal clock (Pck0) divided by the division ratio from 1 through 2 to the sixteenth power minus 1.

28.6.1 BRG Block Diagram

Figure 28.23 shows a block diagram of the BRG.

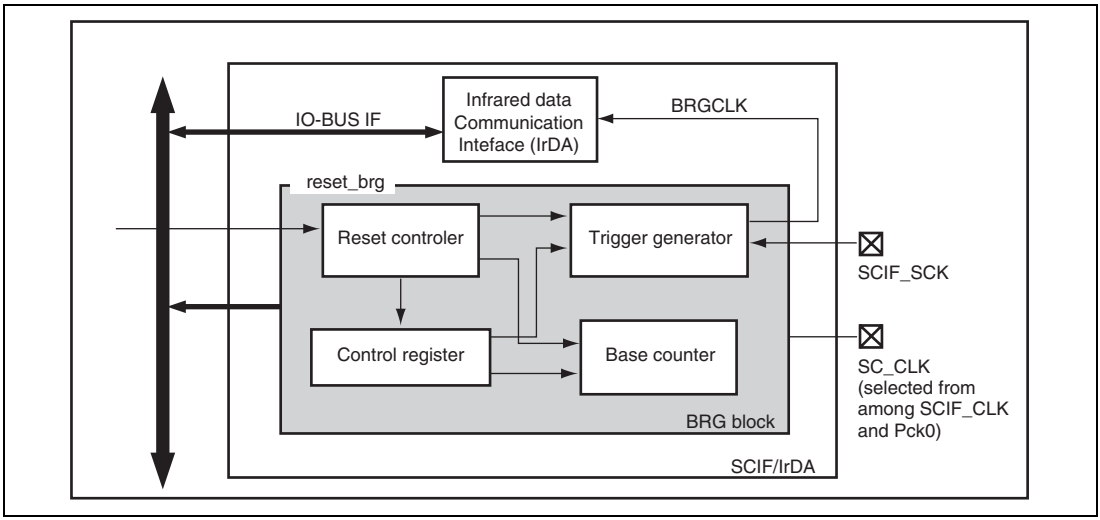


Figure 28.23 BRG Block Diagram

(1) Reset controller

The reset controller controls resetting of the control register, base counter, and trigger generator.

(2) Control register

The control register has the frequency division register and clock select register.

For details, see section 28.3, Register Descriptions.

(3) Base counter

The base counter is a 16-bit CLK (external clock BRG input) synchronization counter.

This counter is used to determine timing of a frequency divided clock when it is generated.

(4) Trigger generator

The trigger generator generates rising-edge/falling-edge triggers for a frequency divided clock, taking timing according to values of the frequency division register and base counter. The triggers generate the frequency divided clock.

The trigger generator also switches the output between SCIF2_CLK (external clock input) and the frequency divided clock.

28.6.2 Restrictions on the BRG

(1) Notes on Frequency Division Register Settings

1. At the first setting of BSGDL2 after a reset, wait time of one bit period or more is required to ensure the clock settling time.
   (Example) Period of one bit when BSGDL2 = 2

   $3.68\ (\text{MHz}) \times 1/2 \times 1/16 = 0.115\ (\text{MHz}) \rightarrow 8695\ (\text{ns})$

2. After the setting stated in 1 above, wait time of one bit period or more is required at the maximum bit rate (BSGDL2 = 65535) before the value of BSGDL2 is changed again.

Set the SCIF registers and BRG registers as shown in the tables below:

- Asynchronous mode (SC_CLK external input)

SCIF	Register/Bit Name	Setting Value	BRG	Register Name	Setting Value
	SCSMR.C/A	0		BSGCKS2	0000
	SCSCR.CKE1	1		BSGDL2	1 to FFFF

- Asynchronous mode (SCK external input)

SCIF	Register/Bit Name	Setting Value	BRG	Register Name	Setting Value
	SCSMR.C/A	0		BSGCKS2	8000
	SCSCR	1		BSGDL2	Don't care

- Clock synchronous mode (external input)

SCIF	Register/Bit Name	Setting Value	BRG	Register Name	Setting Value
	SCSMR	1		BSGCKS2	8000
	SCSCR	1		BSGDL2	Don't care

Section 29 Serial I/O with FIFO (SIOF)

This LSI includes a clock-synchronized serial I/O module with FIFO (SIOF) that comprises three channels.

29.1 Features

- Three channels of serial I/O
- Serial transfer
  16-stage 32-bit FIFOs (transmission and reception are independent of each other)
  Supports 8-bit data/16-bit data/16-bit stereo audio input/output
  MSB first for data transmission
  Supports a maximum of 48-kHz sampling rate
  Synchronization by either frame synchronization pulse or left/right channel switch
  Supports CODEC control data interface
  Connectable to linear, audio, or A-Law or µ-Law CODEC chip
  Supports both master and slave modes
- Serial clock
  An external pin input or peripheral clock (Pck0) can be selected as the clock source.
- Interrupts: One type
- DMA transfer
  Supports DMA transfer by a transfer request for transmission and reception

Figure 29.1 shows a block diagram of the SIOF.

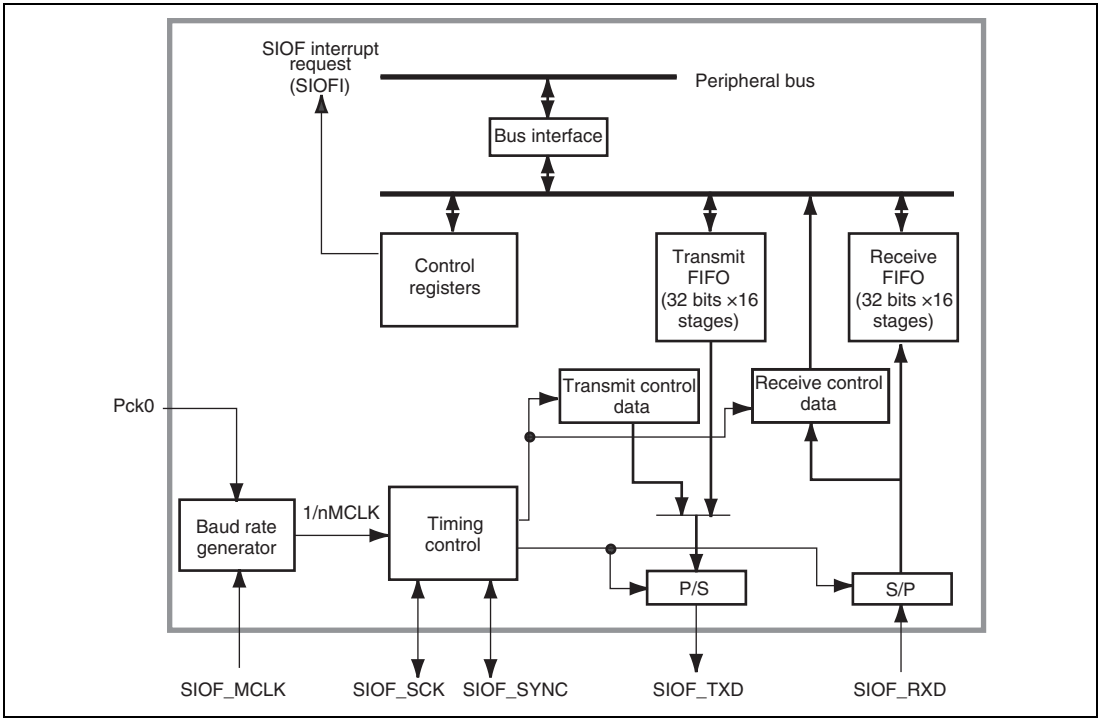


Figure 29.1 Block Diagram of SIOF

29.2 Input/Output Pins

Table 29.1 summarizes the SIOF related external pins.

Table 29.1 Pin Configuration

Channel	Pin Name	Function	I/O	Description
0	SIOF0_MCLK	Master clock	Input	Master clock input pin
	SIOF0_SCK	Serial clock	I/O	Serial clock pin (common to transmission/reception)
	SIOF0_SYNC	Frame synchronous signal	I/O	Frame synchronous signal (common to transmission/reception)
	SIOF0_TXD	Transmit data	Output	Transmit data pin
	SIOF0_RXD	Receive data	Input	Receive data pin
1	SIOF1_MCLK	Master clock	Input	Master clock input pin
	SIOF1_SCK	Serial clock	I/O	Serial clock pin (common to transmission/reception)
	SIOF1_SYNC	Frame synchronous signal	I/O	Frame synchronous signal (common to transmission/reception)
	SIOF1_TXD	Transmit data	Output	Transmit data pin
	SIOF1_RXD	Receive data	Input	Receive data pin
2	SIOF2_MCLK	Master clock	Input	Master clock input pin
	SIOF2_SCK	Serial clock	I/O	Serial clock pin (common to transmission/reception)
	SIOF2_SYNC	Frame synchronous signal	I/O	Frame synchronous signal (common to transmission/reception)
	SIOF2_TXD	Transmit data	Output	Transmit data pin
	SIOF2_RXD	Receive data	Input	Receive data pin

29.3 Register Descriptions

Table 29.2 shows the SIOF register configuration. Table 29.3 shows the register state in each operating mode.

Table 29.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
0	Mode register 0	SIMDR0	R/W	H'FFE3 0000	H'1FE3 0000	16
	Clock select register 0	SISCR0	R/W	H'FFE3 0002	H'1FE3 0002	16
	Transmit data assign register 0	SITDAR0	R/W	H'FFE3 0004	H'1FE3 0004	16
	Receive data assign register 0	SIRDAR0	R/W	H'FFE3 0006	H'1FE3 0006	16
	Control data assign register 0	SICDAR0	R/W	H'FFE3 0008	H'1FE3 0008	16
	Control register 0	SICTR0	R/W	H'FFE3 000C	H'1FE3 000C	16
	FIFO control register 0	SIFCTR0	R/W	H'FFE3 0010	H'1FE3 0010	16
	Status register 0	SISTR0	R/W	H'FFE3 0014	H'1FE3 0014	16
	Interrupt enable register 0	SIIER0	R/W	H'FFE3 0016	H'1FE3 0016	16
	Transmit data register 0	SITDR0	W	H'FFE3 0020	H'1FE3 0020	32
	Receive data register 0	SIRDR0	R	H'FFE3 0024	H'1FE3 0024	32
	Transmit control data register 0	SITCR0	R/W	H'FFE3 0028	H'1FE3 0028	32
	Receive control data register 0	SIRCR0	R/W	H'FFE3 002C	H'1FE3 002C	32
1	Mode register 1	SIMDR1	R/W	H'FFE3 8000	H'1FE3 8000	16
	Clock select register 1	SISCR1	R/W	H'FFE3 8002	H'1FE3 8002	16
	Transmit data assign register 1	SITDAR1	R/W	H'FFE3 8004	H'1FE3 8004	16
	Receive data assign register 1	SIRDAR1	R/W	H'FFE3 8006	H'1FE3 8006	16
	Control data assign register 1	SICDAR1	R/W	H'FFE3 8008	H'1FE3 8008	16
	Control register 1	SICTR1	R/W	H'FFE3 800C	H'1FE3 800C	16
	FIFO control register 1	SIFCTR1	R/W	H'FFE3 8010	H'1FE3 8010	16

Channel	Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
1	Status register 1	SISTR1	R/W	H'FFE3 8014	H'1FE3 8014	16
	Interrupt enable register 1	SIIER1	R/W	H'FFE3 8016	H'1FE3 8016	16
	Transmit data register 1	SITDR1	W	H'FFE3 8020	H'1FE3 8020	32
	Receive data register 1	SIRDR1	R	H'FFE3 8024	H'1FE3 8024	32
	Transmit control data register 1	SITCR1	R/W	H'FFE3 8028	H'1FE3 8028	32
	Receive control data register 1	SIRCR1	R/W	H'FFE3 802C	H'1FE3 802C	32
2	Mode register 2	SIMDR2	R/W	H'FFE4 0000	H'1FE4 0000	16
	Clock select register 2	SISCR2	R/W	H'FFE4 0002	H'1FE4 0002	16
	Transmit data assign register 2	SITDAR2	R/W	H'FFE4 0004	H'1FE4 0004	16
	Receive data assign register 2	SIRDAR2	R/W	H'FFE4 0006	H'1FE4 0006	16
	Control data assign register 2	SICDAR2	R/W	H'FFE4 0008	H'1FE4 0008	16
	Control register 2	SICTR2	R/W	H'FFE4 000C	H'1FE4 000C	16
	FIFO control register 2	SIFCTR2	R/W	H'FFE4 0010	H'1FE4 0010	16
	Status register 2	SISTR2	R/W	H'FFE4 0014	H'1FE4 0014	16
	Interrupt enable register 2	SIIER2	R/W	H'FFE4 0016	H'1FE4 0016	16
	Transmit data register 2	SITDR2	W	H'FFE4 0020	H'1FE4 0020	32
	Receive data register 2	SIRDR2	R	H'FFE4 0024	H'1FE4 0024	32
	Transmit control data register 2	SITCR2	R/W	H'FFE4 0028	H'1FE4 0028	32
	Receive control data register 2	SIRCR2	R/W	H'FFE4 002C	H'1FE4 002C	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 29.3 Register State in Each Operating Mode

Channel	Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
0	Mode register 0	SIMDR0	H'8000	H'8000	Retained	Retained
	Clock select register 0	SISCR0	H'C000	H'C000	Retained	Retained
	Transmit data assign register 0	SITDAR0	H'0000	H'0000	Retained	Retained
	Receive data assign register 0	SIRDAR0	H'0000	H'0000	Retained	Retained
	Control data assign register 0	SICDAR0	H'0000	H'0000	Retained	Retained
	Control register 0	SICTR0	H'0000	H'0000	Retained	Retained
	FIFO control register 0	SIFCTR0	H'1000	H'1000	Retained	Retained
	Status register 0	SISTR0	H'0000	H'0000	Retained	Retained
	Interrupt enable register 0	SIIER0	H'0000	H'0000	Retained	Retained
	Transmit data register 0	SITDR0	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Receive data register 0	SIRDR0	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Transmit control data register 0	SITCR0	H'0000 0000	H'0000 0000	Retained	Retained
	Receive control data register 0	SIRCR0	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
1	Mode register 1	SIMDR1	H'8000	H'8000	Retained	Retained
	Clock select register 1	SISCR1	H'C000	H'C000	Retained	Retained
	Transmit data assign register 1	SITDAR1	H'0000	H'0000	Retained	Retained
	Receive data assign register 1	SIRDAR1	H'0000	H'0000	Retained	Retained
	Control data assign register 1	SICDAR1	H'0000	H'0000	Retained	Retained
	Control register 1	SICTR1	H'0000	H'0000	Retained	Retained
	FIFO control register 1	SIFCTR1	H'1000	H'1000	Retained	Retained
	Status register 1	SISTR1	H'0000	H'0000	Retained	Retained
	Interrupt enable register 1	SIIER1	H'0000	H'0000	Retained	Retained
	Transmit data register 1	SITDR1	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Receive data register 1	SIRDR1	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Transmit control data register 1	SITCR1	H'0000 0000	H'0000 0000	Retained	Retained
	Receive control data register 1	SIRCR1	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained

Channel	Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
2	Mode register 2	SIMDR2	H'8000	H'8000	Retained	Retained
	Clock select register 2	SISCR2	H'C000	H'C000	Retained	Retained
	Transmit data assign register 2	SITDAR2	H'0000	H'0000	Retained	Retained
	Receive data assign register 2	SIRDAR2	H'0000	H'0000	Retained	Retained
	Control data assign register 2	SICDAR2	H'0000	H'0000	Retained	Retained
	Control register 2	SICTR2	H'0000	H'0000	Retained	Retained
	FIFO control register 2	SIFCTR2	H'1000	H'1000	Retained	Retained
	Status register 2	SISTR2	H'0000	H'0000	Retained	Retained
	Interrupt enable register 2	SIIER2	H'0000	H'0000	Retained	Retained
	Transmit data register 2	SITDR2	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Receive data register 2	SIRDR2	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
	Transmit control data register 2	SITCR2	H'0000 0000	H'0000 0000	Retained	Retained
	Receive control data register 2	SIRCR2	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained

29.3.1 Mode Register (SIMDR)

SIMDR is a 16-bit readable/writable register that sets the SIOF operating mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRMD[1:0]		SYN CAT	REDG	FL[3:0]				TXDIZ	RCIM	SYN CAC	SYN CDL	—	—	—	—
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	TRMD[1:0]	10	R/W	Transfer Mode Select transfer mode as shown in table 29.4. 00: Slave mode 1 01: Slave mode 2 10: Master mode 1 11: Master mode 2
13	SYNCAT	0	R/W	SIOF_SYNC Pin Valid Timing Indicates the position of the SIOF_SYNC signal to be output as a synchronization pulse. 0: At the start-bit data of frame 1: At the last-bit data of slot
12	REDG	0	R/W	Receive Data Sampling Edge 0: The SIOF_RXD signal is sampled at the falling edge of SIOF_SCK 1: The SIOF_RXD signal is sampled at the rising edge of SIOF_SCK Note: The timing to transmit the SIOF_TXD signal is at the opposite edge of the timing that samples the SIOF_RXD. This bit is valid only in master mode.
11 to 8	FL[3:0]	0000	R/W	Frame Length Specifies the flame length and transfer data format. For details, refer to table 29.7.

Bit	Bit Name	Initial Value	R/W	Description
7	TXDIZ	0	R/W	SIOF_TXD Pin Output when Transmission is Invalid* 0: High output (1 output) when invalid 1: High-impedance state when invalid Note: Invalid means when disabled, and when a slot that is not assigned as transmit data or control data is being transmitted.
6	RCIM	0	R/W	Receive Control Data Interrupt Mode 0: Sets the RCRDY bit in SISTR when the contents of SIRCR change. 1: Sets the RCRDY bit in SISTR each time when the SIRCR receives the control data.
5	SYNCAC	0	R/W	SIOF_SYNC Pin Polarity Valid when the SIOF_SYNC signal is output as a synchronous pulse. 0: Active-high 1: Active-low
4	SYNCDL	0	R/W	Data Pin Bit Delay for SIOF_SYNC Pin Valid when the SIOF_SYNC signal is output as synchronous pulse. Only one-bit delay is valid for transmission in slave mode. This bit should be set to 1. 0: No bit delay 1: 1-bit delay Note: * When this bit is cleared to 0 (no bit delay is selected) in slave mode, the receive data is sampled at the rising edge of SCK.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 29.4 shows the operation in each transfer mode.

Table 29.4 Operation in Each Transfer Mode

Transfer Mode	Master/Slave	SIOF_SYNC	Bit Delay	Control Data Method*
Slave mode 1	Slave	Synchronous pulse	SYNCDL bit	Slot position
Slave mode 2	Slave	Synchronous pulse		Secondary FS
Master mode 1	Master	Synchronous pulse		Slot position
Master mode 2	Master	L/R	No	Not supported

Note: * The control data method is valid only when the FL bit is specified as B'1xxx. (x: don't care.)

29.3.2 Clock Select Register (SISCR)

SISCR is a 16-bit readable/writable register that sets the serial clock generation conditions for the master clock. SISCR can be specified when the bits TRMD[1:0] in SIMDR are specified as B'10 or B'11.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSSEL	MSIMM	—	BRPS[4:0]					—	—	—	—	—	BRDV[2:0]		
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection The master clock is the clock source input to the baud rate generator (prescaler). 0: Uses the input clock signal of the SIOF_MCLK pin as the master clock 1: Uses Pck0 as the master clock
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate generator as the serial clock 1: Uses the master clock itself as the serial clock
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 8	BRPS[4:0]	00000	R/W	Prescalar Setting Set the master clock division ratio according to the count value of the prescalar of the baud rate generator. The range of settings is from 00000 (× 1/1) to 11111 (× 1/32).
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	BRDV[2:0]	000	R/W	Baud rate generator's Division Ratio Setting Set the frequency division ratio for the output stage of the baud rate generator. 000: Prescalar output × 1/2 001: Prescalar output × 1/4 010: Prescalar output × 1/8 011: Prescalar output × 1/16 100: Prescalar output × 1/32 101: Setting prohibited 110: Setting prohibited 111: Prescalar output × 1/1 • Setting 111 is valid only when the bits BRPS[4:0] are set to 00001. The final frequency division ratio of the baud rate generator is determined by the value of BRPS and BRDV (maximum 1/1024).

29.3.3 Control Register (SICTR)

SICTR is a 16-bit readable/writable register that sets the SIOF operating state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKE	FSE	—	—	—	—	TXE	RXE	—	—	—	—	—	—	TXRST	RXRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 0: Disables the SIOF_SCK output (outputs 0) 1: Enables the SIOF_SCK output If this bit is set to 1, the SIOF initializes the baud rate generator and initiates the operation. At the same time, the SIOF outputs the clock generated by the baud rate generator to the SIOF_SCK pin.
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. 0: Disables the SIOF_SYNC output (outputs 0) 1: Enables the SIOF_SYNC output If this bit is set to 1, the SIOF initializes the frame counter and initiates the operation.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	TXE	0	R/W	Transmit Enable 0: Disables data transmission from the SIOF_TXD pin 1: Enables data transmission from the SIOF_TXD pin • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOF_SYNC signal). • When the 1 setting for this bit becomes valid, the SIOF issues a transmit transfer request according to the setting of the TFWM bit in SIFCTR. When transmit data is stored in the transmit FIFO, transmission of data from the SIOF_TXD pin begins. This bit is initialized upon a transmit reset.
8	RXE	0	R/W	Receive Enable 0: Disables data reception from SIOF_RXD 1: Enables data reception from SIOF_RXD • This bit setting becomes valid at the start of the next frame (at the rising edge of the SIOF_SYNC signal). • When the 1 setting for this bit becomes valid, the SIOF begins the reception of data from the SIOF_RXD pin. When receive data is stored in the receive FIFO, the SIOF issues a reception transfer request according to the setting of the RFWM bit in SIFCTR. This bit is initialized upon receive reset.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TXRST	0	R/W	Transmit Reset 0: Does not reset transmit operation 1: Resets transmit operation • This bit setting becomes valid immediately. For details of transmit reset, refer to table 29.13. • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. For details of receive reset, refer to table 29.13. • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always read as 0.

29.3.4 Transmit Data Register (SITDR)

SITDR is a 32-bit write-only register that specifies the SIOF operating status.

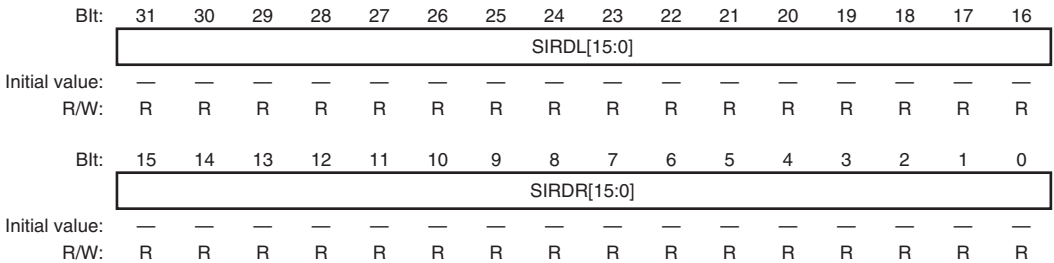
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITDL[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL[15:0]	Undefined	W	Left-Channel Transmit Data Specify data to be output from the SIOF_TXD pin as left-channel data. The position of the left-channel data in the transmit frame is specified by the TDLA bit in SITDAR. • These bits are valid only when the TDLE bit in SITDAR is set to 1.
15 to 0	SITDR[15:0]	Undefined	W	Right-Channel Transmit Data Specify data to be output from the SIOF_TXD pin as right-channel data. The position of the right-channel data in the transmit frame is specified by the TDRA bit in SITDAR. • These bits are valid only when the TDRE bit and TLREP bit in SITDAR are set to 1 and cleared to 0, respectively.

29.3.5 Receive Data Register (SIRDR)

SIRDR is a 32-bit read-only register that reads receive data of the SIOF. SIRDR stores data in the receive FIFO.

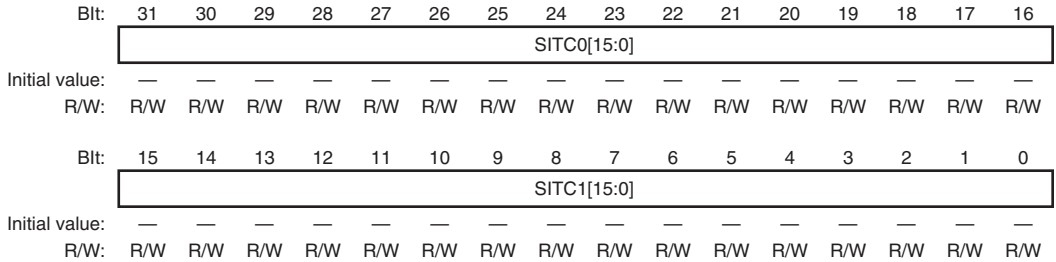


Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL[15:0]	Undefined	R	Left-Channel Receive Data Store data received from the SIOF_RXD pin as left-channel data. The position of the left-channel data in the receive frame is specified by the RDLA bit in SIRDAR. • These bits are valid only when the RDLE bit in SIRDAR is set to 1.
15 to 0	SIRDR[15:0]	Undefined	R	Right-Channel Receive Data Store data received from the SIOF_RXD pin as right-channel data. The position of the right-channel data in the receive frame is specified by the RDRA bit in SIRDAR. • These bits are valid only when the RDRE bit in SIRDAR is set to 1.

29.3.6 Transmit Control Data Register (SITCR)

SITCR is a 32-bit readable/writable register that specifies transmit control data of the SIOF. SITCR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: don't care.).

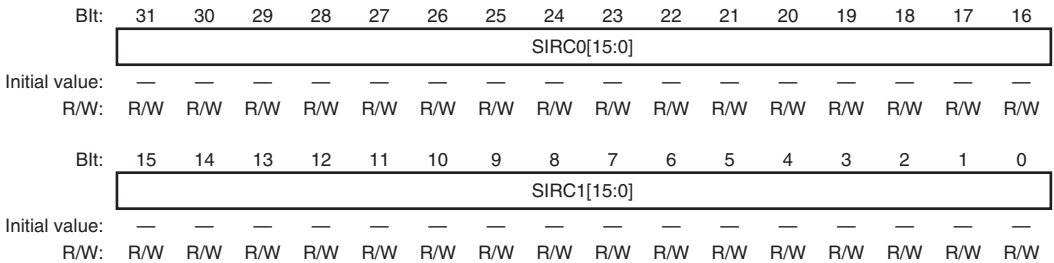
SITCR is initialized by the conditions specified in table 29.3, Register State of SIOF in Each Processing Mode, or by a transmit reset caused by the TXRST bit in SICTR.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC0[15:0]	H'0000	R/W	Control Channel 0 Transmit Data Specify data to be output from the SIOF_TXD pin as control channel 0 transmit data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SITC1[15:0]	H'0000	R/W	Control Channel 1 Transmit Data Specify data to be output from the SIOF_TXD pin as control channel 1 transmit data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

29.3.7 Receive Control Data Register (SIRCR)

SIRCR is a 32-bit readable/writable register that stores receive control data of the SIOF. SIRCR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: don't care.).



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRC0[15:0]	Undefined	R/W	Control Channel 0 Receive Data Store data received from the SIOF_RXD pin as control channel 0 receive data. The position of the control channel 0 data in the transmit or receive frame is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E bit in SICDAR is set to 1.
15 to 0	SIRC1[15:0]	Undefined	R/W	Control Channel 1 Receive Data Store data received from the SIOF_RXD pin as control channel 1 receive data. The position of the control channel 1 data in the transmit or receive frame is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E bit in SICDAR is set to 1.

29.3.8 Status Register (SISTR)

SISTR is a 16-bit readable/writable register that shows the SIOF state. Each bit in this register becomes an SIOF interrupt source when the corresponding bit in SIIER is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared to 0, SITCR is over-written and the previous contents of SITCR are not output from the SIOF_TXD pin. • This bit is valid when the TXE bit in SITCR is set to 1. • This bit indicates a state of the SIOF. If SITCR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if SITDR is written, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
12	TDREQ	0	R	Transmit Data Transfer Request 0: Indicates that the size of empty space in the transmit FIFO does not exceed the size specified by the TFWM bit in SIFCTR. 1: Indicates that the size of empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. A transmit data transfer request is issued when the empty space in the transmit FIFO exceeds the size specified by the TFWM bit in SIFCTR. When using transmit data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the TXE bit in SICTR is 1. • This bit indicates a state; if the size of empty space in the transmit FIFO is less than the size specified by the TFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	RCRDY	0	R	Receive Control Data Ready 0: Indicates that the SIRCR stores no valid data. 1: Indicates that the SIRCR stores valid data. • If SIRCR is written when this bit is set to 1, SIRCR is modified by the latest data. • This bit is valid when the RXE bit in SICTR is set to 1. • This bit indicates a state of the SIOF. If SIRCR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
9	RFFUL	0	R	Receive FIFO Full 0: Receive FIFO not full 1: Receive FIFO full • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if SIRDR is read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid space in the receive FIFO does not exceed the size specified by the RFWM bit in SIFCTR. 1: Indicates that the size of valid space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. A receive data transfer request is issued when the valid data space in the receive FIFO exceeds the size specified by the RFWM bit in SIFCTR. When using receive data transfer through the DMAC, this bit is always cleared by one DMAC access. After DMAC access, when conditions for setting this bit are satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the RXE bit in SICTR is 1. • This bit indicates a state; if the size of valid data space in the receive FIFO is less than the size specified by the RFWM bit in SIFCTR, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SAERR	0	R/W	Slot Assign Error 0: Indicates that no slot assign error occurs 1: Indicates that a slot assign error occurs A slot assign error occurs when the specifications in SITDAR, SIRDAR, and SICDAR overlap. If a slot assign error occurs, the SIOF does not transmit data to the SIOF_TXD pin and does not receive data from the SIOF_RXD pin. Note that the SIOF does not clear the TXE bit or RXE bit in SICTR at a slot assign error. • This bit is valid when the TXE bit or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error occurs 1: Indicates that a frame synchronization error occurs A frame synchronization error occurs when the next frame synchronization timing appears before the previous data or control data transfers have been completed. If a frame synchronization error occurs, the SIOF performs transmission or reception for slots that can be transferred. • This bit is valid when the TXE or RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
3	TFOVF	0	R/W	Transmit FIFO Overflow 0: No transmit FIFO overflow 1: Transmit FIFO overflow A transmit FIFO overflow means that there has been an attempt to write to SITDR when the transmit FIFO is full. When a transmit FIFO overflow occurs, the SIOF indicates overflow, and writing is invalid. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading for transmission has occurred when the transmit FIFO is empty. When a transmit FIFO underflow occurs, the SIOF repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
1	RFUDF	0	R/W	Receive FIFO Underflow 0: No receive FIFO underflow 1: Receive FIFO underflow A receive FIFO underflow means that reading of SIRDR has occurred when the receive FIFO is empty. When a receive FIFO underflow occurs, the value of data read from SIRDR is not guaranteed. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF indicates overflow, and receive data is lost. • This bit is valid when the RXE bit in SICTR is 1. • When 1 is written to this bit, the contents are cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled, an SIOF interrupt is issued.

29.3.9 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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TD MAE	TCR DYE	TFE MPE	TDR EQE	RD MAE	RC RDYE	RF FULE	RD REQE	—	—	SA ERRE	FS ERRE	TF OVFE	TF UDFE	RF UDFE	RF OVFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDMAE	0	R/W	Transmit Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The TDREQE bit can be set as transmit interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC
14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control data ready 1: Enables interrupts due to transmit control data ready
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO empty 1: Enables interrupts due to transmit FIFO empty
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data transfer requests 1: Enables interrupts due to transmit data transfer requests
11	RDMAE	0	R/W	Receive Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CPU/DMA transfer request. The RDREQE bit can be set as receive interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMAC

Bit	Bit Name	Initial Value	R/W	Description
10	RCRDYE	0	R/W	Receive Control Data Ready Enable 0: Disables interrupts due to receive control data ready 1: Enables interrupts due to receive control data ready
9	RFFULE	0	R/W	Receive FIFO Full Enable 0: Disables interrupts due to receive FIFO full 1: Enables interrupts due to receive FIFO full
8	RDREQE	0	R/W	Receive Data Transfer Request Enable 0: Disables interrupts due to receive data transfer requests 1: Enables interrupts due to receive data transfer requests
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SAERRE	0	R/W	Slot Assign Error Enable 0: Disables interrupts due to slot assign error 1: Enables interrupts due to slot assign error
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronization error 1: Enables interrupts due to frame synchronization error
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overflow 1: Enables interrupts due to transmit FIFO overflow
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO underflow 1: Enables interrupts due to transmit FIFO underflow
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO underflow 1: Enables interrupts due to receive FIFO underflow
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overflow 1: Enables interrupts due to receive FIFO overflow

29.3.10 FIFO Control Register (SIFCTR)

SIFCTR is a 16-bit readable/writable register that indicates the area available for the transmit/receive FIFO transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TFWM[2:0]			TFUA[4:0]					RFWM[2:0]			RFUA[4:0]				
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TFWM[2:0]	000	R/W	Transmit FIFO Watermark 000: Issue a transfer request when 16 stages of the transmit FIFO are empty. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or more stages of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more stages of the transmit FIFO are empty. 110: Issue a transfer request when 4 or more stages of the transmit FIFO are empty. 111: Issue a transfer request when 1 or more stages of transmit FIFO are empty. • A transfer request to the transmit FIFO is issued by the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 stages of the FIFO regardless of these bit settings. Note: * When the transmit data is DMA-transferred with the TDMAE bit in the SIIER register set to 1, the TFWM[2:0] bits should not be set to B'111. If these bits are set to B'111, a transmit FIFO overflow may occur.
12 to 8	TFUA[4:0]	10000	R	Transmit FIFO Usable Area Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (full) to B'10000 (empty).

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	RFWM[2:0]	000	R/W	Receive FIFO Watermark 000: Issue a transfer request when 1 stage or more of the receive FIFO are valid. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 4 or more stages of the receive FIFO are valid. 101: Issue a transfer request when 8 or more stages of the receive FIFO are valid. 110: Issue a transfer request when 12 or more stages of the receive FIFO are valid. 111: Issue a transfer request when 16 stages of the receive FIFO are valid. • A transfer request to the receive FIFO is issued by the RDREQE bit in SISTR. • The receive FIFO is always used as 16 stages of the FIFO regardless of these bit settings.
4 to 0	RFUA[4:0]	00000	R	Receive FIFO Usable Area Indicate the number of words that can be transferred by the CPU or DMAC as B'00000 (empty) to B'10000 (full).

29.3.11 Transmit Data Assign Register (SITDAR)

SITDAR is a 16-bit readable/writable register that specifies the position of the transmit data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLE	—	—	—	TDLA[3:0]				TDRE	TLREP	—	—	TDRA[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	TDLA[3:0]	0000	R/W	Transmit Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the left channel is specified in the SITDL bit in SITDR.
7	TDRE	0	R/W	Transmit Right-Channel Data Enable 0: Disables right-channel data transmission 1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit in SITDR as right-channel data 1: Repeatedly transmits data specified in the SITDL bit in SITDR as right-channel data • This bit setting is valid when the TDRE bit is set to 1. • When this bit is set to 1, the SITDR settings are ignored.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	TDRA[3:0]	0000	R/W	Transmit Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the right channel is specified in the SITDR bit in SITDR.

29.3.12 Receive Data Assign Register (SIRDAR)

SIRDAR is a 16-bit readable/writable register that specifies the position of the receive data in a frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLE	—	—	—	RDLA[3:0]				RDRE	—	—	—	RDRA[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RDLE	0	R/W	Receive Left-Channel Data Enable 0: Disables left-channel data reception 1: Enables left-channel data reception
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 8	RDLA[3:0]	0000	R/W	Receive Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a receive frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the left channel is stored in the SIRDL bit in SIRDR.

Bit	Bit Name	Initial Value	R/W	Description
7	RDRE	0	R/W	Receive Right-Channel Data Enable 0: Disables right-channel data reception 1: Enables right-channel data reception
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	RDRA[3:0]	0000	R/W	Receive Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a receive frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the right channel is stored in the SIRDR bit in SIRDR.

29.3.13 Control Data Assign Register (SICDAR)

SICDAR is a 16-bit readable/writable register that specifies the position of the control data in a frame. SICDAR can be specified only when the FL bit in SIMDR is specified as 1xxx (x: don't care.).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD0E	—	—	—	CD0A[3:0]				CD1E	—	—	—	CD1A[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CD0E	0	R/W	Control Channel 0 Data Enable 0: Disables transmission and reception of control channel 0 data 1: Enables transmission and reception of control channel 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	CD0A[3:0]	0000	R/W	Control Channel 0 Data Assigns 3 to 0 Specify the position of control channel 0 data in a receive or transmit frame as 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.
7	CD1E	0	R/W	Control Channel 1 Data Enable 0: Disables transmission and reception of control channel 1 data 1: Enables transmission and reception of control channel 1 data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CD1A[3:0]	0000	R/W	Control Channel 1 Data Assigns 3 to 0 Specify the position of control channel 1 data in a receive or transmit frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the control channel 1 data is specified in the SITD1 bit in SITCR. • Receive data for the control channel 1 data is stored in the SIRD1 bit in SIRCR.

29.4 Operation

29.4.1 Serial Clocks

(1) Master/Slave Modes

The following modes are available as the SIOF clock mode.

- Slave mode: SIOF_SCK, SIOF_SYNC input
- Master mode: SIOF_SCK, SIOF_SYNC output

(2) Baud Rate Generator

In SIOF master mode, the baud rate generator (BRG) is used to generate the serial clock. The division ratio is from 1/1 to 1/1024.

Note that, when using master clock directly as the serial clock without division by BRG (division ratio: 1/1), the MSIMM bit in SISCR should be set to 1.

Figure 29.2 shows connections for supply of the serial clock.

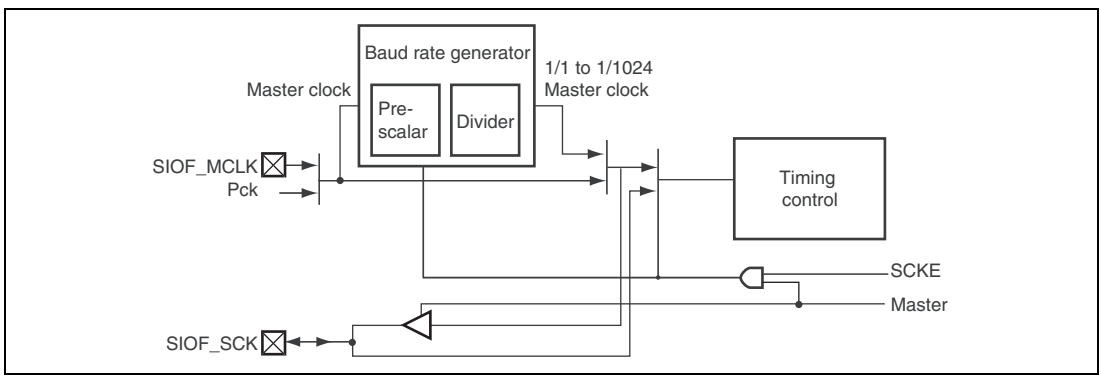


Figure 29.2 Serial Clock Supply

Table 29.5 shows an example of serial clock frequency.

Table 29.5 SIOF Serial Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.2896 MHz	12.288 MHz

29.4.2 Serial Timing

SIOF_SYNC**:** The SIOF_SYNC is a frame synchronous signal. Depending on the transfer mode, it has the following two functions.

- Synchronous pulse: 1-bit-width pulse indicating the start of the frame
- L/R: 1/2-frame-width pulse indicating the left-channel stereo data (L) in high level and the right-channel stereo data (R) in low level

Figure 29.3 shows the SIOF_SYNC synchronization timing.

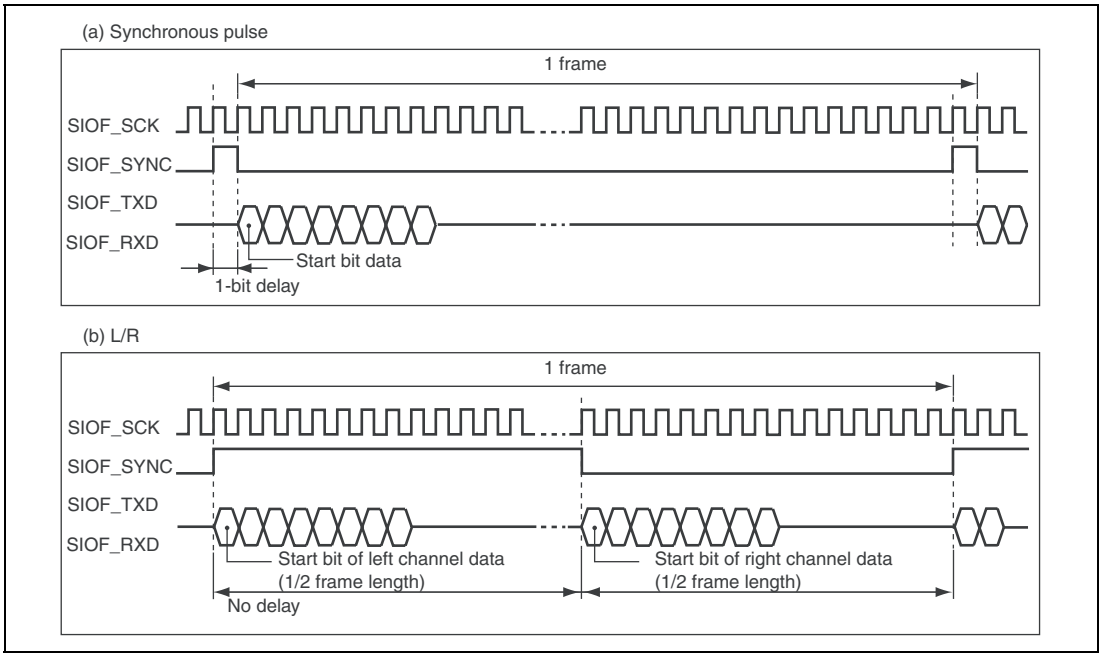


Figure 29.3 Serial Data Synchronization Timing

(1) Transmit/Receive Timing

The SIOF_TXD transmit timing and SIOF_RXD receive timing relative to the SIOF_SCK can be set as the sampling timing in the following two ways. The transmit/receive timing is set using the REDG bit in SIMDR.

- Falling-edge sampling
- Rising-edge sampling

Figure 29.4 shows the transmit/receive timing.

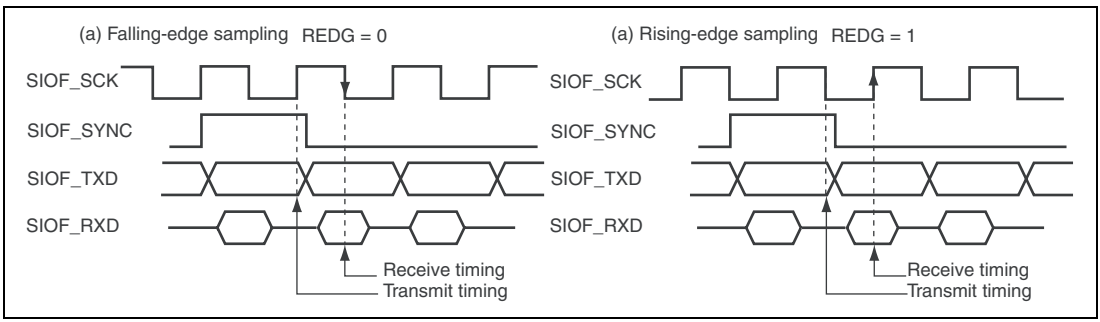


Figure 29.4 SIOF Transmit/Receive Timing

29.4.3 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)

(1) Transfer Mode

The SIOF supports the following four transfer modes as listed in table 29.6. The transfer mode can be specified by the bits TRMD[1:0] in SIMDR.

Table 29.6 Serial Transfer Modes

TRMD[1:0]	Transfer Mode	SIOF_SYNC	Bit Delay	Control Data*
00	Slave mode 1	Synchronous pulse	SYNCDL bit	Slot position
01	Slave mode 2	Synchronous pulse		Secondary FS
10	Master mode 1	Synchronous pulse		Slot position
11	Master mode 2	L/R	No	Not supported

Note: * The control data method is valid only when the FL bit is specified as B'1xxx (x: don't care).

(2) Frame Length

The length of the frame to be transferred by the SIOF is specified by the bits FL[3:0] in SIMDR. Table 29.7 shows the relationship between the bits FL[3:0] settings and frame length.

Table 29.7 Frame Length

FL[3:0]	Slot Length	Number of Bits in a Frame	Transfer Data
00XX	8	8	8-bit monaural data
0100	8	16	8-bit monaural data
0101	8	32	8-bit monaural data
0110	8	64	8-bit monaural data
0111	8	128	8-bit monaural data
10XX	16	16	16-bit monaural data
1100	16	32	16-bit monaural stereo data
1101	16	64	16-bit monaural stereo data
1110	16	128	16-bit monaural stereo data
1111	16	256	16-bit monaural stereo data

Note: X: Don't care.

(3) 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.

29.4.4 Register Allocation of Transfer Data

(1) 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 29.5 shows the transmit/receive data and the SITDR and SIRDR bit alignment.

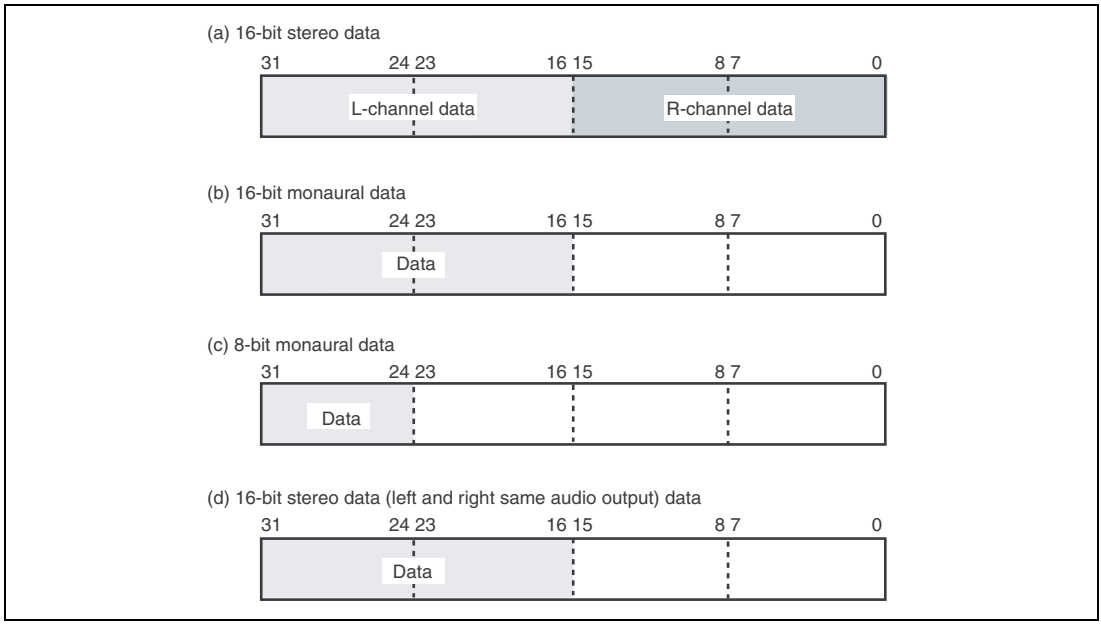


Figure 29.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Therefore, access must be made in byte units for 8-bit data, and in word units for 16-bit data. Data in unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in SITDAR. Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SIRDAR. To achieve left and right same audio output while stereo is specified for transmit data, specify the TLREP bit in SITDAR. Table 29.8 and table 29.9 show the audio mode specification for transmit data and that for receive data, respectively.

Table 29.8 Audio Mode Specification for Transmit Data

	Bit		
Mode	**TDLE**	**TDRE**	**TLREP**
Monaural	1	0	x
Stereo	1	1	0
Left and right same audio output	1	1	1

Note: x: Don't care

Table 29.9 Audio Mode Specification for Receive Data

	Bit	
Mode	**RDLE**	**RDRE**
Monaural	1	0
Stereo	1	1

Note: Left and right same audio mode is not supported in receive data.
To execute 8-bit monaural transmission or reception, use the left channel.

(2) 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 29.6 shows the control data and bit alignment in SITCR and SIRCR.

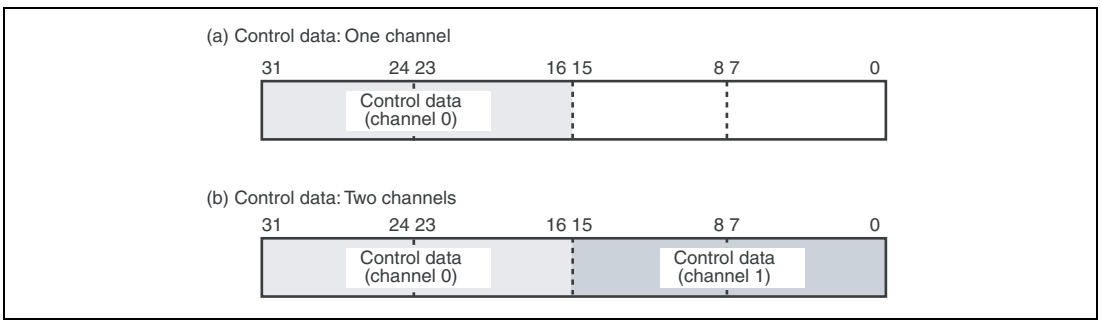


Figure 29.6 Control Data Bit Alignment

The number of channels in control data is specified by the CD0E and CD1E bits in SICDAR. Table 29.10 shows the relationship between the number of channels in control data and bit settings.

Table 29.10 Setting Number of Channels in Control Data

	Bit	
Number of Channels	**CD0E**	**CD1E**
1	1	0
2	1	1

Note: To use only one channel in control data, use channel 0.

29.4.5 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.

(1) 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 29.7 shows an example of the control data interface timing by slot position control.

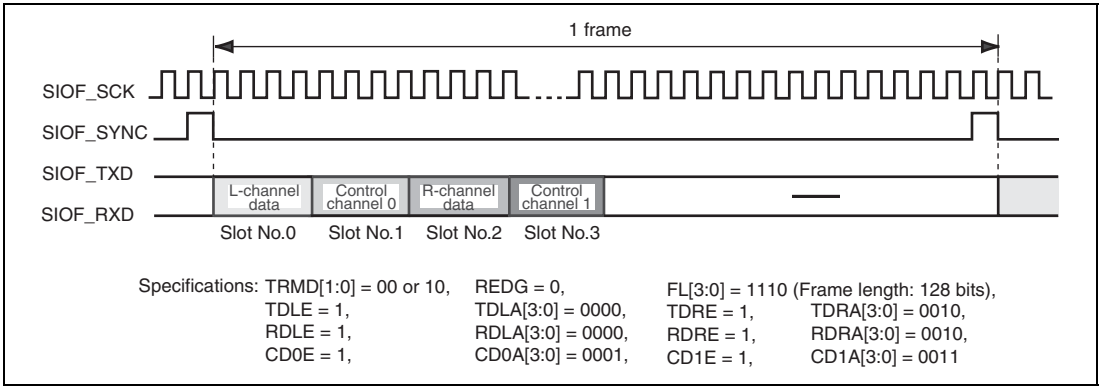


Figure 29.7 Control Data Interface (Slot Position)

(2) Control by Secondary FS (Slave Mode 2)

The CODEC normally outputs the SIOF_SYNC signal as synchronization pulse (FS). In this method, the CODEC outputs the secondary FS specific to the control data transfer after 1/2 frame time has been passed (not the normal FS output timing) to transmit or receive control data. This method is valid for SIOF slave mode. The following summarizes the control data interface procedure by the secondary FS.

- Transmit normal transmit data of LSB = 0 (the SIOF forcibly clears 0).
- To execute control data transmission, send transmit data of LSB = 1 (the SIOF forcibly set to 1 by writing SITCR).
- The CODEC outputs the secondary FS.
- The SIOF transmits or receives (stores in SIRCR) control data (data specified by SITCR) synchronously with the secondary FS.

Figure 29.8 shows an example of the control data interface timing by the secondary FS.

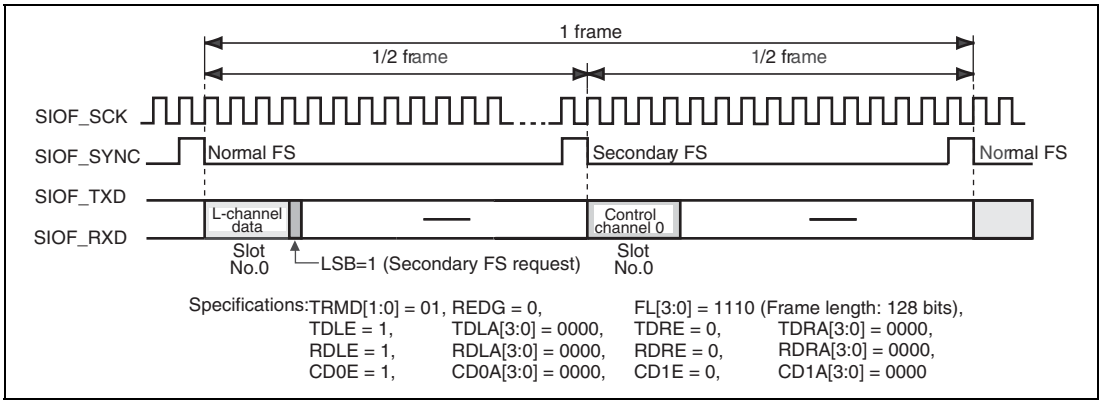


Figure 29.8 Control Data Interface (Secondary FS)

29.4.6 FIFO

(1) 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.)

(2) Transfer Request

The transfer request of the FIFO can be issued to the CPU or DMAC as the following interrupt sources.

- FIFO transmit request: TDREQ (transmit interrupt source)
- FIFO receive request: RDREQ (receive interrupt source)

The request conditions for FIFO transmit or receive can be specified individually. The request conditions for the FIFO transmit and receive are specified by the bits TFWM[2:0] and the bits RFWM[2:0] in SIFCTR, respectively. Table 29.11 and table 29.12 summarize the conditions specified by SIFCTR.

Table 29.11 Conditions to Issue Transmit Request

TFWM[2:0]	Number of Requested Stages	Transmit Request	Used Areas
000	1	Empty area is 16 stages	Smallest
100	4	Empty area is 12 stages or more	
101	8	Empty area is 8 stages or more	
110	12	Empty area is 4 stages or more	
111	16	Empty area is 1 stage or more	Largest

Table 29.12 Conditions to Issue Receive Request

RFWM[2:0]	Number of Requested Stages	Receive Request	Used Areas
000	1	Valid data is 1 stage or more	Smallest
100	4	Valid data is 4 stages or more	
101	8	Valid data is 8 stages or more	
110	12	Valid data is 12 stages or more	
111	16	Valid data is 16 stages	Largest

The number of stages of the FIFO is always sixteen even if the data area or empty area exceeds the FIFO size (the number of FIFOs). Accordingly, an overflow error or underflow error occurs if data area or empty area exceeds sixteen FIFO stages. The FIFO transmit or receive request is canceled when the above condition is not satisfied even if the FIFO is not empty or full.

(3) Number of FIFOs

The number of FIFO stages used in transmission and reception is indicated by the following register.

- Transmit FIFO: The number of empty FIFO stages is indicated by the bits TFUA[4:0] in SIFCTR.
- Receive FIFO: The number of valid data stages is indicated by the bits RFUA[4:0] in SIFCTR.

The above indicate possible data numbers that can be transferred by the CPU or DMAC.

29.4.7 Transmit and Receive Procedures

(1) Transmission in Master Mode

Figure 29.9 shows an example of settings and operation for master mode transmission.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SICDAR, SITCR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit data, slot position for control data, control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	
3	Start SIOF_SCK output		Output serial clock
4	Set the FSE and TXE bits in SICTR to 1	Set the start for frame synchronous signal output and enable transmission	Output frame synchronous signal and issue transmit transfer request*
5	TDREQ = 1? No / Yes		
6	Set SITDR	Set transmit data	
7	Transmit SITDR from SIOF_TXD synchronously with SIOF_SYNC		Transmit
8	Transfer ended? No / Yes Clear the TXE bit in SICTR to 0 End	Set to disable transmission	End transmission

Note: * When interrupts due to transmit data underflow are enabled, after setting the no. 6 transmit data, the TXE bit should be set to 1.

Figure 29.9 Example of Transmit Operation in Master Mode

(2) Reception in Master Mode

Figure 29.10 shows an example of settings and operation for master mode reception.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for receive data, slot position for control data, and FIFO request threshold value	
2	Set the SCKE bit in SICTR to 1	Set operation start for baud rate generator	
3	Start SIOFSCK output		Output serial clock
4	Set the FSE and RXE bits in SICTR to 1	Set the start for frame synchronous signal output and enable reception	Output frame synchronous signal
5	Store SIOFRXD receive data in SIRDR synchronously with SIOF_SYNC		Issue receive transfer request according to the receive FIFO threshold value
6	RDREQ = 1? No / Yes		Reception
7	Read SIRDR	Read receive data	
8	Transfer ended? No / Yes Clear the RXE bit in SICTR to 0 End	Set to disable reception	End reception

Figure 29.10 Example of Receive Operation in Master Mode

(3) Transmission in Slave Mode

Figure 29.11 shows an example of settings and operation for slave mode transmission.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SITDAR, SICDAR, SITCR, and SIFCTR	Set operating mode, serial clock, slot positions for transmit data, slot position for control data, control data, and FIFO request threshold value	
2	Set the TXE bit in SICTR to 1	Set to enable transmission	Issue transmit transfer request to enable transmission when frame synchronous signal is input
3	TDREQ = 1? No Yes		
4	Set SITDR	Set transmit data	
5	Transmit SITDR from SIOFTXD synchronously with SIOF_SYNC		Transmit
6	Transfer ended? No Yes Clear the TXE bit in SICTR to 0 End	Set to disable transmission	End transmission

Figure 29.11 Example of Transmit Operation in Slave Mode

(4) Reception in Slave Mode

Figure 29.12 shows an example of settings and operation for slave mode reception.

No.	Flow Chart	SIOF Settings	SIOF Operation
1	Start Set SIMDR, SISCR, SIRDAR, SICDAR, and SIFCTR	Set operating mode, serial clock, slot positions for receive data, slot position for control data, and FIFO request threshold value	
2	Set the RXE bit in SICTR to 1	Set to enable reception	Enable reception when the frame synchronous signal is input
3	Store SIOFRXD receive data in SIRDR synchronously with SIOF_SYNC		Issue receive transfer request according to the receive FIFO threshold value
4	RDREQ = 1? No / Yes		Reception
5	Read SIRDR	Read receive data	
6	Transfer ended? No / Yes Clear the RXE bit in SICTR to 0 End	Set to disable reception	End reception

Figure 29.12 Example of Receive Operation in Slave Mode

(5) 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 29.13 shows the details of initialization upon transmit or receive reset.

Table 29.13 Transmit and Receive Reset

Type	Objects Initialized
Transmit reset	Stop transmitting form the SIOF_TXD (high level is outputted)
	Transmit FIFO write pointer
	TCRDY, TFEMP, and TDREQ bits in SISTR
	TXE bit in SICTR
Receive reset	Stop receiving form the SIOF_RXD
	Receive FIFO write pointer
	RCRDY, RFFUL, and RDREQ bits in SISTR
	RXE bit in SICTR

29.4.8 Interrupts

The SIOF has one type of interrupt.

(1) Interrupt Sources

Interrupts can be issued by several sources. Each source is shown as an SIOF status in SISTR. Table 29.14 lists the SIOF interrupt sources.

Table 29.14 SIOF Interrupt Sources

No.	Classification	Bit Name	Function Name	Description
1	Transmission	TDREQ	Transmit FIFO transfer request	The transmit FIFO stores data of specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data of specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register is ready to be written.
6		RCRDY	Receive control data ready	The receive control data register stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing has arrived while the transmit FIFO is empty.
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit FIFO is full.
9		RFOVF	Receive FIFO overflow	Serial data is received while the receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read while the receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is input before the specified bit number has been passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in both serial data and control data.

Whether an interrupt is issued or not as the result of an interrupt source is determined by the SIIER settings. If an interrupt source is set to 1 and the corresponding bit in SIIER is set to 1, an SIOF interrupt is issued.

(2) Regarding Interrupt Source

The transmit sources and receive sources are signals indicating the SIOF state; after being set, if the state changes, they are automatically cleared by the SIOF.

When the DMA transfer is used, a DMA transfer request of the FIFO is disabled for one cycle at the end of that DMA transfer.

(3) Processing when Errors Occur

On occurrence of each of the errors indicated as a status in SISTR, the SIOF performs the following operations.

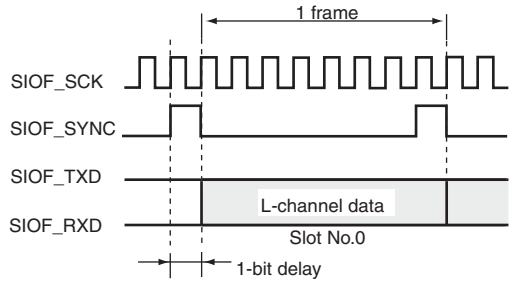
- Transmit FIFO underflow (TFUDF)
  The immediately preceding transmit data is again transmitted.
- Transmit FIFO overflow (TFOVF)
  The contents of the transmit FIFO are protected, and the write operation causing the overflow is ignored.
- Receive FIFO overflow (RFOVF)
  Data causing the overflow is discarded and lost.
- Receive FIFO underflow (RFUDF)
  An undefined value is output on the bus.
- FS error (FSERR)
  The internal counter is reset according to the signal in which an error occurs.
- Assign error (SAERR)
  — If the same slot is assigned to both serial data and control data, the slot is assigned to serial data.
  — If the same slot is assigned to two control data items, data cannot be transferred correctly.

29.4.9 Transmit and Receive Timing

Examples of the SIOF serial transmission and reception are shown in figure 29.13 to figure 29.19.

(1) 8-bit Monaural Data (Case 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, REDG = 0, FL[3:0] = 0000 (frame length: 8 bits)
TDLE = 1, TDLA[3:0] = 0000, TDRE = 0, TDRA[3:0] = 0000,
RDLE = 1, RDLA[3:0] = 0000, RDRE = 0, RDRA[3:0] = 0000,
CD0E = 0, CD0A[3:0] = 0000, CD1E = 0, CD1A[3:0] = 0000

Figure 29.13 Transmit and Receive Timing (8-Bit Monaural Data (1))

(2) 8-bit Monaural Data (Case 2)

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 16 bits

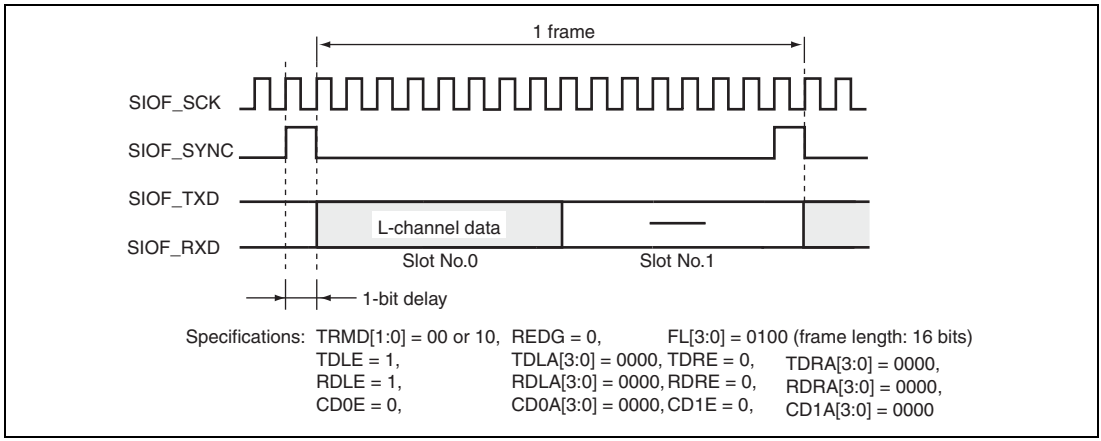


Figure 29.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

(3) 16-bit Monaural Data

Synchronous pulse method, falling edge sampling, slot No.0 used for transmit and receive data, and frame length = 64 bits

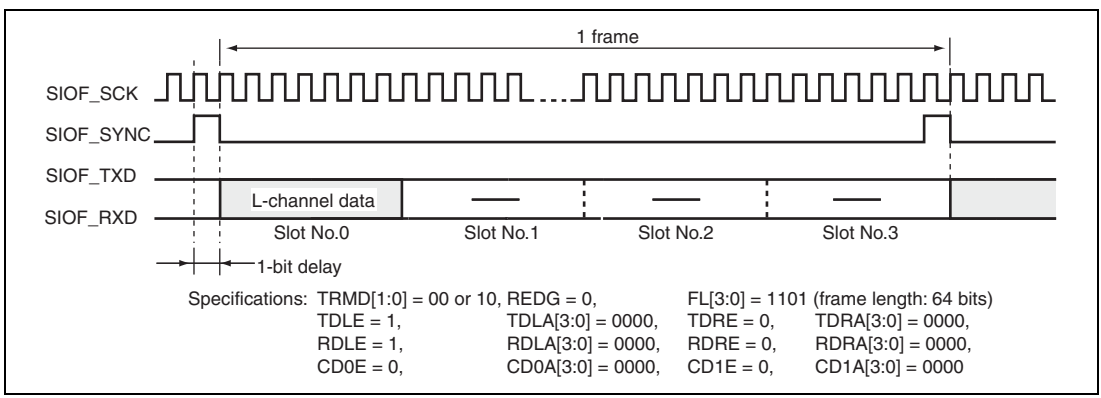


Figure 29.15 Transmit and Receive Timing (16-Bit Monaural Data)

(4) 16-bit Stereo Data (Case 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

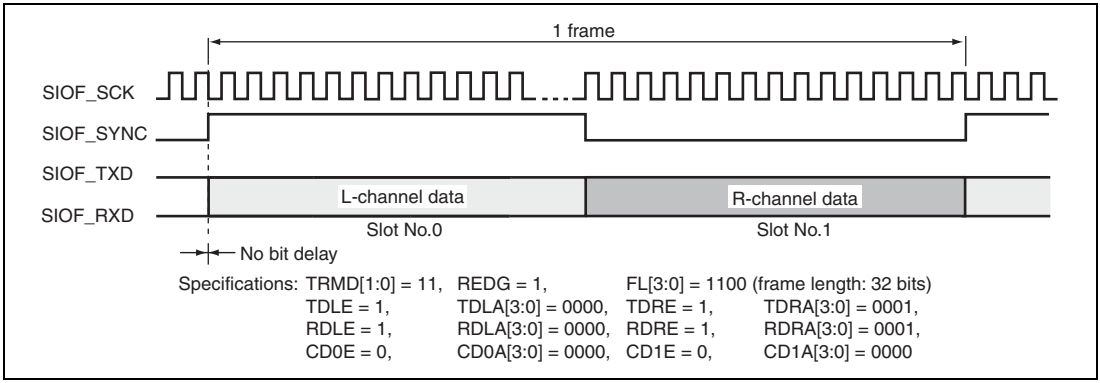


Figure 29.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

(5) 16-bit Stereo Data (Case 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

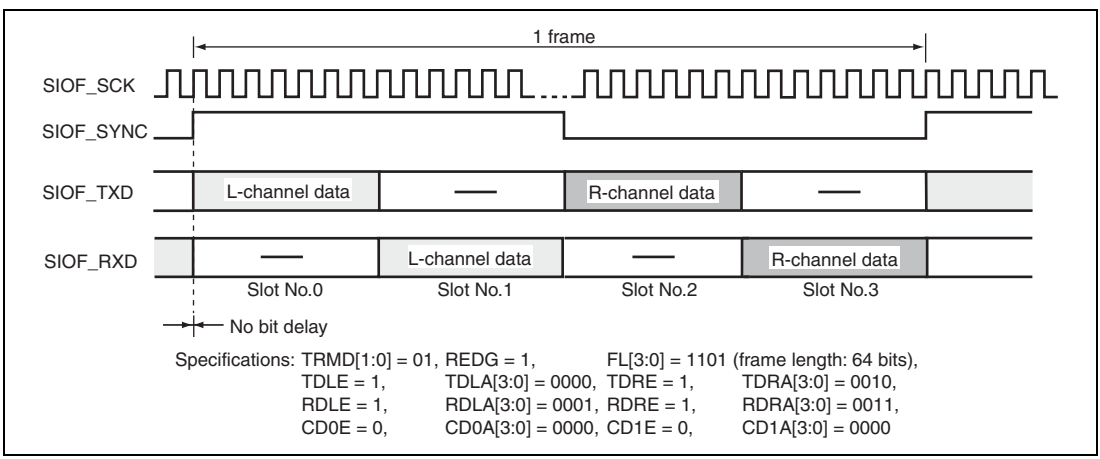


Figure 29.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

(6) 16-bit Stereo Data (Case 3)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.1 used for right-channel data, slot No.2 used for control data for channel 0, slot No.3 used for control data for channel 1, and frame length = 128 bits

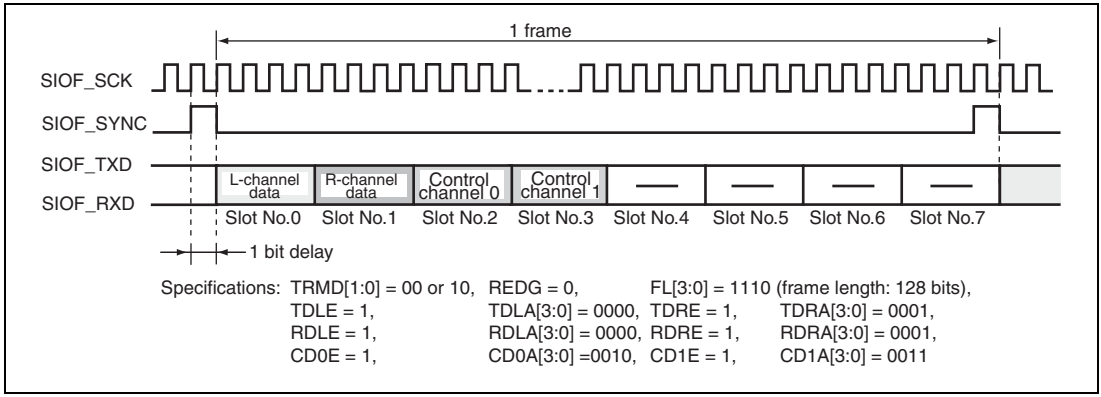


Figure 29.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

(7) 16-bit Stereo Data (Case 4)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.2 used for right-channel data, slot No.1 used for control data for channel 0 , slot No.3 used for control data for channel 1, and frame length = 128 bits

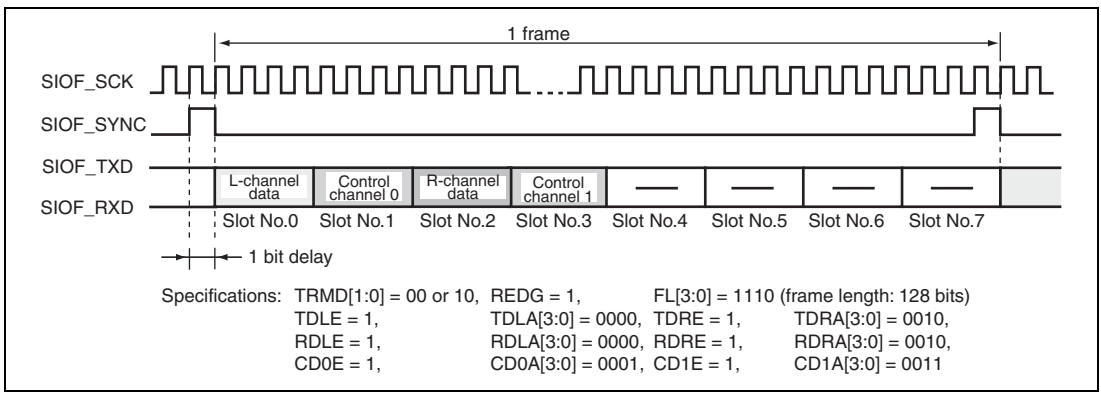


Figure 29.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

(8) Synchronization-Pulse Output Mode at End of Each Slot (SYNCAT Bit = 1)

Synchronous pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.1 used for right-channel data, slot No.2 used for control data for channel 0, slot No.3 used for control data for channel 1, and frame length = 128 bits

In this mode, valid data must be set to slot No. 0.

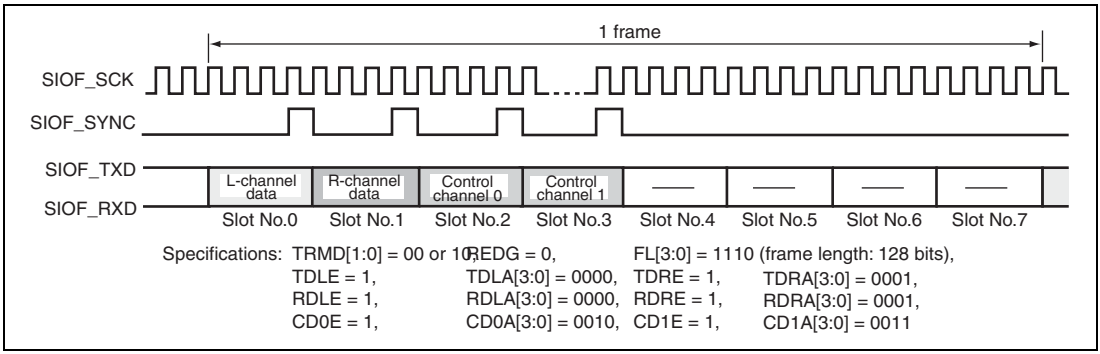


Figure 29.20 Transmit and Receive Timing (16-Bit Stereo Data)

Section 30 SIM Card Module (SIM)

The smart card interface supports IC cards (smart cards) conforming to the ISO/IEC 7816-3 (Identification Card) specification.

30.1 Features

- Communication functions
  Asynchronous half-duplex transmission
  Protocol selectable between T = 0 and T = 1 modes
  Data length: 8 bits
  Parity bit generation and check
  Selectable character protection addition time
  Selectable output clock cycles per etu
  Transmission of error signal (parity error) in receive mode when T = 0
  Detection of error signal and automatic character retransmission in transmit mode when T = 0
  Selectable minimum character interval of 11 etus (N = 255) when T = 1 (etu: Elementary Time Unit)
  Selectable direct convention/inverse convention
  Output clock can be fixed in high or low state
- Freely selectable bit rate by on-chip baud rate generator
- Four types of interrupt source
  Transmit data empty, receive data full, transmit/receive error, transmit complete
- DMA transfer
  Through DMA transfer requests for transmit data empty and receive data full, the direct memory access controller (DMAC) can be started and used for data transfer.
- The time waiting for the operation when T = 0, and the time waiting for a character when T = 1 can be observed.

Figure 30.1 shows a block diagram of the smart card interface.

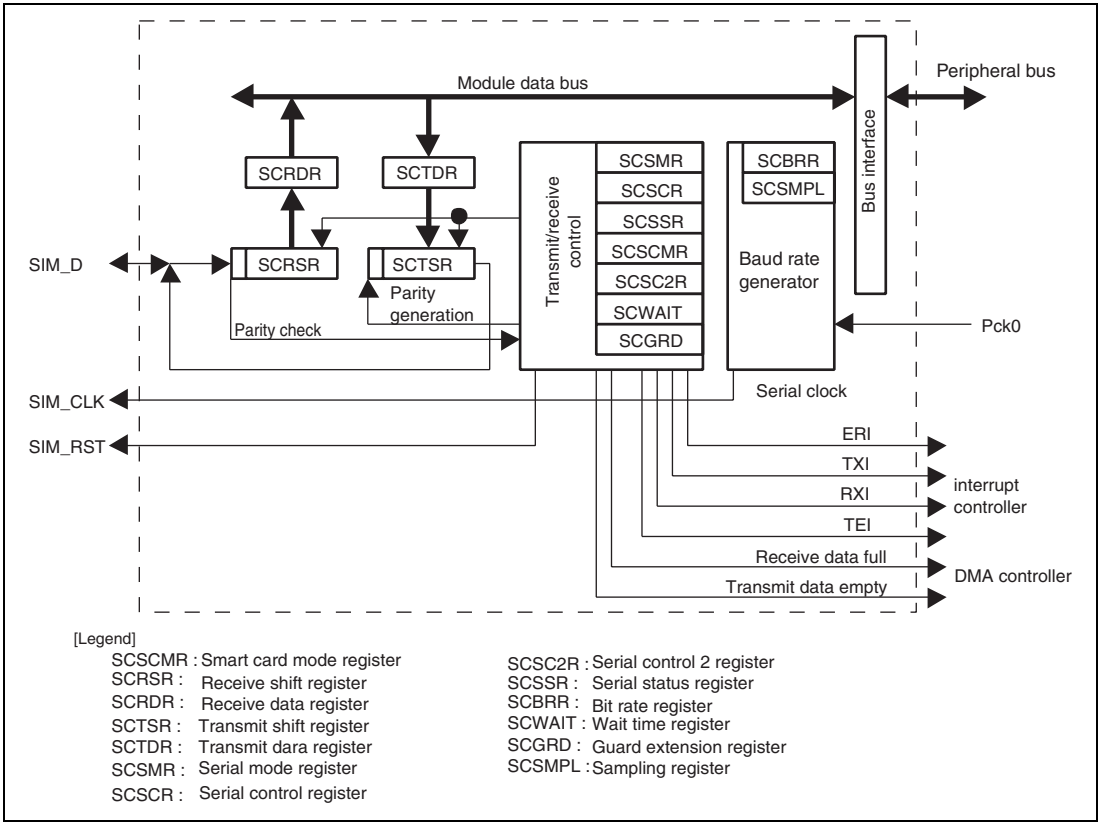


Figure 30.1 Smart Card Interface

30.2 Input/Output Pins

The pin configuration of the smart card interface is shown in table 30.1.

Table 30.1 Pin Configuration

Name	Abbreviation	I/O	Function
Transmit/receive data	SIM_D*	I/O	Transmit/receive data input/output
Clock output	SIM_CLK	Output	Clock output
Smart card reset	SIM_RST	Output	Smart card reset output

Note: * In explaining transmit and receive operations, the transmit data and receive data sides shall be referred to as TxD and RxD, respectively.

30.3 Register Descriptions

Table 30.2 shows the SIM register configuration. Table 30.3 shows the register state in each operating mode.

Table 30.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Serial mode register	SCSMR	R/W	H'FFE9 0000	H'1FE9 0000	8
Bit rate register	SCBRR	R/W	H'FFE9 0002	H'1FE9 0002	8
Serial control register	SCSCR	R/W	H'FFE9 0004	H'1FE9 0004	8
Transmit shift register	SCTSR	—	—	—	—
Transmit data register	SCTDR	R/W	H'FFE9 0006	H'1FE9 0006	8
Serial status register	SCSSR	R/W	H'FFE9 0008	H'1FE9 0008	8
Receive shift register	SCRSR	—	—	—	—
Receive data register	SCRDR	R	H'FFE9 000A	H'1FE9 000A	8
Smart card mode register	SCSCMR	R/W	H'FFE9 000C	H'1FE9 000C	8
Serial control 2 register	SCSC2R	R/W	H'FFE9 000E	H'1FE9 000E	8
Wait time register	SCWAIT	R/W	H'FFE9 0010	H'1FE9 0010	16
Guard extension register	SCGRD	R	H'FFE9 0012	H'1FE9 0012	8
Sampling register	SCSMPL	R/W	H'FFE9 0014	H'1FE9 0014	16

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 30.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Serial mode register	SCSMR	H'20	H'20	Retained	Retained
Bit rate register	SCBRR	H'07	H'07	Retained	Retained
Serial control register	SCSCR	H'00	H'00	Retained	Retained
Transmit shift register	SCTSR	—	—	—	—
Transmit data register	SCTDR	H'FF	H'FF	Retained	Retained
Serial status register	SCSSR	H'84	H'84	Retained	Retained
Receive shift register	SCRSR	—	—	—	—
Receive data register	SCRDR	H'00	H'00	Retained	Retained
Smart card mode register	SCSCMR	H'01	H'01	Retained	Retained
Serial control 2 register	SCSC2R	H'00	H'00	Retained	Retained
Wait time register	SCWAIT	H'0000	H'0000	Retained	Retained
Guard extension register	SCGRD	H'00	H'00	Retained	Retained
Sampling register	SCSMPL	H'0173	H'0173	Retained	Retained

30.3.1 Serial Mode Register (SCSMR)

SCSMR is an 8-bit readable/writable register that selects settings for the communication format of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	—	—	PE	O/$\overline{E}$	—	—	—	—
Initial value:	0	0	1	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PE	1	R	Parity Enable This bit is always read as 1. The write value should always be 1.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects whether even or odd parity is to be used when adding a parity bit and checking parity. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. When set to even parity, during transmission a parity bit is added such that the sum of 1 bits in the parity bit and transmit characters is even. During reception, a check is performed to ensure that the sum of 1 bits in the parity bit and the receive characters is even. 2. When set to odd parity, during transmission a parity bit is added such that the sum of 1 bits in the parity bit and transmit characters is odd. During reception, a check is performed to ensure that the sum of 1 bits in the parity bit and the receive characters is odd.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

30.3.2 Bit Rate Register (SCBRR)

SCBRR is an 8-bit readable/writable register that sets the transmit/receive bit rate.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	BRR[2:0]		
Initial value:	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	BRR2	1	R/W	Set the transmit/receive bit rate 2 to 0.
1	BRR1	1	R/W	
0	BRR0	1	R/W	

The SCBRR setting can be determined from the following formula.

$$\text{sck_frequency} = \frac{\text{Pck0}}{2\ (\text{BRR} + 1)}$$

The units of Pck0 (peripheral clock 0 frequency) and sck_frequency are MHz.

30.3.3 Serial Control Register (SCSCR)

SCSCR is an 8-bit readable/writable register that selects transmit or receive operation, the serial clock output, and whether to enable or disable interrupt requests for the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When serial transmit data is transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR), and the TDRE flag in the serial status register (SCSSR) is set to 1, transmit data empty interrupt (TXI) requests are enabled/disabled. 0: Disables transmit data empty interrupt (TXI) requests* 1: Enables transmit data empty interrupt (TXI) requests Note: * A TXI can be canceled either by clearing the TDRE flag, or by clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When serial receive data is transferred from the receive shift register (SCRSR) to the receive data register (SCRDR), and the RDRF flag in SCSSR is set to 1, receive data full interrupt (RXI) requests, and transmit/receive error interrupt (ERI) requests due to parity errors, overrun errors, and error signal status are enabled/disabled. 0: Disables receive data full interrupt (RXI) requests and transmit/receive error interrupt (ERI) requests*[1]*[2] 1: Enables receive data full interrupt (RXI) requests and transmit/receive error interrupt (ERI) requests*[2] Notes: 1. RXI and ERI interrupt requests can be canceled either by clearing the RDRF, PER, ORER or ERS flag, or by clearing the RIE bit to 0. 2. Wait error interrupt (ERI) requests are enabled or disabled by using the WAIT_IE bit in SCSCR.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable Enables/disables serial transmit operations. 0: Disables transmission*1 1: Enables transmission*2*3 Notes: 1. The TDRE flag in SCSSR is fixed to 1. 2. In this state, if transmit data is written to SCTDR, the transmit operation is initiated. Before setting the TE bit to 1, the serial mode register (SCSMR) and smart card mode register (SCSCMR) must always be set, to determine the transmit format. 3. Even if the TE bit is cleared to 0, the ERS flag is unaffected, and the previous state is retained.
4	RE	0	R/W	Receive Enable Enables/disables serial receive operations. 0: Disables reception*1 1: Enables reception*2 Notes: 1. Clearing the RE bit to 0 has no effect on the RDRF, PER, ERS, ORER, or WAIT_ER flag, and the previous state is retained. 2. If the start bit is detected in this state, serial reception is initiated. Before setting the RE bit to 1, SCSMR and SCSCMR must always be set, to determine the receive format.
3	WAIT_IE	0	R/W	Wait Enable Enables/disables wait error interrupt requests. 0: Disables wait error interrupt (ERI) requests 1: Enables wait error interrupt (ERI) requests
2	TEIE	0	R/W	Transmit End Interrupt Enable When transmission ends and the TEND flag is set to 1, transmit end interrupt (TEI) requests are enabled/disabled. 0: Disables transmit end interrupt (TEI) requests* 1: Enables transmit end interrupt (TEI) requests* Note: * A TEI can be canceled either by writing transmit data to SCTDR and clearing the TEND bit, or by clearing the TEIE bit to 0 after the TDRE flag in SCSSR is read as 1.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	CKE[1:0]	00	R/W	Clock Enable Select the clock source for the smart card interface, and enable/disable clock output from the SIM_CLK pin. 00: Fix the output pin at low 01: Clock output as the output pin 10: Fix the output pin at high 11: Clock output as the output pin

30.3.4 Transmit Shift Register (SCTSR)

SCTSR is a shift register that transmits serial data.

The smart card interface transfers transmit data from the transmit data register (SCTDR) to SCTSR, and then sends the data in order from the LSB or MSB to the SIM_TXD pin to perform serial data transmission.

When data transmission of one byte is completed, transmit data is automatically transferred from SCTDR to SCTSR, and transmission is initiated. When the TDRE flag in the serial status register (SCSSR) is set to 1, no data is transferred from SCTDR to SCTSR.

Direct reading and writing of SCTSR from the CPU or DMAC is not possible.

30.3.5 Transmit Data Register (SCTDR)

SCTDR is an 8-bit readable/writable register that stores data for serial transmission.

When the smart card interface detects a vacancy in the transmit shift register (SCTSR), transmit data written to SCTDR is transferred to SCTSR, and serial transmission is initiated. During SCTSR serial data transmission, if the next transmit data is written to SCTDR, continuous serial transmission is possible.

Bit:	7	6	5	4	3	2	1	0
	SCTD[7:0]							
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCTD[7:0]	All 1	R/W	Transmit Data Store data for serial transmission.

30.3.6 Serial Status Register (SCSSR)

SCSSR is an 8-bit readable/writable register that indicates the operating state of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	–
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Indicates that data was transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR), and that the next serial transmit data can be written to SCTDR. 0: Indicates that valid transmit data is written to SCTDR [Clearing conditions] • When the TE bit in CCSCR is 1, and data is written to SCTDR • When 0 is written to the TDRE bit 1: Indicates that there is no valid transmit data in SCTDR [Setting conditions] • On reset • When the TE bit in SCSCR is 0 • When data is transferred from SCTDR to SCTSR, and data can be written to SCTDR

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/W	Receive Data Register Full Indicates that received data is stored in the receive data register (SCRDR). 0: Indicates that no valid received data is stored in SCRDR [Clearing conditions] • On reset • When data is read from SCRDR • When 0 is written to RDRF 1: Indicates that valid received data is stored in SCRDR [Setting condition] When serial reception is completed normally, and received data is transferred from SCRSR to SCRDR. Note: In T = 0 mode, when a parity error is detected during reception, the SCRDR contents and RDRF flag are unaffected, and the previous state is retained. On the other hand, in T = 1 mode, when a parity error is detected during reception, the received data is transferred to SCRDR, and the RDRF flag is set to 1. In both T = 0 and T = 1 modes, even if the RE bit in the serial control register (SCSCR) is cleared to 0, the SCRDR contents and RDRF flag are unaffected, and the previous state is retained.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/W	Overrun Error Indicates that an overrun error occurred during reception, resulting in abnormal termination. 0: Indicates that reception is in progress, or that reception was completed normally*1 [Clearing conditions] • On reset • When 0 is written to the ORER bit 1: Indicates that an overrun error occurred during reception*2 [Setting condition] When the RDRF bit is set to 1 and the next serial reception is completed. Notes: 1. When the RE bit in SCSCR is cleared to 0, the ORER flag is unaffected and the previous state is retained. 2. In SCRDR, the received data before the overrun error occurred is lost, and the data that had been received at the time when the overrun error occurred is retained. Further, with the ORER bit set to 1, subsequent serial reception cannot be continued.
4	ERS	0	R/W	Error Signal Status Indicates the status of error signals returned from the receive side during transmission. In T = 1 mode, this flag is not set. 0: Indicates that an error signal indicating detection of a parity error was not sent from the receive side [Clearing conditions] • On reset • When 0 is written to the ERS bit 1: Indicates that an error signal indicating detection of a parity error was sent from the receive side [Setting condition] When an error signal is sampled. Note: Even if the TE bit in SCSCR is cleared to 0, the ERS flag is unaffected, and the previous state is retained.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/W	Parity Error Indicates that a parity error has occurred during reception, resulting in abnormal termination. 0: Indicates that reception is in progress, or that reception was completed normally*1 [Clearing conditions] • On reset • When 0 is written to the PER bit 1: Indicates that a parity error occurred during reception*2 [Setting condition] When the sum of 1 bits in the received data and parity bit does not match the even or odd parity specified by the O/E bit in the serial mode register (SCSMR). Notes: 1. When the RE bit in SCSCR is cleared to 0, the PER flag is unaffected, and the previous state is retained. 2. In T = 0 mode, the data received when a parity error occurs is not transferred to SCRDR, and the RDRF flag is not set. On the other hand, in T = 1 mode, the data received when a parity error occurs is transferred to SCRDR, and the RDRF flag is set. When a parity error occurs, the PER flag should be cleared to 0 before the sampling timing for the next parity bit.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End Indicates that transmission is ended. The TEND flag is read-only, and cannot be written. 0: Indicates that transmission is in progress [Clearing condition] When transmit data is transferred from SCTDR to SCTSR, and serial transmission is initiated. 1: Indicates that transmission is ended [Setting conditions] • On reset • When the ERS flag is 0 (normal transmission) after one byte of serial character and a parity bit are transmitted Note: The TEND flag is set 1 etu before the end of the character protection time.

Bit	Bit Name	Initial Value	R/W	Description
1	WAIT_ER	0	R/W	Wait Error Indicates the wait timer error status. 0: Indicates that the interval between the start of two successive characters has not exceeded the etu set by SCWAIT. [Clearing conditions] • On reset • When 0 is written to the WAIT_ER flag 1: Indicates that the interval between the start of two successive characters has exceeded the etu set by SCWAIT. [Setting conditions] • In T = 0 mode, when the interval between the start of a character to be received and immediately preceding transmitted or received character exceeds the (value of 60 × SCWAIT: Operation wait time) etu. • In T = 1 mode, when the interval between the start of two successive received characters exceeds the (SCWAIT value: Character protection time) etu. Notes: 1. Even if the RE bit in SCSCR is cleared to 0, the WAIT_ER flag is unaffected, and the previous state is retained. 2. In T = 0 mode, even if the setting condition for the WAIT_ER flag is satisfied when the RE bit is set to 1, the WAIT_ER flag may not be set to 1. In this case, the RE bit has been set to 1, then the WAIT_ER flag is set to 1 after 60 × (SCWAIT + n) etu (n ≥ 0: depending on the timing for setting the RE bit to 1) since the last transmission or reception. 3. In T = 0 mode, if the WAIT_ER flag does not need to be set to 1 after 60 × (SCWAIT + n) etu since the last transmission or reception, the mode should be changed from T = 0 to T = 1, and changed to T = 0 again by the PB bit in SCSCMR. In T = 1 mode, if the WAIT_ER flag does not need to be set to 1 after (SCWAIT) etu since the last reception, the mode should be changed from T = 1 to T = 0, and changed to T = 1 again by the PB bit in SCSCMR.

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.

30.3.7 Receive Shift Register (SCRSR)

SCRSR is a register that receives serial data.

The smart card interface receives serial data input from the SIM_RXD pin in order, from the LSB or MSB, and sets it in SCRSR, converting it to parallel data. When reception of one byte of data is completed, the data is automatically transferred to SCRDR. The CPU or DMAC cannot directly read from or write to SCRSR.

30.3.8 Receive Data Register (SCRDR)

SCRDR is an 8-bit read-only register that stores received serial data.

When reception of one byte of serial data is completed, the smart card interface transfers the received serial data from the receive shift register (SCRSR) to SCRDR for storage, and completes the receive operation. Thereafter, SCRSR can receive data. In this way, SCRSR and SCRDR constitute a double buffer, enabling continuous reception of data. SCRDR cannot be written to by the CPU or DMAC.

Bit:	7	6	5	4	3	2	1	0
	SCRD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCRD[7:0]	All 0	R	Receive Data Store received serial data.

30.3.9 Smart Card Mode Register (SCSCMR)

SCSCMR is an 8-bit readable/writable register that selects functions of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	—	LCB	PB	—	SDIR	SINV	RST	SMIF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
6	LCB	0	R/W	Last Character When this bit is set to 1, the character protection time is 2 etus, and the setting of the guard extension register is invalid. 0: The character protection time is determined by the value of the guard extension register. 1: The character protection time is 2 etus.
5	PB	0	R/W	Protocol Selects the T = 0 or T = 1 protocol. 0: The smart card interface operates according to the T = 0 protocol. 1: The smart card interface operates according to the T = 1 protocol.
4	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the format for serial/parallel conversion. 0: Transmits the SCTDR contents in LSB-first. Received data is stored in SCRDR as LSB-first. 1: Transmits the SCTDR contents in MSB-first. Received data is stored in SCRDR as MSB-first.

Bit	Bit Name	Initial Value	R/W	Description
2	SINV	0	R/W	Smart Card Data Inversion Specifies inversion of the data logic level. In combination with the function of bit 3, used for transmission to or reception from the inverse convention card. The SINV bit does not affect the parity bit. 0: Transmits the SCTDR contents without change. Stores received data in SCRDR without change. 1: Inverts the SCTDR contents and transmits it. Inverts received data and stores it in SCRDR.
1	RST	0	R/W	Smart Card Reset Controls the output of the SIM_RST pin of the smart card interface. 0: The SIM_RST pin of the smart card interface outputs low level. 1: The SIM_RST pin of the smart card interface outputs high level.
0	SMIF	1	R/W	Smart Card Interface Mode Select This bit is always read as 1. The write value should always be 1.

30.3.10 Serial Control 2 Register (SCSC2R)

SCSC2R is an 8-bit readable/writable register that enables or disables receive data full interrupt (RXI) requests.

Bit:	7	6	5	4	3	2	1	0
	EIO	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	EIO	0	R/W	Error Interrupt Only When the EIO bit is 1, even if the RIE bit is set to 1, a receive data full interrupt (RXI) request is not sent to the CPU. When the DMAC is used with this setting, the CPU processes only ERI requests. Receive data full interrupt (RXI) requests are determined by the RIE bit setting.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

30.3.11 Guard Extension Register (SCGRD)

SCGRD is an 8-bit readable/writable register that sets the time added for character protection.

Bit:	7	6	5	4	3	2	1	0
	SCGRD[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SCGRD [7:0]	All 0	R/W	Guard Extension Indicate the time added for character protection after transmitting a character to the smart card. The interval between the start of two successive characters is 12 etus (no addition) when the value of this register is H'00, is 13 etus when the value is H'01, and so on, up to 266 etus for H'FE. If the value of this register is H'FF, the interval between the start of two successive characters is 11 etus in T = 1 mode and is 12 etus in T = 0 mode.

30.3.12 Wait Time Register (SCWAIT)

SCWAIT is a 16-bit readable/writable register. If the interval between the start of two successive characters exceeds the set value (in etu units), a wait time error is generated.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	SCWAIT [15:0]	All 0	R/W	Wait Time Register • T = 0 In this mode, the operation wait time can be set in this register. If the interval between the start of characters to be received and transmitted or received characters immediately before exceeds the (60 × the value set in this register) etu, the WAIT_ER flag is set to 1. However, if SCWAIT is set to H'0000, the WAIT_ER flag is set after 60 etus. • T = 1 In this mode, the character wait time can be set in this register. If the interval between the start of two successive received characters exceeds the (the value set in this register) etu, the WAIT_ER flag is set to 1. However, if SCWAIT is set to H'0000, the WAIT_ER flag is set after 1 etu.

30.3.13 Sampling Register (SCSMPL)

SCSMPL is a 16-bit readable/writable register that sets the number of serial clock cycles per etu.

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

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 to 0	SCSMPL [10:0]	H'173	R/W	Setting for Number of Serial Clock Cycles per Etu The number of serial clock cycles per etu is (SCSMPL value + 1). The value written to SCSMPL should always be H'0007 or greater.

30.4 Operation

30.4.1 Overview

The main functions of the smart card interface are as follows.

- One frame consists of 8-bit data and one parity bit.
- During transmission, a character protection time, set using SCGRD and the LCB and PB bits in SCSCMR, is inserted between the end of each parity bit and the beginning of the next frame.
- During reception in T = 0 mode, when a parity error is detected, low level is output for a duration of 1 etu as an error signal, 10.5 etus after the start bit.
- During transmission in T = 0 mode, if an error signal is sampled, after 2 etus or more have elapsed, the same data is automatically transmitted.
- Only asynchronous communication functions are supported; there is no clocked synchronous communication function.

30.4.2 Data Format

Figure 30.2 shows the data format used by the smart card interface. The smart card interface performs a parity check for each frame during reception.

During reception in T = 0 mode, if a parity error is detected, an error signal is returned to the transmit side, requesting data retransmission. When the transmit side samples the error signal, it retransmits the same data.

During reception in T = 1 mode, if a parity error is detected, an error signal is not returned. During transmission, error signals are not sampled and data is not retransmitted.

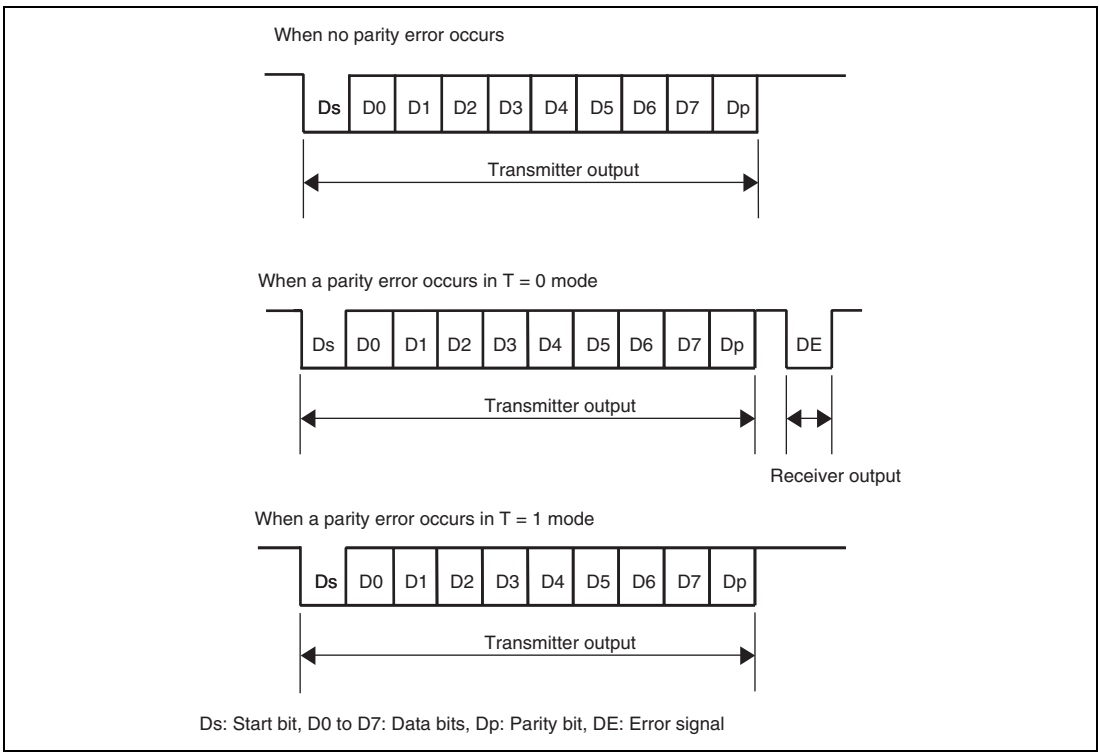


Figure 30.2 Data Format Used by Smart Card Interface

The operation sequence is as follows.

1. When not in use, the data line is in a high-impedance state and fixed at high level by a pull-up resistance.

2. The transmit side initiates transmission of one frame of data. The data frame begins with the start bit (Ds: low level). This is followed by eight data bits (D0 to D7) and the parity bit (Dp).
3. The smart card interface then returns the data line to high impedance. The data line is held at high level by the pull-up resistance.
4. The receive side performs a parity check.

   If there is no parity error and reception is normal, reception of the next frame is awaited, without further action.

   On the other hand, when a parity error has occurred in T = 0 mode, an error signal (DE: low level) is output, requesting data retransmission. After output of an error signal with the specified duration, the receive side again sets the signal line to the high-impedance state. The signal line returns to high level by means of the pull-up resistance. If in T = 1 mode, however, no error signal is output even if a parity error occurs.
5. If the transmit side does not receive an error signal, the next frame is transmitted.

   On the other hand, if in T = 0 mode and an error signal is received, the data for which the error occurred is retransmitted as in step 2 above. In T = 1 mode, however, error signals are not received and retransmission is not performed.

30.4.3 Register Settings

Table 30.4 shows a map of the bits in the registers used by the smart card interface.

Bits for which 0 or 1 is shown must always be set to the value shown. The method for setting the bits other than these is explained below.

Table 30.4 Register Settings for Smart Card Interface

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SCSMR	0	0	PE	O/$\overline{E}$	0	0	0	0
SCBRR	0	0	0	0	0	BRR2	BRR1	BRR0
SCSCR	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0
SCTDR	SCTD7	SCTD6	SCTD5	SCTD4	SCTD3	SCTD2	SCTD1	SCTD0
SCSSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	0
SCRDR	SCRD7	SCRD6	SCRD5	SCRD4	SCRD3	SCRD2	SCRD1	SCRD0
SCSCMR	0	LCB	PB	0	SDIR	SINV	RST	1
SCSC2R	EIO	0	0	0	0	0	0	0
SCWAIT	SCWAIT15 to SCWAIT0							
SCGRD	SCGRD7 to SCGRD0							
SCSMPL	SCSMPL10 to SCSMPL0, bits 11 to 15 are 0							

- Serial mode register (SCSMR) setting

  When the IC card is set for the direct convention, the O/$\overline{E}$ bit is cleared to 0; for the inverse convention, it is set to 1.
- Bit rate register (SCBRR) setting

  Sets the bit rate. For the method of computing settings, refer to section 30.4.4, Clocks.
- Serial control register (SCSCR) settings

  Each interrupt can be enabled and disabled using the TIE, RIE, TEIE, and WAIT_IE bits.

  By setting either the TE or RE bit to 1, transmission or reception is selected.

  The CKE[1] and CKE[0] bits are used to select the clock output state. For details, refer to section 30.4.4, Clocks.
- Smart card mode register (SCSCMR) settings

  When the IC card is set for the direct convention, both the SDIR and SINV bits are cleared to 0; for the inverse convention, both are set to 1. The SMIF bit is always set to 1.

  Figure 30.3 below shows the register settings and waveform examples at the start character for two types of IC cards (a direct-convention type and an inverse-convention type).

  For the direct-convention type, the logical level 1 is assigned to the Z state, and the logical level 0 to the A state, and transmission and reception are performed in LSB-first. The data of the above start character is then H'3B. Even parity is used according to the smart card specification, and so the parity bit is 1.

For the inverse-convention type, the logical level 1 is assigned to the A state, and the logical level 0 to the Z state, and transmission and reception are performed in MSB-first. The data of the start character shown in figure 30.3 is then H'3F. Even parity is used according to the smart card specification, and so the parity bit is 0 corresponding to the Z state.

In addition, the only D7 to D0 bits are inverted by the SINV bit. The O/$\overline{E}$ bit in SCSMR is set to odd parity mode to invert the parity bit. In transmission and reception, the setting condition is similar.

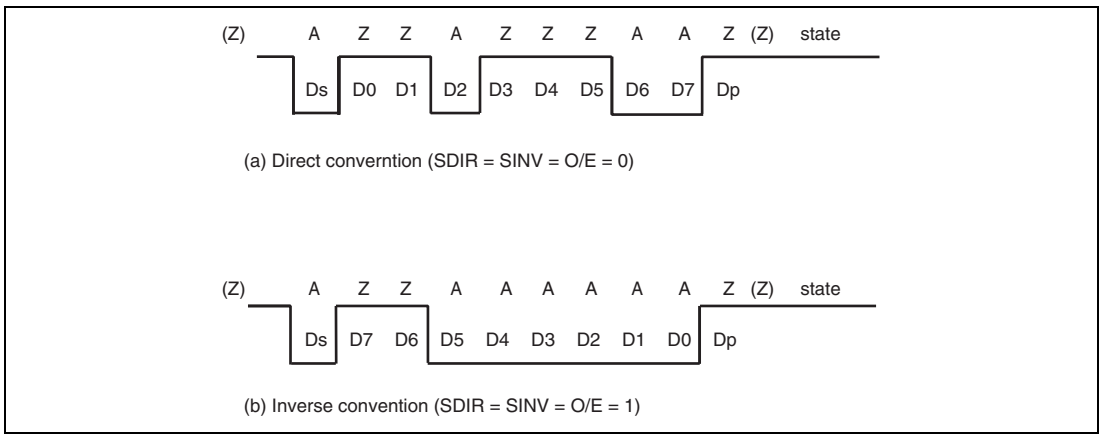


Figure 30.3 Examples of Start Character Waveforms

30.4.4 Clocks

Only the internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock in the smart card interface. The bit rate is set using the bit rate register (SCBRR) and the sampling register (SCSMPL), using the formula indicated below. Examples of bit rates are listed in table 30.5

Here, when the CKE0 bit is set to 1 and the clock output is selected, a clock signal is output from the SIM_CLK pin with frequency equal to (SCSMPL + 1) times the bit rate.

$$B = Pck0 \times 10^6 / \{(S+1) \times 2\,(N+1)\}$$

where

B = Bit rate (bits/s)

Pck0 = Peripheral clock0

S = SCSMPL setting ($0 \leq S \leq 2047$)

N = SCBRR setting ($0 \leq N \leq 7$).

Table 30.5 Example of Bit Rates (bits/s) for SCBRR Settings (Pck0 = 66.6 MHz, SCSMPL = 371)

SCBRR Setting	SCK Frequency (MHz)	Bit Rate (bits/s)
7	4.16	11190
6	4.76	12788
5	5.55	14919
4	6.66	17903

Note: The bit rate is a value that is rounded off below the decimal point.

30.4.5 Data Transmit/Receive Operation

(1) Initialization

Prior to data transmission and reception, the following procedure should be used to initialize the smart card interface. Initialization is also necessary when switching from transmit mode to receive mode, and when switching from receive mode to transmit mode. An example of the initialization process is shown in the flowchart of figure 30.4.

Step (1) to step (7) of figure 30.4 correspond to the following operation.

1. Clear the TE and RE bits in the serial control register (SCSCR) to 0.
2. Clear the error flags PER, ORER, ERS, and WAIT_ER in the serial status register (SCSSR) to 0.
3. Set the parity bit (O/$\overline{E}$ bit) in the serial mode register (SCSMR).
4. Set the LCB, PB, SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR).
5. Set the value corresponding to the bit rate to the bit rate register (SCBRR).
6. Set the clock source select bits (CKE[1] and CKE[0] bits) in the serial control register (SCSCR). At this time, the TIE, RIE, TE, RE, TEIE, and WAIT_IE bits should be cleared to 0. If the CKE[0] bit is set to 1, a clock signal is output from the SIM_CLK pin.
7. After waiting at least 1 etu, set the TIE, RIE, TE, RE, TEIE, and WAIT_IE bits in SCSCR. Except for self-check, the TE bit and RE bit should not be set simultaneously.

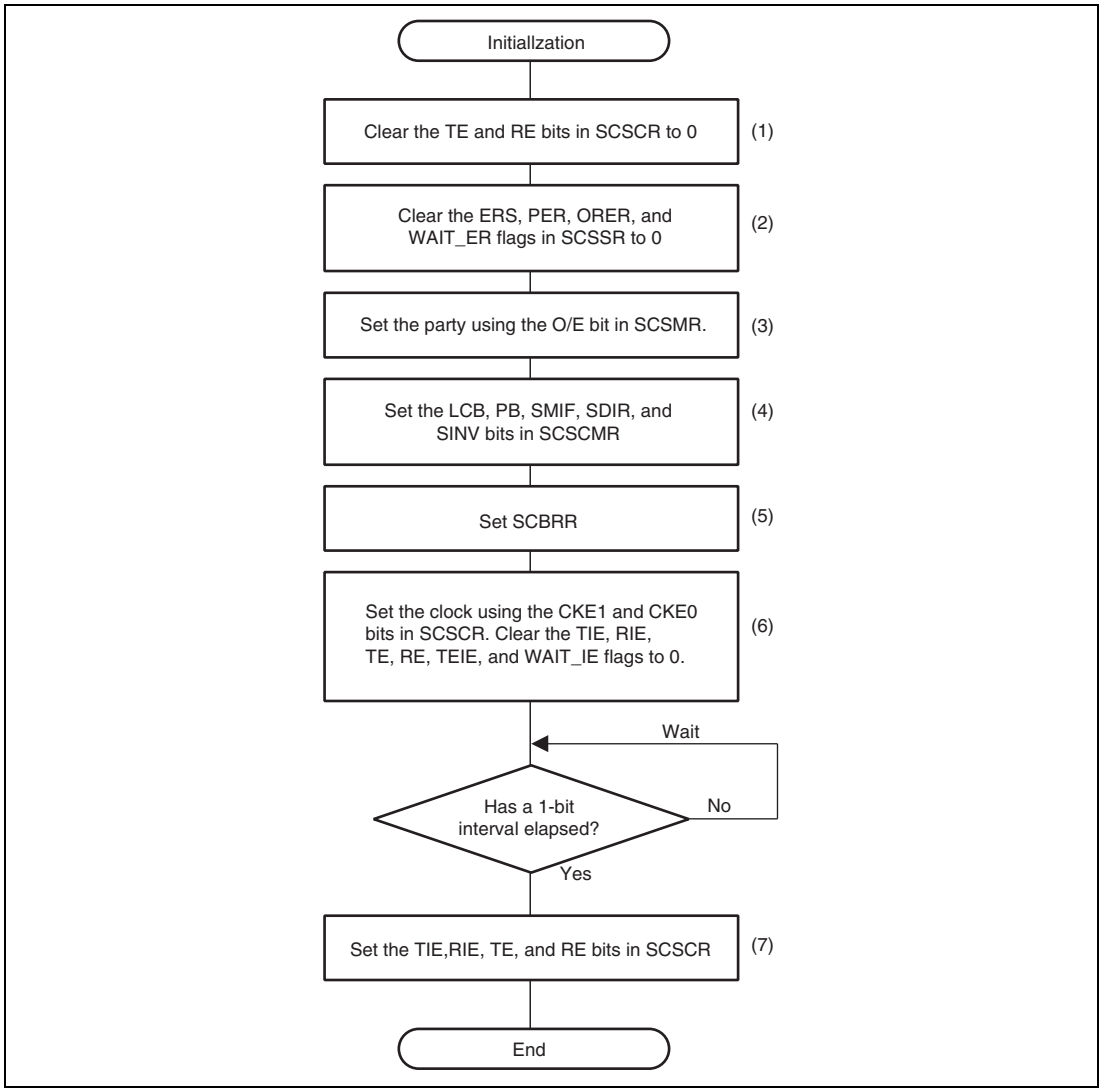


Figure 30.4 Example of Initialization Flow

(2) Serial Data Transmission

Data transmission in smart card mode includes error signal sampling and retransmit processing. An example of transmit processing is shown in figure 30.5.

Step (1) to step (6) of figure 30.5 correspond to the following operation.

1. Follow the initialization procedure above to initialize the smart card interface.
2. Confirm that the ERS bit (error flag) in SCSSR is cleared to 0.
3. Repeat steps (2) and (3) until it can be confirmed that the TDRE flag in SCSSR is set to 1.
4. Write transmit data to SCTDR, and perform transmission. At this time, the TDRE flag is automatically cleared to 0. When transmission of the start bit is started, the TEND flag is automatically cleared to 0, and the TDRE flag is automatically set to 1.
5. When performing continuous data transmission, return to step (2).
6. When transmission is ended, clear the TE bit to 0.

Interrupt processing can be performed in the above series of processing.

When the TIE bit is set to 1 to enable interrupt requests and if transmission is started and the TDRE flag is set to 1, a transmit data empty interrupt (TXI) request is issued. When the RIE bit is set to 1 to enable interrupt requests and if an error occurs during transmission and the ERS flag is set to 1, a transmit/receive error interrupt (ERI) request is issued.

For details, refer to Interrupt Operations in section 30.4.5, Data Transmit/Receive Operation.

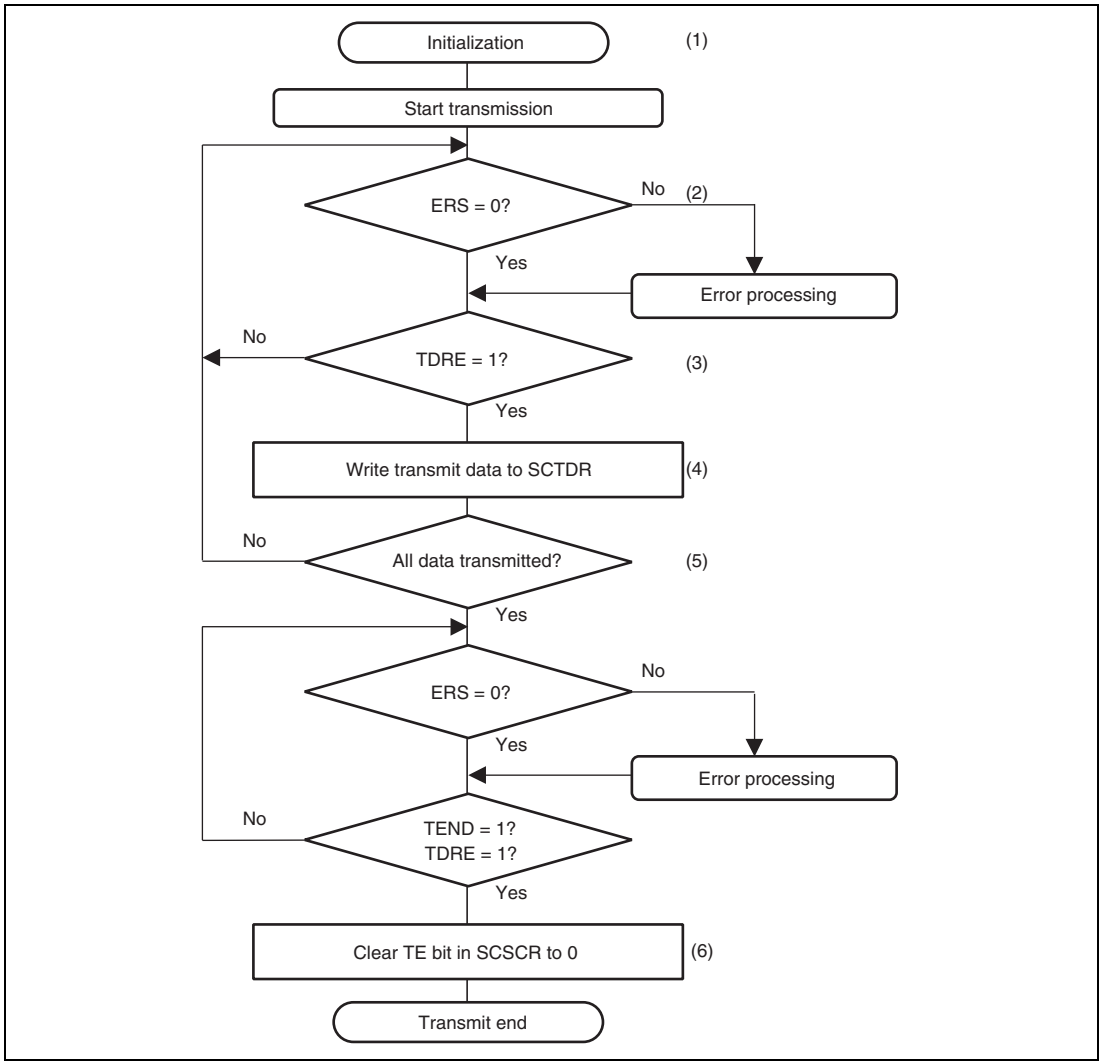


Figure 30.5 Example of Transmit Processing

(3) Serial Data Reception

An example of data receive processing in smart card mode is shown in figure 30.6.

Step (1) to step (6) of figure 30.6 correspond to the following operation.

1. Follow the initialization procedure above to initialize the smart card interface.

2. Confirm that the PER, ORER, and WAIT_ER flags in SCSSR are 0. If one of these flags is set, after performing the prescribed receive error processing, clear the PER, ORER, and WAIT_ER flags to 0.
3. Repeat steps (2) and (3) in the figure until it can be confirmed that the RDRF flag is set to 1.
4. Read received data from SCRDR.
5. When receiving data continuously, return to step (2).
6. When reception is ended, clear the RE bit to 0.

Interrupt processing can be performed in the above series of processing.

When the RIE bit is set to 1 and the EIO bit is cleared to 0 and if the RDRF flag is set to 1, a receive data full interrupt (RXI) request is issued. If the RIE bit is set to 1, an error occurs during reception, and either the ORER, PER, or WAIT_ER flag is set to 1, a transmit/receive error interrupt (ERI) request is issued.

For details, refer to, Interrupt Operations in section 30.4.5, Data Transmit/Receive Operation.

If a parity error occurs during reception and the PER flag is set to 1, in T = 0 mode the received data is not transferred to SCRDR, and so this data cannot be read. In T = 1 mode, received data is transferred to SCRDR, and so this data can be read.

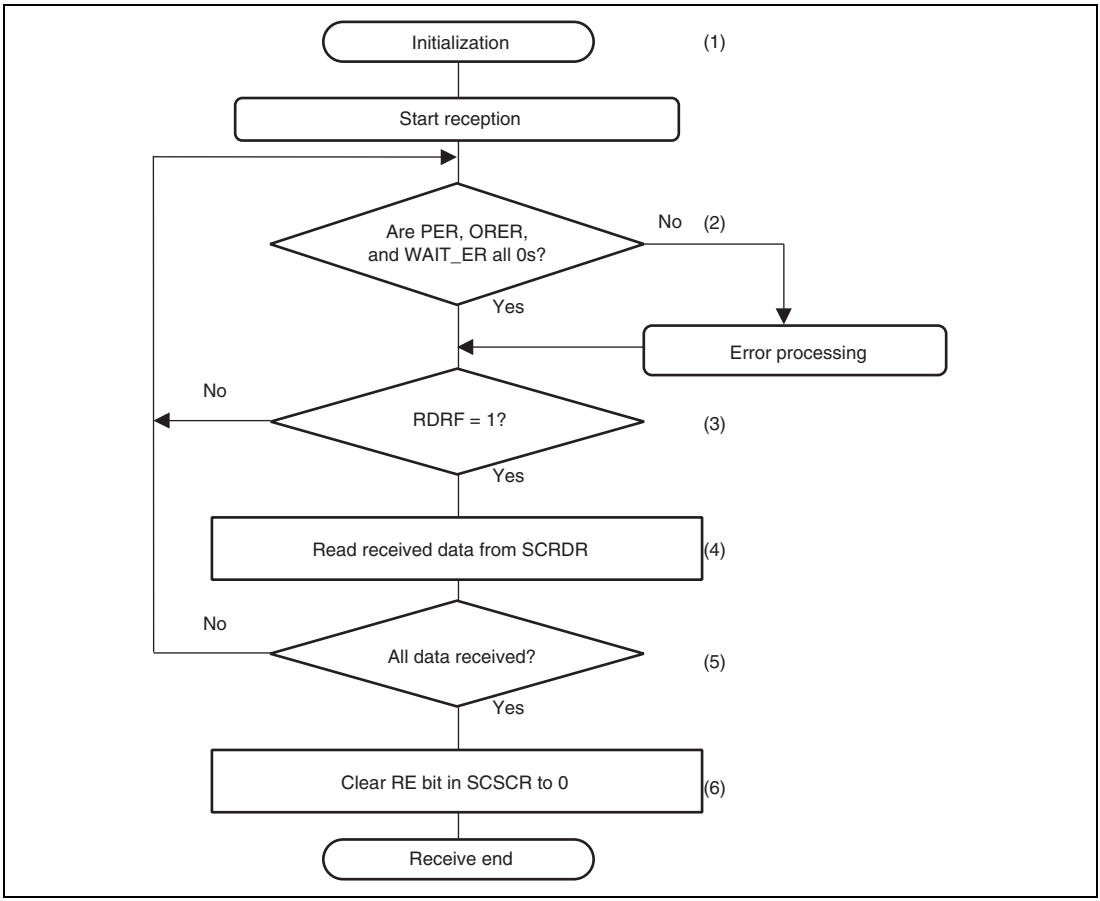


Figure 30.6 Example of Receive Processing

(4) Switching Modes

When switching from receive mode to transmit mode, after confirming that reception has been completed, start initialization, and then clear the RE bit to 0 and set the TE bit to 1. Completion of reception can be confirmed through the RDRF flag.

When switching from transmit mode to receive mode, after confirming that transmission has been completed, start initialization, and then clear the TE bit to 0 and set the RE bit to 1. Completion of transmission can be confirmed through the TDRE and TEND flags.

(5) Interrupt Operations

The smart card interface has four types of interrupt requests: transmit data empty interrupt (TXI) requests, transmit/receive error interrupt (ERI) requests, receive data full interrupt (RXI) requests, and transmit end interrupt (TEI) requests.

- When the TDRE flag in SCSSR is set to 1, a TXI request is issued.
- When the RDRF flag in SCSSR is set to 1, an RXI request is issued.
- When the ERS, ORER, PER, or WAIT_ER flag in SCSSR is set to 1, an ERI request is issued.
- When the TEND flag in SCSSR is set, a TEI request is issued.

Table 30.6 lists the interrupt sources for the smart card interface. Each of the interrupt requests can be enabled or disabled using the TIE, RIE, TEIE, and WAIT_IE bits in SCSCR and the EIO bit in SCSC2R. In addition, each interrupt request can be sent independently to the interrupt controller.

Table 30.6 Interrupt Sources of Smart Card Interface

Operating State		Flags	Mask Bits	Interrupt Sources
Transmit mode	Normal operation	TDRE	TIE	TXI
		TEND	TEIE	TEI
	Error	ERS	RIE	ERI
Receive mode	Normal operation	RDRF	RIE, EIO	RXI
	Error	ORER, PER	RIE	ERI
		WAIT_ER	WAIT_IE	ERI

(6) Data Transfer Using DMAC

The smart card interface enables reception and transmission using the DMAC.

In transmission, when the TDRE flag in SCSSR is set to 1, a DMA transfer request for transmit data empty is issued. If a DMA transfer request for transmit data empty is set in advance as a DMAC activation source, the DMAC can be activated and made to transfer data when a DMA transfer request for transmit data empty occurs.

When in T = 0 mode and if an error signal is received during transmission, the same data is automatically retransmitted. At the time of this retransmission, no DMA transfer request is issued, and so the number of bytes specified to the DMAC can be transmitted.

When using the DMAC for transmit data processing and performing error processing as a result of an interrupt request sent to the CPU, the TIE bit should be cleared to 0 so that no TXI requests are

generated, and the RIE bit should be set to 1 so that an ERI request is issued. The ERS flag set when an error signal is received is not cleared automatically, and so should be cleared by sending an interrupt request to the CPU.

In receive operation, when the RDRF flag in SCSSR is set to 1, a DMA transfer request for receive data full is issued. By setting a DMA transfer request for receive data full in advance as a DMAC activation source, the DMAC can be activated and made to transfer data when a DMA transfer request for receive data full occurs.

When in T = 0 mode and if a parity error occurs during reception, a data retransmit request is issued. At this time the RDRF flag is not set, and a DMA transfer request is not issued, so the number of bytes specified to the DMAC can be received.

When using the DMAC for receive data processing and performing error processing as a result of an interrupt request sent to the CPU, the RIE bit should be set to 1 and the EIO bit to 1, so that no RXI requests are generated and only ERI requests are generated.

The PER, ORER, and WAIT_ER flags that are set by a receive error are not automatically cleared, and so should be cleared by sending an interrupt request to the CPU.

When using the DMAC for transmission and reception, the DMAC should always be set first and put into the enabled state, before setting the smart card interface.

30.5 Usage Notes

The following matters should be noted when using the smart card interface.

(1) Receive Data Timing and Receive Margin

When SCSMPL holds its initial value, the smart card interface operates at a basic clock frequency 372 times the transfer rate.

During reception, the smart card interface samples the falling edge of the start bit using the serial clock for internal synchronization. Receive data is captured internally at the rising edge of the 186th serial clock pulse. This is shown in figure 30.7.

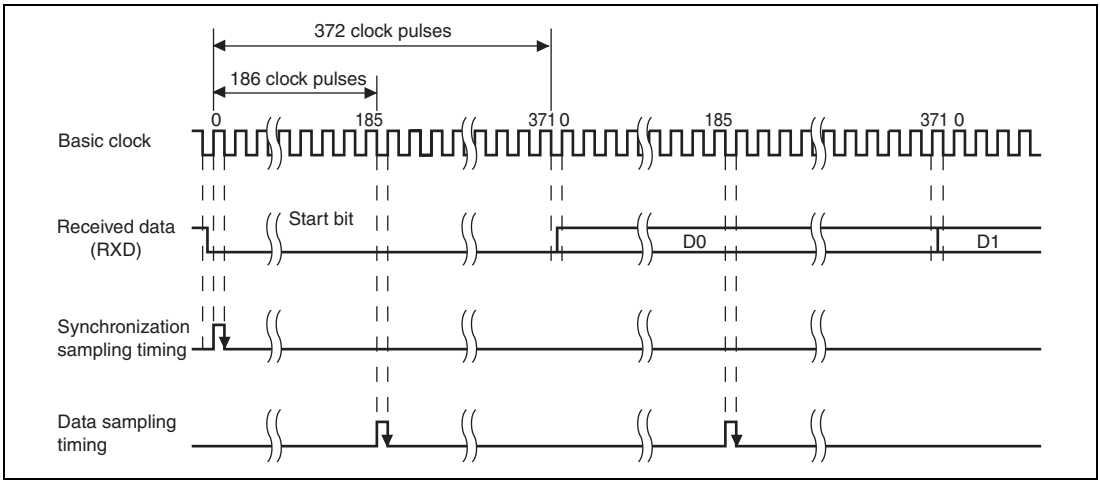


Figure 30.7 Receive Data Sampling Timing in Smart Card Mode

Hence the receive margin can be expressed as follows.

Formula for receive margin in smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (L + F) \right| \times 100\%$$

where

M: Receive margin (%)

N: Ratio of the bit rate to the clock (N = 372)

D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of the deviation of the clock frequency

In the above formula, if F = 0 and D = 0.5, then the receive margin is as follows.

When D = 0.5, F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

(2) Retransmit Operation

Retransmit operations when the smart card interface is in receive mode and in transmit mode are described below.

(a) Retransmission when the smart card interface is in receive mode (T = 0)

Figure 30.8 shows retransmit operations when the smart card interface is in receive mode. Step (1) to step (5) of figure 30.8 correspond to the following operation.

1. If an error is detected as a result of checking the received parity bit, the PER bit in SCSSR is automatically set to 1. At this time, if the RIE bit in SCSCR is set to enable, an ERI request is issued. The PER bit in SCSSR should be cleared to 0 before the sampling timing for the next parity bit.
2. The RDRF bit in SCSSR is not set for frames in which a parity error occurs.
3. If no error is detected as a result of checking the received parity bit, the PER bit in SCSSR is not set.
4. If no error is detected as a result of checking the received parity bit, it is assumed that reception was completed normally, and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is 1 and the EIO bit is 0, an RXI request is generated.
5. If a normal frame is received, the pin retains its high-impedance state at the timing for transmission of error signals.

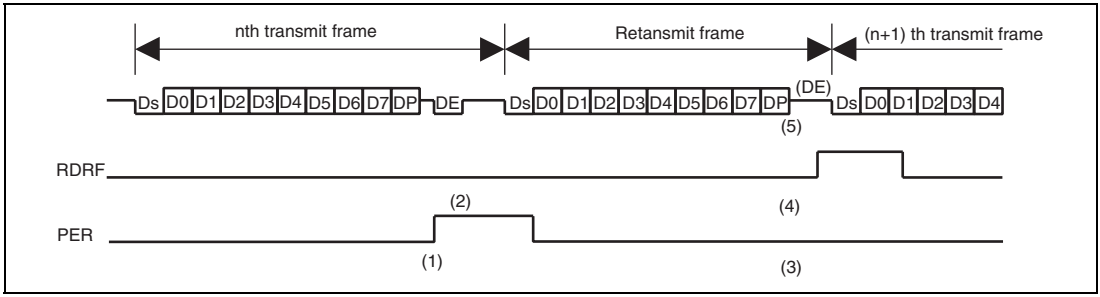


Figure 30.8 Retransmission when Smart Card Interface is in Receive Mode

(b) Retransmission when the smart card interface is in transmit mode (T = 0)

Figure 30.9 shows retransmit operations when the smart card interface is in transmit mode. Step (1) to step (4) of figure 30.9 correspond to the following operation

1. After completion of transmission of one frame, if an error signal is returned from the receive side, the ERS bit in SCSSR is set to 1. If the RIE bit in SCSCR is set to enable, an ERI request is issued. The ERS bit in SCSSR should be cleared to 0 before the sampling timing for the next parity bit.
2. In T = 0 mode, the TEND bit in SCSSR is not set for a frame when an error signal indicating an error is received.
3. If no error signal is returned from the receive side, the ERS bit in SCSSR is not set.
4. If no error signal is returned from the receive side, it is assumed that transmission of one frame, including retransmission, is completed, and the TEND bit in SCSSR is set to 1. At this time, if the TIE bit in SCSCR is set to enable, a TEI interrupt request is issued.

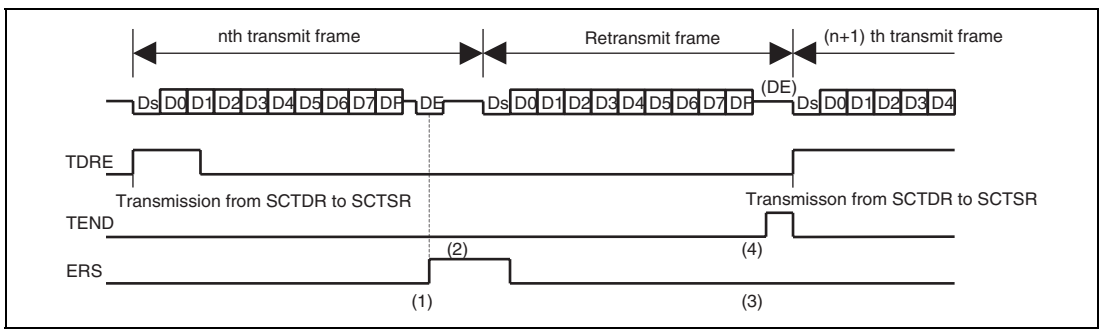


Figure 30.9 Retransmit Standby Mode (Clock Stopped) when Smart Card Interface is in Transmit Mode

(3) Standby Mode Switching

When switching between smart card interface mode and standby mode, in order to retain the clock duty, the following switching procedure should be used. Step (1) to step (7) of figure 30.10 correspond to the following operation.

- When switching from smart card interface mode to standby mode

1. Write 0 to the TE and RE bits in the serial control register (SCSCR), to stop transmit and receive operations. At the same time, set the CKE1 bit to the value for the output-fixed state in standby mode.
2. Write 0 to the CKE0 bit in SCSCR to stop the clock.
3. Wait for one cycle of the serial clock. During this interval, the duty is retained, and the clock output is fixed at the specified level.
4. Make the transition to standby mode.

- To return from standby mode to smart card interface mode

5. Cancel the standby state.
6. Set the CKE1 bit in the serial control register (SCSCR) to the value of the output-fixed state at the beginning of standby (the current SIM_CLK pin state).
7. Write 1 to the CKE0 bit in SCSCR to output a clock signal. Clock signal generation begins at normal duty.

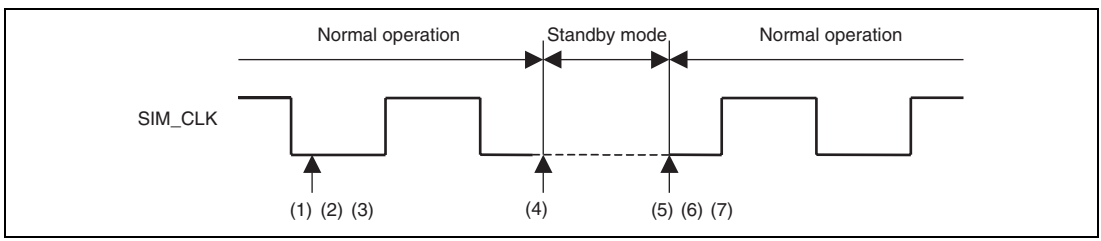


Figure 30.10 Procedure for Stopping Clock and Restarting

(4) Power-On and Clock Output

In order to retain the clock duty from power-on, the following switching procedure should be used.

1. The initial state is set to port-input with high impedance. In order to fix the potential, a pull-up resistance/pull-down resistance is used.
2. Use the CKE1 bit in the serial control register (SCSCR) to fix the specified output.
3. Set the CKE0 bit in SCSCR to 1 to start clock output.

(5) Pin Connections

An example of pin connections for the smart card interface is shown in figure 30.11.

In communication with the smart card, transmission and reception are performed using a single data transmit line. The data transmit line should be pulled up by a resistance on the power supply V_{cc} side.

When using the clock generated by the smart card interface with the IC card, the SIM_CLK pin output is input to the CLK pin of the IC card. If an internal clock of the IC card is used, this connection is not needed.

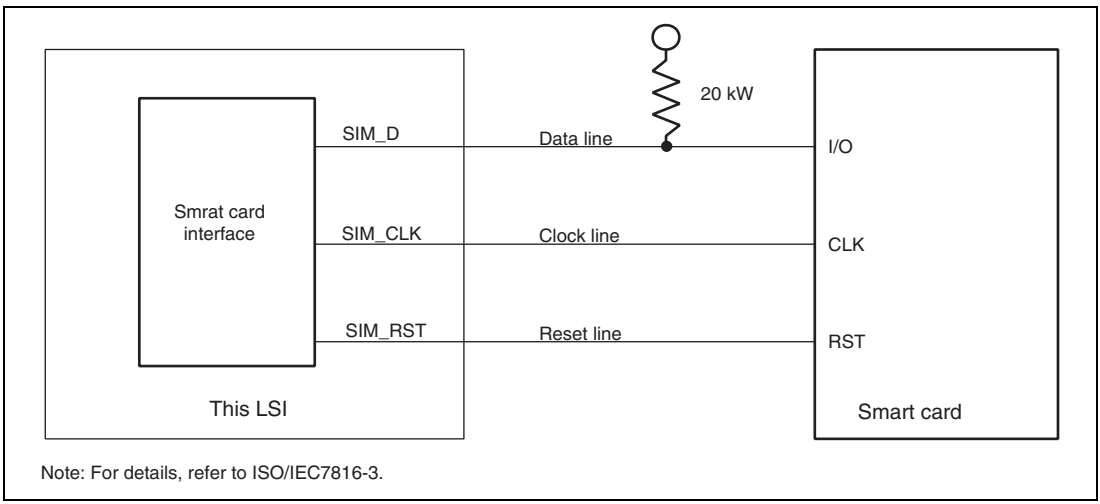


Figure 30.11 Example of Pin Connections in Smart Card Interface

Note: The transmission/reception in loop can perform self-check when the RE and TE bits are set to 1 without connecting to the IC card.

(6) Transmit End Interrupt

In continuous transmission, when the TEIE bit is always set to 1, the TEND bit is set to 1 at a transmit end. Therefore, the unnecessary transmit end interrupt (TEI) request occurs.

When SCTSR starts transmitting after the last transmit data is written to SCTDR, the TEIE bit in SCSCR should be set to 1 so that the occurrence of the unnecessary TEI interrupt request can be prevented.

The waveform of the timing to set the TEIE bit to 1 is shown in figure 30.12.

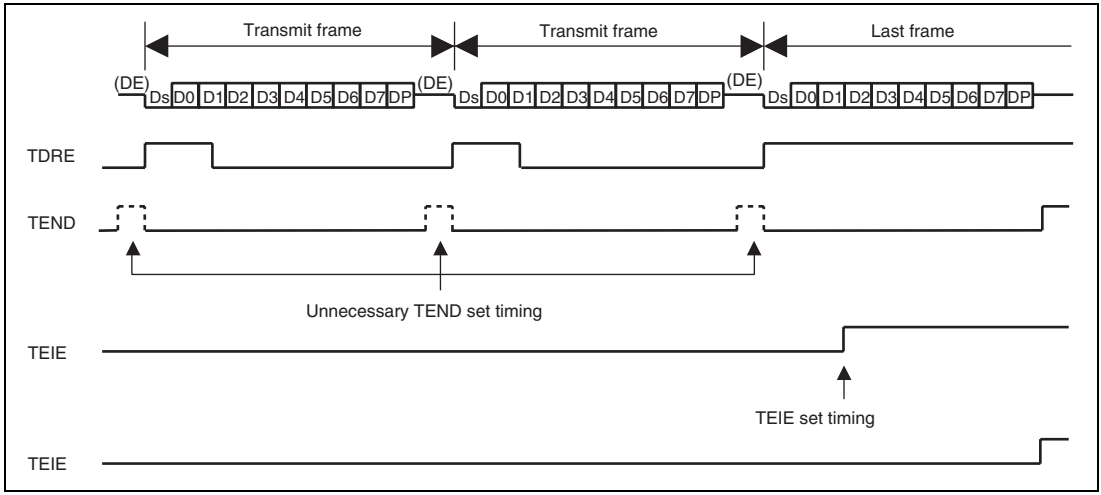


Figure 30.12 TEIE Set Timing

Section 31 Multimedia Card Interface (MMCIF)

This LSI supports a multimedia card interface (MMCIF). The MMC mode interface can be utilized. The MMCIF is a clock-synchronous serial interface that transmits/receives data which is distinguished in terms of command and response. A number of commands and responses are predefined in the multimedia card. As the MMCIF specifies a command code and command type/response type upon the issuance of a command, commands extended by the secure multimedia card (Secure-MMC) and additional commands can be supported in the future within the range of combinations of currently defined command types/response types.

31.1 Features

- Interface that complies with the MultiMediaCard System Specification Version 3.1
- MMC mode supported.
  Interface via the CLK output (transfer clock output) pin, the CMD input/output (command output/response input) pin, and the DAT input/output (data input/output) pin
- 16.7-Mbps bit rate (max.) for the card interface (at peripheral clock 1 of 33.3 MHz)
- Incorporates 64 data-transfer FIFOs of 16 bits
- DMA transfer requests issuance supported.
  Note that DMA transfer is disabled when the card conforming to the Multimedia Card System Specification (version 2.2 or lower) is connected.
- Four interrupt sources
  FIFO empty/full, command/response/data transfer complete, transfer error, and FIFO ready
- Card identification function
- Stream transfer unsupported

Figure 31.1 shows a block diagram of the MMCIF.

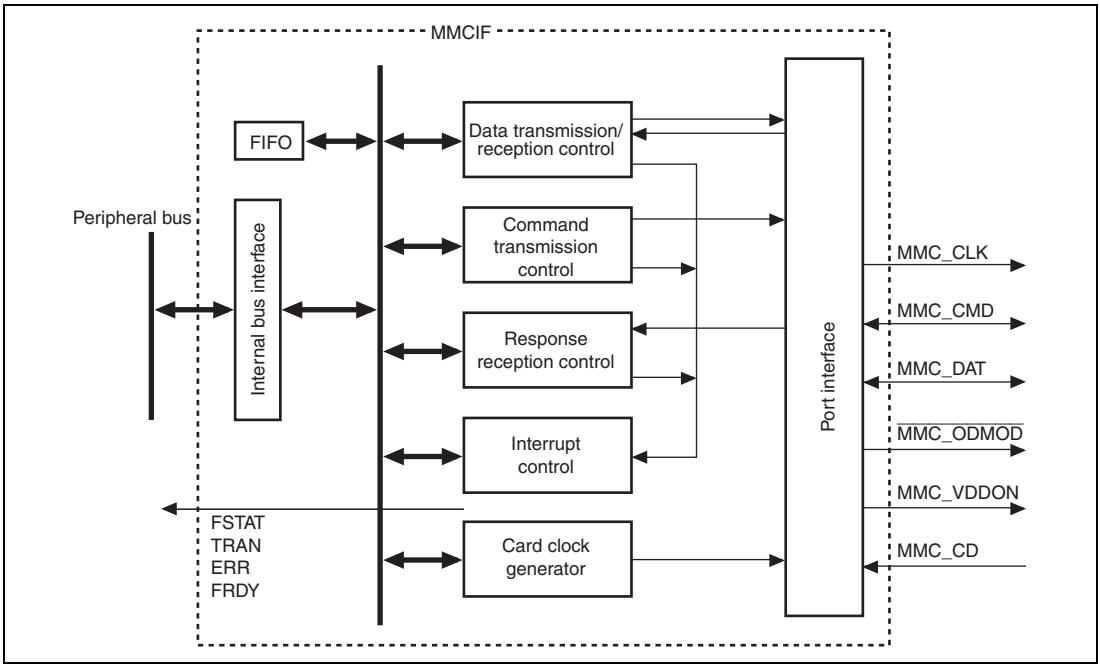


Figure 31.1 MMCIF Block Diagram

31.2 Input/Output Pins

Table 31.1 summarizes the pins of the MMCIF.

Table 31.1 Pin Configuration

Pin Name	Abbrev. (MMC)	I/O	Function
MMC_CLK	CLK	Output	Clock output pin
MMC_CMD	CMD	Input/Output	Command output/response input pin
MMC_DAT	DAT	Input/Output	Data input/output pin
$\overline{\text{MMC_ODMOD}}$	$\overline{\text{MMC_ODMOD}}$	Output	Open drain mode control
MMC_VDDON	MMC_VDDON	Output	Card power supply control
MMC_CD	MMC_CD	Input	Card identification signal

Note: For easier understanding of transmission and reception, the data transmission side and reception side are specified as MCTXD and MCRXD.

31.3 Register Descriptions

Figure 31.2 shows the MMCIF register configuration. Figure 31.3 shows the register states in each processing mode.

Table 31.2 Register Configuration

Register Name	Abbrev.	R/W	Area P4 Address	Area 7 Address	Access Size
Command type register	CMDTYR	R/W	H'FFF9 0018	H'1FF9 0018	8
Response type register	RSPTYR	R/W	H'FFF9 0019	H'1FF9 0019	8
Transfer byte number count register	TBCR	R/W	H'FFF9 0014	H'1FF9 0014	8
Transfer block number counter	TBNCR	R/W	H'FFF9 001A	H'1FF9 001A	16
Command register 0	CMDR0	R/W	H'FFF9 0000	H'1FF9 0000	8
Command register 1	CMDR1	R/W	H'FFF9 0001	H'1FF9 0001	8
Command register 2	CMDR2	R/W	H'FFF9 0002	H'1FF9 0002	8
Command register 3	CMDR3	R/W	H'FFF9 0003	H'1FF9 0003	8
Command register 4	CMDR4	R/W	H'FFF9 0004	H'1FF9 0004	8
Command register 5	CMDR5	R	H'FFF9 0005	H'1FF9 0005	8
Response register 0	RSPR0	R/W	H'FFF9 0020	H'1FF9 0020	8
Response register 1	RSPR1	R/W	H'FFF9 0021	H'1FF9 0021	8
Response register 2	RS PR2	R/W	H'FFF9 0022	H'1FF9 0022	8
Response register 3	RSPR3	R/W	H'FFF9 0023	H'1FF9 0023	8
Response register 4	RSPR4	R/W	H'FFF9 0024	H'1FF9 0024	8
Response register 5	RSPR5	R/W	H'FFF9 0025	H'1FF9 0025	8
Response register 6	RSPR6	R/W	H'FFF9 0026	H'1FF9 0026	8
Response register 7	RSPR7	R/W	H'FFF9 0027	H'1FF9 0027	8
Response register 8	RSPR8	R/W	H'FFF9 0028	H'1FF9 0028	8
Response register 9	RSPR9	R/W	H'FFF9 0029	H'1FF9 0029	8
Response register 10	RSPR10	R/W	H'FFF9 002A	H'1FF9 002A	8
Response register 11	RSPR11	R/W	H'FFF9 002B	H'1FF9 002B	8
Response register 12	RSPR12	R/W	H'FFF9 002C	H'1FF9 002C	8
Response register 13	RSPR13	R/W	H'FFF9 002D	H'1FF9 002D	8

Register Name	Abbrev.	R/W	Area P4 Address	Area 7 Address	Access Size
Response register 14	RSPR14	R/W	H'FFF9 002E	H'1FF9 002E	8
Response register 15	RSPR15	R/W	H'FFF9 002F	H'1FF9 002F	8
Response register 16	RSPR16	R/W	H'FFF9 0030	H'1FF9 0030	8
Response register D	RSPRD	R/W	H'FFF9 0031	H'1FF9 0031	8
Command start register	CMDSTRT	R/W	H'FFF9 0006	H'1FF9 0006	8
Operation control register	OPCR	R/W	H'FFF9 000A	H'1FF9 000A	8
Command timeout control register	CTOCR	R/W	H'FFF9 0011	H'1FF9 0011	8
Data timeout register	DTOUTR	R/W	H'FFF9 0032	H'1FF9 0032	16
Card status register	CSTR	R	H'FFF9 000B	H'1FF9 000B	8
Interrupt control register 0	INTCR0	R/W	H'FFF9 000C	H'1FF9 000C	8
Interrupt control register 1	INTCR1	R/W	H'FFF9 000D	H'1FF9 000D	8
Interrupt status register 0	INTSTR0	R/W	H'FFF9 000E	H'1FF9 000E	8
Interrupt status register 1	INTSTR1	R/W	H'FFF9 000F	H'1FF9 000F	8
Transfer clock control register	CLKON	R/W	H'FFF9 0010	H'1FF9 0010	8
VDD/open drain control register	VDCNT	R/W	H'FFF9 0012	H'1FF9 0012	8
Data register	DR	R/W	H'FFF9 0040	H'1FF9 0040	16
FIFO pointer clear register	FIFOCLR	W	H'FFF9 0042	H'1FF9 0042	8
DMA control register	DMACR	R/W	H'FFF9 0044	H'1FF9 0044	8
Interrupt control register 2	INTCR2	R/W	H'FFF9 0046	H'1FF9 0046	8
Interrupt status register 2	INTSTR2	R/W	H'FFF9 0048	H'1FF9 0048	8
Card switch register	CSWR	R	H'FFF9 004A	H'1FF9 004A	8
Switch status register	SWSR	R/W	H'FFF9 004C	H'1FF9 004C	8
Chattering elimination pulse setting register	CHATR	R/W	H'FFF9 004E	H'1FF9 004E	8

Table 31.3 Register States in Each Processing Mode

Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Module Standby
Command type register	CMDTYR	H'00	H'00	Retained	Retained
Response type register	RSPTYR	H'00	H'00	Retained	Retained
Transfer byte number count register	TBCR	H'00	H'00	Retained	Retained
Transfer block number counter	TBNCR	H'0000	H'0000	Retained	Retained
Command register 0	CMDR0	H'00	H'00	Retained	Retained
Command register 1	CMDR1	H'00	H'00	Retained	Retained
Command register 2	CMDR2	H'00	H'00	Retained	Retained
Command register 3	CMDR3	H'00	H'00	Retained	Retained
Command register 4	CMDR4	H'00	H'00	Retained	Retained
Command register 5	CMDR5	H'00	H'00	Retained	Retained
Response register 0	RSPR0	H'00	H'00	Retained	Retained
Response register 1	RSPR1	H'00	H'00	Retained	Retained
Response register 2	RSPR2	H'00	H'00	Retained	Retained
Response register 3	RSPR3	H'00	H'00	Retained	Retained
Response register 4	RSPR4	H'00	H'00	Retained	Retained
Response register 5	RSPR5	H'00	H'00	Retained	Retained
Response register 6	RSPR6	H'00	H'00	Retained	Retained
Response register 7	RSPR7	H'00	H'00	Retained	Retained
Response register 8	RSPR8	H'00	H'00	Retained	Retained
Response register 9	RSPR9	H'00	H'00	Retained	Retained
Response register 10	RSPR10	H'00	H'00	Retained	Retained
Response register 11	RSPR11	H'00	H'00	Retained	Retained
Response register 12	RSPR12	H'00	H'00	Retained	Retained
Response register 13	RSPR13	H'00	H'00	Retained	Retained
Response register 14	RSPR14	H'00	H'00	Retained	Retained
Response register 15	RSPR15	H'00	H'00	Retained	Retained
Response register 16	RSPR16	H'00	H'00	Retained	Retained
Response register D	RSPRD	H'00	H'00	Retained	Retained

Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Module Standby
Command start register	CMDSTRT	H'00	H'00	Retained	Retained
Operation control register	OPCR	H'00	H'00	Retained	Retained
Command timeout control register	CTOCR	H'00	H'00	Retained	Retained
Data timeout register	DTOUTR	H'FFFF	H'FFFF	Retained	Retained
Card status register	CSTR	H'0x	H'0x	Retained	Retained
Interrupt control register 0	INTCR0	H'00	H'00	Retained	Retained
Interrupt control register 1	INTCR1	H'00	H'00	Retained	Retained
Interrupt status register 0	INTSTR0	H'00	H'00	Retained	Retained
Interrupt status register 1	INTSTR1	H'00	H'00	Retained	Retained
Transfer clock control register	CLKON	H'00	H'00	Retained	Retained
VDD/open drain control register	VDCNT	H'00	H'00	Retained	Retained
Data register	DR	H'xxxx	H'xxxx	Retained	Retained
FIFO pointer clear register	FIFOCLR	H'00	H'00	Retained	Retained
DMA control register	DMACR	H'00	H'00	Retained	Retained
Interrupt control register 2	INTCR2	H'00	H'00	Retained	Retained
Interrupt status register 2	INTSTR2	H'0x	H'0x	Retained	Retained
Card switch register	CSWR	H'02	H'02	Retained	Retained
Switch status register	SWSR	H'00	H'00	Retained	Retained
Chattering elimination pulse setting register	CHATR	H'13	H'13	Retained	Retained

31.3.1 Command Type Register (CMDTYR)

CMDTYR specifies the command format in conjunction with RSPTYR. The bits TY1 and TY0 specify the existence and direction of transfer data, and the bits TY6 to TY4, and TY2 specify the additional settings. The bits TY6 to TY4, and TY2 should all be cleared to 0 or only one of them should be set to 1. The bits TY6 to TY4, and TY2 can only be set to 1 if the corresponding settings in the bits TY1 and TY0 allow that setting. If these bits are not set correctly, operation cannot be guaranteed. When transferring a single block, set TY1 and TY0 to 01 or 10, and all of TY6 to TY4, and TY2 to 0.

Bit:	7	6	5	4	3	2	1	0
	—	TY6	TY5	TY4	—	TY2	TY[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

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	TY6	0	R/W	Specifies predefined multiblock transfer. Bits TY1 and TY0 should be set to 01 or 10. When using a command to set this bit, it is necessary to specify the transfer block size and the transfer block number in TBCR and TBNCR, respectively.
5	TY5	0	R/W	Specifies multiblock transfer when using secure MMC. Bits TY1 and TY0 should be set to 01 or 10. When using a command to set this bit, it is necessary to specify the transfer block size and the transfer block number in TBCR and TBNCR, respectively.
4	TY4	0	R/W	Set this bit to 1 when issuing the CMD12 command. Bits TY1 and TY0 should be set to 00. To issue Stop Tran (SPI multiblock write end data token), set this bit to 1 and the bits TY1 and TY0 to 11.
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	TY2	0	R/W	Specifies open-ended multiblock transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the multiblock transfer specified by this bit ends when it is aborted by the CMD12 command.
1, 0	TY[1:0]	0	R/W	These bits specify the existence and direction of transfer data. 00: A command without data transfer 01: A command with read data reception 10: A command with write data transmission 11: Set this bit when transmitting Stop Tran

Table 31.4 summarizes the correspondence between the commands described in the MultiMediaCard System Specification Version 3.1 and the settings of the CMDTYR and RSPTYR registers.

31.3.2 Response Type Register (RSPTYR)

RSPTYR specifies the command format in conjunction with CMDTYR. The bits RTY[2:0] specify the number of response bytes, and the bits RTY5 and RTY4 specify the additional settings.

Bit:	7	6	5	4	3	2	1	0
	—	—	RTY5	RTY4	—	RTY[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RTY5	0	R/W	Set this bit when using a command with an R1b response

Bit	Bit Name	Initial Value	R/W	Description
4	RTY4	0	R/W	Specifies that the command response CRC (other than an R2 response) be checked through CRC7. Bits RTY2 to RTY0 should be set to 100.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	RTY[2:0]	000	R/W	These bits specify the number of command response bytes. 000: A command requiring no command responses. 001: Setting prohibited. 010: Setting prohibited. 011: Setting prohibited. 100: A command requiring 6-byte command response. Specified by R1, R1b, R3, R4, and R5 responses in MMC mode. 101: A command requiring a 17-byte command response. Specified by the R2 response in MMC mode. 110: Setting prohibited 111: Setting prohibited

Note: The purpose of a CRC check through RTY4 is to check the CRC that is attached to a command response, not to check the bit in the event of a CRC error regarding a command response. A CRC check for an R2 command response in MMC mode cannot be performed.

Table 31.3 summarizes the correspondence between the commands described in the MultiMediaCard System Specification Version 3.1 and the settings of the CMDTYR and RSPTYR registers.

Table 31.4 Correspondence between Commands and CMDTYR and RSPTYR Settings

- MMC Mode

CMD			CMDTYR						RSPTYR			
INDEX	Abbreviation	resp	6	5	4	3	2	1, 0	6	5	4	2 to 0
CMD0	GO_IDLE_STATE	—						00				000
CMD1	SEND_OP_COND	R3						00				100
CMD2	ALL_SEND_CID	R2						00				101
CMD3	SET_RELATIVE_ADDR	R1						00			*4	100
CMD4	SET_DSR	—						00				000
CMD7	SELECT/DESELECT_CARD	R1b						00		1	*4	100
CMD9	SEND_CSD	R2						00				101
CMD10	SEND_CID	R2						00				101
CMD11	READ_DAT_UNTIL_STOP	R1						01			*4	100
CMD12	STOP_TRANSMISSION	R1b			1			00		1	*4	100
CMD13	SEND_STATUS	R1						00			*4	100
CMD15	GO_INACTIVE_STATE	—						00				000
CMD16	SET_BLOCKLEN	R1						00			*4	100
CMD17	READ_SINGLE_BLOCK	R1		*3				01			*4	100
CMD18	READ_MULTIPLE_BLOCK	R1	*2				*2	01			*4	100
CMD20	WRITE_DAT_UNTIL_STOP	R1						10			*4	100
CMD23	SET_BLOCK_COUNT	R1						00			*4	100
CMD24	WRITE_BLOCK	R1		*3				10			*4	100
CMD25	WRITE_MULTIPLE_BLOCK	R1	*2				*2	10			*4	100
CMD26	PROGRAM_CID	R1						10			*4	100
CMD27	PROGRAM_CSD	R1						10			*4	100
CMD28	SET_WRITE_PROT	R1b						00		1	*4	100
CMD29	CLR_WRITE_PROT	R1b						00		1	*4	100
CMD30	SEND_WRITE_PROT	R1						01			*4	100

CMD			CMDTYR						RSPTYR			
INDEX	Abbreviation	resp	6	5	4	3	2	1, 0	6	5	4	2 to 0
CMD32*1	TAG_SECTOR_START	R1						00			*4	100
CMD33*1	TAG_SECTOR_END	R1						00			*4	100
CMD34*1	UNTAG_SECTOR	R1						00			*4	100
CMD35	TAG_ERASE_GROUP_START	R1						00			*4	100
CMD36	TAG_ERASE_GROUP_END	R1						00			*4	100
CMD37*1	UNTAG_ERASE_GROUP	R1						00			*4	100
CMD38	ERASE	R1b						00		1	*4	100
CMD39	FAST_IO	R4						00			*4	100
CMD40	GO_IRQ_STATE	R5						00			*4	100
CMD42	LOCK_UNLOCK	R1b						10		1	*4	100
CMD55	APP_CMD	R1						00			*4	100
CMD56	GEN_CMD	R1b						*5		1	*4	100

Notes: 1. These commands are not supported after MMCA Ver3.1 specification cards.
2. Set the TY6 bit when the transfer block number is set in advance, set the TY2 bit when the transfer block number is not set.
3. Set this bit when using secure MMC multiple block transaction.
4. Set these bits to 1 when the CRC of a command and response other than R2 is checked. (Checking the CRC of an R2 command and response is not possible).
5. Set these bits to 01 in read and 10 in write access.

Blank spaces denote a value of 0.

31.3.3 Transfer Byte Number Count Register (TBCR)

TBCR specifies the number of bytes to be transferred (the block size) for each single block transfer command. TBCR specifies the net number of data block bytes not including the start bit, end bit, and CRC.

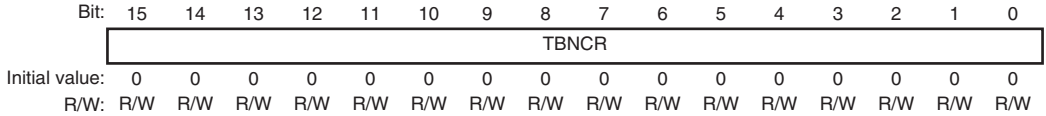
The multiblock transfer command corresponds to the number of bytes of each data block.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	C[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CS[3:0]	0000	R/W	Transfer Data Block Size 0000: 1 byte 0001: 2 bytes 0010: 4 bytes 0011: 8 bytes 0100: 16 bytes 0101: 32 bytes 0110: 64 bytes 0111: 128 bytes 1000: 256 bytes 1001: 512 bytes 1010: 1024 bytes 1011: 2048 bytes 1100 to 1111: Setting prohibited

31.3.4 Transfer Block Number Counter (TBNCR)

A value other than 0 must be written to the TBNCR register if multiple block transfer is selected by the bits TY5 and TY6 in CMDTYR. Set the transfer block number in TBNCR. Every time a block transfer ends, the value of TBNCR is decremented by one, and the command sequence finishes when the TBNCR value reaches 0.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TBNCR	All 0	R/W	Transfer Block Number Counter [Clearing condition] When a specified block number is transferred or 0 is written to TBNCR.

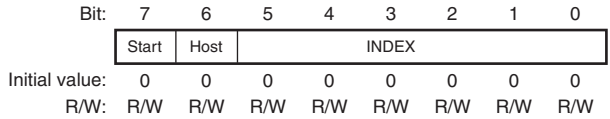
31.3.5 Command Registers 0 to 5 (CMDR0 to CMDR5)

The CMDR registers are six 8-bit registers. A command is written to CMDR as shown in table 31.5, and the command is transmitted when the START bit in CMDSTRT is set to 1.

Table 31.5 CMDR Configuration

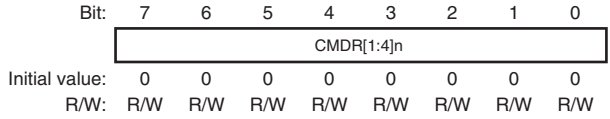
Register	Contents	Operation
CMDR0	Start bit, Host bit, and command index	Write command index Set the Start bit to 0 and the Host bit to 1
CMDR[1:4]	Command argument	Write command arguments.
CMDR5	CRC and End bits	Setting of CRC is unnecessary (automatic calculation). Setting of the End bit is unnecessary.

- CMDR0



Bit	Bit Name	Initial Value	R/W	Description
7	Start	0	R/W	Start bit (set to 0)
6	Host	0	R/W	Transmission bit (set to 1)
5 to 0	INDEX	All 0	R/W	Command index

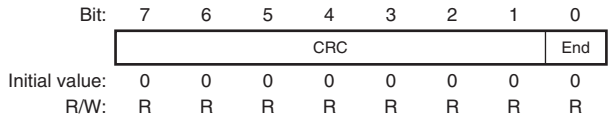
- CMDR1 to CMDR4



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CMDR[1:4]n	All 0	R/W	Command arguments

Note: n = 0 to 7

- CMDR5



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	CRC	All 0	R	Setting unnecessary. These bits are always read as 0.
0	End	0	R	Setting unnecessary. This bit is always read as 0.

31.3.6 Response Registers 0 to 16, D (RSPR0 to RSPR16, RSPRD)

RSPR0 to RSPR16 are 8-bit command response registers. RSPRD is a 5-bit data response register.

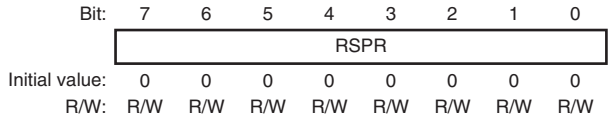
The number of command response bytes differs according to the command. The number of command response bytes can be specified through RSPTYR in the MMCIF. The command response is shifted-in from the bit 0 in RSPR16, and shifted to the number of command response bytes × 8 bits. Table 31.6 summarizes the correspondence between the number of command response bytes and valid RSPR registers.

Table 31.6 Correspondence between Command Response Byte Number and RSPR

	MMC Mode Response	
RSPR registers	**6 bytes (R1, R1b, R3, R4, R5)**	**17 bytes (R2)**
RSPR0	—	1st byte
RSPR1	—	2nd byte
RSPR2	—	3rd byte
RSPR3	—	4th byte
RSPR4	—	5th byte
RSPR5	—	6th byte
RSPR6	—	7th byte
RSPR7	—	8th byte
RSPR8	—	9th byte
RSPR9	—	10th byte
RSPR10	—	11th byte
RSPR11	1st byte	12th byte
RSPR12	2nd byte	13th byte
RSPR13	3rd byte	14th byte
RSPR14	4th byte	15th byte
RSPR15	5th byte	16th byte
RSPR16	6th byte	17th byte

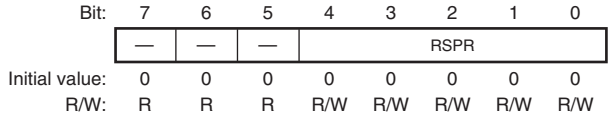
RSPR0 to RSPR16 are simple shift registers with the initial value H'00. A command response that has been shifted in is not automatically cleared, and it is continuously shifted until it is shifted out from bit 7 in RSPR0. To clear unnecessary bytes to H'00, write an arbitrary value to each RSPR.

- RSPR0 to RSPR16



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RSPR	All 0	R/W	These bits are cleared to H'00 by writing an arbitrary value. RSPR0 to RSPR16 make up a continuous 17-byte shift registers in which a command response is stored.

- RSPRD



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	RSPR	All 0	R/W	All of these bits are cleared to 00 by writing an arbitrary value. Data response is stored.

31.3.7 Command Start Register (CMDSTRT)

CMDSTRT is an 8-bit readable/writable register that triggers the start of command transmission or Stop Tran transmission, representing the start of a command sequence. The following operations should be completed before the command sequence starts.

Command transmission:

- Analysis of prior command response, clearing the command response register write if necessary
- Analysis/transfer of receive data of prior command if necessary
- Preparation of transmit data of the next command if necessary
- Setting of CMDTYR, RSPTYR, TBCR, and TBNCR

  The CMDR0 to CMDR4, CMDTYR, RSPTYR, TBCR, and TBNCR registers should not be changed until command transmission has ended (during the time the CWRE flag in CSTR is set to 1).
- Setting of CMDR0 to CMDR4

Stop Tran transmission:

- Setting of CMDTYR and RSPTYR

  The CMDTYR and RSPTYR registers should not be changed until command transmission has ended (during the time the CWRE flag in CSTR is set to 1).

Command sequences are controlled by the sequencers on both the MMCIF side and the MMC card side. Normally, these operate synchronously. However, if an error occurs or a command is aborted, these may become temporarily unsynchronized. Be careful when setting the CMDOFF bit in OPCR, issuing the CMD12 command, or processing an error in MMC mode. A new command sequence should be started only after the end of the command sequence on both the MMCIF and card sides is confirmed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	START
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	START	0	R/W	Starts command or Stop Tran transmission when 1 is written. This bit is cleared by hardware.

31.3.8 Operation Control Register (OPCR)

OPCR aborts command operation, and suspends or continues data transfer.

Bit:	7	6	5	4	3	2	1	0
	CMD OFF	—	RD_ CONTI	DATEN	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CMDOFF	0	R/W	Command Off Aborts all command operations (MMCIF command sequence) when 1 is written to it after a command is transmitted. This bit is then cleared by hardware. Write enabled period: From command transmission completion to command sequence end Write 0: Operation is not affected. Write 1: Command sequence is forcibly aborted.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	RD_CONTI	0	R/W	Read Continue This bit is cleared by hardware when 1 is written and the MMCIF resumes data read. Read data reception is resumed while the sequence has been halted by FIFO full or termination of block reading in multiblock read. Write enabled period: While read data reception is halted Write 0: Operation is not affected. Write 1: Resumes read data reception.

Bit	Bit Name	Initial Value	R/W	Description
4	DATAEN	0	R/W	Data Enable Starts write data transmission by a command with write data. Resumes write data transmission while the sequence has been halted by FIFO empty or termination of block writing in multiblock write. Write enabled period: (1) after receiving a response to a command with write data, (2) while sequence is halted by FIFO empty, (3) when one block writing in multiblock write is terminated Write 0: Operation is not affected. Write 1: Starts or resumes write data transmission.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

In write data transmission, the contents of the command response and data response should be analyzed, and then transmission should be triggered. In addition, the data transmission should be temporarily halted by FIFO full/empty, and it should resume when the preparation has been completed.

In multiblock transfer, the transfer should be temporarily halted at every block break to select whether to continue to the next block or to abort the multiblock transfer command by issuing the CMD12 command or Stop Tran. To continue to the next block, the RD_CONTI and DATAEN bits should be set to 1. To issue the CMD12 command or Stop Tran, the CMDOFF bit should be set to 1 to abort the command sequence on the MMCIF side. When using the auto-mode for pre-defined multiblock transfer, the setting of the RD_CONTI bit or the DATAEN bit between blocks can be omitted.

31.3.9 Command Timeout Control Register (CTOCR)

CTOCR specifies the period for generating a timeout for the command response.

When the command response is received, CTOCR continues counting the transfer clock and enters the command timeout error state when the number of transfer clock cycles reaches the number specified in CTOCR. When the CTERIE bit in INTCR1 is set to 1, the CTERI flag in INTSTR1 is set. To perform command timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1 and then clearing the CTERI flag.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTSEL0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CTSEL0	1	R/W	0: 128 transfer clock cycles from command transmission completion to response reception completion 1: 256 transfer clock cycles from command transmission completion to response reception completion

Note: If an R2 response (17-byte command response) in MMC mode is requested and CTSEL0 is cleared to 0, a timeout is generated during response reception. Therefore, set CTSEL0 to 1.

31.3.10 Data Timeout Register (DTOUTR)

DTOUTR specifies the period for generating a data timeout. The 16-bit counter (DTOUTC) and a prescaler count peripheral clock 1 to monitor the data timeout. The prescaler always counts peripheral clock 1, and outputs a count pulse at every 10,000 peripheral clock 1 cycles. The initial value of DTOUTC is 0, and DTOUTC starts counting the prescaler output at the start of the command sequence. DTOUTC is cleared when the command sequence has ended, or when the command sequence has been aborted by setting the CMDOFF bit to 1, after which the DTOUTC stops counting the prescaler output.

When the command sequence does not end, DTOUTC continues counting the prescaler output, and enters the data timeout error states when the number of prescaler outputs reaches the number specified in DTOUTR. When the DTERIE bit in INTCR1 is set to 1, the DTERI flag in INTSTR1 is set. To perform data timeout error handling, abort the command sequence by setting the CMDOFF bit to 1, and then clear the DTERI flag.

For a command with data busy status, data timeout cannot be monitored since the command sequence is terminated before entering the data busy state. Timeout in the data busy state should be monitored by firmware. Setting DTOUTR to 0 will cause a timeout immediately after the start of the command sequence.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DTOUTR	All 1	R/W	Data Timeout Time/10,000 Data timeout time: Peripheral clock 1 cycle × DTOUTR setting value × 10,000.

31.3.11 Card Status Register (CSTR)

CSTR indicates the MMCIF status during command sequence execution.

Bit:	7	6	5	4	3	2	1	0
	BUSY	FIFO_FULL	FIFO_EMPTY	CWRE	DTBUSY	DTBUSY_TU	—	REQ
Initial value:	0	0	0	0	0	—	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUSY	0	R	Command Busy Indicates command execution status. When the CMDOFF bit in OPCR is set to 1, this bit is cleared to 0 because the MMCIF command sequence is aborted. 0: Idle state waiting for a command, or data busy state. 1: Command sequence execution in progress.
6	FIFO_FULL	0	R	FIFO Full This bit is set to 1 when the FIFO becomes full while data is being received, and cleared to 0 when RD_CONTI is set to 1 or the command sequence is completed. 0: The FIFO is empty. 1: The FIFO is full.
5	FIFO_EMPTY	0	R	FIFO Empty This bit is set to 1 when the FIFO becomes empty while data is being sent, and cleared to 0 when DATAEN is set to 1 or the command sequence is completed. Indicates whether the FIFO holds data or not. 0: The FIFO includes data. 1: The FIFO is empty.

Bit	Bit Name	Initial Value	R/W	Description
4	CWRE	0	R	Command Register Write Enable Indicates whether the CMDR command is being transmitted or has been transmitted. 0: The CMDR command has been transmitted, or the START bit in CMDSTRT has not been set yet, so the new command can be written. 1: The CMDR command is waiting to be transmitted or is being transmitted. If a new command is written, a malfunction will result.
3	DTBUSY	0	R	Data Busy Indicates command execution status. Indicates that the card is in the busy state after the command sequence of a command without data transfer which includes the busy state in the response has ended or a command with write data has ended. 0: Idle state waiting for a command, or command sequence execution in progress 1: Card is in the data busy state after command sequence termination.
2	DTBUSY_TU	Undefined	R	Data Busy Pin Status Monitors the levels of the DAT pin in MMC mode. By reading this bit, whether the card is in the busy state can be monitored after the card in the busy state has been deselected and then selected again afterwards. 0: Card indicates data busy. 1: Card indicates not data busy.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	REQ	0	R	Interrupt Request Indicates whether an interrupt is requested. An interrupt request is the logical OR of the INTSTR0, INTSTR1, and INTSTR2 flags. Settings of the INTSTR0, INTSTR1, and INTSTR2 flags are controlled by the enable bits in INTCR0, INTSTR1, and INTCR2. 0: No interrupt requested. 1: Interrupt requested.

31.3.12 Interrupt Control Registers 0 and 1 (INTCR0, INTCR1)

The INTCR registers enable or disable the INTSTR0 and INTSTR1 flags and control the interrupt outputs.

- INTCR0

Bit:	7	6	5	4	3	2	1	0
	FEIE	FFIE	DRPIE	DTIE	CRPIE	CMDIE	DBSYIE	BTIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	FEIE	0	R/W	FIFO Empty Flag Enable 0: Disables FIFO empty flag setting. 1. Enables FIFO empty flag setting.
6	FFIE	0	R/W	FIFO Full Flag Enable 0: Disables FIFO full flag setting. 1: Enables FIFO full flag setting.
5	DRPIE	0	R/W	Data Response End Flag Enable 0: Disables data response end flag setting. 1: Enables data response end flag setting.
4	DTIE	0	R/W	Data Transfer End Flag Enable 0: Disables data transfer end flag setting. 1: Enables data transfer end flag setting.
3	CRPIE	0	R/W	Command Response End Flag Enable 0: Disables command response end flag setting. 1: Enables command response end flag setting.
2	CMDIE	0	R/W	Command Output End Flag Enable 0: Disables command output end flag setting. 1: Enables command output end flag setting.
1	DBSYIE	0	R/W	Data Busy End Flag Enable 0: Disables data busy end flag setting. 1: Enables data busy end flag setting.

Bit	Bit Name	Initial Value	R/W	Description
0	BTIE	0	R/W	Multiblock Transfer End Flag Enable. 0: Disables multiblock transfer end flag setting 1: Enables multiblock transfer end flag setting

- INTCR1

Bit:	7	6	5	4	3	2	1	0
	INTRQ2E	INTRQ1E	INTRQ0E	—	WRERIE	CRCERIE	DTERIE	CTERIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ2E	0	R/W	ERR Interrupt Enable 0: Disables ERR interrupt. 1: Enables ERR interrupt.
6	INTRQ1E	0	R/W	TRAN Interrupt Enable 0: Disables TRAN interrupt. 1: Enables TRAN interrupt.
5	INTRQ0E	0	R/W	FSTAT Interrupt Enable 0: Disables FSTAT interrupt. 1: Enables FSTAT interrupt.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	WRERIE	0	R/W	Write Error Flag Enable 0: Disables write error flag setting. 1: Enables write error flag setting.
2	CRCERIE	0	R/W	CRC Error Flag Enable 0: Disables CRC error flag setting. 1: Enables CRC error flag setting.
1	DTERIE	0	R/W	Data Timeout Error Flag Enable 0: Disables data timeout error flag setting. 1: Enables data timeout error flag setting.

Bit	Bit Name	Initial Value	R/W	Description
0	CTERIE	0	R/W	Command Timeout Error Flag Enable 0: Disables command timeout error flag setting. 1: Enables command timeout error flag setting.

31.3.13 Interrupt Status Registers 0 and 1 (INTSTR0, INTSTR1)

The INTSTR registers control MMCIF interrupts.

- INTSTR0

Bit:	7	6	5	4	3	2	1	0
	FEI	FFI	DRPI	DTI	CRPI	CMDI	DBSYI	BTI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
7	FEI	0	R/(W)*	FIFO Empty Flag [Setting 1 condition] When FIFO becomes empty while FEIE = 1 and write data is being transmitted (when the FIFO_EMPTY bit in CSTR is set) [Clearing 0 condition] Write 0 after reading FEI = 1.	FSTAT
6	FFI	0	R/(W)*	FIFO Full Flag [Setting 1 condition] When FIFO becomes full while FFIE = 1 and read data is being received (when the FIFO_FULL bit in CSTR is set) [Clearing 0 condition] Write 0 after reading FFI = 1.	FSTAT

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
5	DRPI	0	R/(W)*	Data Response Flag [Setting 1 condition] When the CRC status is received while DRPIE = 1. [Clearing 0 condition] Write 0 after reading DRPI = 1.	TRAN
4	DTI	0	R/(W)*	Data Transfer End Flag [Setting 1 condition] When the number of bytes of data transfer specified in TBCR ends while DTIE = 1. [Clearing 0 condition] Write 0 after reading DTI = 1.	TRAN
3	CRPI	0	R/(W)*	Command Response Receive End Flag [Setting 1 condition] When command response reception ends while CRPIE = 1. [Clearing 0 condition] Write 0 after reading CRPI = 1.	TRAN
2	CMDI	0	R/(W)*	Command Transmit End Flag [Setting 1 condition] When command transmission ends while CMDIE = 1. [Clearing 0 condition] Write 0 after reading CMDI = 1.	TRAN
1	DBSYI	0	R/(W)*	Data Busy End Flag [Setting 1 condition] When data busy state is canceled while DBSYIE = 1 [Clearing 0 condition] Write 0 after reading DBSYI = 1.	TRAN
0	BTI	0	R/(W)*	Multiblock Transfer End Flag [Setting 1 condition] When the number of bytes of data transfer specified in TBCR ends after TBNCR is decremented to 0 while BTIE = 1. [Clearing 0 condition] Write 0 after reading BTI = 1.	TRAN

Note: * Cleared by writing 0 after reading 1

- INTSTR1

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	WRERI	CRCERI	DTERI	CTERI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	
3	WRERI	0	R/(W)*	Write Error Flag [Setting 1 condition] When a status error (write error) for transmit data response is detected while WREIE = 1. [Clearing 0 condition] Write 0 after reading WREI = 1. Note: When the write error occurs, abort the command sequence by setting the CMDOFF bit to 1.	ERR
2	CRCERI	0	R/(W)*	CRC Error Flag [Setting 1 condition] When a CRC error for command response or receive data or a CRC status error for transmit data response is detected while CRCERIE = 1. For the command response other than R2, CRC is checked when the RTY4 in RSPTYR is enabled. For the R2 command response, CRC is not checked, thus this flag is not set. [Clearing 0 condition] Write 0 after reading CRCERI = 1. Note: When the CRC error occurs, abort the command sequence by setting the CMDOFF bit to 1.	ERR

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
1	DTERI	0	R/(W)*	Data Timeout Error Flag [Setting 1 condition] When a data timeout error specified in DTOUTR occurs while DTERIE = 1. [Clearing 0 condition] Write 0 after reading DTERI = 1. Note: When the data timeout error occurs, abort the command sequence by setting the CMDOFF bit to 1 and then clear the DTERI flag.	ERR
0	CTERI	0	R/(W)*	Command Timeout Error Flag [Setting 1 condition] When a command timeout error specified in TOCR occurs while CTERIE = 1. [Clearing condition] Write 0 after reading CTERI = 1. Note: When the command timeout error occurs, abort the command sequence by setting the CMDOFF bit to 1 and then clear the CTERI flag.	ERR

Note: * Cleared by writing 0 after reading 1

31.3.14 Transfer Clock Control Register (CLKON)

CLKON controls the transfer clock frequency and clock ON/OFF.

Bits CSEL[3:0] must be set to 0001 for the peripheral clock of 33.3 MHz in order to achieve a 16.7-Mbps transfer clock in the MMCIF. At this time, use a sufficiently slow clock for transfer in open-drain type output in MMC mode.

In a command sequence, do not perform clock ON/OFF or frequency modification.

Bit:	7	6	5	4	3	2	1	0
	CLKON	—	—	—	CSEL[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CLKON	0	R/W	Clock On 0: Fixes the transfer clock output from the MMC_CLK pin to low level. 1: Outputs the transfer clock from the MMC_CLK pin.
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CSEL[3:0]	0000	R/W	Transfer Clock Frequency Select 0000: Setting prohibited 0001: Uses the 1/2-divided system clock as a transfer clock. 0010: Uses the 1/4-divided system clock as a transfer clock. 0011: Uses the 1/8-divided system clock as a transfer clock. 0100: Uses the 1/16-divided system clock as a transfer clock. 0101: Uses the 1/32-divided system clock as a transfer clock. 0110: Uses the 1/64-divided system clock as a transfer clock. 0111: Uses the 1/128-divided system clock as a transfer clock. 1000: Uses the 1/256-divided system clock as a transfer clock. 1001 to 1111: Setting prohibited

31.3.15 VDD/Open-Drain Control Register (VDCNT)

VDCNT can use the $\overline{\text{MMC_ODMOD}}$ pin to control open-drain in MMC mode. $\overline{\text{MMC_VDDON}}$ output is available for ON/OFF of the card power supply (VDD).

Bit:	7	6	5	4	3	2	1	0
	VDDON	ODMOD	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	VDDON	0	R/W	Available as a control signal for card power supply (VDD) 0: Low level signal is output to $\overline{\text{MMC_VDDON}}$. 1: High level signal is output to $\overline{\text{MMC_VDDON}}$.
6	ODMOD	0	R/W	Available to control open-drain of CMD output in MMC mode 0: Low level signal is output to $\overline{\text{MMC_ODMOD}}$. 1: High level signal is output to $\overline{\text{MMC_ODMOD}}$.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

31.3.16 Data Register (DR)

DR is a register to read/write the FIFO data.

Word- or byte-access is possible.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0 (7 to 0)	DR	Undefined	R/W	Register to read/write FIFO data. Word- or byte-access is possible. However, address 2n + 1 cannot be accessed in bytes.

31.3.17 FIFO Pointer Clear Register (FIFOCLR)

The FIFO write/read pointer is cleared by writing an arbitrary value to FIFOCLR.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	FIFOCLR	Undefined	W	The FIFO pointer is cleared by writing an arbitrary value to this register.

31.3.18 DMA Control Register (DMACR)

DMACR sets DMA request signal output. DMAEN enables or disables a DMA request signal. The DMA request signal is output based on a value that has been set to SET[2:0].

This register should be set before a multiblock transfer command (CMD18 or CMD25) is executed. Auto mode cannot be used for open-ended multiblock transfers.

Bit:	7	6	5	4	3	2	1	0
	DMAEN	AUTO	—	—	—	SET[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DMAEN	0	R/W	0: Disables output of DMA request signal. 1: Enables output of DMA request signal.
6	AUTO	0	R/W	Auto Mode for pre-define multiblock transfer using DMA transfer. For details on auto mode operation, see section 14, Direct Memory Access Controller (DMAC). 0: Disable auto mode 1: Enable auto mode
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	SET[2:0]	000	R/W	Sets DMA request signal assert condition. 000: Not output 001: Remaining data in FIFO is 1/4 or less of FIFO capacity. 010: Remaining data in FIFO is 1/2 or less of FIFO capacity. 011: Remaining data in FIFO is 3/4 or less of FIFO capacity. 100: Remaining data in FIFO is 1 byte or more. 101: Remaining data in FIFO is 1/4 or more of FIFO capacity. 110: Remaining data in FIFO is 1/2 or more of FIFO capacity. 111: Remaining data in FIFO is 3/4 or more of FIFO capacity.

31.3.19 Interrupt Control Register 2 (INTCR2)

INTCR2 controls the enable/disable of interrupts.

Bit:	7	6	5	4	3	2	1	0
	INTREQ3E	—	—	—	—	—	CDIE	FRDYIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ3E	0	R/W	FRDY Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	CDIE	0	R/W	Card Detection Flag Enable 0: Setting of card detection flag disabled 1: Setting of card detection flag enabled
0	FRDYIE	0	R/W	FIFO Ready Completion Flag Enable 0: Setting of FIFO ready completion flag disabled 1: Setting of FIFO ready completion flag enabled

31.3.20 Interrupt Status Register 2 (INTSTR2)

INTSTR2 controls the interrupt output of the MMCIF.

FRDYI is set even in the set condition after a clear. To clear FRDYI, disable the flag setting through FRDYIE in INTCR2.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CDI	FRDY_TU	FRDYI
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R/(W)*	R	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description	Interrupt output
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	
2	CDI	0	R/(W)*	Card Identification Flag Identifies insert/pullout of card (variation between high and low of card identification signal) [Setting 1 condition] When insert/pullout of card is identified while CDIE = 1. [Clearing 0 condition] Write 0 after reading CDI = 1.	FRDY
1	FRDY_TU	1	R	When the set condition of FRDYI is met Read value 0: Remaining data in FIFO meets the assert condition specified by DMACR. 1: Remaining data in FIFO does not meet the assert condition specified by DMACR.	
0	FRDYI	0	R/(W)*	FIFI Ready Completion Flag [Setting 1 condition] When the DMAEN bit is set while FRDYIE = 1 and the remaining data in FIFO does not meet the assert condition specified by DMACR. [Clearing 0 condition] Write 0 after reading FRDYI = 1.	FRDY

Note: * Cleared by writing 0 after reading 1

31.3.21 Card Switch Register (CSWR)

CSWR indicates the state of the card identification signal.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CDB*
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	CDB*	0	R	Card Identification State Indication Indicates insert/pullout of card 0: Card not inserted (mmc_cd = 1) 1: Card inserted (mmc_cd = 0)

Note: * mmc_cd = 0 indicates the card is inserted. The polarity of the card identification state indication changes depending on the connector types.

31.3.22 Switch Status Register (SWSR)

SWSR controls the peripheral clock 1 and the internally divided clock when the card is identified.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	GATE_CDB	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	GATE_CDB	0	R/W	Clock Control at Card Identification Stops the clock supply to FF that is not required at card identification 0: Peripheral clock 1 and internal divided clock operating 1: Peripheral clock 1 and internal divided clock halted
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

31.3.23 Chattering Elimination Pulse Setting Register (CHATR)

CHATR specifies the cycle of the chattering elimination pulse to eliminate the chattering that arises in the card identification signal.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	CHAT				
Initial value:	0	0	0	1	0	0	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	000	R	The read value is 0. The write value should be 0.
4 to 0	CHAT	10011	R/W	Frequency for Chattering Elimination Pulse Cycle 00000 to 01000 and 11000 to 11111: setting prohibited

Chattering elimination pulse cycle Term (sec) = 2^{char}/Pck1 frequency (MHz)

Pck1: Peripheral clock 1

Chattering: An unstable state in which ON and OFF repeatedly occur immediately after the switching of the card identification signal.

Table 31.7 List of Chattering Elimination Pulse Cycles

chat	Decimal Notation	2^{chat}	Chattering Elimination Pulse Cycle Pck1 = 33 MHz
01001	(9)	512	0.015 (ms)
01010	(10)	1024	0.031 (ms)
01011	(11)	2048	0.061 (ms)
01100	(12)	4096	0.123 (ms)
01101	(13)	8192	0.246 (ms)
01110	(14)	16384	0.492 (ms)
01111	(15)	32768	0.983 (ms)
10000	(16)	65536	1.966 (ms)
10001	(17)	131072	3.932 (ms)
10010	(18)	262144	7.864 (ms)
10011	(19)	524288	15.729 (ms)
10100	(20)	1048576	31.457 (ms)
10101	(21)	2097152	62.915 (ms)
10110	(22)	4194304	125.829 (ms)
10111	(23)	8388608	251.658 (ms)

31.4 Operation

The multimedia card is an external storage media that can be easily connected or disconnected. The MMCIF controls data transfer with the multimedia card and operates in MMC mode.

Insert a card and supply power to it. Then operate the MMCIF by applying the transfer clock after setting an appropriate transfer clock frequency. In this case, MMC_VDDON and $\overline{\text{MMC_ODMOD}}$ are available for card power supply control and open-drain mode control, respectively.

A series of operations — starting with sending out commands, receiving a command response, data transmit/receive, receiving data responses, etc.— is called the command sequence. The command response starts with sending out commands by setting the START bit in CMDSTRT and ends with the completion of all data transmit/receive and responses. The multimedia card may go into the data busy state in which the card receives only special commands caused by writing to or erasing the flash memory in the card. The data busy state is indicated by a '0' output from the card through the DAT pin.

Notes: Do not connect or disconnect the card during command sequence execution or in the data busy state.

31.4.1 Operations in MMC Mode

MMC mode is an operating mode in which the transfer clock is output from the MMC_CLK pin, command transmission/response reception occurs via the MMC_CMD pin, and data is transmitted/received via the MMC_DAT pin. In this pin configuration, the next command can be issued while data is being transmitted or received.

This feature is utilized for multiblock or stream transfer. The CMD12 command is an example of where the current command sequence is aborted.

In MMC mode, broadcast commands that simultaneously issue commands to multiple cards are supported. After information from the inserted cards is recognized through a broadcast command, a relative address is given to each card. One card is selected by the relative address with other cards deselected, and then various commands are issued to the selected card.

Commands in MMC mode are basically classified into three types: broadcast, relative address, and flash memory operation commands. The card can be operated by issuing these commands appropriately according to the card state.

(1) Operation of Broadcast Commands

CMD0, CMD1, CMD2, and CMD4 are broadcast commands. These commands and the CMD3 command make up a sequence where relative addresses are assigned to individual cards. In this sequence, the CMD output format is open drain and the command response is wired-OR. The transfer clock frequency should be set low enough by dividing the transfer clock for the CLKON register by 28.

- All cards are initialized to the idle state by CMD0.
- The operation condition registers (OCR) of all cards are read via wired-OR and cards that cannot operate are deactivated by CMD1.
  The cards that are not deactivated enter the ready state.
- The card identification (CID) of each card in the ready state is read via wired-OR through CMD2.
  Each card compares its own CID and data on the CMD, and if they differ, the card immediately aborts the CID output. One card that has completed its output enters the acknowledge state. When the R2 response is necessary, set the CTOCR register to H'01.
- A relative address (RCA) is given to the card in the acknowledge state through CMD3.
  A card that can acquire an RCA enters the standby state.
- By repeating CMD2 and CMD3, an RCA is given to all cards in the ready state to put the cards in the standby state.

Note: When requesting the R2 response (a 17-byte command response) in MMC mode, set CTSEL0 to 1. Setting CTSEL0 to 0 causes a timeout while receiving the response.

(2) Operation of Relative Address Commands

CMD7, CMD9, CMD10, CMD13, CMD15, CMD39, and CMD55 are relative address commands that address the card through its RCA. The relative address commands are used to read card administration information and original information, and to change the specific card states.

CMD7 sets one addressed card to the transfer state, and the other cards to the standby state. Only the card in the transfer state can execute flash-memory operation commands, other than broadcast or relative-address commands.

(3) Operation of Commands Not Requiring Command Response

Some broadcast commands do not require a command response.

Figure 31.2 shows an example of the command sequence for commands that do not require a command response.

Figure 31.3 shows the operational flow for commands that do not require a command response.

- Create settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or through the command output end interrupt (CMDI).

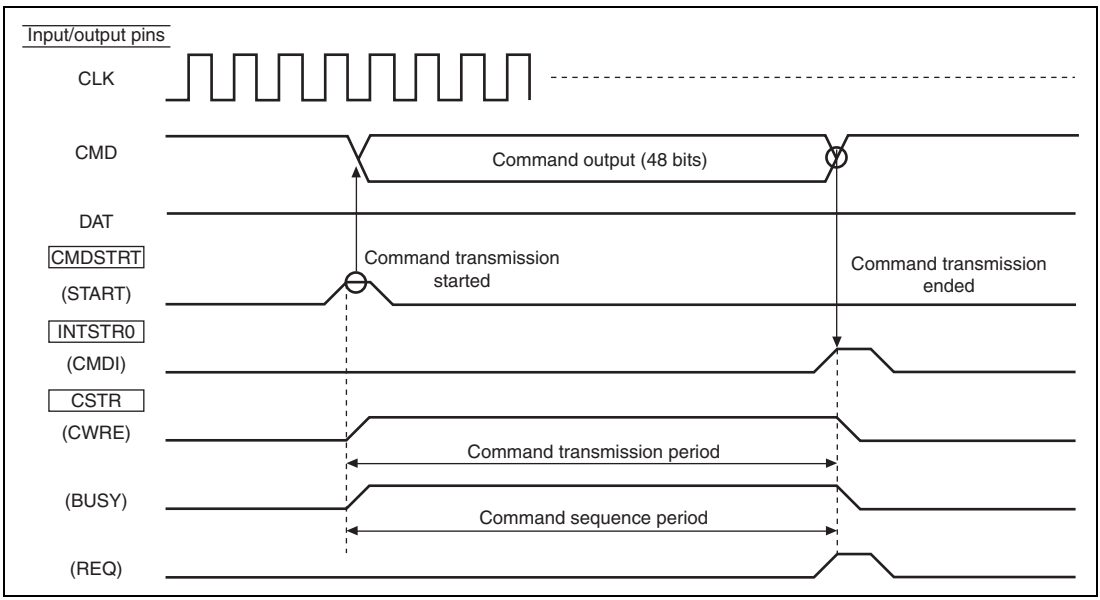


Figure 31.2 Example of Command Sequence for Commands Not Requiring Command Response

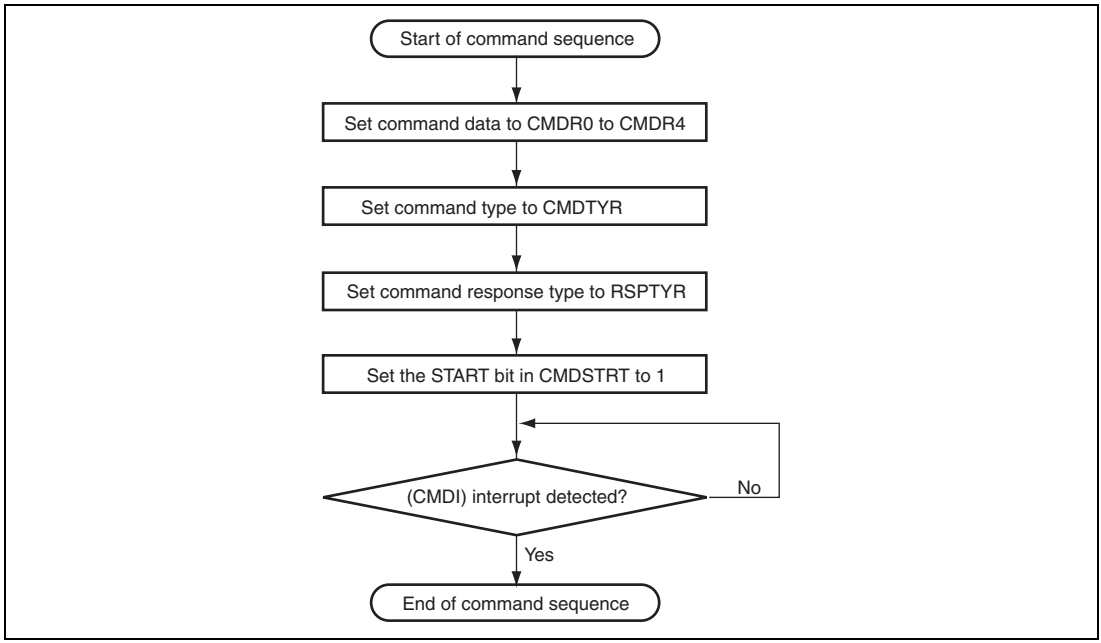


Figure 31.3 Example of Operational Flow for Commands Not Requiring Command Response

(4) Operation of Commands without Data Transfer

Broadcast, relative address, and flash memory operation commands include a number of commands that do not include data transfer. Such commands execute the desired data transfer using command arguments and command responses. For a command that is related to time-consuming processing such as flash memory write/erase, the card indicates the data busy state via the DAT.

Figures 31.4 and 31.5 show examples of the command sequence for commands without data transfer.

Figure 31.6 shows the operational flow for commands without data transfer.

- Create settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission.
- Command transmission completion can be confirmed through the command transmit end interrupt (CMDI).
- The command response is received from the card.

- If the card returns no command response, the command response is detected through the command timeout error (CTERI).
- The end of the command sequence is detected by poling the BUSY flag in CSTR or through the command response receive end interrupt (CRPI).
- Check whether the state is data busy or not through the DTBUSY_TU bit in CSTR. If data busy is detected, the end of data busy state is detected through the data busy end interrupt (DBSYI).
- Write the CMDOFF bit to 1 when a CRC error (CRCERI) or command timeout error (CTERI) occurs.

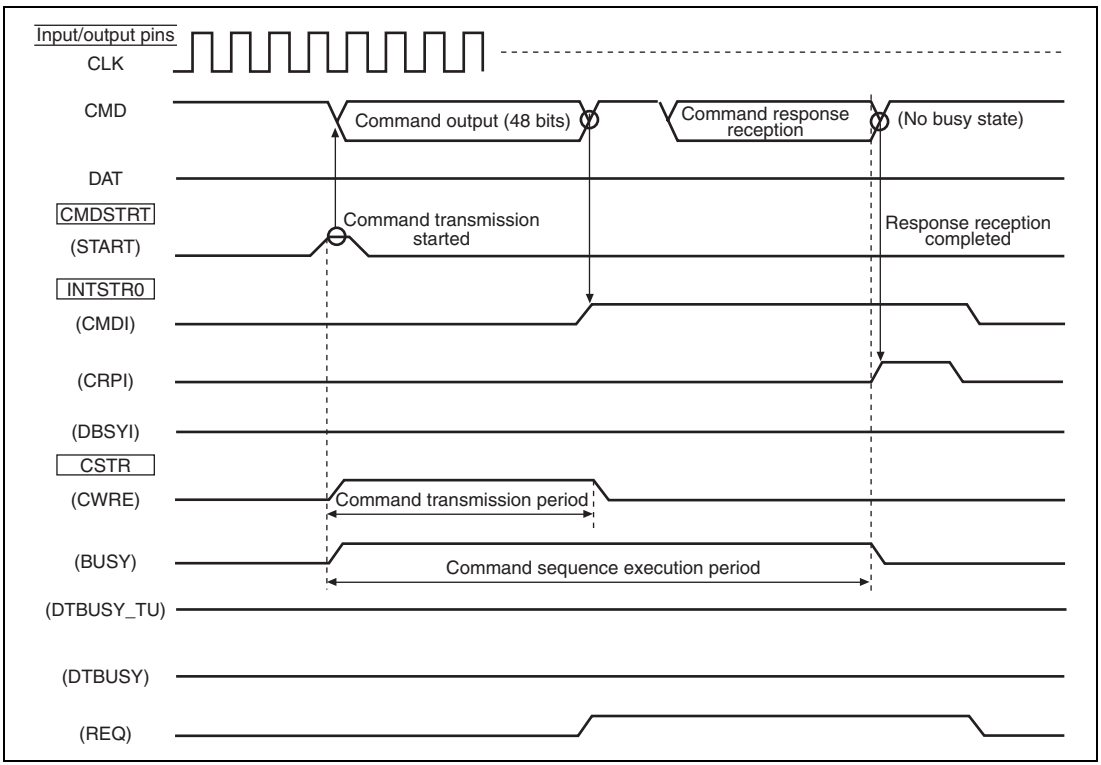


Figure 31.4 Example of Command Sequence for Commands without Data Transfer (No Data Busy State)

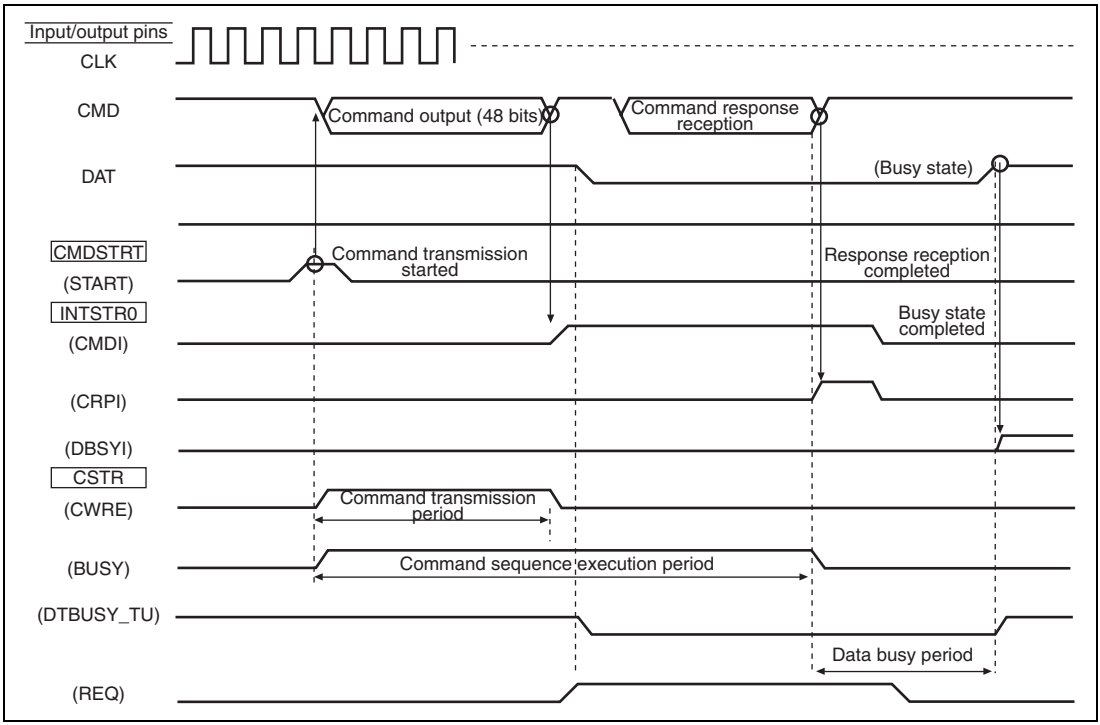


Figure 31.5 Example of Command Sequence for Commands without Data Transfer (with Data Busy State)

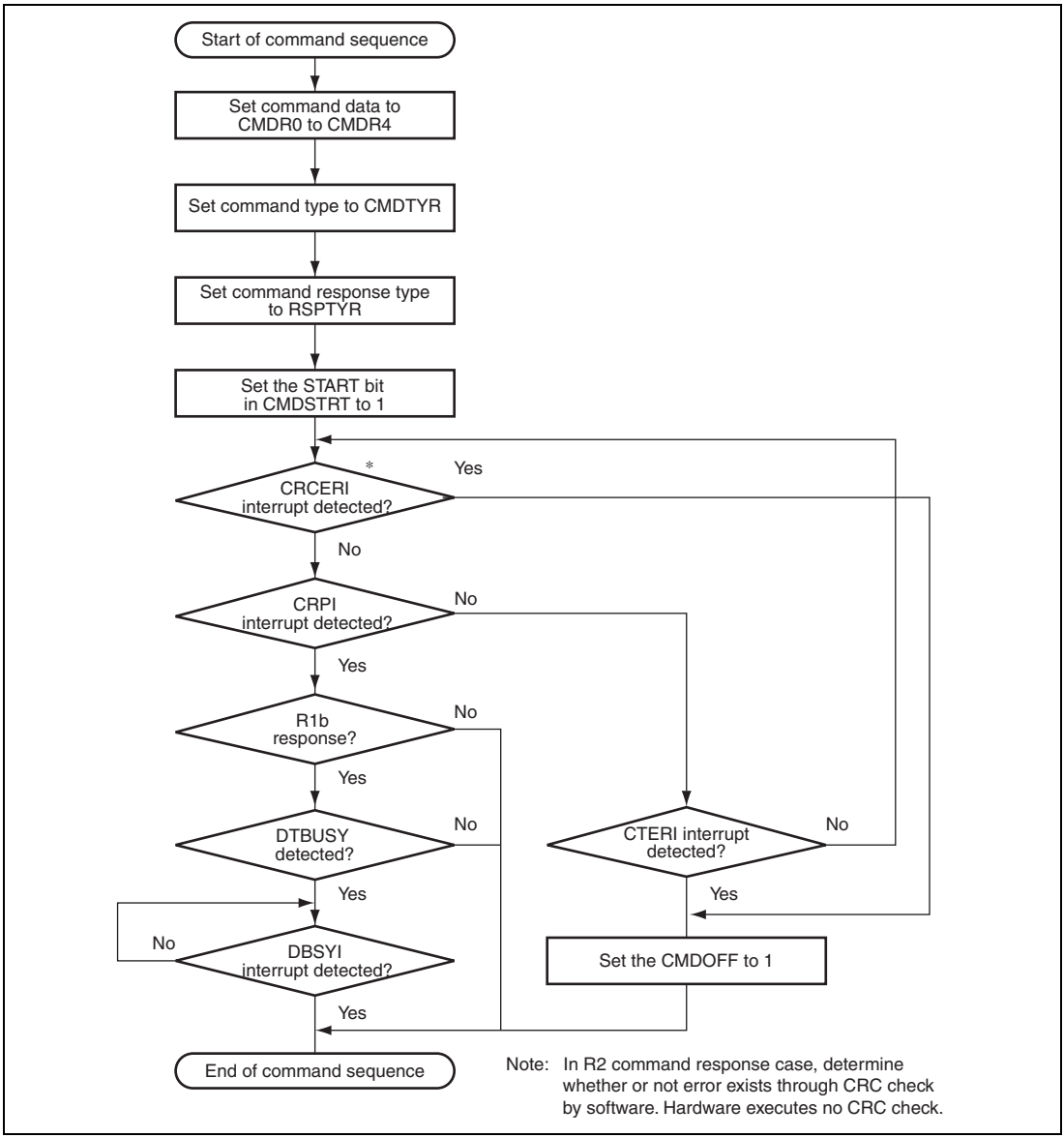


Figure 31.6 Example of Operational Flow for Commands without Data Transfer

(5) Commands with Read Data

Flash memory operation commands include a number of commands involving read data. Such commands confirm the card status through the command argument and command response, and receive card information and flash memory data from the MMC_DAT pin.

In multiblock transfer, there are two methods of transfer: the open-ended and pre-defined methods. The open-ended method suspends operation at each block transfer and waits for an instruction as to whether to continue the command sequence. The pre-defined method starts transferring with the block number set beforehand.

When the FIFO is full between blocks in multiblock transfer, the command sequence is suspended. Once the command sequence is suspended, any necessary processing of the data in FIFO must be done before the command sequence is continued.

Figures 31.7 to 31.9 show examples of the command sequence for commands with read data.

Figures 31.10 to 31.12 show the operational flows for commands with read data.

- Create settings to issue the command, and clear FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission.
- Command transmission completion can be confirmed through the command transmit end interrupt (CMDI).
- The command response is received from the card.
- If the card returns no command response, the command response is detected through the command timeout error (CTERI).
- Read data is received from the card.
- The inter-block suspension in multiblock transfer and suspension due to FIFO full are detected through the data transfer end interrupt (DTI) and FIFO full interrupt (FFI), respectively.
  To continue the command sequence, the RD_CONTI bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and CMD12 should be issued. Unless the sequence is suspended in a pre-defined multiblock transfer, CMD12 is not needed.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or the data transfer end interrupt (DTI).

- Write 1 to the CMDOFF bit when a CRC error (CRCERI) or command timeout error (CTERI) occurs in the command response reception.
- Clear the FIFO by writing 1 to the CMDOFF bit when a CRC error (CRCERI) or data timeout error (DTERI) occurs in the read data reception.

Note: In multiblock transfer, when the command sequence is ended (write 1 to the CMDOFF bit) before command response reception (CRPI), the command response may not be received correctly. Therefore, to receive the command response, the command sequence must be continued (set the RD_CONTI bit to 1) until the command response reception ends.

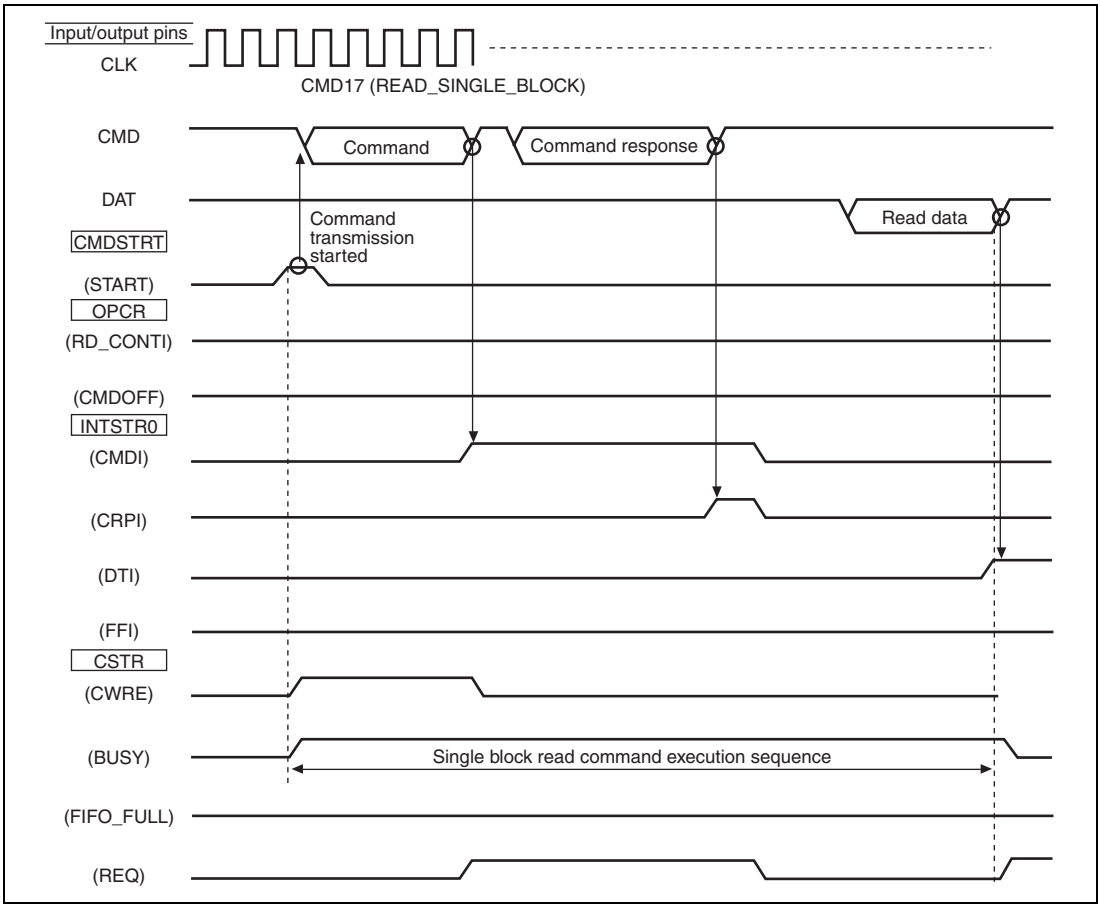


Figure 31.7 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size)

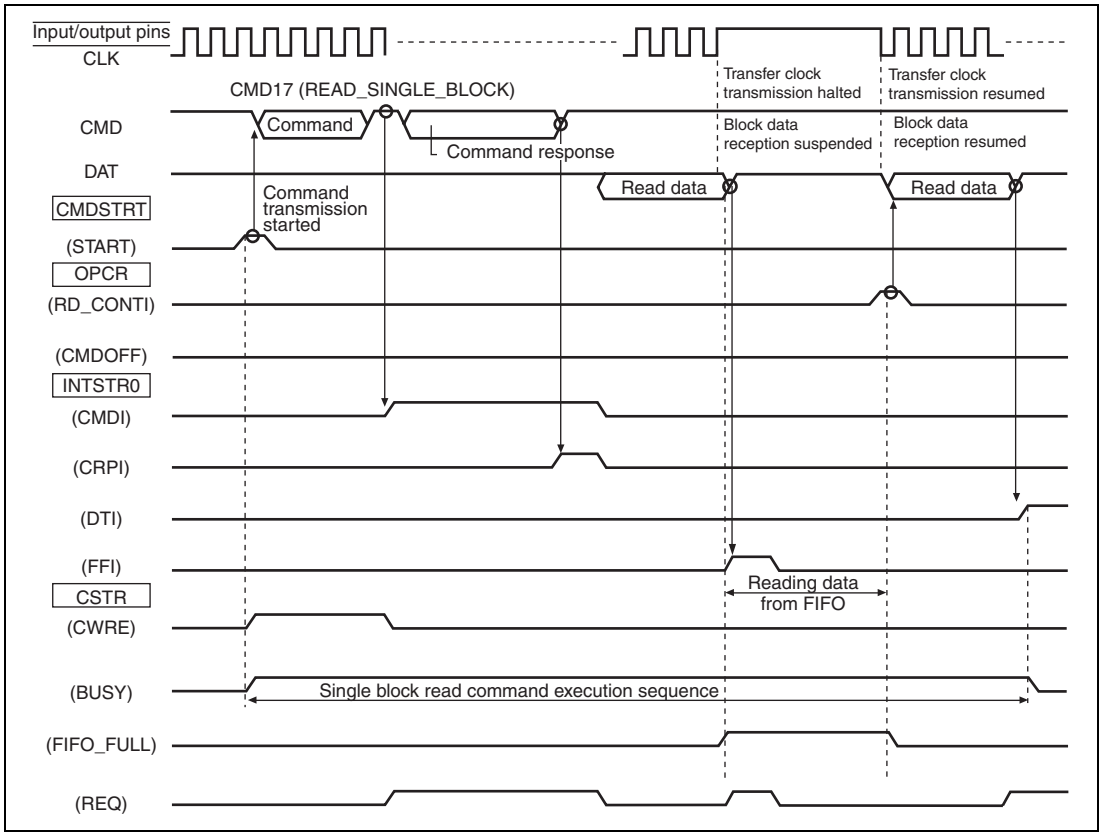


Figure 31.8 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size)

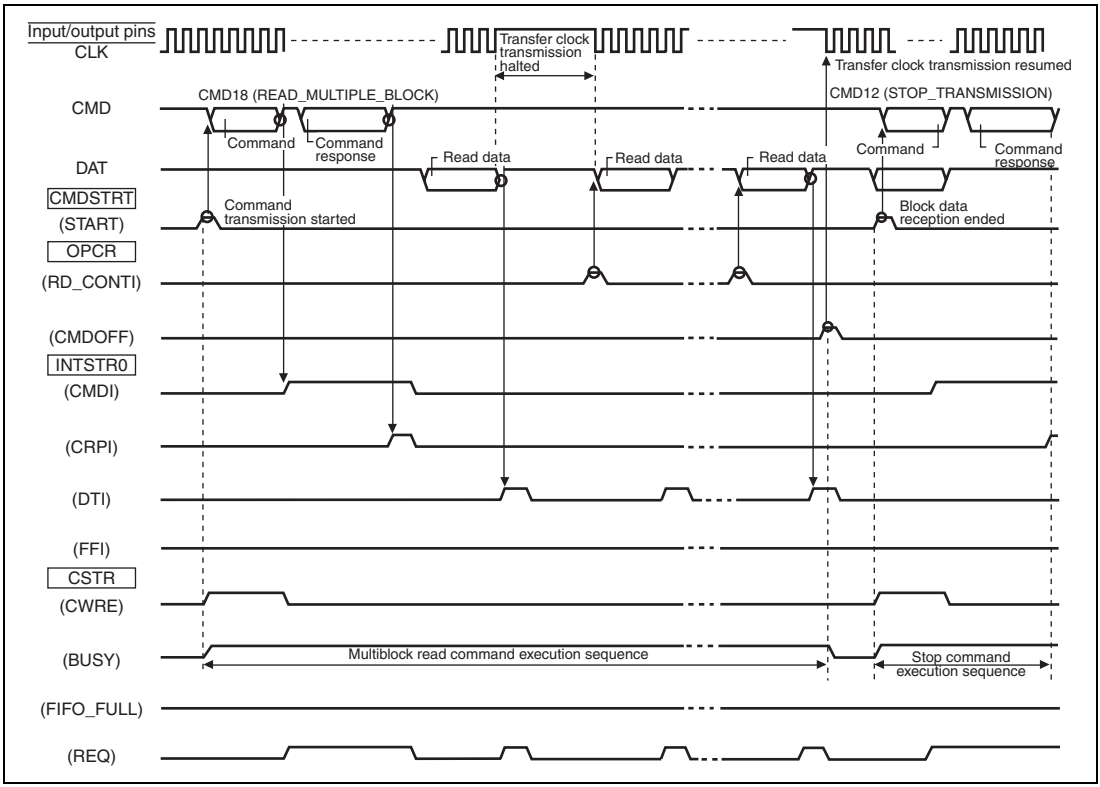


Figure 31.9 Example of Command Sequence for Commands with Read Data (Multiblock Transfer)

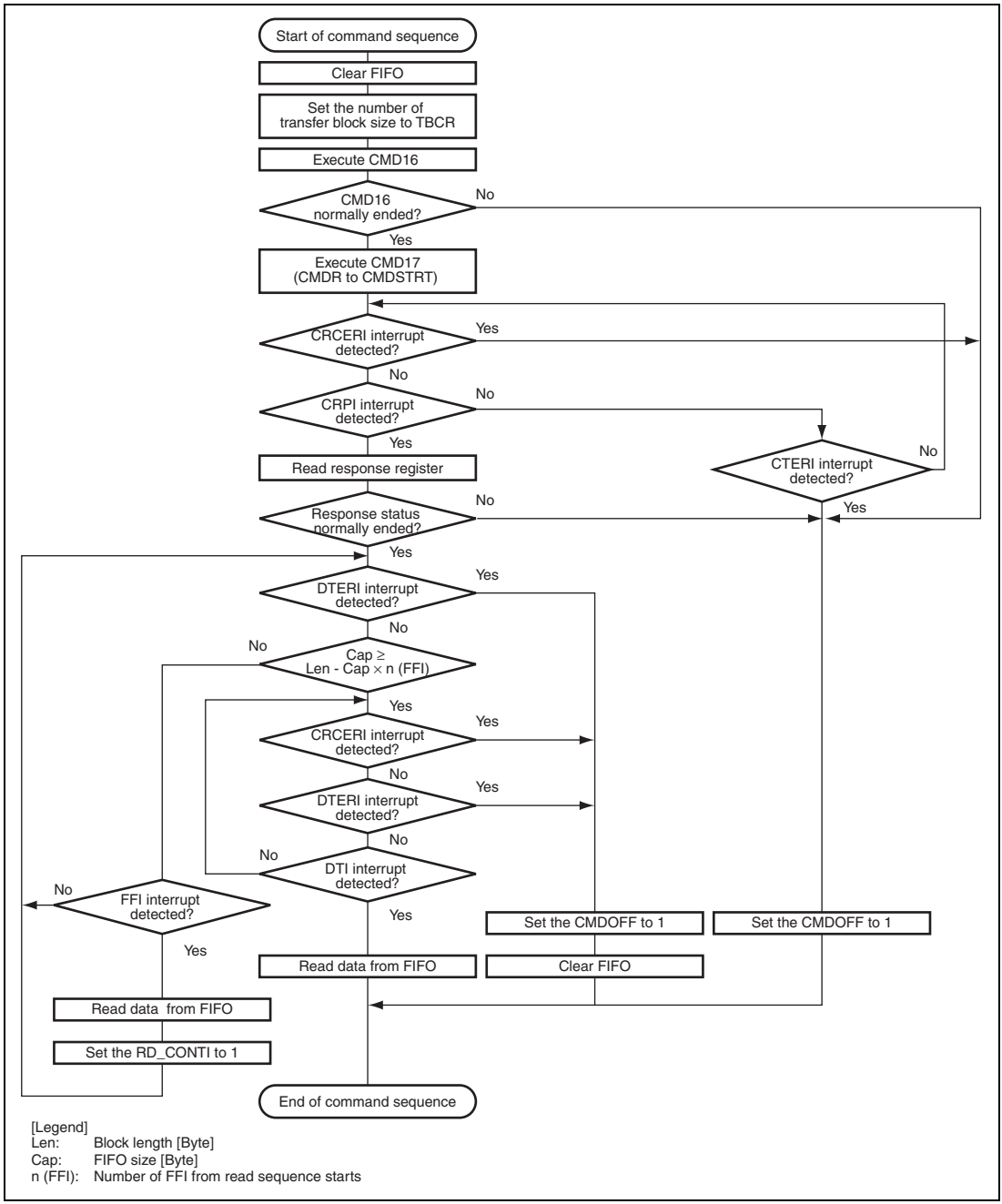


Figure 31.10 Example of Operational Flow for Commands with Read Data (Single Block Transfer)

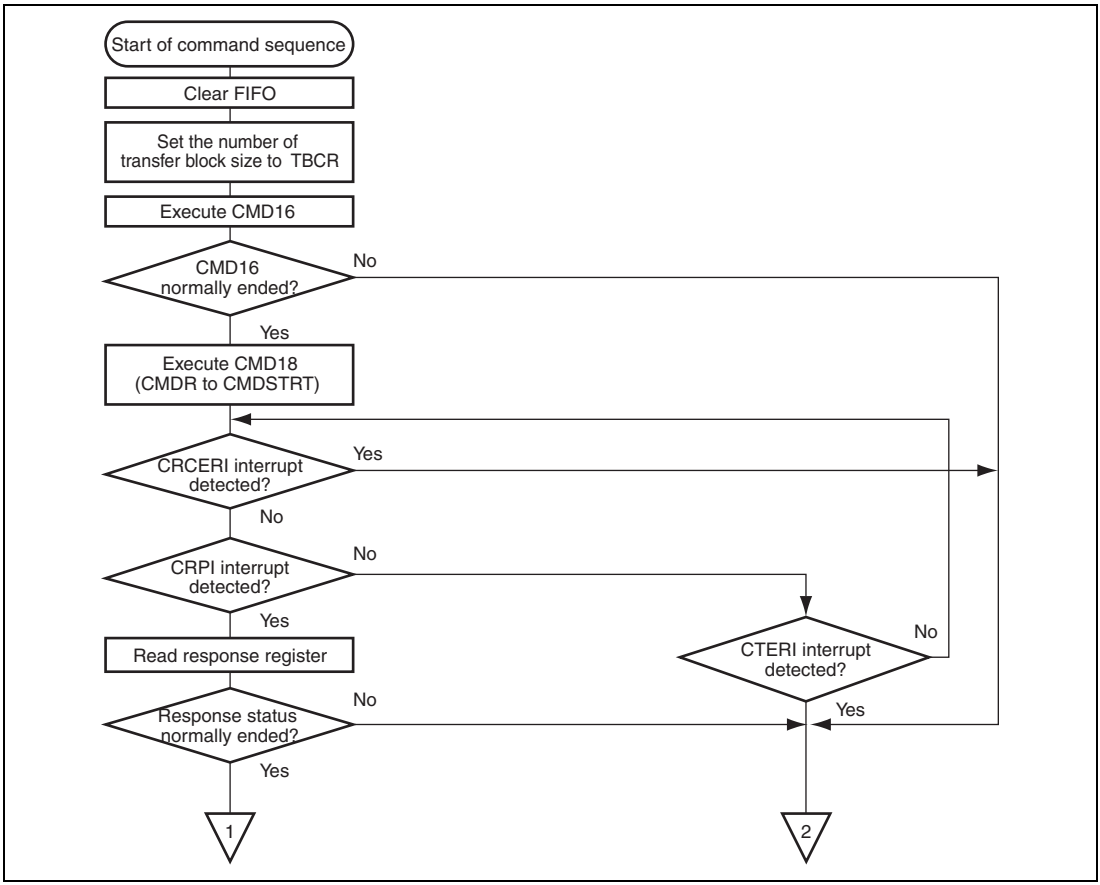


Figure 31.11 (1) Example of Operational Flow for Commands with Read Data (Open-ended Multiblock Transfer)

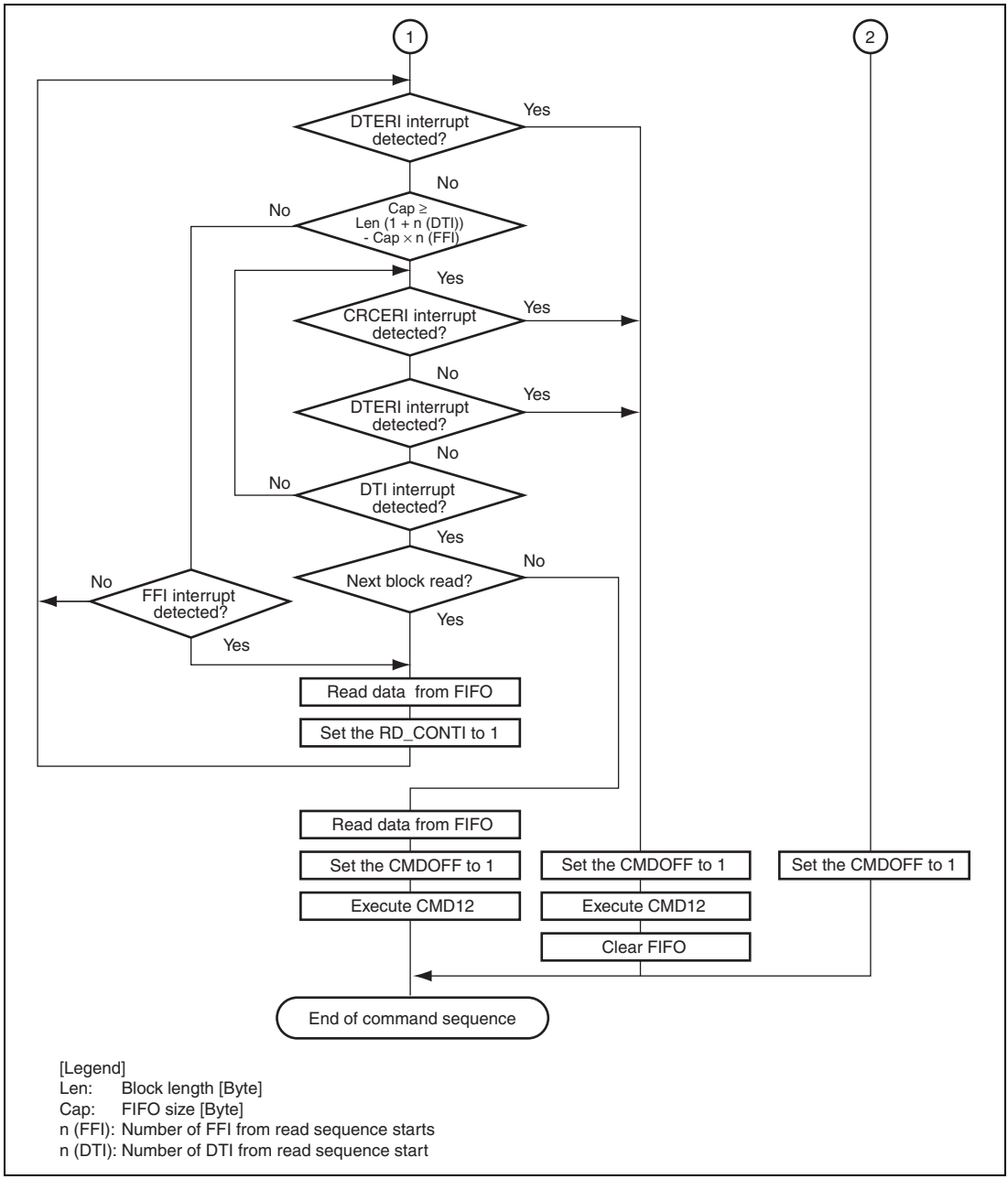


Figure 31.11 (2) Example of Operational Flow for Commands with Read Data (Open-ended Multiblock Transfer)

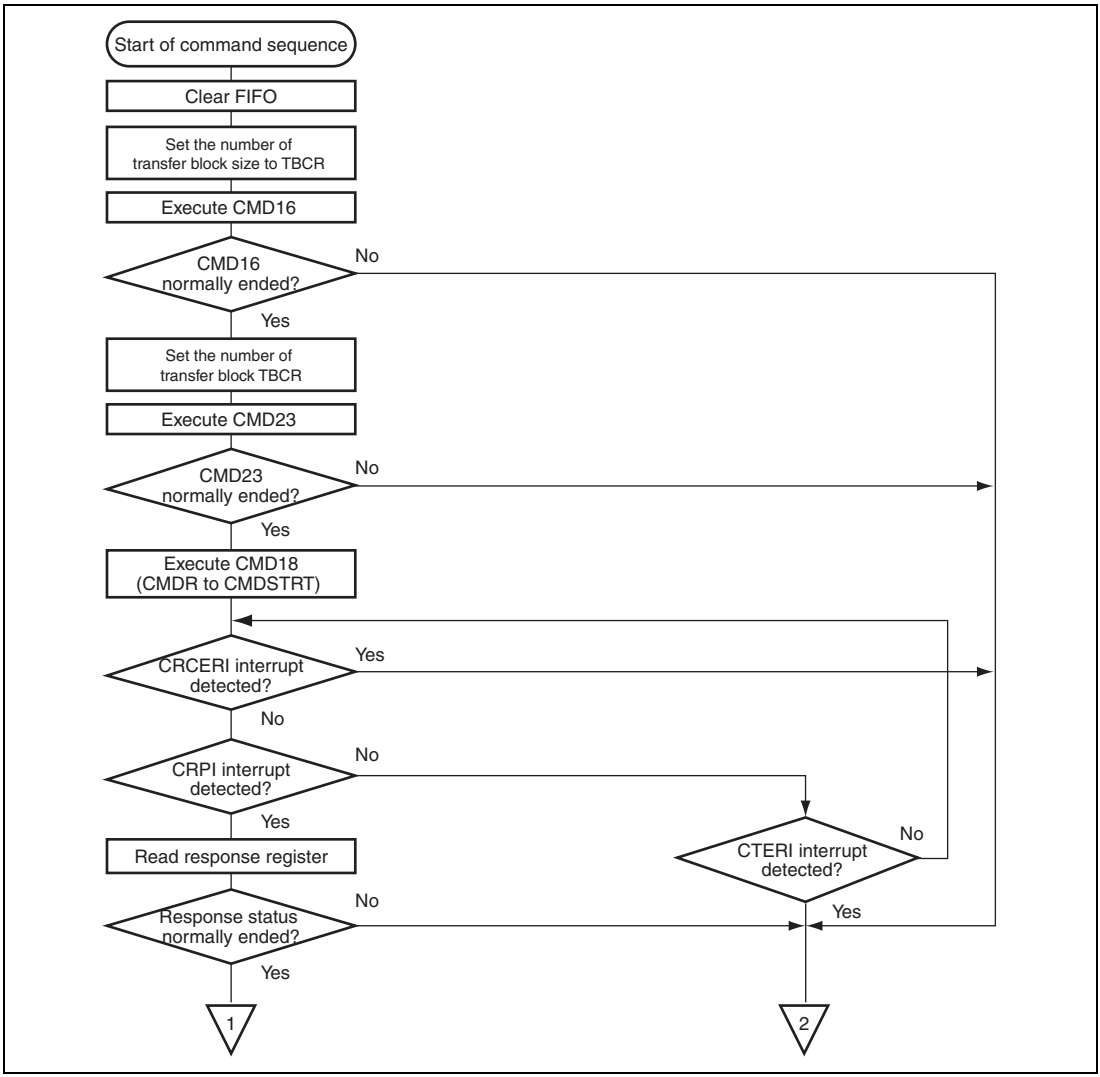


Figure 31.12 (1) Example of Operational Flow for Commands with Read Data (Pre-defined Multiblock Transfer)

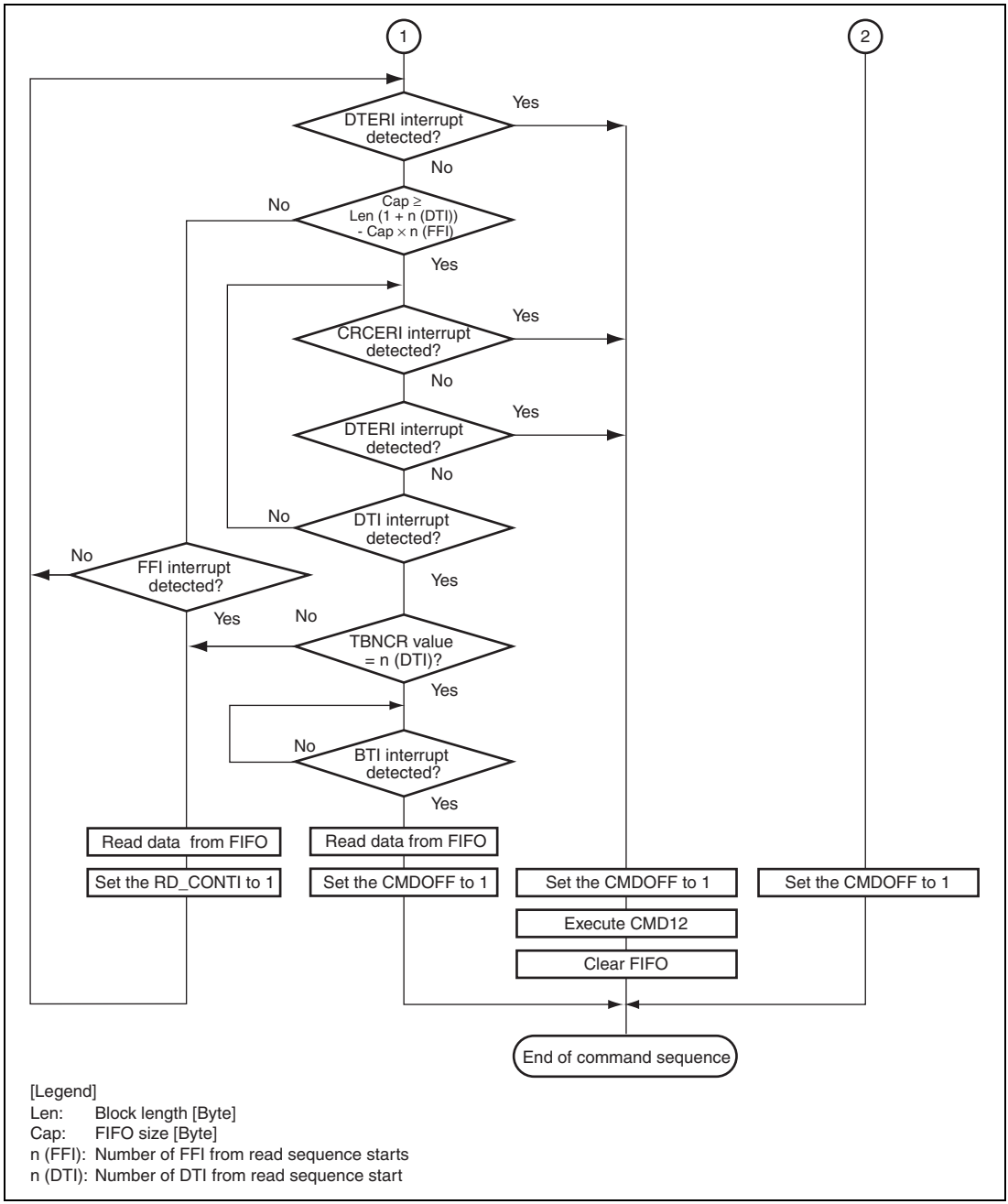


Figure 31.12 (2) Example of Operational Flow for Commands with Read Data (Pre-defined Multiblock Transfer)

(6) Commands with Write Data

Flash memory operation commands include a number of commands involving write data. Such commands confirm the card status through the command argument and command response, and transmit card information and flash memory data via the MMC_DAT pin. For a command that is related to time-consuming processing such as flash memory write, the card indicates the data busy state to the MMC_DAT pin.

In multiblock transfer, there are two methods of transfer: the open-ended and pre-defined methods. The open-ended method suspends operation at each block transfer and waits for an instruction as to whether to continue the command sequence. The pre-defined method starts transferring with the block number set beforehand.

When the FIFO is empty between blocks in multiblock transfer, the command sequence is suspended. Once the command sequence is suspended, any necessary processing of data in FIFO must be done before the command sequence is continued.

Figures 31.13 to 31.15 show examples of the command sequence for commands with write data.

Figures 31.16 to 31.18 show the operational flows for commands with write data.

- Create settings to issue a command, and clear FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission.
- The command response is received from the card.
- If the card returns no command response, the command response is detected through the command timeout error (CTERI).
- Set the write data to FIFO.
- Set the DATAEN bit in OPCR to 1 to start write data transmission.
- Inter-block suspension in multiblock transfer and suspension by the FIFO empty are detected through the data response completion flag (DRPI) and FIFO empty flag (FEI), respectively. To continue the command sequence, fill FIFO with write data and set the DATAEN bit in OPCR to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1 and CMD12 should be issued. Unless the sequence is suspended in pre-defined multiblock transfer, CMD12 is not needed.
- The end of the command sequence is detected by poling the BUSY flag in CSTR, data response completion flag (DRPI), or pre-defined multiblock transfer completion (BTI).
- After the data transfer end (after DPRI detection), the data busy state is checked through DTBUSY in CSTR. If the card is in the data busy state, the release of the data busy state is detected through the data busy end flag (DBSYI).

- Write the CMDOFF bit to 1 when a CRC error (CRCERI) or command timeout error (CTERI) occurs in the command response reception.
- Write the CMDOFF bit to 1 when a CRC error (CRCERI) or data timeout error (DTERI) occurs in the write data transmission.

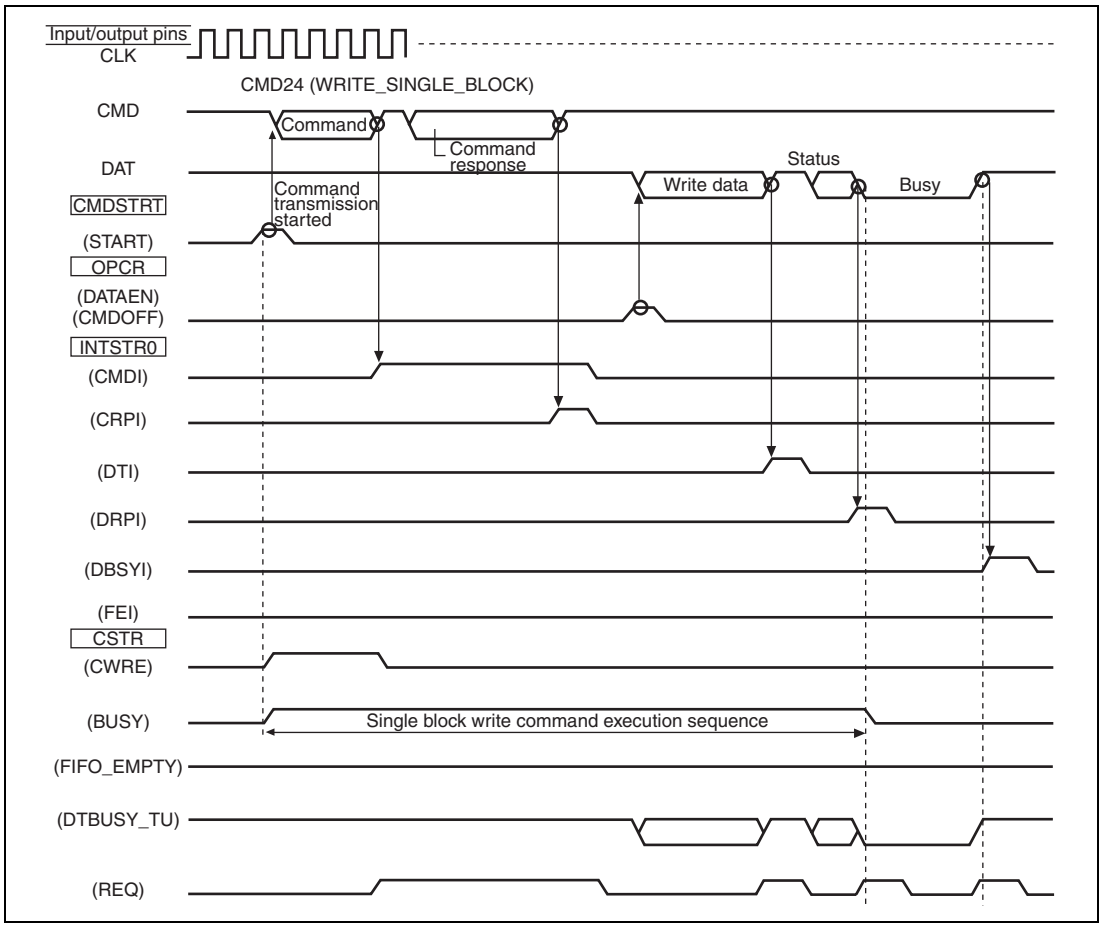


Figure 31.13 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size)

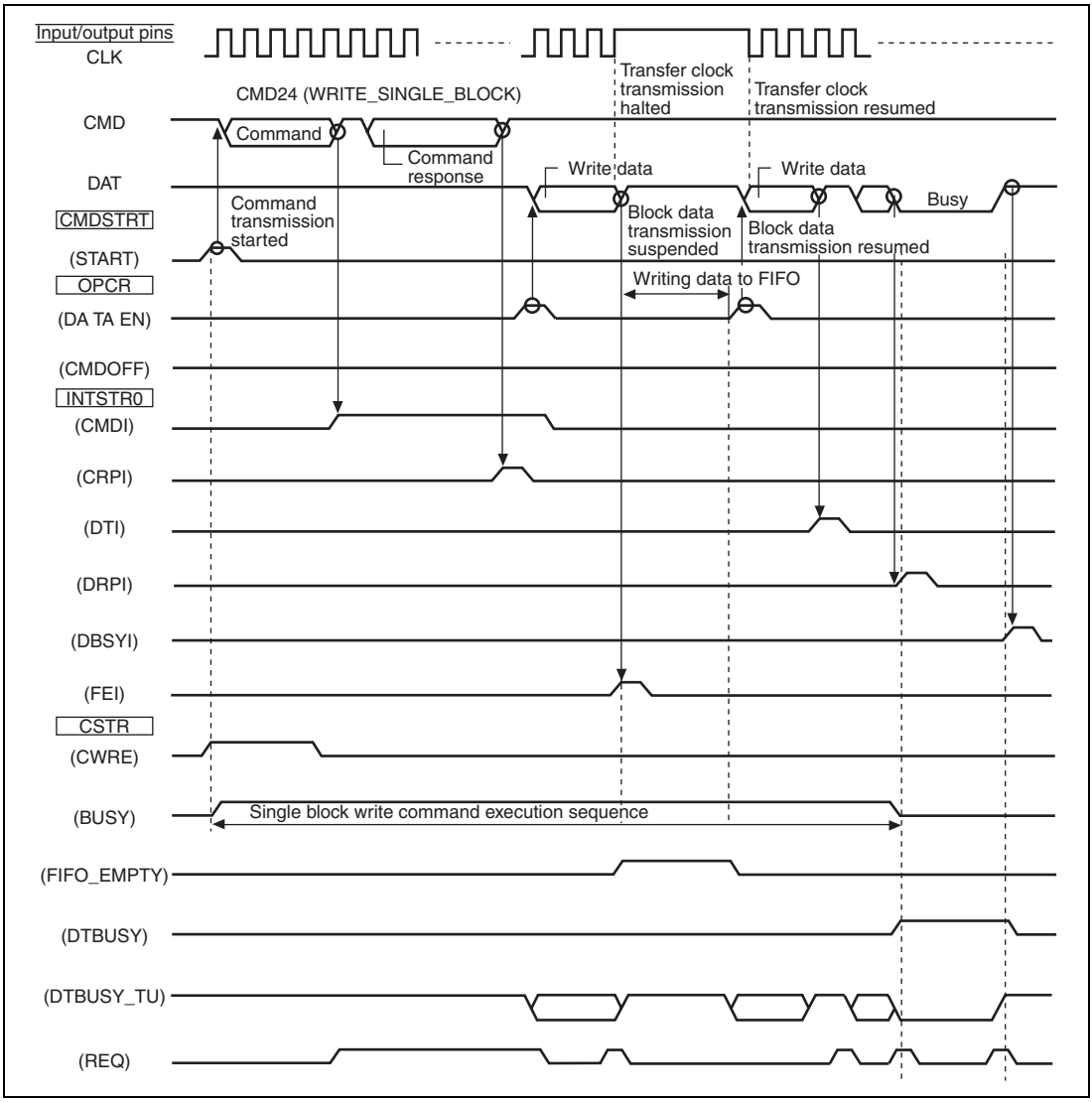


Figure 31.14 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size)

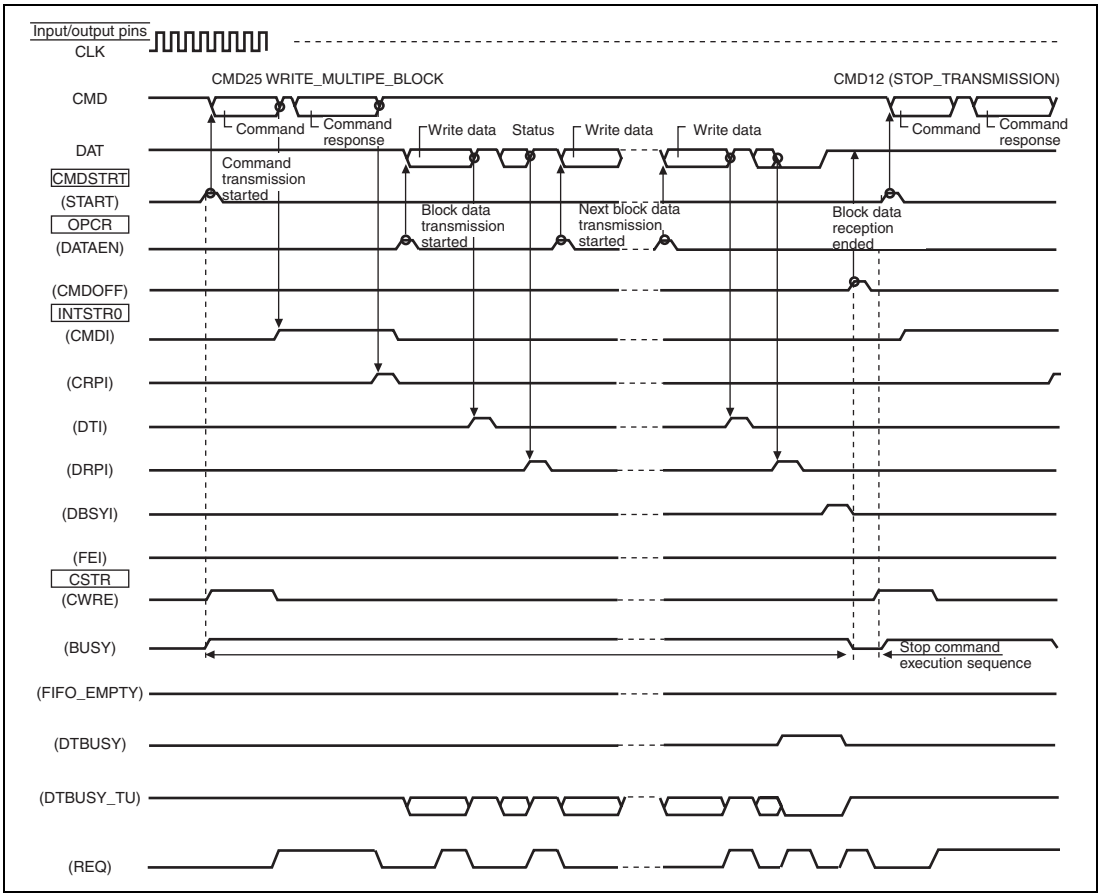


Figure 31.15 Example of Command Sequence for Commands with Write Data (Multiblock Transfer)

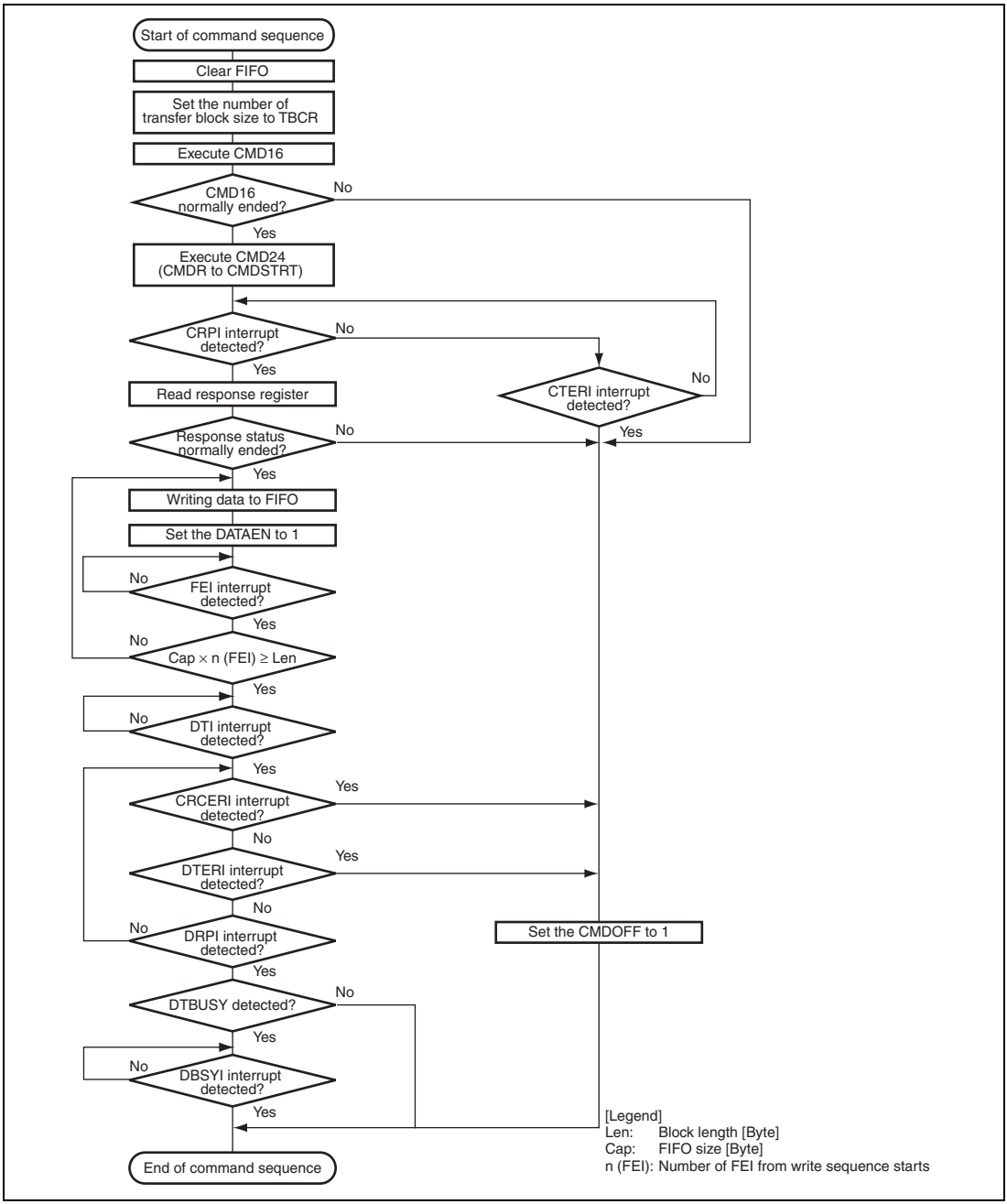


Figure 31.16 Example of Operational Flow for Commands with Write Data (Single Block Transfer)

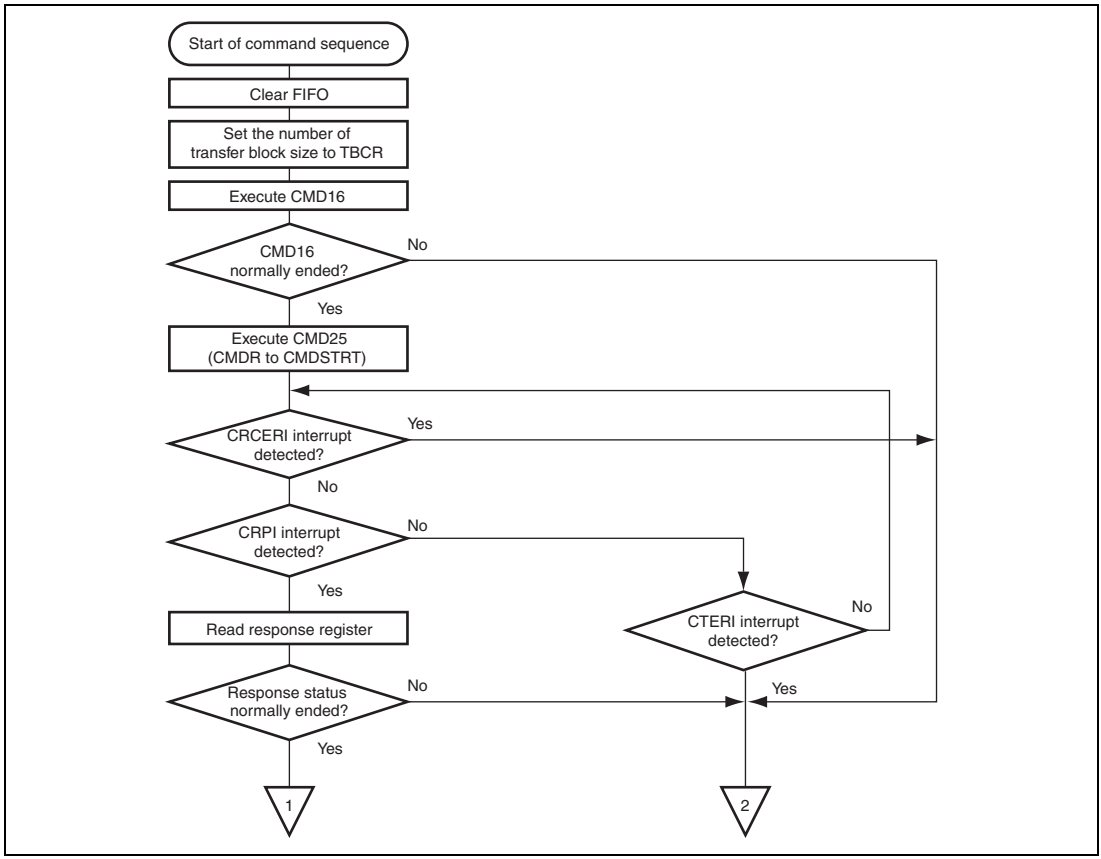
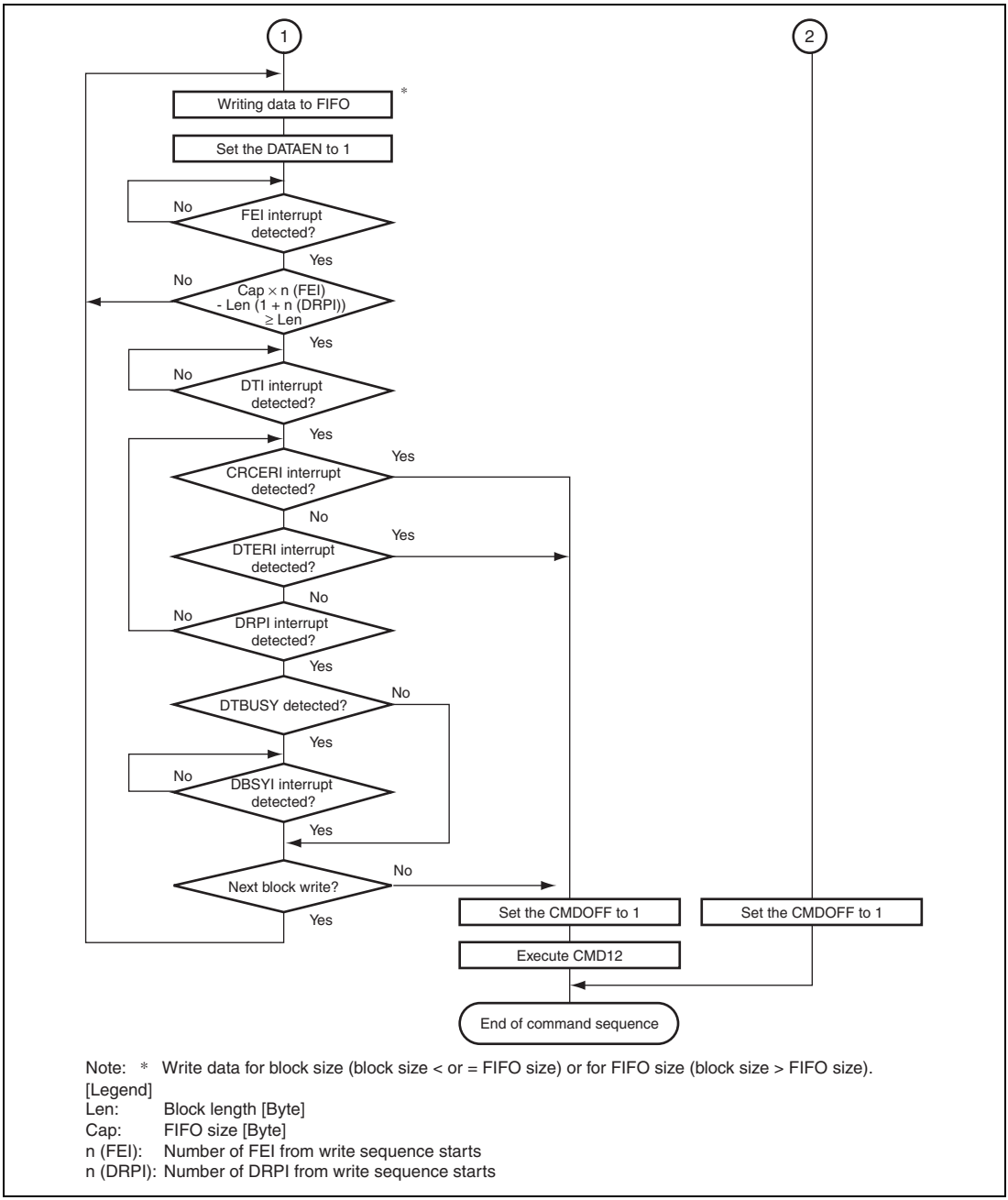


Figure 31.17 (1) Example of Operational Flow for Commands with Write Data (Open-ended Multiblock Transfer)



Note: * Write data for block size (block size < or = FIFO size) or for FIFO size (block size > FIFO size).
[Legend]
Len: Block length [Byte]
Cap: FIFO size [Byte]
n (FEI): Number of FEI from write sequence starts
n (DRPI): Number of DRPI from write sequence starts

Figure 31.17 (2) Example of Operational Flow for Commands with Write Data (Open-ended Multiblock Transfer)

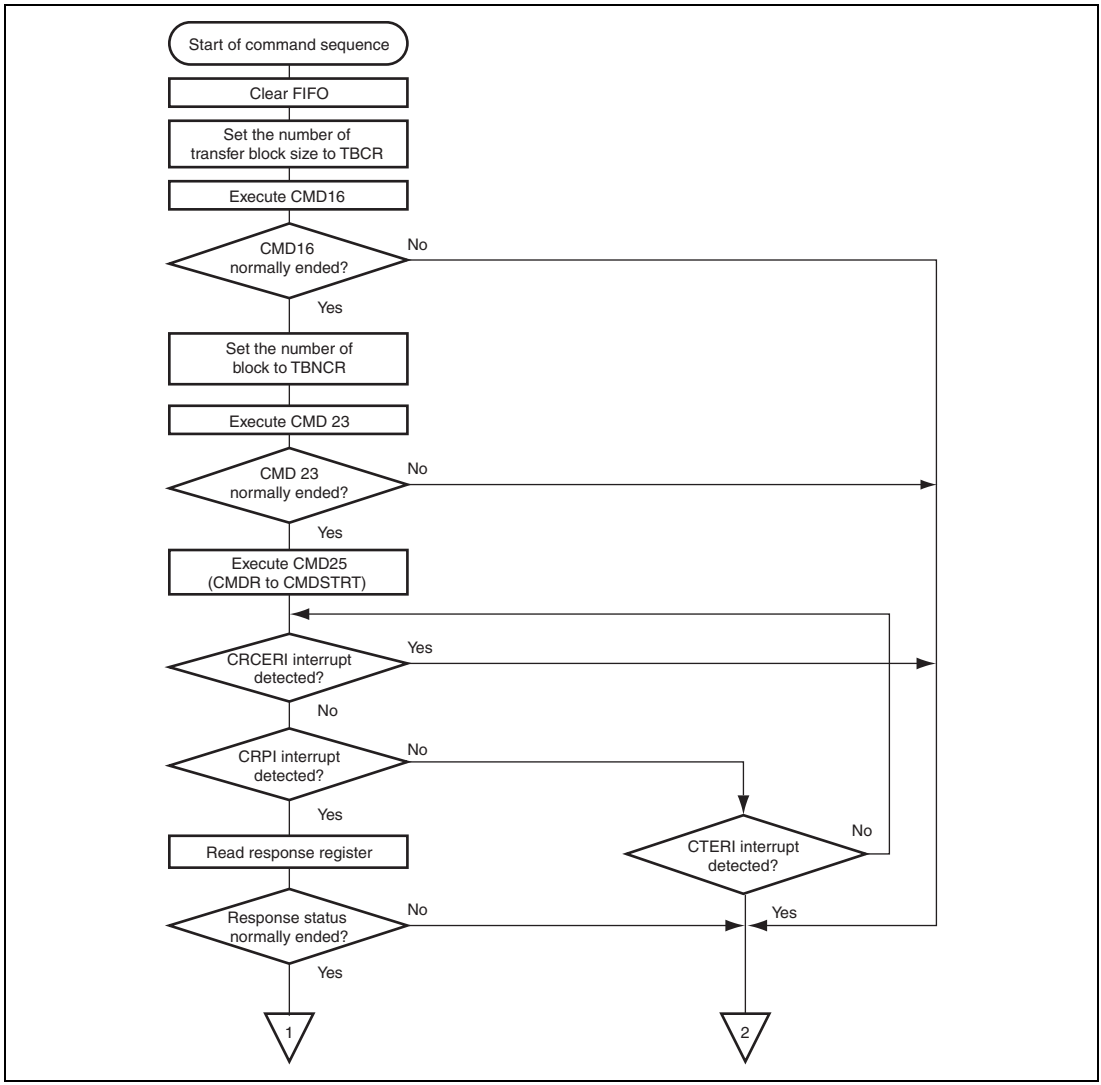
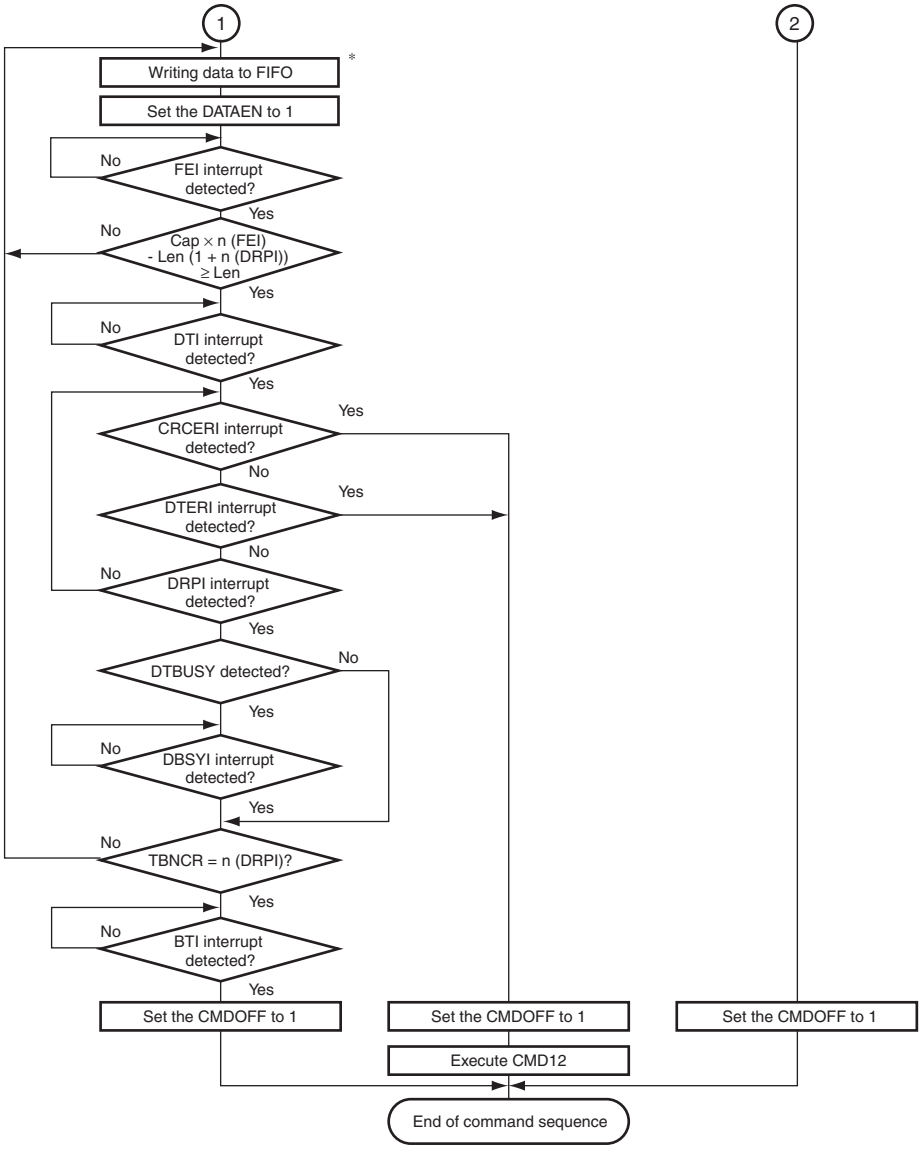


Figure 31.18 (1) Example of Operational Flow for Commands with Write Data (Pre-defined Multiblock Transfer)



Note: * Write data for block size (block size < or = FIFO size) or for FIFO size (block size > FIFO size).

[Legend]

Len: Block length [Byte]
Cap: FIFO size [Byte]
n (FEI): Number of FEI from write sequence starts
n (DRPI): Number of DRPI from write sequence starts

Figure 31.18 (2) Example of Operational Flow for Commands with Write Data (Pre-defined Multiblock Transfer)

31.5 Operations when Using DMAC

31.5.1 Operation in Read Sequence

To transfer data using the DMAC, set MMCIF (DMACR) after setting the DMAC. Transmit the read command after setting DMACR.

When using DMA, next block read is resumed automatically when the AUTO bit in DMACR is set to 1 under the condition that normal read is detected after a block transfer end of a pre-defined multiblock transfer. Figure 31.19 shows an example of the operational flow for a pre-defined multiblock read in MMC mode using auto-mode.

- Clear FIFO.
- Set the block number to TBNCR.
- Set DMACR.
- Read command transmission is started.
- Command response and read data are received from card.
- When the card does not return the command response, the command response is detected by the command timeout error (CTERI).
- The end of the command sequence is detected by poling the BUSY flag in CSTR or through the pre-defined multiblock transfer end flag (BTI).
- An error in a command sequence (during data reception) is detected through the CRC error flag or data timeout flag. When these flags are detected, set the CMDOFF bit in OPCR to 1, issue CMD12, and suspend the command sequence.
- The data remains in FIFO after the read sequence end. Set the SET[2:0] bits in DMACR to 100 to read all data left in FIFO if necessary.
- Confirm the DMAC transfer completion and clear the DMAEN bit in DMACR to 0.
- Set the CMDOFF bit to 1 and clear DMACR to H'00 when a CRC error (CRCERI) or command timeout error (CTERI) occurs in the command response reception.
- Set the CMDOFF bit to 1, clear DMACR to H'00, and clear FIFO when a CRC error (CRCERI) or data timeout error (DTERI) occurs in the read data reception.

Note: * In multiblock transfer, when the command sequence is ended (1 is written to the CMDOFF bit) before command response reception ends (CRPI), the command response may not be received correctly. Therefore, to receive the command response, the command sequence must be continued (set the RD_CONTI bit to 1) until the command response reception ends.
Access from the DMAC to FIFO must be done in bytes or words.

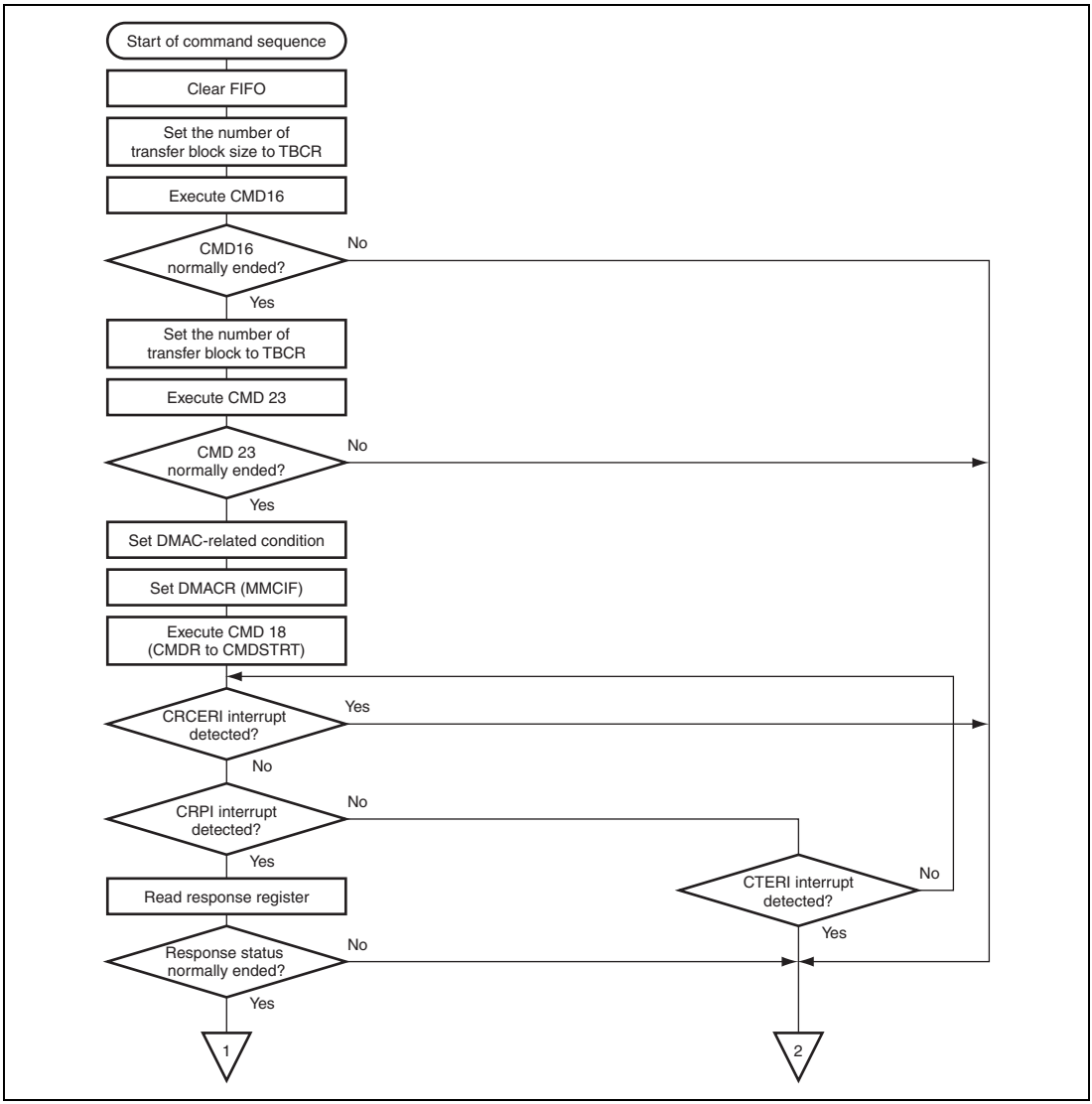


Figure 31.19 (1) Example of Operational Flow for Auto-mode Pre-defined Multiblock Read Transfer

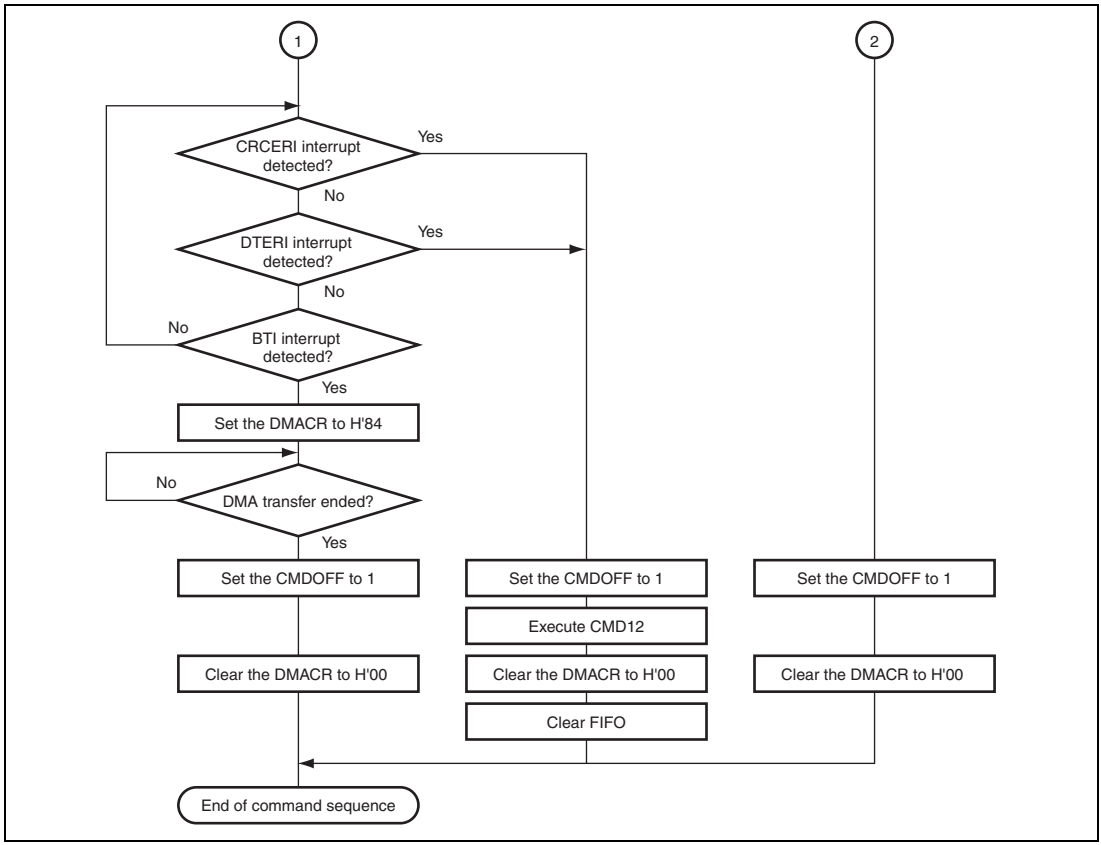


Figure 31.19 (2) Example of Operational Flow for Auto-mode Pre-defined Multiblock Read Transfer

31.5.2 Operation in Write Sequence

To transfer data using the DMAC, set MMCIF (DMACR) after setting the DMAC.

When using DMA, it is possible to process the inter-block interrupt by hardware in pre-defined multiblock transfer by setting the AUTO bit in DMACR to 1. Figure 31.20 shows an example of the operational flow for a pre-defined multiblock write sequence in MMC mode using auto-mode.

- Clear FIFO.
- Set the block number to TBNCR.
- Set the START bit in CMDSTRT to 1 and command transmission will begin.
- Command response is received from the card.
- If the card does not return the command response, the command response is detected through the command timeout error (CTERI).
- Set the DMACR and write data in FIFO.
- Confirm the DMAC transfer completion and clear the DMAEN bit in DMACR to 0.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or through the pre-defined multiblock transfer end flag (BTI).
- An error in a command sequence (during data transmission) is detected through the CRC error flag (CRCERI) or data timeout error flag. When these flags are detected, set the CMDOFF bit in OPCR to 1, issue CMD12, and suspend the command sequence.
- Confirm there is no data busy condition. Detect the data busy state through the data busy end flag (DBSYI).
- Detect whether the current state is the data busy state through the DTBUSY bit in CSTR after the data transfer end (after DRPI detection). If it is still the data busy state, use the DBSYI flag to confirm the end of the data busy condition.
- Set the CMDOFF bit to 1 and end the command sequence.
- Set the CMDOFF bit to 1 when a CRC error (CRCERI) or command timeout error (CTERI) occurs in the command response reception.
- Set the CMDOFF bit to 1, clear the DMACR to H'00, and clear FIFO when a CRC error (CRCERI) or data timeout error (DTERI) occurs in the write data transmission.

Note: Access from the DMAC to FIFO must be done in bytes or words.

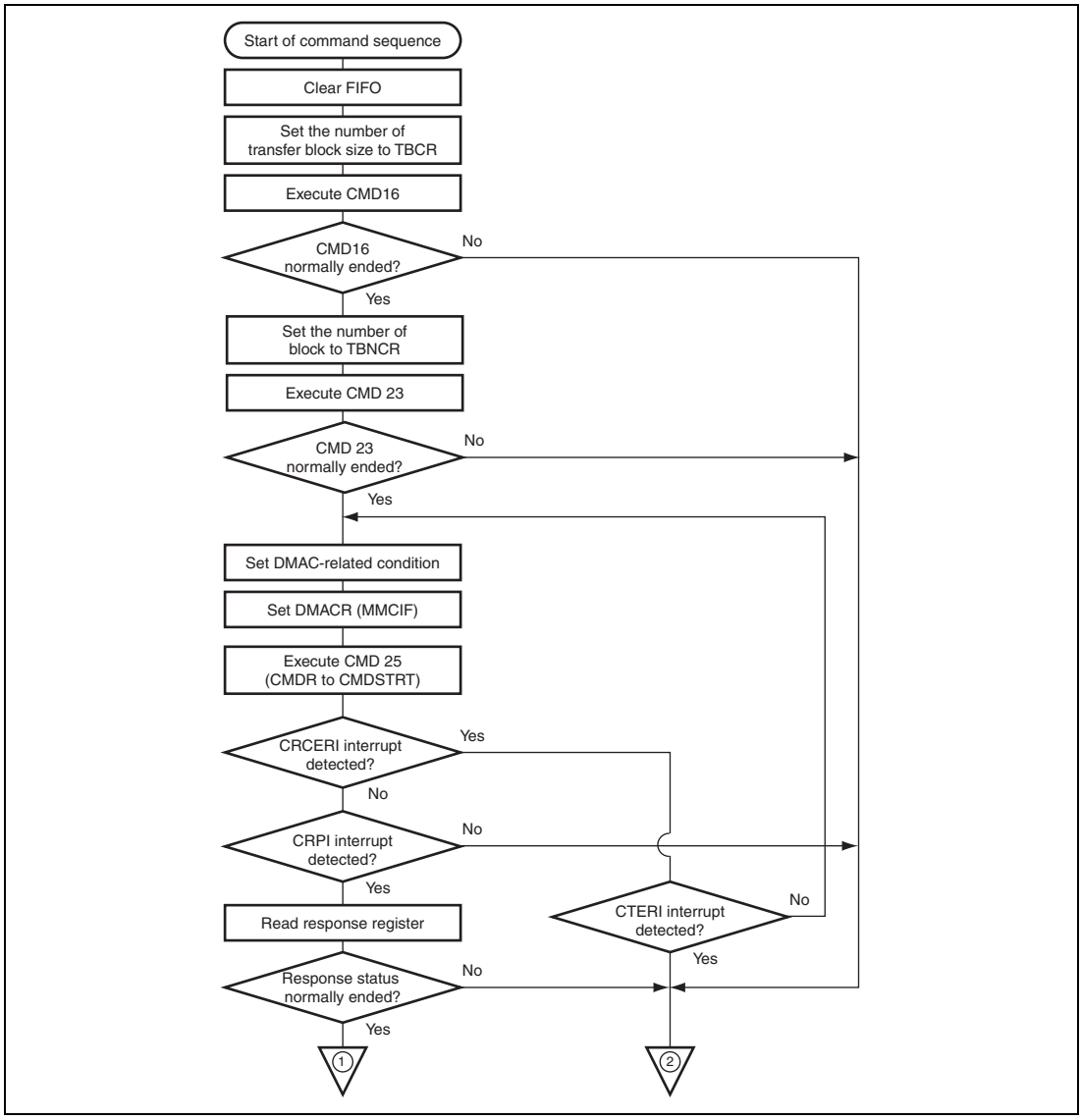


Figure 31.20 (1) Example of Operational Flow for Auto-mode Pre-defined Multiblock Write Transfer

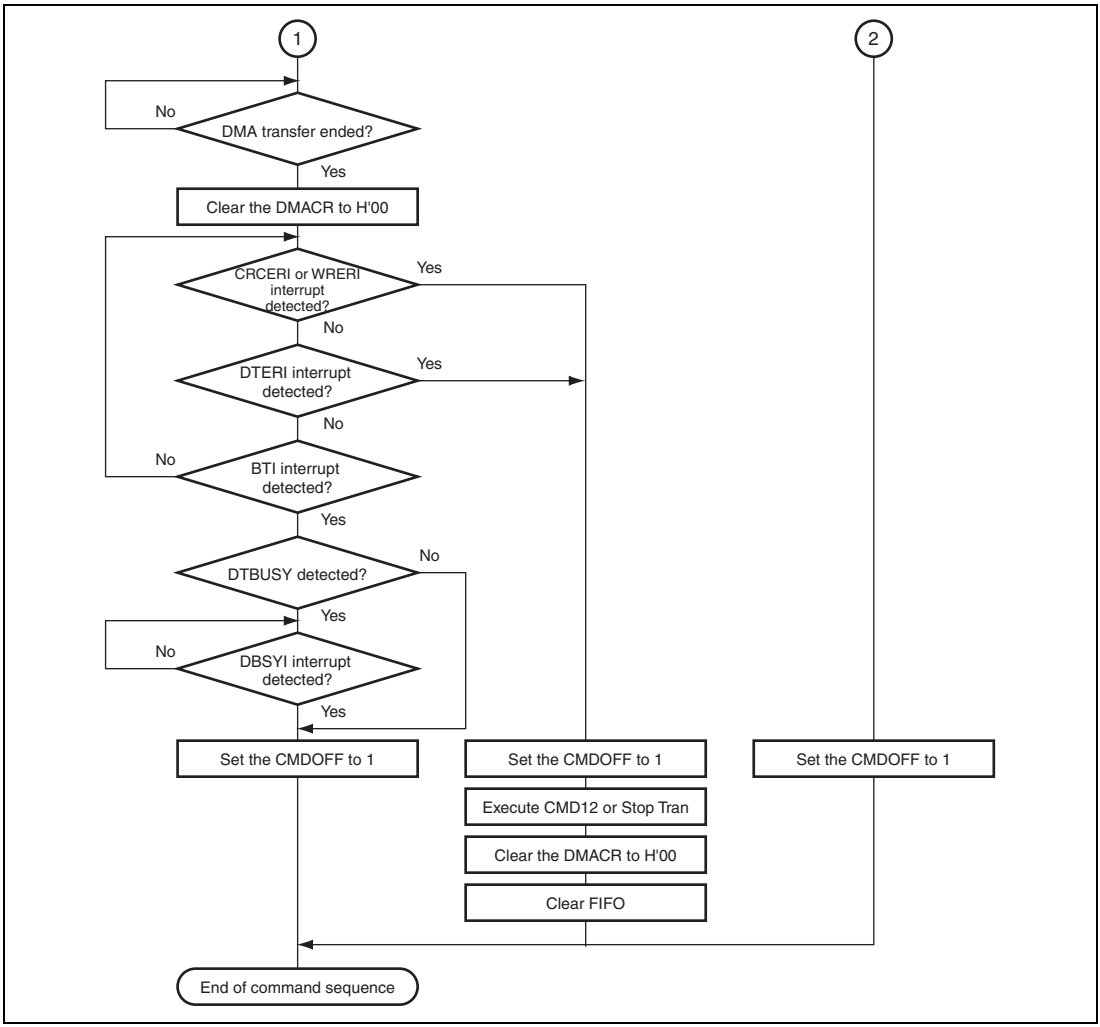


Figure 31.20 (2) Example of Operational Flow for Auto-mode Pre-defined Multiblock Write Transfer

31.6 MMCIF Interrupt Sources

Table 31.8 lists the MMCIF interrupt sources. The interrupt sources are classified into four groups, and four interrupt vectors are assigned. Each interrupt source can be individually enabled by the enable bits in INTCR0 to INTCR2. Disabled interrupt sources do not set a flag.

Table 31.8 MMCIF Interrupt Sources

Name	Interrupt source	Interrupt flag
ERR	Write error	WRERI
	CRC error*	CRCERI
	Data timeout error	DTERI
	Command timeout error	CTERI
FSTAT	FIFO empty	FEI
	FIFO full	FFI
TRAN	Data response	DRPI
	Data transfer end	DTI
	Command response end	CRPI
	Command output end	CMDI
	Data busy end	DBSYI
	Block transfer end	BTI
FRDY	FIFO ready	FRDYI
	Card identification	CDI

Note: * Excluding the CRC error in the R2 command response

31.7 Procedure to Apply the Card Detection Function

Figure 31.21 shows the operation flow to apply the card detection function.

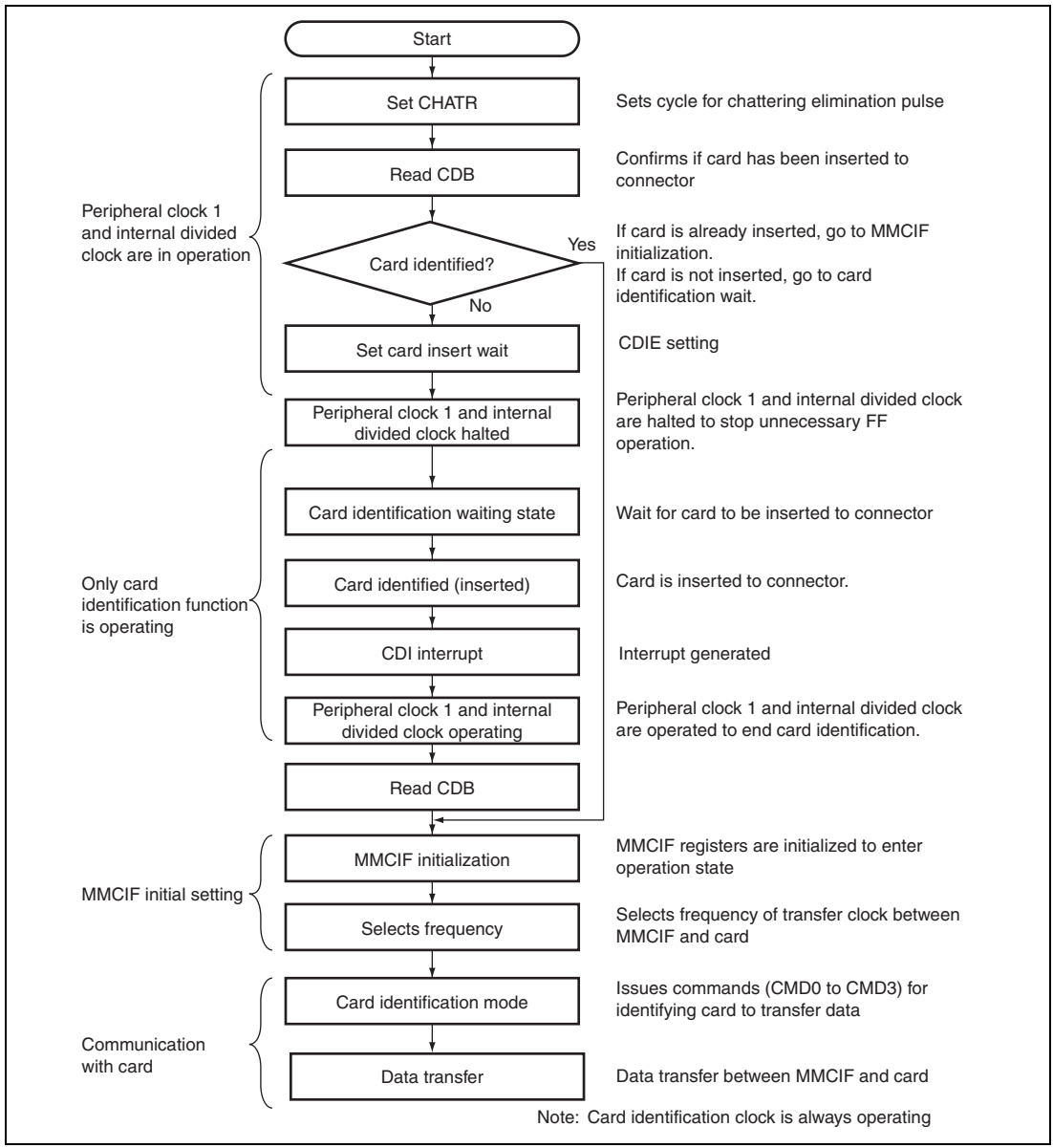


Figure 31.21 Operation Flow to Apply the Card Identification Function

Section 32 PC Card Controller (PCC)

The PC card controller (PCC) controls the external buffer, interrupts, and exclusive ports of the PC card interface to be connected to this LSI. Using the PCC enables two slots of PC cards that conform to the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard to be easily connected to this LSI.

32.1 Features

- As a PC card interface to be connected to physical area 6, an IC memory card interface and an I/O card interface are supported.
- Outputs control signals for the external buffer ($\overline{\text{PCC_DRV}}$).
- Supports a preemptive operating system by switching attribute memory, common memory, and I/O space by using addresses.
- Provides a segment bit (an address bit for the PC card) for common memory, enabling access to a 64-Mbyte space fully conforming to PCMCIA specifications.
- Disables the PCC operation and supports only a bus interface of a PC card interface (by using the P0USE bit of PCC0GCR).

Figure 32.1 shows a block diagram of the PC card controller.

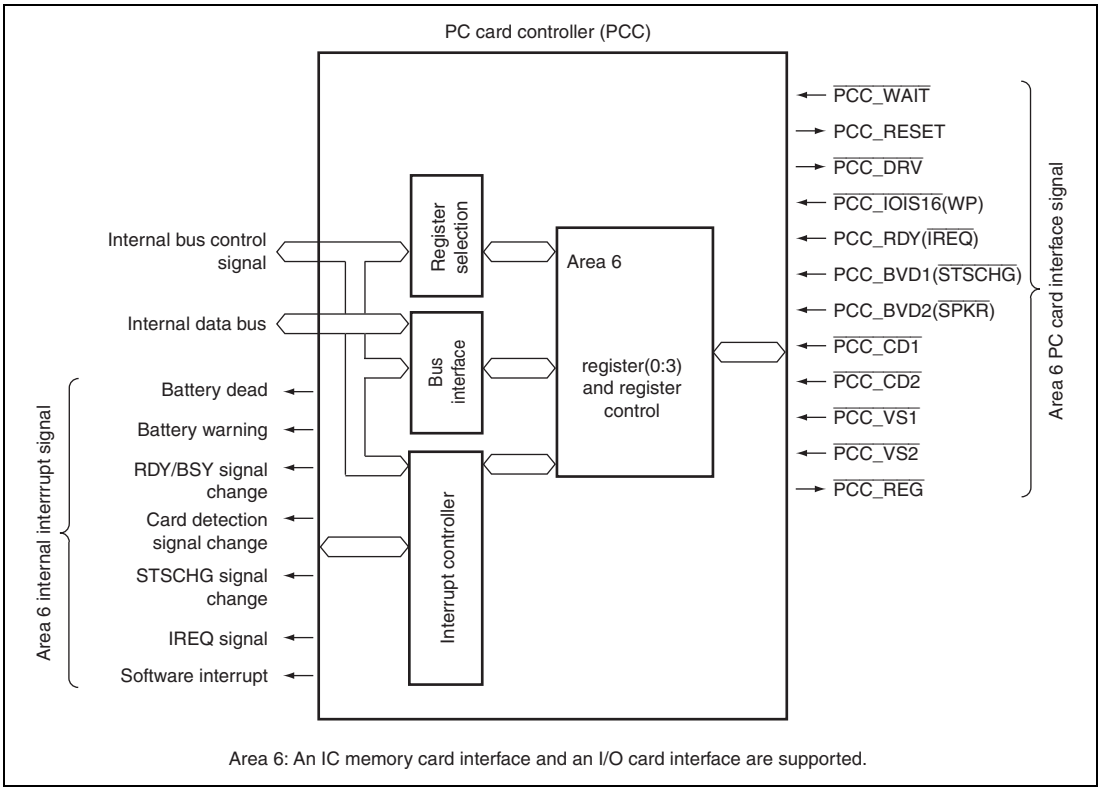


Figure 32.1 PC Card Controller Block Diagram

32.1.1 PCMCIA Support

This LSI supports an interface based on PCMCIA specifications for physical areas 6. Interfaces supported are the IC memory card interface and I/O card interface defined in the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard. Both the IC memory card interface and I/O card interface are supported in area 6.

Table 32.1 Features of the PCMCIA Interface

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Masked ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Common memory capacity	Maximum 64 Mbytes (Supports full PCMCIA specifications by using a segment bit (an address bit for the PC card))
Attribute memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Others	Dynamic bus sizing for I/O bus width* The PCMCIA interface can be accessed from the address-conversion region and non-address-conversion region.

Note: * Dynamic bus sizing for the I/O bus width is supported only in little-endian mode.

This LSI can directly access 32- and 64-Mbyte physical areas in a 64-Mbyte memory space and an I/O space of the PC card (continuous 32/16-Mbyte area mode). This LSI provides a segment bit (an address bit for the PC card) in the general control register for area 6 to support a common memory space with full PCMCIA specifications (64 Mbytes).

Continuous 32-Mbyte Area Mode: Setting 0 (initial value) in bit 3 (P0MMOD) of the general control register enables the continuous 32-Mbyte area mode. In this mode, the attribute memory space and I/O memory space are 32 Mbytes and the common memory space is 64 Mbytes. In the common memory space, set 1 in bit 2 (P0PA25) of the general control register to access an address of more than 32 Mbytes. By this operation, 1 is output to A25 pin, enabling an address space of more than 32 Mbytes to be accessed. When an address of 32 Mbytes or less is accessed, no setting is required (initial value: 0). This bit does not affect access to attribute memory space or I/O memory space.

Figure 32.2 shows the relationship between the memory space of this LSI and the memory and I/O spaces of the PC card in the continuous 32-Mbyte area mode. Although memory space and I/O space are supported in area 6.

In area 6, set 1 in bit 0 (P0REG) of the general control register to access the common memory space of the PC card, and set 0 in bit 0 to access the attribute memory space (initial value: 0). By this operation, the set value is output to $\overline{\text{PCC_REG}}$ pin, enabling any space to be accessed. When the I/O space is accessed in area 6, the output of $\overline{\text{PCC_REG}}$ pin is always 0 regardless of the value of bit 0 (P0REG).

See the register descriptions in section 32.3, Register Descriptions for details of register settings.

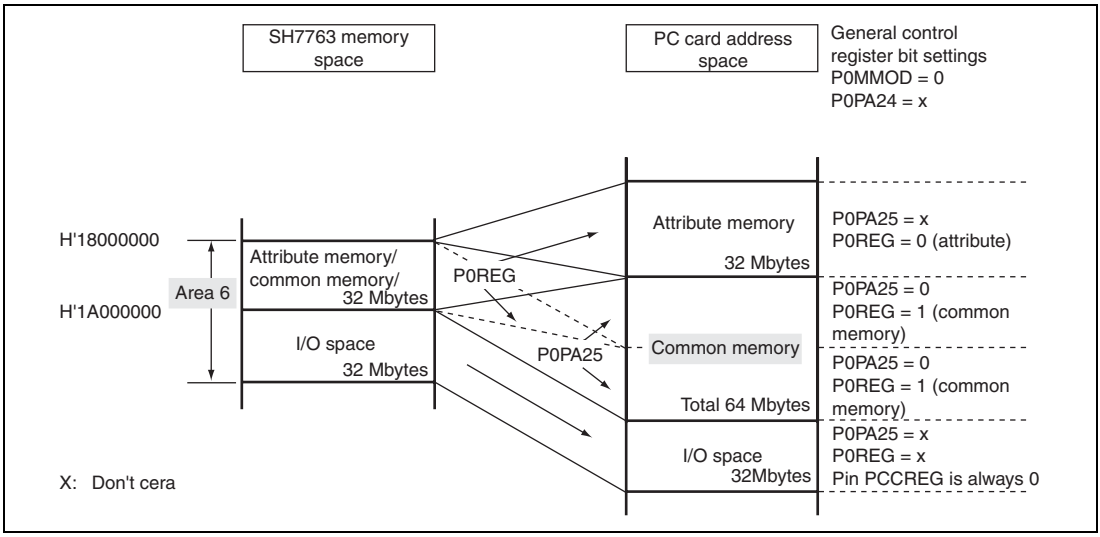


Figure 32.2 Continuous 32-Mbyte Area Mode

Continuous 16-Mbyte Area Mode: Setting 1 in bit 3 (P0MMOD) of the general control register enables the continuous 16-Mbyte area mode. In this mode, the attribute memory space and I/O memory space are 16 Mbytes, and the common memory space is 64 Mbytes. In the common memory space, set the PC card address in bit 2 (P0PA25) and bit 1 (P0PA24) of the general control register to access an address of more than 16 Mbytes. By this operation, values are output to A25 and A24 pins, enabling an address space of more than 16 Mbytes to be accessed (initial value: 0 for P0PA25 and P0PA24). When an address of 16 Mbytes or less is accessed, no settings are required. This bit does not affect access to attribute memory space or I/O memory space.

Figure 32.3 shows the relationship between the memory space of this LSI and the memory and I/O spaces of the PC card in the continuous 16-Mbyte area mode. Although memory space and I/O space are supported in area 6.

The attribute memory space, common memory space, and I/O space of the PC card are provided as 16-Mbyte physical spaces in this mode. Therefore, this LSI automatically controls $\overline{\text{PCC_REG}}$ pin (the value of bit 0 (P0REG) in the general control register is ignored). In area 6, the output of $\overline{\text{PCC_REG}}$ pin is 0 when the attribute memory space or I/O space is accessed, and 1 when the common memory space is accessed.

See the register descriptions in section 32.3, Register Descriptions for details of register settings.

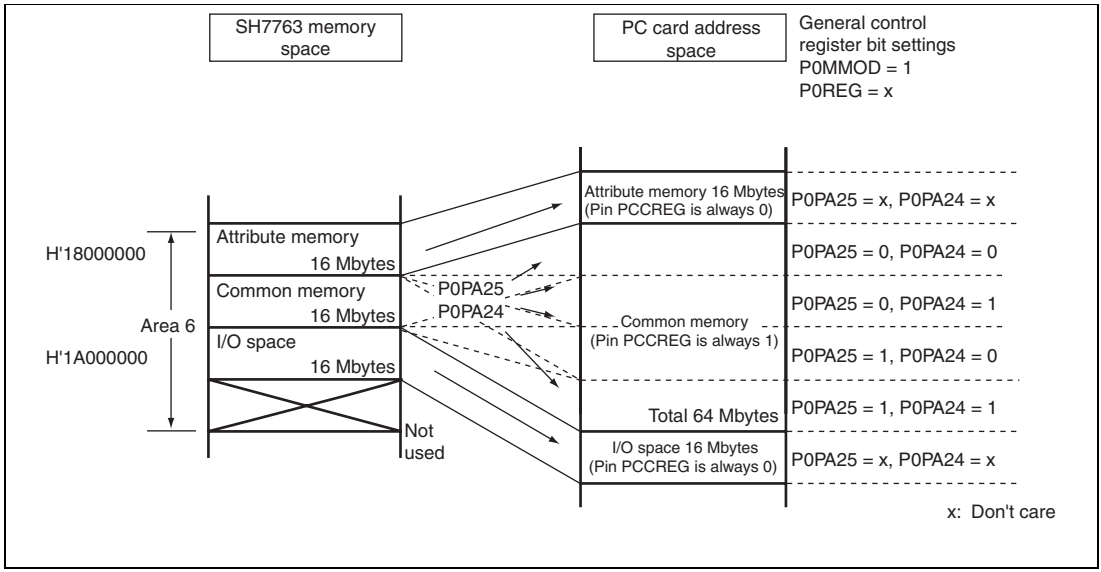


Figure 32.3 Continuous 16-Mbyte Area Mode (Area 6)

32.2 Input/Output Pins

PCC related external pins are listed below.

Table 32.2 PCC Pin Configuration

Pin Name	Abbreviation	I/O	Description
Hardware wait request pin	$\overline{\text{PCC_WAIT}}$	Input	Hardware wait request signal
PCMCIA 16-bit input/output pin	$\overline{\text{PCC_IOIS16}}$	Input	Write protection signal from PC card when IC memory interface is connected
			Signal to indicate 16-bit I/O from PC card when I/O card interface is connected
PCMCIA ready pin	PCC_RDY	Input	Ready/busy signal form PC card when IC memory interface is connected
			Interrupt request signal from PC card when I/O card interface is connected
PCMCIA BVD1 pin	PCC_BVD1	Input	Buttery voltage detect 1 signal from PC card when IC memory interface is connected
			Card status change signal from PC card when I/O card interface is connected
PCMCIA BVD2 pin	PCC_BVD2	Input	Buttery voltage detect 2 signal from PC card when IC memory interface is connected
			Digital sound signal from PC card when I/O card interface is connected
PCMCIA CD1 pin	$\overline{\text{PCC_CD1}}$	Input	Card detect 1 signal from PC card
PCMCIA CD2 pin	$\overline{\text{PCC_CD2}}$	Input	Card detect 2 signal from PC card
PCMCIA VS1 pin	$\overline{\text{PCC_VS1}}$	Input	Voltage sense 1 signal from PC card
PCMCIA VS2 pin	$\overline{\text{PCC_VS2}}$	Input	Voltage sense 2 signal from PC card
PCMCIA REG pin	$\overline{\text{PCC_REG}}$	Output	Area indicate signal for PC card
PCMCIA buffer control pin	$\overline{\text{PCC_DRV}}$	Output	Buffer control signal
PCMCIA reset pin	PCC_RESET	Output	Reset signal for PC card

32.3 Register Descriptions

Table 32.3 shows the PCC register configuration. Table 32.4 shows the register state in each operating mode.

Table 32.3 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Area 6 interface status register	PCC0ISR	R	H'FFE9 8000	H'1FE9 8000	8
Area 6 general control register	PCC0GCR	R/W	H'FFE9 8002	H'1FE9 8002	8
Area 6 card status change register	PCC0CSCR	R/W	H'FFE9 8004	H'1FE9 8004	8
Area 6 card status change interrupt enable register	PCC0CSCIER	R/W	H'FFE9 8006	H'1FE9 8006	8

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 32.4 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Area 6 interface status register	PCC0ISR	Undefined	Undefined	Retained	Retained
Area 6 general control register	PCC0GCR	H'00	H'00	Retained	Retained
Area 6 card status change register	PCC0CSCR	H'00	H'00	Retained	Retained
Area 6 card status change interrupt enable register	PCC0CSCIER	H'00	H'00	Retained	Retained

32.3.1 Area 6 Interface Status Register (PCC0ISR)

PCC0ISR is an 8-bit read-only register which is used to read the status of the PC card connected to area 6. The initial value of PCC0ISR depends on the PC card status.

Bit:	7	6	5	4	3	2	1	0
	P0RDY/ IREQ	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2/ P0SPKR	P0BVD1/ P0STSCHG
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	P0RDY/ IREQ	Undefined*	R	PCC0 Ready The value of RDY/$\overline{\text{BSC}}$ pin of the PC card connected to area 6 is read when the IC memory card interface is connected. The value of $\overline{\text{IREQ}}$ pin of the PC card connected to area 6 is read when the I/O card interface is connected. This bit cannot be written to. Indicates that the value of RDY/$\overline{\text{BSC}}$ pin is 0 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of $\overline{\text{IREQ}}$ pin is 0 when the PC card connected to area 6 is the I/O card interface type. Indicates that the value of RDY/$\overline{\text{BSC}}$ pin is 1 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of $\overline{\text{IREQ}}$ pin is 1 when the PC card connected to area 6 is the I/O card interface type.

Bit	Bit Name	Initial Value	R/W	Description
6	P0MWP	Undefined*	R	PCC0 Write Protect The value of WP pin of the PC card connected to area 6 is read when the IC memory card interface is connected. 0 is read when the I/O card interface is connected. This bit cannot be written to. Indicates that the value of WP pin is 0 when the PC card connected to area 6 uses the IC memory card interface. The value of bit 6 is always 0 when the PC card connected to area 6 is the I/O card interface type. Indicates that the value of WP pin is 1 when the PC card connected to area 6 is the IC memory card interface type.
5	P0VS2	Undefined*	R	PCC0 Voltage Sense 2 The value of $\overline{VS2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 1
4	P0VS1	Undefined*	R	PCC0 Voltage Sense 1 The value of $\overline{VS1}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 1
3	P0CD2	Undefined*	R	PCC0 Card Detect 2 The value of $\overline{CD2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 1

Bit	Bit Name	Initial Value	R/W	Description
2	P0CD1	Undefined*	R	PCC0 Card Detect 1 The value of $\overline{CD1}$ pin of the PC card connected to area 6 is read. This bit cannot be written to. 0: The value of $\overline{CD1}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{CD1}$ pin of the PC card connected to area 6 is 1
1	P0BVD2/ P0SPKR	Undefined*	R	PCC0 Battery Voltage Detect 2 and 1 The values of BVD2 and BVD1 pins of the PC card connected to area 6 are read when the IC memory card interface is connected. The values of $\overline{SPKR}$ and $\overline{STSCHG}$ pins of the PC card connected to area 6 are read when the I/O card interface is connected. These bits cannot be written to. • IC memory interface 11: The battery voltage of the PC card connected to area 6 is normal (Battery Good) 01: The battery must be changed although data is guaranteed for the PC card connected to area 6 (Battery Warning) x0: The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead) • I/O card interface 0: The value of $\overline{SPKR}$ or $\overline{STSCHG}$ pin of the PC card connected to area 6 is 0 1: The value of $\overline{SPKR}$ or $\overline{STSCHG}$ pin of the PC card connected to area 6 is 1
0	P0BVD1/ P0STSCHG	Undefined*	R	

Note: * Differs according to the PC card status.

32.3.2 Area 6 General Control Register (PCC0GCR)

PCC0GCR is an 8-bit readable/writable register which controls the external buffer, resets, address A25 and A24 pins, and $\overline{\text{PCC_REG}}$ pin, and sets the PC card type for the PC card connected to area 6. PCC0GCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	P0DRVE	P0PCCR	P0PCCT	P0USE	P0MMOD	P0PA25	P0PA24	P0REG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	P0DRVE	0	R/W	PCC0 Buffer Control Controls the external buffer for the PC card connected to area 6. 0: High-level setting for control $\overline{\text{PCC_DRV}}$ pin of the external buffer for the PC card connected to area 6 1: Low-level setting for control $\overline{\text{PCC_DRV}}$ pin of the external buffer for the PC card connected to area 6
6	P0PCCR	0	R/W	PCC0 Card Reset Controls resets for the PC card connected to area 6. 0: Low-level setting for reset PCC_RESET pin for the PC card connected to area 6 1: High-level setting for reset PCC_RESET pin for the PC card connected to area 6
5	P0PCCT	0	R/W	PCC0 Card Type Specifies the type of the PC card connected to area 6. Cleared to 0 when the PC card is the IC memory card interface type; set to 1 when the PC card is the I/O card interface type. 0: The PC card connected to area 6 is handled as the IC memory card interface type 1: The PC card connected to area 6 is handled as the I/O card interface type

Bit	Bit Name	Initial Value	R/W	Description
4	P0USE	0	R/W	PCC0 Use/Not Use Specifies that the PC Card Controller to be worked or not worked. 0: PC Card Controller doesn't work 1: PC Card Controller works Note: When setting P0USE to 1, following settings are required. When P0USE is set to 1 and P0PCCT is set to 0, bits 21 and 20 (SA1 and SA0) in the CS6BWCR register of BSC should be set to 0. When P0USE and P0PCCT are set to 1, bits 21 and 20 (SA1 and SA0) in the CS6BWCR register of BSC should be set to 1. Before P0USE is set to 1, bits 15 to 12 (TYPE3 to TYPE0) in CS6BBCR of BSC should be set to 0101.
3	P0MMOD	0	R/W	PCC0 Mode Controls $\overline{\text{PCC_REG}}$ and A24 pins for the PC card connected to area 6. Specifies either A24 of the address to be accessed or bit P0REG for outputting to $\overline{\text{PCC_REG}}$ pin. When the common memory space is accessed, specifies either A24 of the address to be accessed or bit P0PA24 for outputting to A24 pin. By this operation, continuous 32 or 16 Mbytes can be selected for the address area of the common memory space of the PC card. 0: Bit P0REG is output to $\overline{\text{PCC_REG}}$ pin, and A24 of address to be accessed is output to A24 pin (continuous 32-Mbyte area mode) 1: A24 of address to be accessed is output to $\overline{\text{PCC_REG}}$ pin. When the common memory space is accessed, P0PA24 is output to A24 pin (continuous 16-Mbyte area mode)

Bit	Bit Name	Initial Value	R/W	Description
2	P0PA25	0	R/W	PC Card Address Controls A25 pin for the PC card connected to area 6. When the common memory space is accessed for the PC card connected to area 6, this bit is output to A25 pin. When the attribute memory space or I/O space is accessed, this bit is meaningless. 0: When the common memory space is accessed for the PC card connected to area 6, 0 is output to A25 pin 1: When the common memory space is accessed for the PC card connected to area 6, 1 is output to A25 pin
1	P0PA24	0	R/W	PC Card Address Controls A24 pin for the PC card connected to area 6. When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, this bit is output to A24 pin. When bit P0MMOD is 0 or the attribute memory space or I/O space is accessed, this bit is meaningless. 0: When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 0 is output to A24 pin 1: When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 1 is output to A24 pin
0	P0REG	0	R/W	PCC0REG Space Indication Controls $\overline{\text{PCC_REG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 0, this bit is output to $\overline{\text{PCC_REG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 1 or the I/O card interface is accessed, this bit is meaningless. 0: When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 0 is output to $\overline{\text{PCC_REG}}$ pin 1: When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 1 is output to $\overline{\text{PCC_REG}}$ pin

32.3.3 Area 6 Card Status Change Register (PCC0CSCR)

PCC0CSCR is an 8-bit readable/writable able register. PCC0CSCR bits are set to 1 by interrupt sources of the PC card connected to area 6 (only bit 7 can be set to 1 as required). PCC0CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	P0SCDI	—	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	P0SCDI	0	R/W	PCC0 Software Card Detect Change Interrupt A PCC0 software card detect change interrupt can be generated by writing 1 to this bit. When this bit is set to 1, the same interrupt as the PCC0 card detect change interrupt (bit 3 set status) occurs if bit 3 (PCC0 card detect change enable) in the area 6 card status change interrupt enable register (PCC0CSCIER) is set to 1. If bit 3 is cleared to 0, no interrupt occurs. 0: No software card detect change interrupt occurs for the PC card connected to area 6 1: Software card detect change interrupt occurs for the PC card connected to area 6
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	P0IREQ	0	R/W	PCC0IREQ Request Indicates the interrupt request for the $\overline{\text{IREQ}}$ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. The P0IREQ bit is set to 1 when an interrupt request signal in pulse mode or level mode is input to the $\overline{\text{IREQ}}$ pin. The mode is selected by bits 5 and 6 (PCC0IREQ interrupt enable bits) in the area 6 card status change interrupt enable register (PCC0CSCIER). This bit can be cleared to 0 only in pulse mode. Write 0 to bit 5 to clear the bit to 0. This bit is not changed if 1 is written. In level mode, bit 5 is a read-only bit which reflects the $\overline{\text{IREQ}}$ pin state (if the $\overline{\text{IREQ}}$ pin is low, 1 is read). This bit always reads 0 on the IC memory card interface. 0: No interrupt request on the $\overline{\text{IREQ}}$ pin of the PC card when the PC card is on the I/O card interface 1: An interrupt request on the $\overline{\text{IREQ}}$ pin of the PC card has occurred when the PC card is on the I/O card interface
4	P0SC	0	R/W	PCC0 Status Change Indicates a change in the value of the $\overline{\text{STSCHG}}$ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. When the $\overline{\text{STSCHG}}$ pin is changed from 1 to 0, the SC bit is set to 1. When $\overline{\text{STSCHG}}$ pin is not changed, the P0SC bit remains at 0. Write 0 to bit 4 when this bit is set to 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the IC memory card interface. 0: $\overline{\text{STSCHG}}$ pin of the PC card is not changed when the PC card is on the I/O card interface 1: $\overline{\text{STSCHG}}$ pin of the PC card is changed from 1 to 0 when the PC card is on the I/O card interface

Bit	Bit Name	Initial Value	R/W	Description
3	P0CDC	0	R/W	PCC0 Card Detect Change Indicates a change in the value of the $\overline{CD1}$ and $\overline{CD2}$ pins in the PC card connected to area 6. When the $\overline{CD1}$ and $\overline{CD2}$ values are changed, the P0CDC bit is set to 1. When the values are not changed, the P0CDC bit remains at 0. Write 0 to bit 3 in order to clear this bit to 0. This bit is not changed if 1 is written. 0: $\overline{CD1}$ and $\overline{CD2}$ pins in the PC card are not changed 1: $\overline{CD1}$ and $\overline{CD2}$ pins in the PC card are changed
2	P0RC	0	R/W	PCC0 Ready Change Indicates a change in the value of the RDY/$\overline{BSY}$ pin of the PC card when the PC card connected to area 6 is the IC memory card interface type. When the RDY/$\overline{BSY}$ pin is changed from 0 to 1, the P0RC bit is set to 1. When the RDY/$\overline{BSY}$ pin is not changed, the P0RC bit remains at 0. Write 0 to bit 2 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: RDY/$\overline{BSY}$ pin in the PC card is not changed when the PC card is on the IC memory card interface 1: RDY/$\overline{BSY}$ pin in the PC card is changed from 0 to 1 when the PC card is on the IC memory card interface

Bit	Bit Name	Initial Value	R/W	Description
1	P0BW	0	R/W	PCC0 Battery Warning Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which “the battery must be changed although the data is guaranteed” when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 0 and 1, respectively, the P0BW bit is set to 1; in other cases, the P0BW bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: BVD2 and BVD1 of the PC card are not in the battery warning state when the PC card is in the IC memory card interface 1: BVD2 and BVD1 of the PC card are in the battery warning state and “the battery must be changed although the data is guaranteed” when the PC card is on the IC memory card interface
0	P0BD	0	R/W	PCC0 Battery Dead Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which “the battery must be changed since the data is not guaranteed” when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 1 and 0 or 0 and 0, the P0BD bit is set to 1; in other cases, the P0BD bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 0 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface. 0: BVD2 and BVD1 of the PC card are not in the state in which “the battery must be changed since the data is not guaranteed” when the PC card is on the IC memory card interface 1: BVD2 and BVD1 of the PC card are in the state in which “the battery must be changed since the data is not guaranteed” when the PC card is on the IC memory card interface

32.3.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER)

The area 6 card status change interrupt enable register (PCC0CSCIER) is an 8-bit readable/writable register. PCC0CSCIER enables or disables interrupt requests for interrupt sources for the PC card connected to area 6. When a PCC0CSCIER is set to 1, the corresponding interrupt is enabled, and when the bit is cleared to 0, the interrupt is disabled. PCC0CSCIER is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	P0CRE	IREQE[1:0]		P0SCE	P0CDE	P0RE	P0BWE	P0BDE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	P0CRE	0	R/W	PCC0 Card Reset Enable When this bit is set to 1, and when the $\overline{CD1}$ and $\overline{CD2}$ pins detect that a PC card is connected to area 6, the area 6 general control register (PCC0GCR) is initialized. 0: The area 6 general control register (PCC0GCR) is not initialized even if a PC card is detected in area 6 1: The area 6 general control register (PCC0GCR) is initialized when a PC card is detected connected to area 6

Bit	Bit Name	Initial Value	R/W	Description
6, 5	IREQE[1:0]	00	R/W	PCC0IREQ Request Enable These bits enable or disable $\overline{\text{IREQ}}$ pin interrupt requests and select the interrupt mode when the PC card connected to area 6 is the I/O card interface type. Note that bit 5 (P0IREQ) in the area 6 card status change register (PCC0CSCR) is cleared if the values in bits 6 and 5 in this register are changed. These bits have no meaning on the IC memory card interface. 00: IREQ requests are not accepted for the PC card connected to area 6. Bit 5 in the status change register (PCC0CSCR) functions as a read-only bit that indicates the inverse of the $\overline{\text{IREQ}}$ pin signal. 01: The level-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In level mode, an interrupt occurs when level 0 of the signal input from the $\overline{\text{IREQ}}$ pin is detected. 10: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the $\overline{\text{IREQ}}$ pin is detected. 11: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the $\overline{\text{IREQ}}$ pin is detected.

Bit	Bit Name	Initial Value	R/W	Description
4	P0SCE	0	R/W	PCC0 Status Change Enable When the PC card connected to area 6 is on the I/O card interface, bit 4 enables or disables the interrupt request when the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) is changed. This bit has no meaning in the IC memory card interface. 0: No interrupt occurs for the PC card connected to area 6 regardless of the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) 1: An interrupt occurs for the PC card connected to area 6 when the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) is changed from 1 to 0
3	P0CDE	0	R/W	PCC0 Card Detect Change Enable Bit 3 enables or disables the interrupt request when the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins are changed. 0: No interrupt occurs for the PC card connected to area 6 regardless of the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins 1: An interrupt occurs for the PC card connected to area 6 when the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins are changed
2	P0RE	0	R/W	PCC0 Ready Change Enable When the PC card connected to area 6 is on the IC memory card interface, bit 2 enables or disables the interrupt request when the value of the RDY/$\overline{\text{BSY}}$ pin is changed. This bit has no meaning on the I/O card interface. 0: No interrupt occurs for the PC card connected to area 6 regardless of the value of the RDY/$\overline{\text{BSY}}$ pin 1: An interrupt occurs for the PC card connected to area 6 when the value of the RDY/$\overline{\text{BSY}}$ pin is changed from 0 to 1

Bit	Bit Name	Initial Value	R/W	Description
1	P0BWE	0	R/W	PCC0 Battery Warning Enable When the PC card connected to area 6 is on the IC memory card interface, bit 1 enables or disables the interrupt request when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”. This bit has no meaning on the I/O card interface. 0: No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed” 1: An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”
0	P0BDE	0	R/W	PCC0 Battery Dead Enable When the PC card connected to area 6 is on the IC memory card interface, bit 0 enables or disables the interrupt request when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed”. This bit has no meaning on the I/O card interface. 0: No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed” 1: An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed”

32.4 Operation

32.4.1 PC card Connection Specification (Interface Diagram, Pin Correspondence)

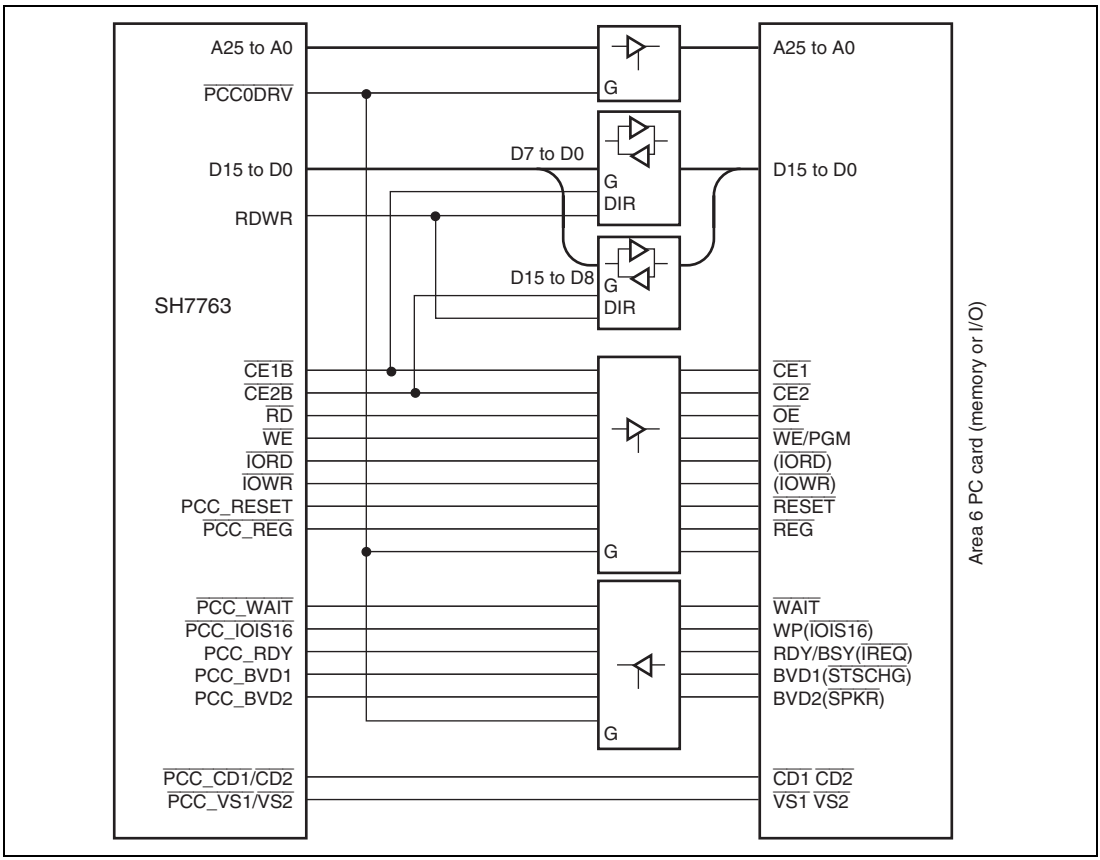


Figure 32.4 SH7763 Interface

Table 32.5 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	
1	GND		Ground	GND		Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CE1B}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	RD
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$
16	RDY/$\overline{BSY}$	O	Ready/busy	$\overline{IREQ}$	O	Interrupt request	PCC_RDY
17	VCC		Power supply	VCC		Power supply	—
18	VPP1		Programming power supply	VPP1		Programming and peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP	O	Write protect	$\overline{\text{IOIS16}}$	O	16-bit I/O port	$\overline{\text{PCC_IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{PCC_CD1}}$
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2B}}$
43	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{PCC_VS1}}$
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{IORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{IOWR}}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—

	IC Memory Card Interface			I/O Card Interface			This LSI Corresponding Pin
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	
52	VPP2		Programming power supply	VPP2		Programming and peripheral power supply	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{VS2}$	O	Voltage sense	$\overline{VS2}$	O	Voltage sense	$\overline{PCC_VS2}$
58	RESET	I	Reset	RESET	I	Reset	PCC_RESET
59	$\overline{WAIT}$	O	Wait request	$\overline{WAIT}$	O	Wait request	$\overline{PCC_WAIT}$
60	RFU		Reserved	$\overline{INPACK}$	O	Input acknowledge	—
61	$\overline{REG}$	I	Attribute memory space select	$\overline{REG}$	I	Attribute memory space select	$\overline{PCC_REG}$
62	BVD2	O	Battery voltage detection	$\overline{SPKR}$	O	Digital sound signal	PCC_BVD2
63	BVD1	O	Battery voltage detection	$\overline{STSCHG}$	O	Card status change	PCC_BVD1
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{CD2}$	O	Card detection	$\overline{CD2}$	O	Card detection	$\overline{PCC_CD2}$
68	GND		Ground	GND		Ground	—

32.4.2 PC Card Interface Timing

(1) Memory Card Interface Timing

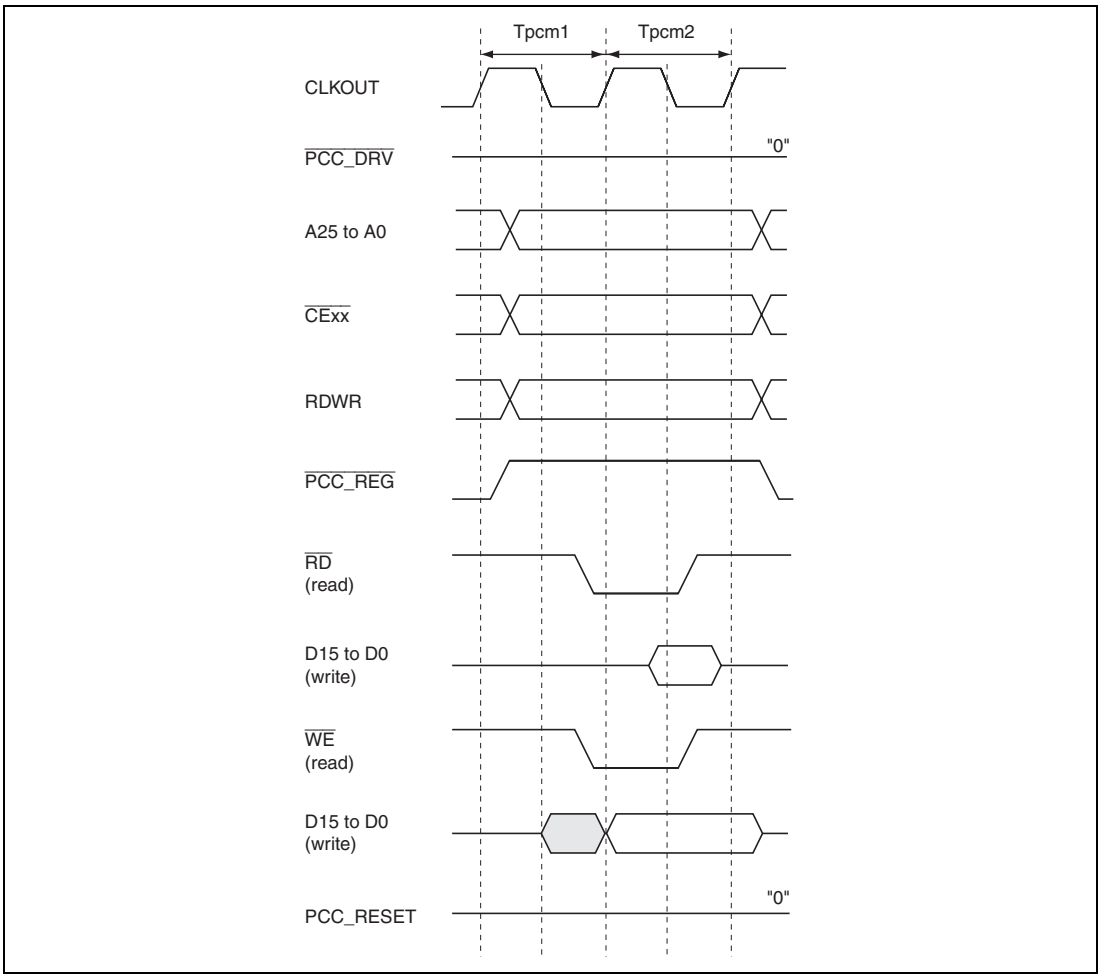


Figure 32.5 PCMCIA Memory Card Interface Basic Timing

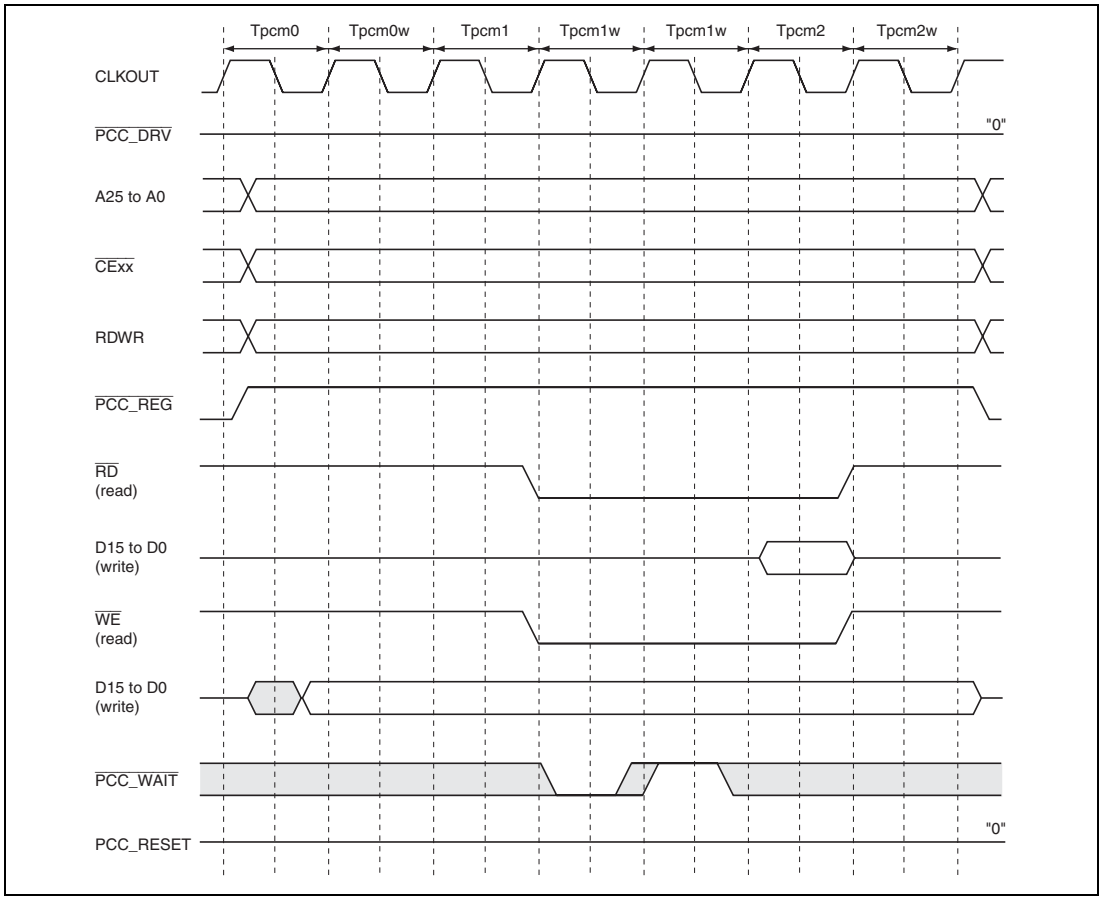


Figure 32.6 PCMCIA Memory Card Interface Wait Timing

(2) I/O Card Interface Timing

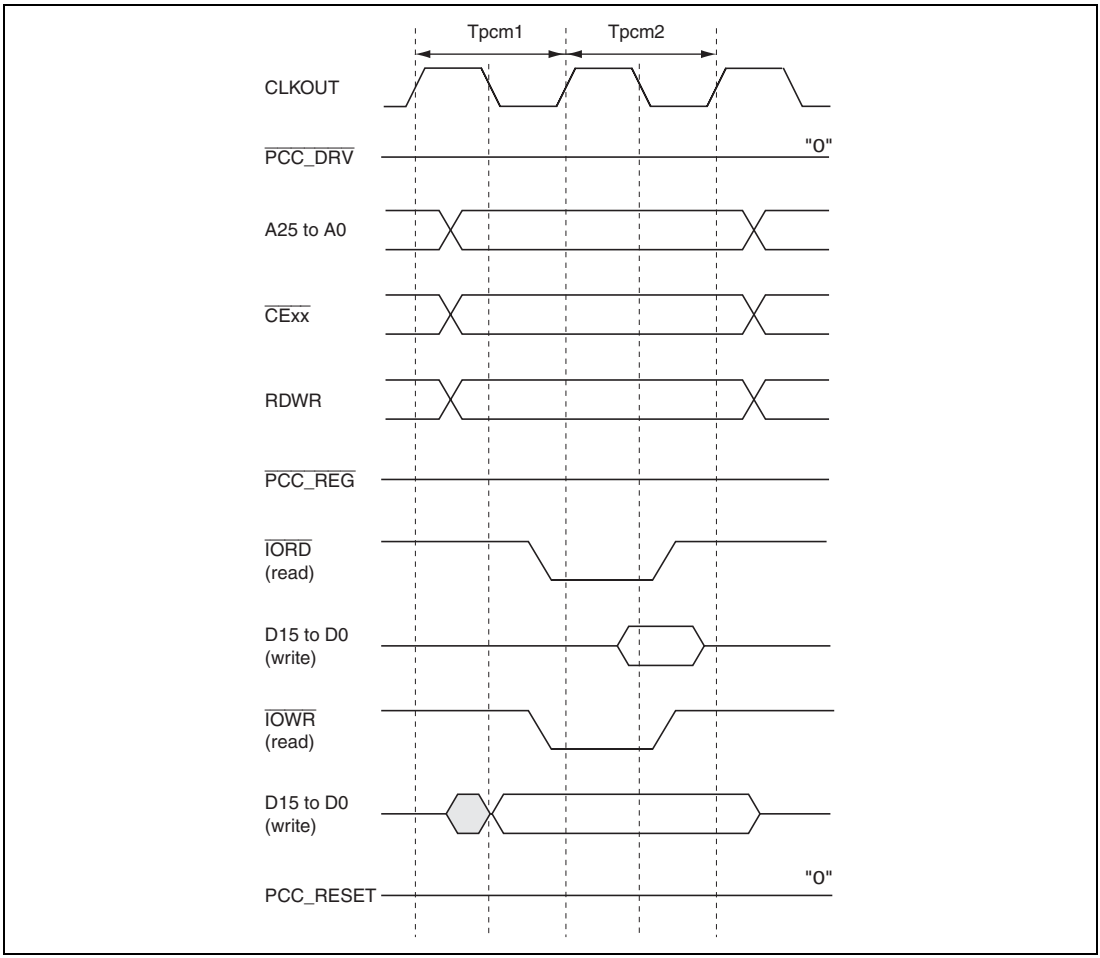


Figure 32.7 PCMCIA I/O Card Interface Basic Timing

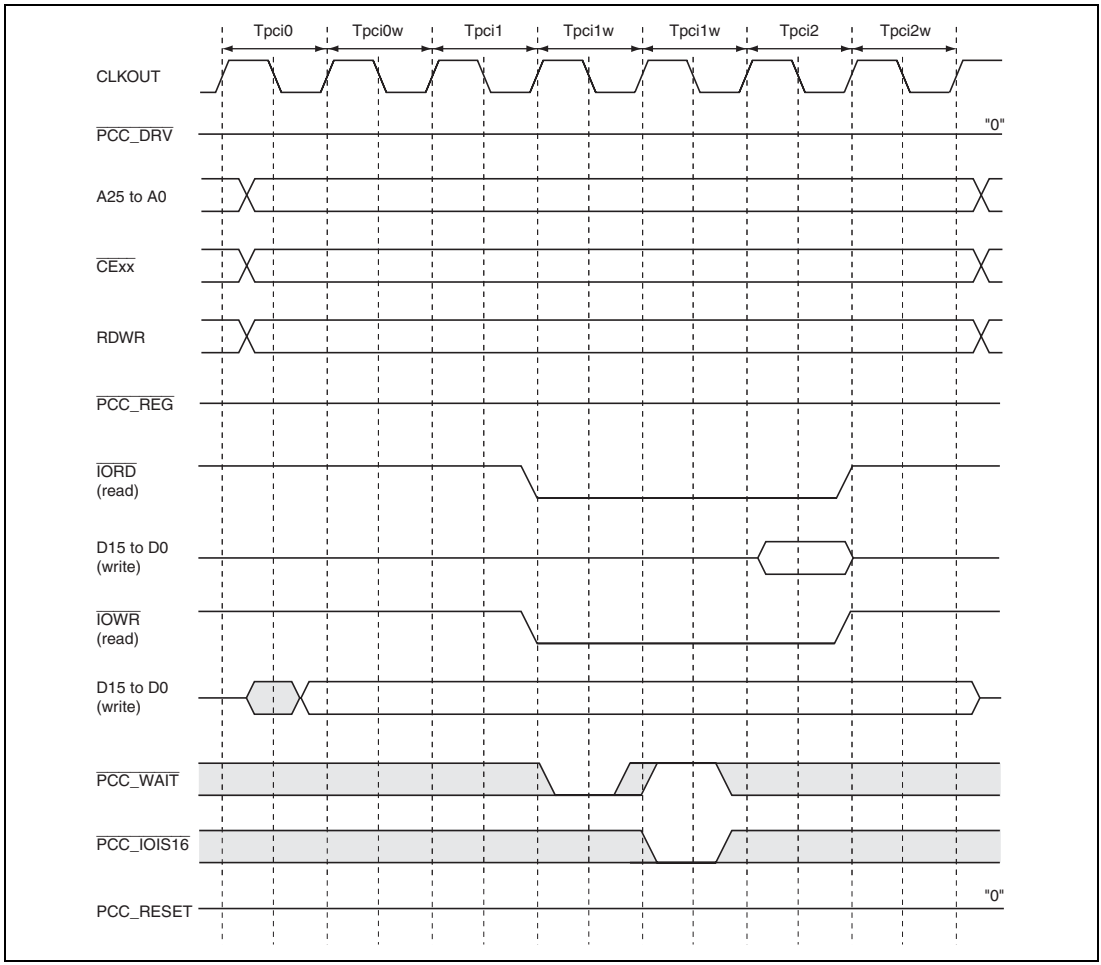


Figure 32.8 PCMCIA I/O Card Interface Wait Timing

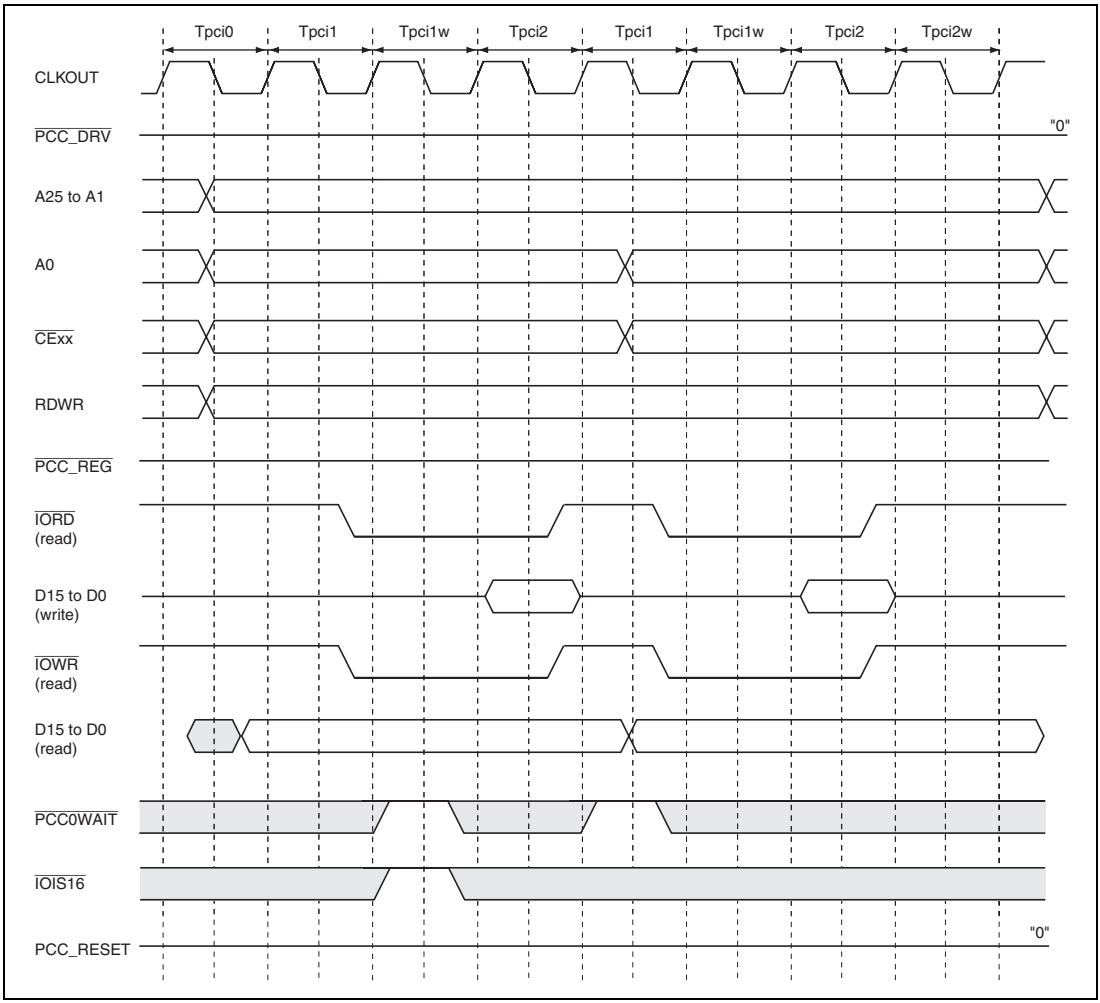


Figure 32.9 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

32.5 Usage Notes

(1) External Bus Frequency Limit when Using PC Card

According to the PC card standard, the attribute memory access time is specified as 600 ns (3.3 V)/300 ns (5 V). Therefore, when this LSI accesses attribute memory, the bus cycle must be coordinated with the PC card interface timing. In this LSI, the timing can be adjusted by setting the TED, TEH, and PCW values in the CS6BWCR register, allowing a PC card to be used within the above frequency ranges.

The common memory access time and I/O access time (based on the IORD and IOWR signals) are also similarly specified (see table below), and a PC card must be used within the above ranges in order to satisfy all these specifications.

PC Card Space	Access Time (5 V Operation)	Access Time (3.3 V Operation)
Attribute memory	300 ns	600 ns
Common memory	250 ns	600 ns
I/O space (–$\overline{\text{IORD}}$/–$\overline{\text{IOWR}}$ pulse width)	165 ns	165 ns

(2) Pin Function Control and Card Type Switching

When setting pin function controller pin functions to dedicated PC card use ("other function"), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made. However, this restriction does not apply to the card detection pins ($\overline{\text{CD1}}$, $\overline{\text{CD2}}$).

When changing the card type bit (P0PCCT) in the area 6 general control register (PCC0GCR), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made.

Reason: When PC card controller settings are modified, the functions of PC card pins that generate various interrupts change, with the result that unnecessary interrupts may be generated.

(3) Setting Procedure when Using PC Card Controller

The following steps should be followed when using a card controller:

1. Set bit 12 (MAP) in the common control register (CMNCR) of bus state controller to 1.
2. Set bits 15 to 12 (TYPE3 to TYPE0) in the bus control register for CS6B (CS6BBCR) of the bus state controller to B′0101.
3. Set bit 4 (P0USE) in the area 6 general control register in the PC card controller to 1.
4. Set the pin function controller to custom PC card pin functions (“other functions”).

Section 33 Audio Codec Interface (HAC)

The HAC, the audio codec digital controller interface, supports bidirectional data transfer compliant with Audio Codec 97 (AC'97) Version 2.1. The HAC provides serial transmission to /reception from the AC97 codec. Each channel of the HAC can be connected to a single audio codec device.

The HAC carries out data extraction from/insertion into audio frames. For data slots within both receive and transmit frames, the PIO transfer by the CPU or the DMA transfer by the DMAC can be used.

33.1 Features

The HAC has the following features:

- Supports Digital interface to a single AC'97 version 2.1 Audio Codec
- PIO transfer of status slots 1 and 2 in Rx frames
- PIO transfer of command slots 1 and 2 in Tx frames
- PIO transfer of data slots 3 and 4 in Rx frames
- PIO transfer of data slots 3 and 4 in Tx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Rx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Tx frames
- Accommodates various sampling rates by qualifying slot data with tag bits and monitoring the Tx frame request bits of Rx frames
- Generates data ready, data request, overrun and underrun interrupts
- Supports cold reset, warm reset, and power-down mode

Figure 33.1 shows a block diagram of the HAC.

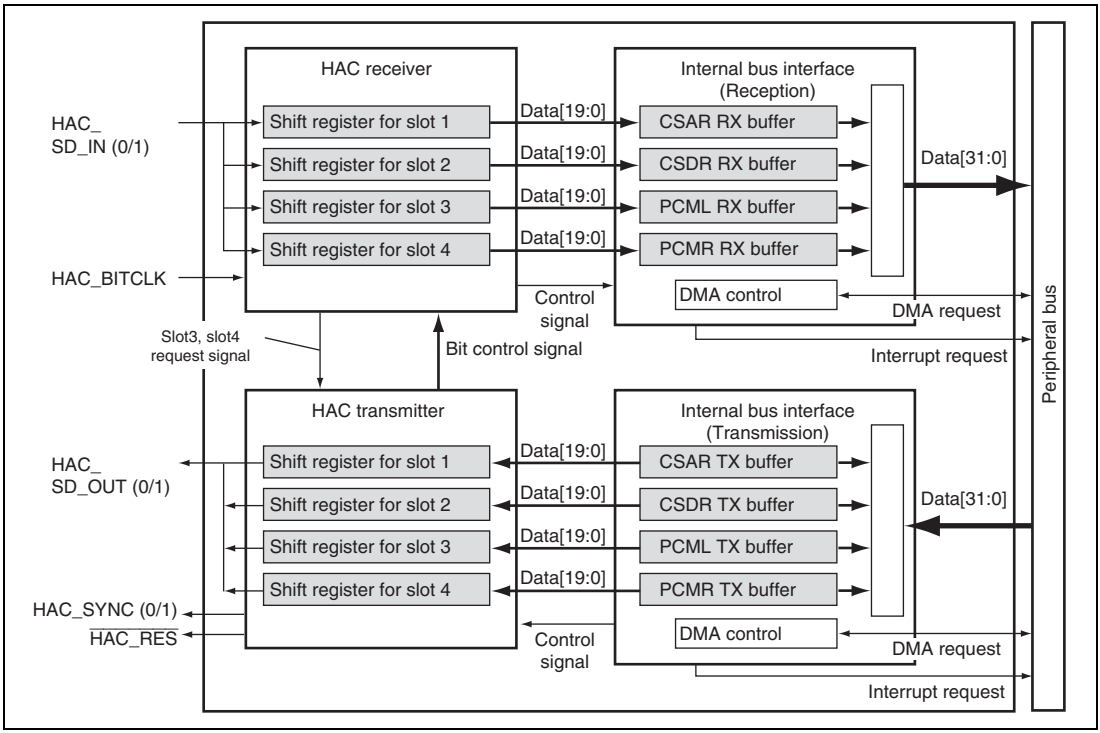


Figure 33.1 Block Diagram

33.2 Input/Output Pins

Table 33.1 describes the HAC pin configuration.

Table 33.1 Pin Configuration

Pin Name	I/O	Function
HAC_BITCLK	Input	HAC serial data clock
HAC_SD_IN	Input	HAC serial data incoming to Rx frame
HAC_SD_OUT	Output	HAC serial data outgoing from Tx frame
HAC_SYNC	Output	HAC frame sync
$\overline{\text{HAC_RES}}$	Output	HAC reset (negative logic signal)

33.3 Register Descriptions

Table 33.2 shows the HAC register configuration. Table 33.3 shows the register state in each operating mode.

Table 33.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address*	Area 7 Address*	Size
Control and status register	HACCR	R/W	H'FFEB 0008	H'1FEB 0008	32
Command/status address register	HACCSAR	R/W	H'FFEB 0020	H'1FEB 0020	32
Command/status data register	HACCSDR	R/W	H'FFEB 0024	H'1FEB 0024	32
PCM left channel register	HACPCML	R/W	H'FFEB 0028	H'1FEB 0028	32
PCM right channel register	HACPCMR	R/W	H'FFEB 002C	H'1FEB 002C	32
TX interrupt enable register	HACTIER	R/W	H'FFEB 0050	H'1FEB 0050	32
TX status register	HACTSR	R/W	H'FFEB 0054	H'1FEB 0054	32
RX interrupt enable register	HACRIER	R/W	H'FFEB 0058	H'1FEB 0058	32
RX status register	HACRSR	R/W	H'FFEB 005C	H'1FEB 005C	32
HAC control register	HACACR	R/W	H'FFEB 0060	H'1FEB 0060	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 33.3 Register State in Each Operating Mode

Register Name	Abbrev.	Power-on Reset	Manual Reset	Sleep	Standby
Control and status register	HACCR	H'0000 0200	H'0000 0200	Retained	Retained
Command/status address register	HACCSAR	H'0000 0000	H'0000 0000	Retained	Retained
Command/status data register	HACCSDR	H'0000 0000	H'0000 0000	Retained	Retained
PCM left channel register	HACPCML	H'0000 0000	H'0000 0000	Retained	Retained
PCM right channel register	HACPCMR	H'0000 0000	H'0000 0000	Retained	Retained
TX interrupt enable register	HACTIER	H'0000 0000	H'0000 0000	Retained	Retained
TX status register	HACTSR	H'F000 0000	H'F000 0000	Retained	Retained
RX interrupt enable register	HACRIER	H'0000 0000	H'0000 0000	Retained	Retained
RX status register	HACRSR	H'0000 0000	H'0000 0000	Retained	Retained
HAC control register	HACACR	H'8400 0000	H'8400 0000	Retained	Retained

33.3.1 Control and Status Register (HACCR)

HACCR is a 32-bit read/write register for controlling input/output and monitoring the interface status.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved Always 0 for read and write.
15	CR	0	R	Codec Ready 0: The HAC-connected codec is not ready. 1: The HAC-connected codec is ready.
14 to 12	—	All 0	R	Reserved Always read as 0. Write prohibited.
11	CDRT	0	W	HAC Cold Reset Use a cold reset only after power-on, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a cold reset on the HAC. [Read] Always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
10	WMRT	0	W	HAC Warm Reset Use a warm reset only after power-up, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a warm reset on the HAC. [Read] Always read as 0.
9	—	1	R	Reserved Always 1 for read and write.
8 to 6	—	All 0	R	Reserved Always 0 for read and write.
5	ST	0	W	Start Transfer [Write] 1: Starts data transmission/reception. 0: Stops data transmission/reception at the end of the current frame. Do not take this action to terminate transmission/reception in normal operation. [Read access] Always read as 0.
4 to 0	—	All 0	R	Reserved Always 0 for read and write.

To place the off-chip codec device into the power-down mode, write 1 to bit 12 of the register index 26 in the off-chip codec via the HAC. When entering the power-down mode, the off-chip codec stops HAC_BITCLK and suspends the normal operation. The off-chip codec acts in the same manner at power-on. To resume the normal operation, perform a cold reset or a warm reset on the off-chip codec.

33.3.2 Command/Status Address Register (HACCSAR)

HACCSAR is a 32-bit read/write register that specifies the address of the codec register to be read /written. When requesting a write to/read from a codec register, write the command register address to HACCSAR. Then the HAC transmits this register address to the codec via slot 1.

After the codec has responded to a read request (HACRSR.STARY = 1), the status address received via slot 1 can be read out from HACCSAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RW	CA/SA[6:4]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CA/SA[3:0]				SLREQ[3:12]										—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19	RW	0	R/W	Codec Read/Write Command 0: Notifies the off-chip codec device of a write access to the register specified in the address field (CA6/SA6 to CA0/SA0). Write the data to HACCSDR in advance. When HACACR.TX12_ATOMIC is 1, the HAC transmits HACCSAR and HACCSDR as a pair in the same Tx frame. When HACACR.TX12_ATOMIC is 0, transmission of HACCSAR and HACCSDR in the same Tx frame is not guaranteed. 1: Notifies the off-chip codec device of a read access to the register specified in the address field (CA6/SA6 to CA0/SA0).

Bit	Bit Name	Initial Value	R/W	Description
18 to 12	CA/SA[6:0]	All 0	R/W	Codec Control Register Addresses 6 to 0 /Codec Status Register Addresses 6 to 0 [Write] Specify the address of the codec register to be written. [Read] Indicate the status address received via slot 1, corresponding to the codec register whose data has been returned in HACCSDR.
11 to 2	SLREQ [3:12]	All 0	R	Slot Requests 3 to 12 Valid only in the Rx frame. Indicate whether the codec is requesting slot data in the next Tx frame. Automatically set by hardware, and correspond to bits 11 to 2 of slot 1 in the Rx frame. 0: Slot data is requested. 1: Slot data is not requested.
1, 0	—	All 0	R	Reserved Always 0 for read and write.

33.3.3 Command/Status Data Register (HACCSDR)

HACCSDR is a 32-bit read/write data register used for accessing the codec register. Write the command data to HACCSDR. The HAC then transmits the data to the codec via slot 2.

After the codec has responded to a read request (HACRSR.STDRY = 1), the status data received via slot 2 can be read out from HACCSDR. In both read and write, HACCSAR stores the related codec register address.

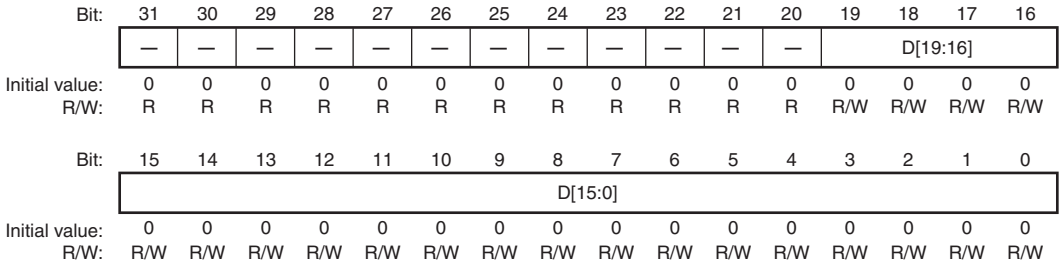
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	CD/SD[15:12]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 4	CD/SD[15:0]	All 0	R/W	Command Data 15 to 0/Status Data 15 to 0 Write data to these bits and then write the codec register address in HACCSAR. The HAC then transmits the data to the codec. Read these bits to get the contents of the codec register indicated by HACCSAR.
3 to 0	—	All 0	R	Reserved Always 0 for read and write.

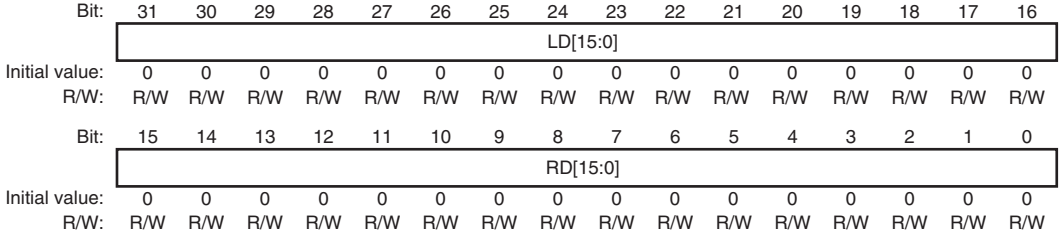
33.3.4 PCM Left Channel Register (HACPCML)

HACPCML is a 32-bit read/write data register used for accessing the left channel of the codec in digital audio recording or stream playback. To transmit the PCM playback left channel data to the codec, write the data to HACPCML. To receive the PCM record left channel data from the codec, read HACPCML. The data is left justified to accommodate a codec with ADC/DAC resolution of 20 bits or less.



Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D[19: 0]	All 0	R/W	Data 19 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.

In 16-bit packed DMA mode, HACPCML is defined as follows:



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	LD[15:0]	All 0	R/W	Left Data 15 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.
15 to 0	RD[15:0]	All 0	R/W	Right Data 15 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

33.3.5 PCM Right Channel Register (HACPCMR)

HACPCMR is a 32-bit read/write register used for accessing the right channel of the codec in digital audio recording or stream playback. To transmit the PCM playback right channel data to the codec, write the data to HACPCMR. To receive the PCM record right channel data from the codec, read HACPCMR. The data is left justified to accommodate a codec with ADC/DAC resolution of 20-bit or less.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D[19:0]	All 0	R/W	Data 19 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

33.3.6 TX Interrupt Enable Register (HACTIER)

HACTIER is a 32-bit read/write register that enables or disables HAC TX interrupts.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved Always 0 for read and write.
29	PLTFRQIE	0	R/W	PCML TX Request Interrupt Enable 0: Disables PCML TX request interrupts 1: Enables PCML TX request interrupts
28	PRTFRQIE	0	R/W	PCMR TX Request Interrupt Enable 0: Disables PCMR TX request interrupts 1: Enables PCMR TX request interrupts
27 to 10	—	All 0	R	Reserved Always 0 for read and write.
9	PLTFUNIE	0	R/W	PCML TX Underrun Interrupt Enable 0: Disables PCML TX underrun interrupts 1: Enables PCML TX underrun interrupts
8	PRTFUNIE	0	R/W	PCMR TX Underrun Interrupt Enable 0: Disables PCMR TX underrun interrupts 1: Enables PCMR TX underrun interrupts
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

33.3.7 TX Status Register (HACTSR)

HACTSR is a 32-bit read/write register that indicates the status of the HAC TX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD AMT	CMD DMT	PLT FRQ	PRT FRQ	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31	CMDAMT	1	R/W*[2]	Command Address Empty 0: CSAR Tx buffer contains untransmitted data. 1: CSAR Tx buffer is empty and ready to store data.*[1]
30	CMDDMT	1	R/W*[2]	Command Data Empty 0: CSDR Tx buffer contains untransmitted data. 1: CSDR Tx buffer is empty and ready to store data.*[1]
29	PLTFRQ	1	R/W*[2]	PCML TX Request 0: PCML Tx buffer contains untransmitted data. 1: PCML TX buffer is empty and needs to store data. In DMA mode, writing to HACPCML will automatically clear this bit to 0.
28	PRTFRQ	1	R/W*[2]	PCMR TX Request 0: PCMR Tx buffer contains untransmitted data. 1: PCMR TX buffer is empty and needs to store data. In DMA mode, writing to HACPCMR will automatically clear this bit to 0.
27 to 10	—	All 0	R	Reserved Always 0 for read and write.

Bit	Bit Name	Initial Value	R/W	Description
9	PLTFUN	0	R/W*2	PCML TX Underrun 0: No PCML TX underrun has occurred. 1: PCML TX underrun has occurred because the codec has requested slot 3 data but new data is not written to PCML.
8	PRTFUN	0	R/W*2	PCMR TX Underrun 0: No PCMR TX underrun has occurred. 1: PCMR TX underrun has occurred because the codec has requested slot 4 data but new data is not written to PCMR.
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

Notes: 1. CMDAMT and CMDDMT have no associated interrupts. Poll these bits until they are read as 1 before writing a new command to HACCSAR/HACCSDR. When bit 19 (RW) of HACCSAR is 0 and TX12_ATOMIC is 1, take the following steps:
   1. Initialize CMDDMT and CMDAMT before first accessing a codec register after HAC initialization by any reset event.
   2. After making the settings in HACCSDR and HACCSAR, poll CMDDMT and CMDAMT until they are cleared to 1, and then initialize these bits.
   3. Now the next write to a register is available.

2. These bits are read/write. Writing 0 to the bit initializes it but writing 1 has no effect.

33.3.8 RX Interrupt Enable Register (HACRIER)

HACRIER is a 32-bit read/write register that enables or disables HAC RX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PLRFOVIE	PRRFOVIE	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.
22	STARYIE	0	R/W	Status Address Ready Interrupt Enable 0: Disables status address ready interrupts. 1: Enables status address ready interrupts.
21	STDRYIE	0	R/W	Status Data Ready Interrupt Enable 0: Disables status data ready interrupts. 1: Enables status data ready interrupts.
20	PLRFRQIE	0	R/W	PCML RX Request Interrupt Enable 0: Disables PCML RX request interrupts. 1: Enables PCML RX request interrupts.
19	PRRFRQIE	0	R/W	PCMR RX Request Interrupt Enable 0: Disables PCMR RX request interrupts. 1: Enables PCMR RX request interrupts.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.
13	PLRFOVIE	0	R/W	PCML RX Overrun Interrupt Enable 0: Disables PCML RX overrun interrupts. 1: Enables PCML RX overrun interrupts.
12	PRRFOVIE	0	R/W	PCMR RX Overrun Interrupt Enable 0: Disables PCMR RX overrun interrupts. 1: Enables PCMR RX overrun interrupts.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

33.3.9 RX Status Register (HACRSR)

HACRSR is a 32-bit read/write register that indicates the status of the HAC RX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	STARY	STDRY	PLRFRQ	PRRFRQ	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PLRFOV	PRRFOV	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.
22	STARY	0	R/W	Status Address Ready 0: HACCSAR (status address) is not ready. 1: HACCSAR (status address) is ready.
21	STDRY	0	R/W	Status Data Ready 0: HACCSDR (status data) is not ready. 1: HACCSDR (status data) is ready.
20	PLRFRQ	0	R/W	PCML RX Request 0: PCML RX data is not ready. 1: PCML RX data is ready and must be read. In DMA mode, reading HACPCML automatically clears this bit to 0.
19	PRRFRQ	0	R/W	PCMR RX Request 0: PCMR RX data is not ready. 1: PCMR RX data is ready and must be read. In DMA mode, reading HACPCMR automatically clears this bit to 0.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.

Bit	Bit Name	Initial Value	R/W	Description
13	PLRFOV	0	R/W	PCML RX Overrun 0: No PCML RX data overrun has occurred. 1: PCML RX data overrun has occurred because the HAC has received new data from slot 3 before PCML data is not read out.
12	PRRFOV	0	R/W	PCMR RX Overrun 0: No PCMR RX data overrun has occurred. 1: PCMR RX data overrun has occurred because the HAC has received new data from slot 4 before PCMR data is not read out.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

Note: * This register is read/write. Writing 0 to the bit initializes it but writing 1 has no effect.

33.3.10 HAC Control Register (HACACR)

HACACR is a 32-bit read/write register used for controlling the HAC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	DMA RX16	DMA TX16	—	—	TX12_ ATOMIC	—	RXD MAL_ EN	TXD MAL_ EN	RXD MAR_ EN	TXD MAR_ EN	—	—	—	—	—
Initial value:	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	1	R	Reserved Always 1 for read and write..
30	DMARX16	0	R/W	16-bit RX DMA Enable 0: Disables 16-bit packed RX DMA mode. Enables the RXDMAL_EN and RXDMAR_EN settings. 1: Enables 16-bit packed RX DMA mode. Disables the RXDMAL_EN and RXDMAR_EN settings.

Bit	Bit Name	Initial Value	R/W	Description
29	DMATX16	0	R/W	16-bit TX DMA Enable 0: Disables 16-bit packed TX DMA mode. Enables the TXDMAL_EN and TXDMAR_EN settings. 1: Enables 16-bit packed TX DMA mode. Disables the TXDMAL_EN and TXDMAR_EN settings.
28, 27	—	All 0	R	Reserved Always 0 for read and write.
26	TX12_ATOMIC	1	R/W	TX Slot 1 and 2 Atomic Control 0: Transmits TX data in HACCSAR and that in HACCSDR separately. (Setting prohibited) 1: Transmits TX data in HACCSAR and that in HACCSDR in the same frame if bit 19 in HACCSAR is 0 (write). (HACCSAR must be written last.)
25	—	0	R	Reserved Always 0 for read and write.
24	RXDMAL_EN	0	R/W	RX DMA Left Enable 0: Disables 20-bit RX DMA for HACPCML. 1: Enables 20-bit RX DMA is for HACPCML.
23	TXDMAL_EN	0	R/W	TX DMA Left Enable 0: Disables 20-bit TX DMA for HACPCML. 1: Enables 20-bit TX DMA for HACPCML.
22	RXDMAR_EN	0	R/W	RX DMA Right Enable 0: Disables 20-bit RX DMA for HACPCMR. 1: Enables 20-bit RX DMA for HACPCMR.
21	TXDMAR_EN	0	R/W	TX DMA Right Enable 0: Disables 20-bit TX DMA for HACPCMR. 1: Enables 20-bit TX DMA for HACPCMR.
20 to 0	—	All 0	R	Reserved Always 0 for read and write.

33.4 AC 97 Frame Slot Structure

Figure 33.2 shows the AC97 frame slot structure. This LSI supports slots 0 to 4 only. Slots 5 to 12 are out of scope.

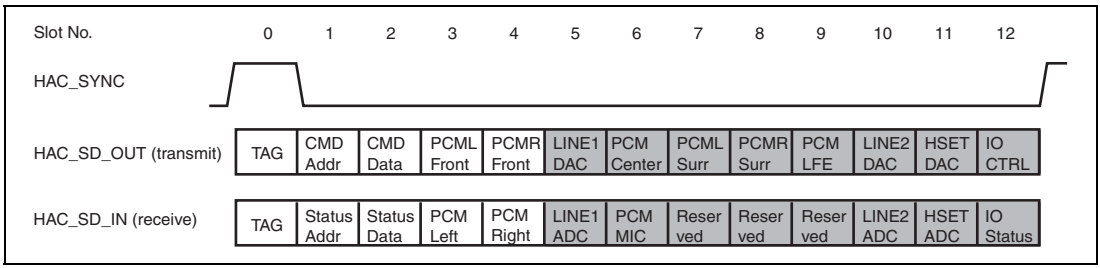


Figure 33.2 AC97 Frame Slot Structure

Table 33.4 AC97 Transmit Frame Structure

Slot	Name	Description
0	SDATA_OUT TAG	Codec IDs and Tags indicating valid data
1	Control CMD Addr write port	Read/write command and register address
2	Control DATA write port	Register write data
3	PCM L DAC playback	Left channel PCM output data
4	PCM R DAC playback	Right channel PCM output data
5	Modem Line 1 DAC	Modem 1 output data (unsupported)*
6	PCM Center	Center channel PCM data (unsupported)*
7	PCM Surround L	Surround left channel PCM data (unsupported)*
8	PCM Surround R	Surround right channel PCM data (unsupported)*
9	PCM LFE	LFE channel PCM data (unsupported)*
10	Modem Line 2 DAC	Modem 2 output data (unsupported)*
11	Modem handset DAC	Modem handset output data (unsupported)*
12	Modem IO control	Modem control IO output (unsupported)*

Notes: * There is no register for unsupported functions.

Table 33.5 AC97 Receive Frame Structure

Slot	Name	Description
0	SDATA_IN TAG	Tags indicating valid data
1	Status ADDR read port	Register address and slot request
2	Status DATA read port	Register read data
3	PCM L ADC record	Left channel PCM input data
4	PCM R ADC record	Right channel PCM input data
5	Modem Line 1 ADC	Modem 1 input data (unsupported)*
6	Dedicated Microphone ADC	Optional PCM data (unsupported)*
7 to 9	Reserved	Reserved
10	Modem Line 2 ADC	Modem 2 input data (unsupported)*
11	Modem handset input DAC	Modem handset input data (unsupported)*
12	Modem IO status	Modem control IO input (unsupported)*

Notes: * There is no register for unsupported functions.

33.5 Operation

33.5.1 Receiver

The HAC receiver receives serial audio data input on the HAC_SD_IN pin, synchronous to HAC_BITCLK. From slot 0, the receiver extracts tag bits that indicate which other slots contain valid data. It will update the receive data only when receiving valid slot data indicated by the tag bits.

Supporting data only in slots 1 to 4, the receiver ignores tag bits and data related to slots 5 to 12. It loads valid slot data to the corresponding shift register to hold the data for PIO or DMA transfer, and sets the corresponding status bits. It is possible to read 20-bit data within a 32-bit register using PIO.

In the case of RX overrun, the new data will overwrite the current data in the RX buffer of the HAC.

33.5.2 Transmitter

The HAC transmitter outputs serial audio data on the HAC_SDOUT pin, synchronous to The transmitter sets the tag bits in slot 0 to indicate which slots in the current frame contain valid data. It loads data slots to the current TX frame in response to the corresponding slot request bits from the previous RX frame.

The transmitter supports data only in slots 1 to 4. The TX buffer holds data that has been transferred using PIO or DMA, and sets the corresponding status bit. It is possible to write 20-bit data within a 32-bit register using PIO.

In the case of a TX underrun, the HAC will transmit the current TX buffer data until the next data arrives.

33.5.3 DMA

The HAC supports DMA transfer for slots 3 and 4 of both the RX and TX frames. Specify the slot data size for DMA transfer, 16 or 20 bits, with the DMARX16 and DMATX16 bits in HACACR.

When the data size is 20 bits, transfer of data slots 3 and 4 requires two local bus access cycles. Since each of the receiver and transmitter has its DMA request, the stereo mode generates a DMA request for slots 3 and 4 separately. The mono mode generates a DMA request for just one slot.

When the data size is 16 bits, data from slots 3 and 4 are packed into a single 32-bit quantity (left data and right data are in PCML), which requires only one local bus access cycle.

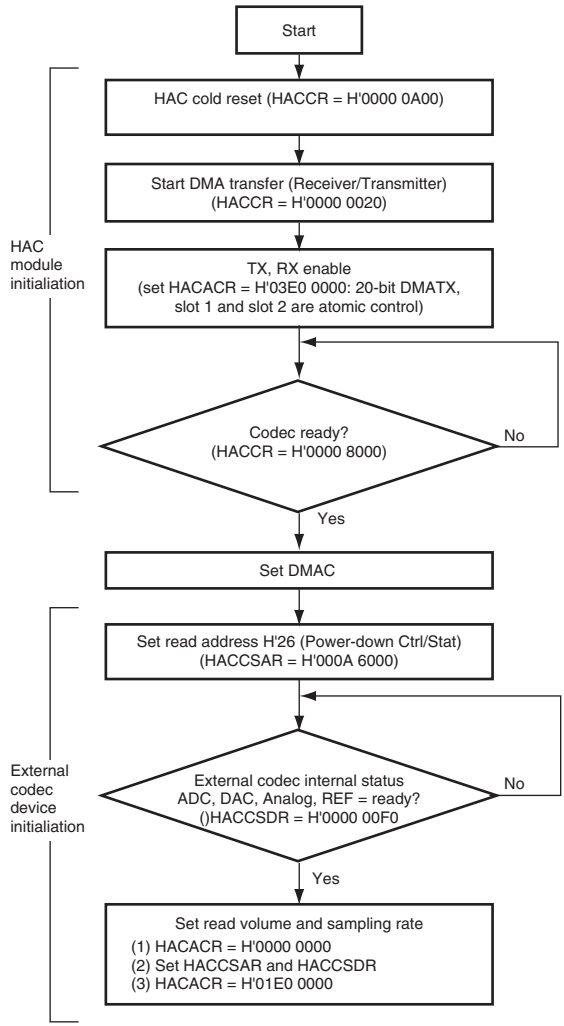
It may be necessary to halt a DMA transfer before the end count is reached, depending on system applications. If so, clear the corresponding DMA bit in HACACR to 0 (DMA disabled). To resume a DMA transfer, reprogram the DMAC and then set the corresponding DMA bit to 1 (DMA enabled).

33.5.4 Interrupts

Interrupts can be used for flag events from the receiver and transmitter. Make the setting for each interrupt in the corresponding interrupt enable register. Interrupts include a request to the CPU to read/write slot data, overrun and underrun. To get the interrupt source, read the status register. Writing 0 to the bit will clear the corresponding interrupt.

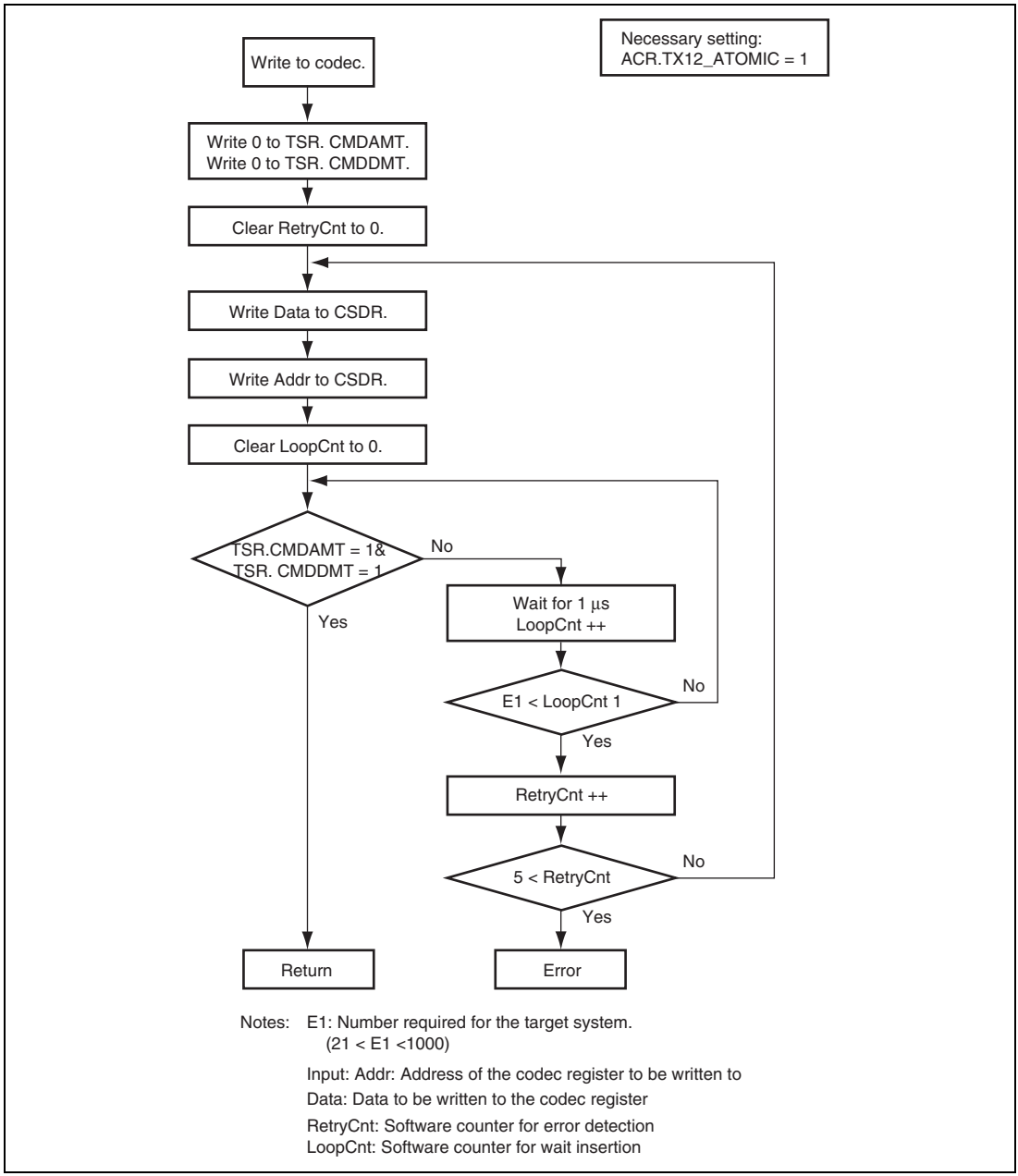
33.5.5 Initialization Sequence

Figure 33.3 shows an example of the initialization sequence.



Note: Refer to section 14, Direct Memory Access Controller (DMAC).

Figure 33.3 Initialization Sequence



Notes: E1: Number required for the target system.
(21 < E1 <1000)

Input: Addr: Address of the codec register to be written to

Data: Data to be written to the codec register

RetryCnt: Software counter for error detection

LoopCnt: Software counter for wait insertion

Figure 33.4 Sample Flowchart for Off-Chip Codec Register Write

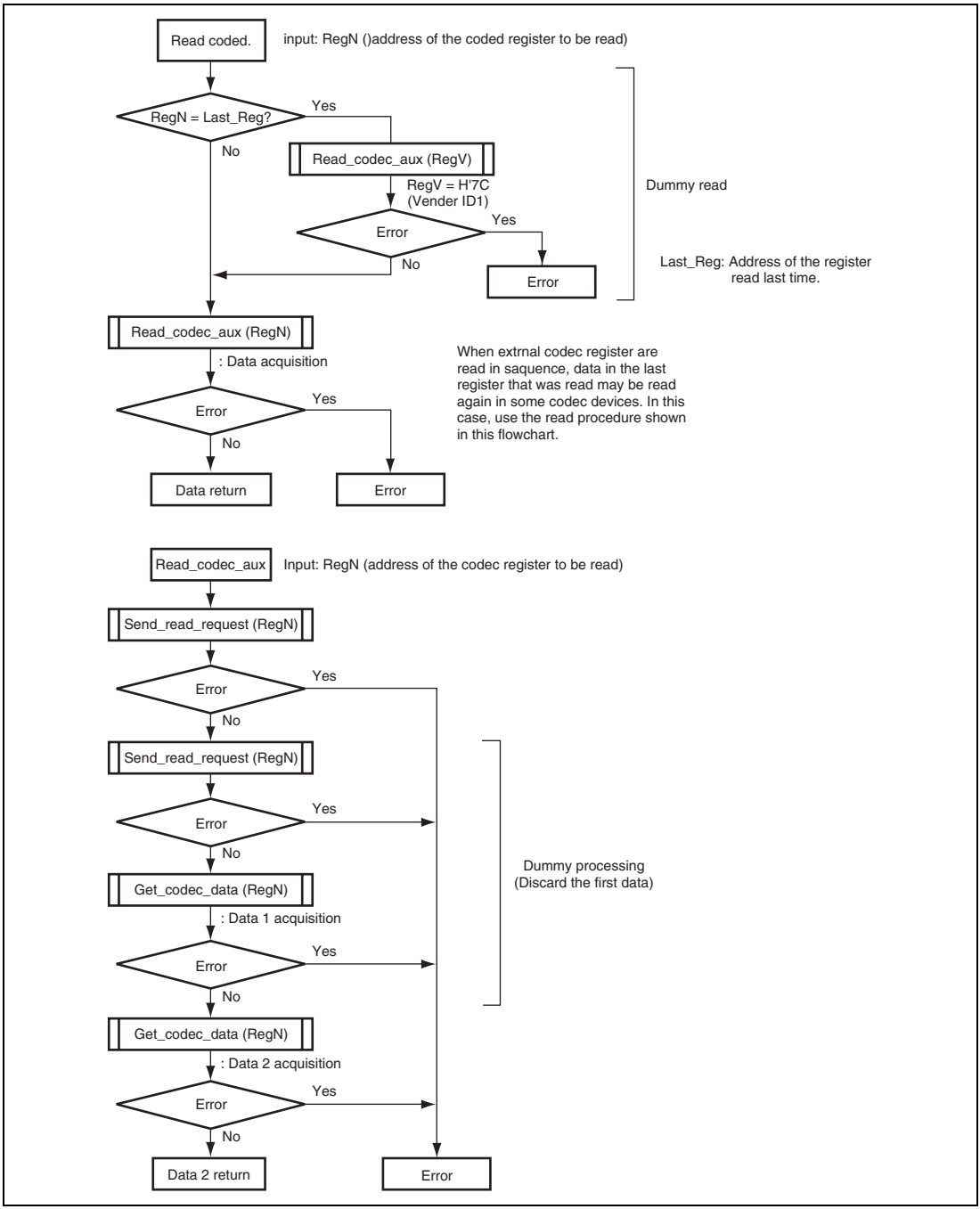


Figure 33.5 Sample Flowchart for Off-Chip Codec Register Read (1)

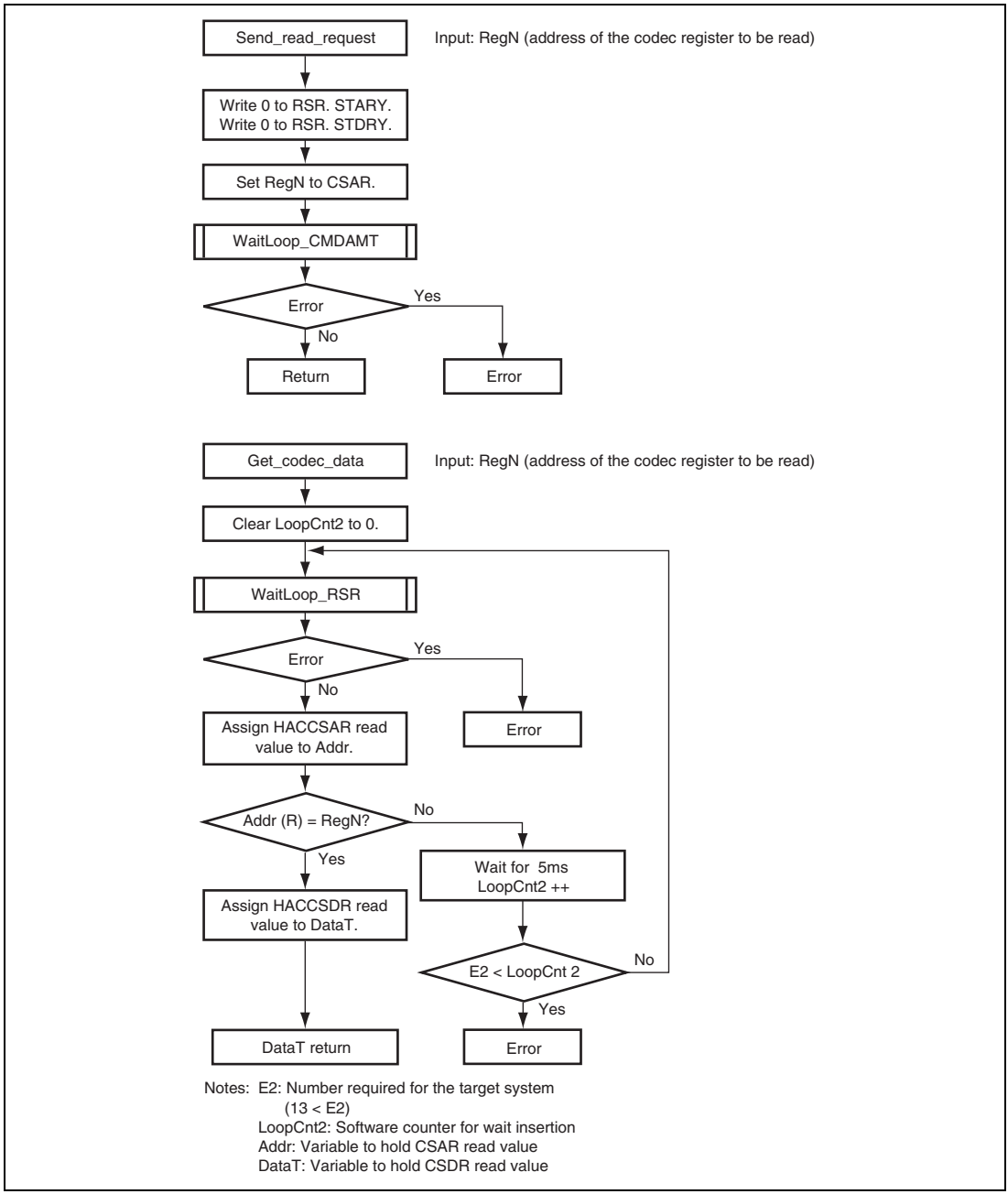
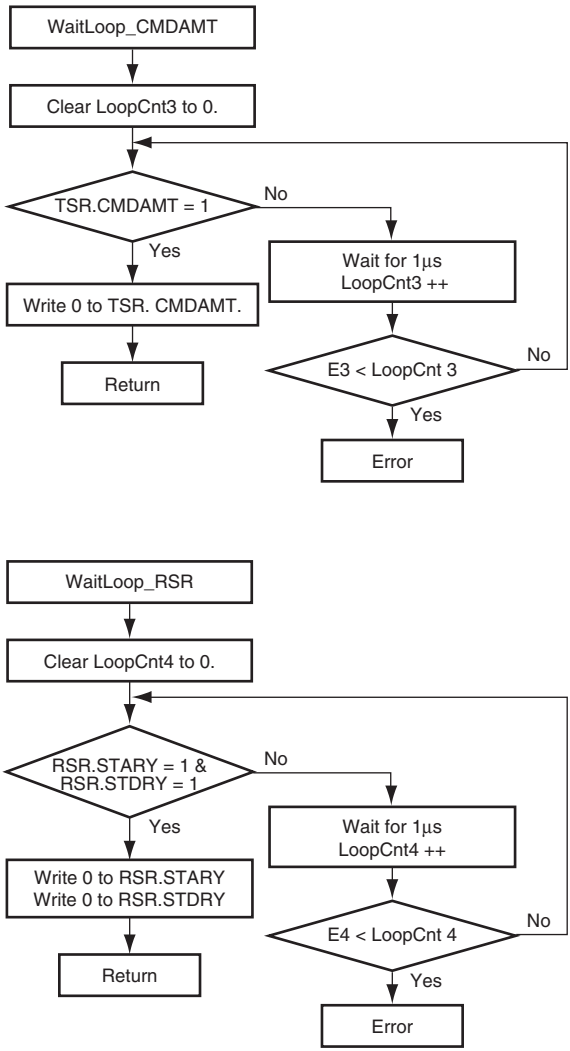


Figure 33.6 Sample Flowchart for Off-Chip Codec Register Read (2)



Notes: E3 and E4: Number required for the target system
(21 < E3, 21 < E4 < 1000)
LoopCnt3: Software counter for wait insertion
LoopCnt4: Software counter for wait insertion

Figure 33.7 Sample Flowchart for Off-Chip Codec Register Read (3)

33.5.6 Notes

The HAC_SYNC signal is generated by the HAC to indicate the position of slot 0 within a frame.

33.5.7 Reference

AC'97 Component Specification, Revision 2.1

Section 34 Serial Sound Interface (SSI)

The serial sound interface (SSI) is a module designed to send or receive audio data interface with a variety of devices offering Philips format compatibility. It also provides additional modes for other common formats, as well as support for a multi-channel mode.

34.1 Features

The SSI has the following features.

- Number of channels: Four channel
- Operating modes: Non-compressed mode
  The non-compressed mode supports all serial audio streams divided into channels.
- The SSI module is configured as any of a transmitter or receiver. The serial bus format can be used.
- Asynchronous transfer between the buffer and the shift register
- Division ratios of the serial bus interface clock can be selected.
- Data transmission/reception can be controlled from the DMAC or interrupt.

Figure 34.1 is a block diagram of the SSI module.

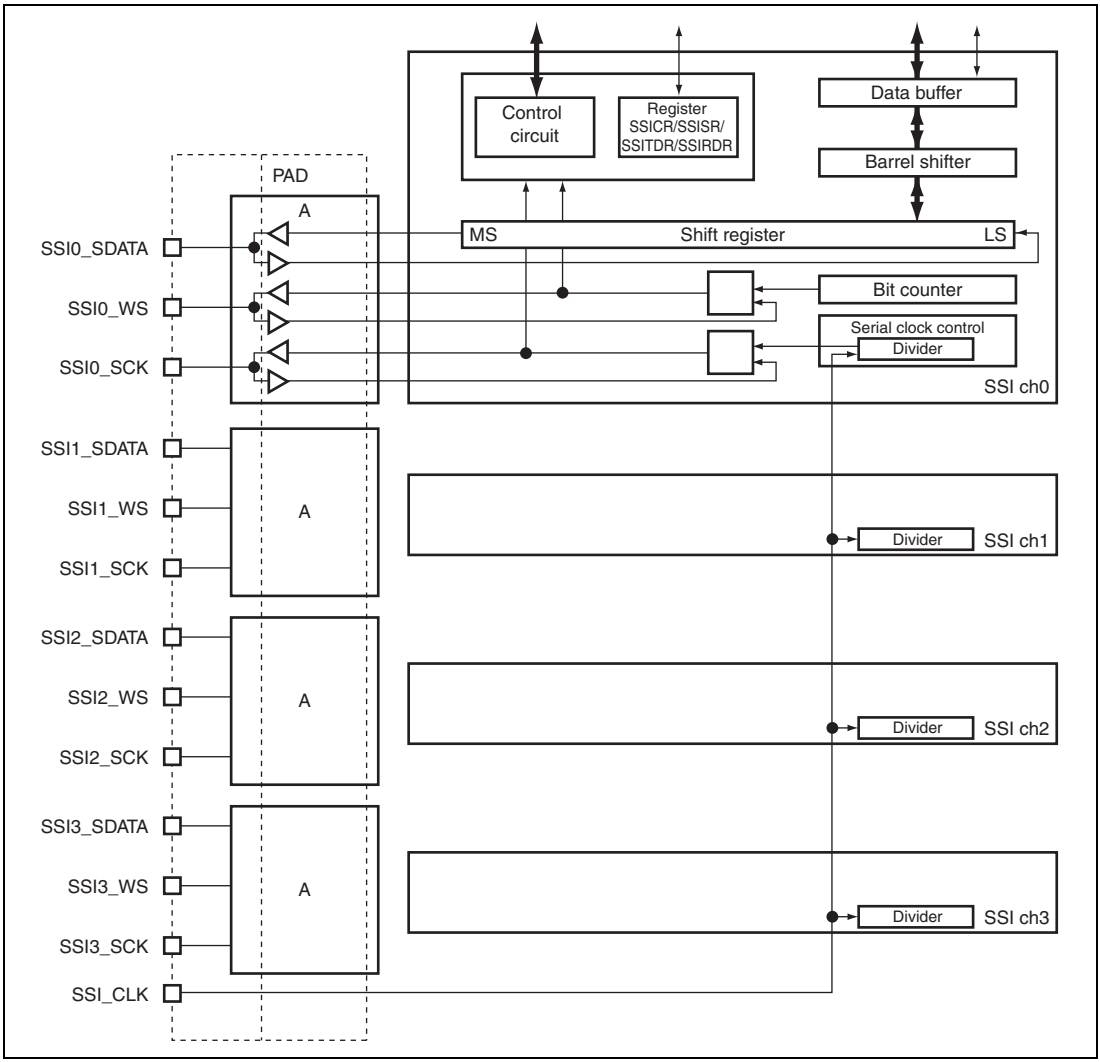


Figure 34.1 Block Diagram of SSI Module

34.2 Input/Output Pins

Table 34.1 lists the pin configurations relating to the SSI module.

Table 34.1 Pin Configuration

Channel	Pin Name	I/O	Function
Common	SSI_CLK	Input	Divider input clock (oversampling clock 256/384/512fs input)
0	SSI0_WS	I/O	Word select
	SSI0_SDATA	I/O	Serial data input/output
	SSI0_SCK	I/O	Serial bit clock
1	SSI1_WS	I/O	Word select
	SSI1_SDATA	I/O	Serial data input/output
	SSI1_SCK	I/O	Serial bit clock
2	SSI2_WS	I/O	Word select
	SSI2_SDATA	I/O	Serial data input/output
	SSI2_SCK	I/O	Serial bit clock
3	SSI3_WS	I/O	Word select
	SSI3_SDATA	I/O	Serial data input/output
	SSI3_SCK	I/O	Serial bit clock

34.3 Register Descriptions

Table 34.2 shows the SSI register configuration. Table 34.3 shows the register state in each operating mode.

Table 34.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Area P4 Address	Area 7 Address	Access Size
0	Control register 0	SSICR0	R/W	H'FFE5 0000	H'1FE5 0000	32
	Status register 0	SSISR0	R/W*	H'FFE5 0004	H'1FE5 0004	32
	Transmit data register 0	SSITDR0	R/W	H'FFE5 0008	H'1FE5 0008	32
	Receive data register 0	SSIRDR0	R	H'FFE5 000C	H'1FE5 000C	32
1	Control register 1	SSICR1	R/W	H'FFE5 8000	H'1FE5 8000	32
	Status register 1	SSISR1	R/W*	H'FFE5 8004	H'1FE5 8004	32
	Transmit data register 1	SSITDR1	R/W	H'FFE5 8008	H'1FE5 8008	32
	Receive data register 1	SSIRDR1	R	H'FFE5 800C	H'1FE5 800C	32
2	Control register 2	SSICR2	R/W	H'FFE6 0000	H'1FE6 0000	32
	Status register 2	SSISR2	R/W*	H'FFE6 0004	H'1FE6 0004	32
	Transmit data register 2	SSITDR2	R/W	H'FFE6 0008	H'1FE6 0008	32
	Receive data register 2	SSIRDR2	R	H'FFE6 000C	H'1FE6 000C	32
3	Control register 3	SSICR3	R/W	H'FFE6 8000	H'1FE6 8000	32
	Status register 3	SSISR3	R/W*	H'FFE6 8004	H'1FE6 8004	32
	Transmit data register 3	SSITDR3	R/W	H'FFE6 8008	H'1FE6 8008	32
	Receive data register 3	SSIRDR3	R	H'FFE6 800C	H'1FE6 800C	32

Note: * Bits 27 and 26 in the status register are readable/writable bits, and the other bits are read-only bits. For details, refer to section 34.3.2, Status Register (SSISR).

Table 34.3 Register State in Each Operating Mode

Channel	Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
0	Control register 0	SSICR0	H'0000 0000	H'0000 0000	Retained	Retained
	Status register 0	SSISR0	H'0010 A003	H'0x10 A00x	Retained	Retained
	Transmit data register 0	SSITDR0	H'0000 0000	H'0000 0000	Retained	Retained
	Receive data register 0	SSIRDR0	H'0000 0000	H'0000 0000	Retained	Retained
1	Control register 1	SSICR1	H'0000 0000	H'0000 0000	Retained	Retained
	Status register 1	SSISR1	H'0010 A003	H'0x10 A00x	Retained	Retained
	Transmit data register 1	SSITDR1	H'0000 0000	H'0000 0000	Retained	Retained
	Receive data register 1	SSIRDR1	H'0000 0000	H'0000 0000	Retained	Retained
2	Control register 2	SSICR2	H'0000 0000	H'0000 0000	Retained	Retained
	Status register 2	SSISR2	H'0010 A003	H'0x10 A00x	Retained	Retained
	Transmit data register 2	SSITDR2	H'0000 0000	H'0000 0000	Retained	Retained
	Receive data register 2	SSIRDR2	H'0000 0000	H'0000 0000	Retained	Retained
3	Control register 3	SSICR3	H'0000 0000	H'0000 0000	Retained	Retained
	Status register 3	SSISR3	H'0010 A003	H'0x10 A00x	Retained	Retained
	Transmit data register 3	SSITDR3	H'0000 0000	H'0000 0000	Retained	Retained
	Receive data register 3	SSIRDR3	H'0000 0000	H'0000 0000	Retained	Retained

34.3.1 Control Register (SSICR)

SSICR is a 32-bit readable/writable register that controls the IRQ, selects each polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	—	CKDV			MUEN	—	TRMD	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables or disables the DMA request. 0: DMA request disabled. 1: DMA request enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt disabled 1: Underflow interrupt enabled
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt disabled 1: Overflow interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle interrupt disabled 1: Idle interrupt enabled
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt disabled 1: Data interrupt enabled
23, 22	CHNL[1:0]	00	R/W	Channels These bits indicate the number of channels in each system word. 00: 1 channel per system word 01: 2 channels per system word 10: 3 channels per system word 11: 4 channels per system word
21 to 19	DWL[2:0]	000	R/W	Data Word Length These bits indicate the encoded number of bits in a data word. 000: 8 Bits 001: 16 Bits 010: 18 Bits 011: 20 Bits 100: 22 Bits 101: 24 Bits 110: 32 Bits 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	SWL[2:0]	000	R/W	System Word Length These bits indicate the encoded number of bits in a system word. 000: 8 Bits 001: 16 Bits 010: 24 Bits 011: 32 Bits 100: 48 Bits 101: 64 Bits 110: 128 Bits 111: 256 Bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial clock input, slave mode 1: Serial clock output, master mode Note: The SCKD and SWSD bits can be set both 0s or 1s ((0,0) or (1,1)). Other settings are prohibited.
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select input, slave mode 1: Serial word select output, master mode Note: The SCKD and SWSD bits can be set both 0s or 1s ((0,0) or (1,1)). Other settings are prohibited.

Bit	Bit Name	Initial Value	R/W	Description
13	SCKP	0	R/W	Serial Clock Polarity 0: SSI_WS and SSI_SDATA change on falling edge of SSI_SCK (sampled on rising edge of SCK) 1: SSI_WS and SSI_SDATA change on rising edge of SSI_SCK (sampled on falling edge of SCK)

	SCKP = 0	SCKP = 1
SSI_SDATA input sampling timing in receive mode (TRMD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_SDATA output change timing in transmit mode (TRMD = 1)	SSI_SCK falling edge	SSI_SCK rising edge
SSI_WS input sampling in slave mode (SWSD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_WS output change timing in master mode (SWSD = 1)	SSI_SCK falling edge	SSI_SCK rising edge

12	SWSP	0	R/W	Serial WS Polarity 0: SSI_WS is low for the first channel, high for the second channel 1: SSI_WS is high for the first channel, low for the second channel
11	SPDP	0	R/W	Serial Padding Polarity 0: Padding bits are low 1: Padding bits are high When MUEN = 1, padding bits are low. (The MUTE function is given priority)

Bit	Bit Name	Initial Value	R/W	Description
10	SDTA	0	R/W	Serial Data Alignment 0: Serial data is transmitted/ received first, followed by padding bits. 1: Padding bits are transmitted/ received first, followed by serial data.
9	PDTA	0	R/W	Parallel Data Alignment When the data word length is 32, 16 or 8 bit, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. 0: Parallel data (SSITDR, SSIRDR) is left-aligned 1: Parallel data (SSITDR, SSIRDR) is right-aligned. • DWL = 000 (with a data word length of 8 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted or received at each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is derived from bits 31 to 24. • DWL = 001 (with a data word length of 16 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted or received at each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is derived from bits 31 to 16.

Bit	Bit Name	Initial Value	R/W	Description
9 (cont.)	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 0 (left-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits 31 down to (32 minus the number of bits in the data word length specified by DWL). That is, If DWL = 011, the data word length is 20 bits; therefore, bits 31 to 12 in either SSIRDR or SSITDR are used. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 1 (right-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits (the number of bits in the data word length specified by DWL minus 1) to 0 i.e. if DWL = 011, then DWL = 20 and bits 19 to 0 are used in either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (with a data word length of 32 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV	All 0	R/W	Serial Oversampling Clock Division Ratio These bits define the division ratio between oversampling Clock (SSI_CLK) and the serial bit clock. These bits are ignored if SCKD = 0. The Serial Bit Clock is used for the shift register and is provided on the SSI_SCK pin. 000: (Serial bit clock frequency = oversampling clock frequency/1) 001: (Serial bit clock frequency = oversampling clock frequency/2) 010: (Serial bit clock frequency = oversampling clock frequency/4) 011: (Serial bit clock frequency = oversampling clock frequency/8) 100: (Serial bit clock frequency = oversampling clock frequency/16) 101: (Serial bit clock frequency = oversampling clock frequency/6) 110: (Serial bit clock frequency = oversampling clock frequency/12) 111: Setting prohibited
3	MUEN	0	R/W	Mute Enable 0: The SSI module is not muted 1: The SSI module is muted
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: The SSI module is in receive mode 1: The SSI module is in transmit mode
0	EN	0	R/W	SSI Module Enable 0: The SSI module is disabled 1: The SSI module is enabled

34.3.2 Status Register (SSISR)

SSISR is configured by status flags that indicate the operating status of the SSI module and bits that indicate the current channel number and word number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	—	—	—	—	—	—	—	—
Initial value:	—	—	—	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R/W*[1]	R/W*[1]	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO[1:0]		SWNO	IDST
Initial value:	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the status of the DMA request of SSI module. TRMD = 0 (Receive Mode): • If DMRQ = 1 then SSIRDR has unread data. • If SSIRDR is read then DMRQ = 0 until there is new unread data. TRMD = 1 (Transmit Mode): • If DMRQ = 1, SSITDR requests data to be written to continue the transmission onto the audio serial bus. • Once data is written to SSITDR then DMRQ = 0 until further transmit data is requested.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/W*1	Underflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a lower rate than the required rate. This bit is set to 1 regardless of the setting of UIEN bit. In order to clear it to 0, write 0 in it. If UIRQ = 1 and UIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If UIRQ = 1, it indicates that SSIRDR was read out before DMRQ and DIRQ bits would indicate the existence of new unread data. In this instance, the same received data may be stored twice by the host, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If UIRQ = 1, it indicates that the transmitted data was not written in SSITDR. By this, the same data may be transmitted one time too often, which can lead to destruction of multi-channel data. Consequently, erroneous SSI data will be output, which makes this error more serious than underflow in the receive mode. Note: When underflow error occurs, the data in the data buffer will be transmitted until the next data is written in.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W*1	Overflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a higher rate than the required rate. This bit is set to 1 regardless of the setting of OIEN bit. In order to clear it to 0, write 0 in it. If OIRQ = 1 and OIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If OIRQ = 1, it indicates that the previous unread data had not been read out before new unread data was written in SSIRDR. This may cause the loss of data, which can lead to destruction of multi-channel data. Note: When overflow error occurs, the data in the data buffer will be overwritten by the next data sent from the SSI interface. When TRMD = 1 (Transmit Mode): If OIRQ = 1, it indicates that SSITDR had data written in before the data in SSITDR was transferred to the shift register. This may cause the loss of data, which can lead to destruction of multi-channel data.
25	IIRQ	0*2	R	Idle Mode Interrupt Status Flag This status flag indicates whether the SSI module is in the idle status. This bit is set to 1 regardless of the setting of IIEN bit, so that polling will be possible. The interrupt can be masked by clearing IIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If IIRQ = 1 and IIEN = 1, then an interrupt will be generated. 0: The SSI module is not in the idle status. 1: The SSI module is in the idle status.

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the SSI module requires that data be either read out or written in. This bit is set to 1 regardless of the setting of DIEN bit, so that polling will be possible. The interrupt can be masked by clearing DIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If DIRQ = 1 and DIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): 0: No unread data exists in SSIRDR. 1: Unread data exists in SSIRDR. When TRMD = 1 (Transmit Mode): 0: The transmit buffer is full. 1: The transmit buffer is empty, and requires that data be written in SSITDR.
23 to 4	—	H'10A00	R	Reserved These bits are always read as H'10A00. The write value should always be 0.
3, 2	CHNO[1:0]	00	R	Channel Number The number indicates the current channel. When TRMD = 0 (Receive Mode): This bit indicates to which channel the current data in SSIRDR belongs. When the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates the data of which channel should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	Serial Word Number The number indicates the current word number. When TRMD = 0 (Receive Mode): This bit indicates which system word the current data in SSIRDR is. Regardless whether the data has been read out from SSIRDR, when the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates which system word should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.
0	IDST	1*[2]	R	Idle Mode Status Flag Indicates that the serial bus activity has ceased. This bit is cleared if EN = 1 and the Serial Bus is currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 and TRMD = 1): This bit is set to 1 if no more data has been written to SSITDR and the current system word has been completed. It can also be set to 1 by clearing the EN bit after sufficient data has been written to SSITDR to complete the system word currently being output. SSI = Serial bus master receiver (SWSD = 1 and TRMD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. SSI = slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external device stops the serial bus clock before the current system word is completed then this bit will never be set.

Note: 1. These bits are readable/writable bits. If writing 0, these bits are initialized, although writing 1 is ignored.
2. At manual reset, the bit is undefined.

34.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to SSITDR is transferred to the shift register as it is required for transmission. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit.

Reading this register will return the data in the buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

34.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores the received data.

Data in SSIRDR is transferred from the shift register as each data word is received. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit in SSICR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

34.4 Operation

34.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus formats can be one of eight major modes as shown in table 34.4.

Table 34.4 Bus Formats of SSI Module

Bus Format	TPMD	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compressed Slave Receiver	0	0	0	Control bits						Configuration bits								
Non-Compressed Slave Transmitter	1	0	0															
Non-Compressed Master Receiver	0	1	1															
Non-Compressed Master Transmitter	1	1	1															

34.4.2 Non-Compressed Modes

The non-compressed mode is designed to support all serial audio streams which are split into channels. It can support Philips, Sony and Matsushita modes as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(2) Slave Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(3) Master Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these signals is as defined in the SSI module. If the incoming data does not conform to the defined format then operation is not guaranteed.

(4) Master Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these signals is as defined in the configuration bits in the SSI module.

(5) Configuration Fields - Word Length Related

All configuration bits relating to the word length of SSICR are valid in non-compressed modes.

There are many configurations that the SSI module can support and it is not sensible to show all of the Serial Data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, and Matsushita.

1. Philips Format

Figures 34.2 and 34.3 show the supported Philips protocol both with padding and without. Padding occurs when the data word length is smaller than the system word length.

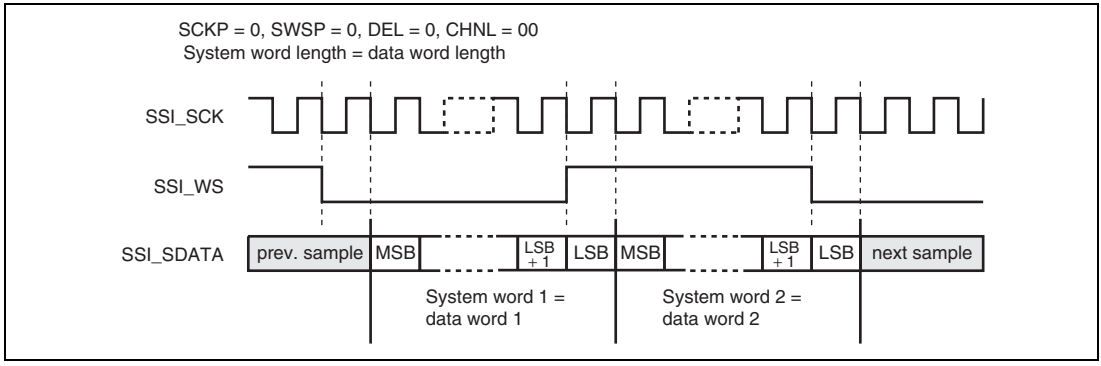


Figure 34.2 Philips Format (with no Padding)

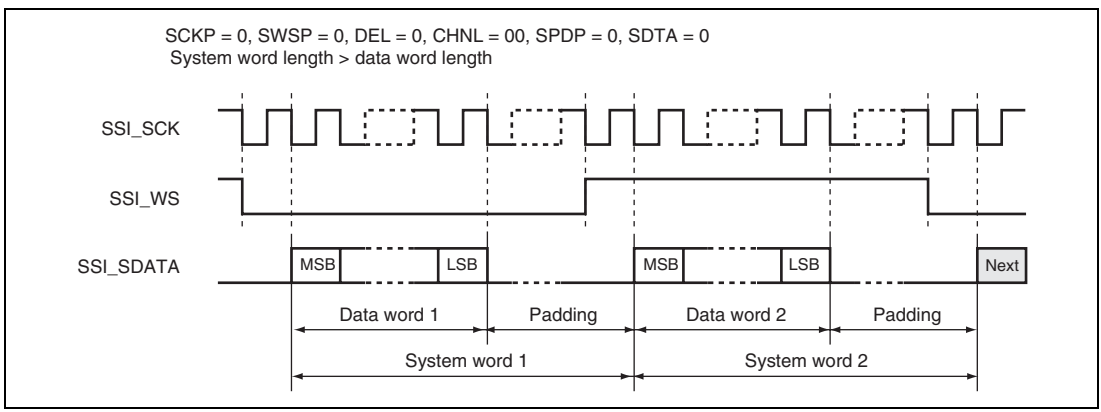


Figure 34.3 Philips Format (with Padding)

Figure 34.4 shows the format used by Sony. Figure 34.5 shows the format used by Matsushita. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

2. Sony Format

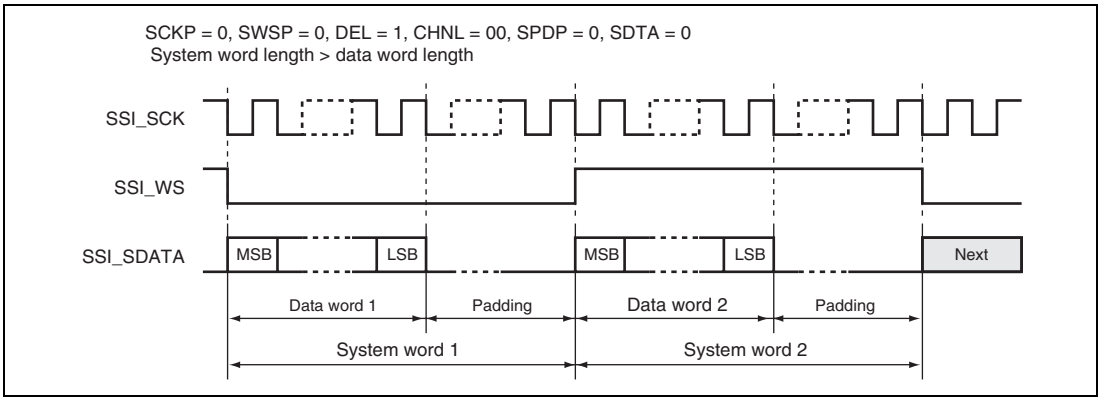


Figure 34.4 Sony Format (with Serial Data First, Followed by Padding Bits)

3. Matsushita Format

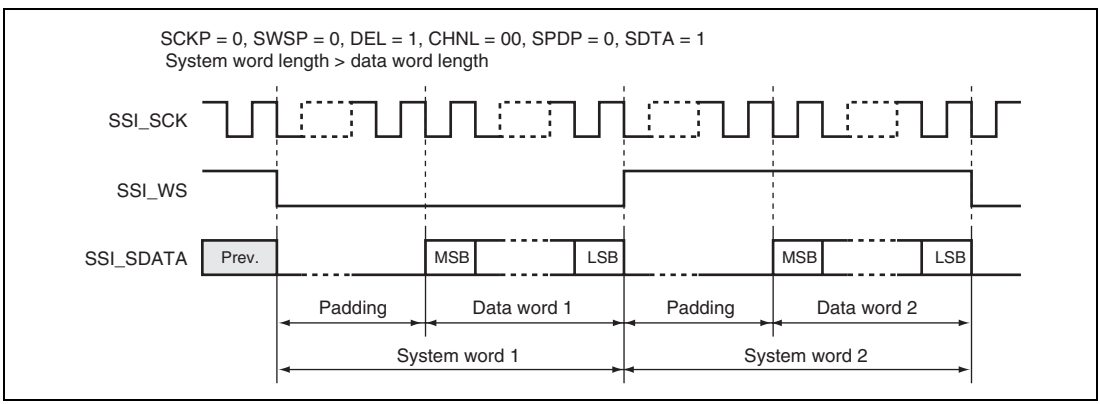


Figure 34.5 Matsushita Format (with Padding Bits First, Followed by Serial Data)

(6) Multi-Channel Formats

Some devices extend the definition of the specification by Philips and allow more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 2, 3 and 4 channels by the use of the CHNL, SWL and DWL bits. It is important that the system word length (SWL) is greater than or equal to the number of channels (CHNL) times the data word length (DWL).

Table 34.5 shows the number of padding bits for each of the valid configurations. If a setup is not valid it does not have a number in the following table and has instead a dash.

Table 34.5 Number of Padding Bits for Each Valid Configuration

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

In the case of the SSI module configured as a transmitter then each word that is written to SSITDR is transmitted in order on the serial audio bus.

In the case of the SSI module configured as a receiver each word received on the Serial Audio Bus is presented for reading in order by SSIRDR.

Figures 34.6 to 34.8 show how 2, 3 and 4 channels are transferred on the serial audio bus.

Figures 34.6 to 34.8 show how data on multiple channels (2, 3 or 4 channels) is transferred on the serial audio bus. Figure 34.6 shows the data transfer using no padding bits. Figure 34.7 shows the data transfer in which data is left-aligned with padding bits. Figure 34.8 shows the data transfer in which data is right-aligned with padding bits. This selection is purely arbitrary.

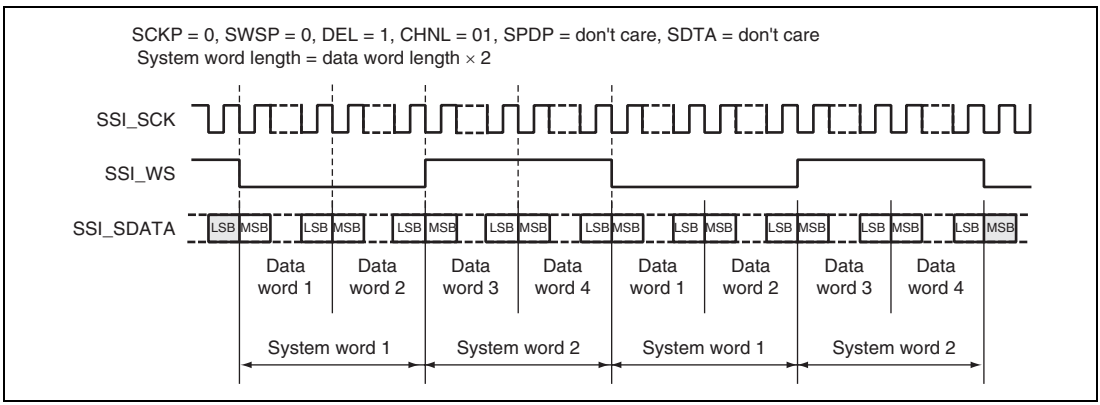


Figure 34.6 Multichannel Format (2 Channels, No Padding)

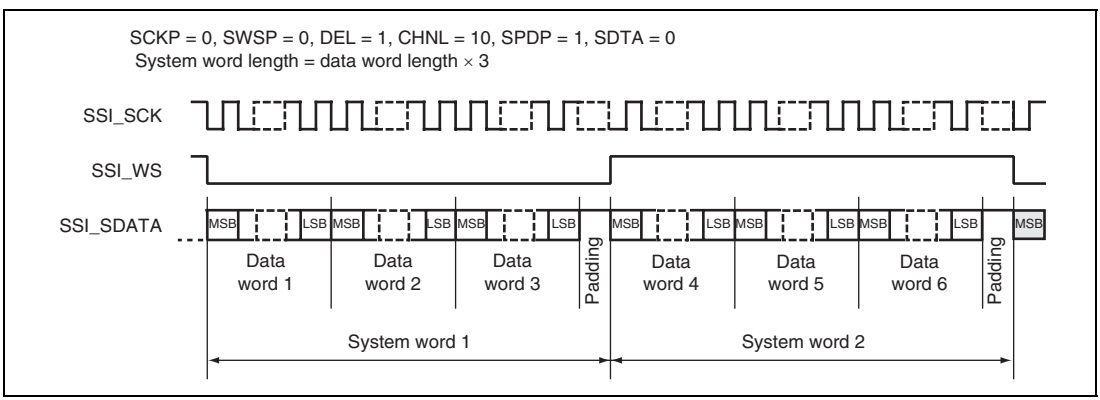


Figure 34.7 Multichannel Format (3 Channels with High Padding)

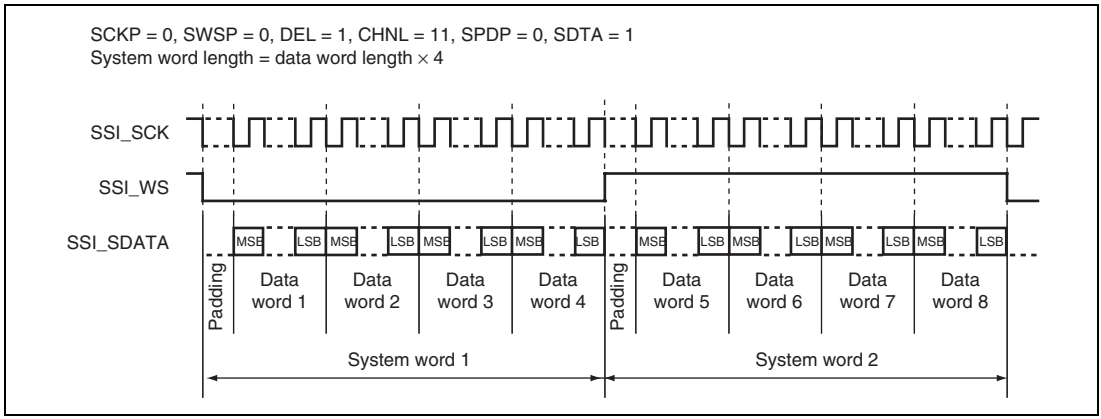


Figure 34.8 Multichannel Format (4 Channels, with Padding Bits First, Followed by Serial Data, with Padding)

(7) Configuration Fields - Signal Format Fields

There are several more configuration bits in non-compressed mode which will now be demonstrated. These bits are NOT mutually exclusive, however some configurations will probably not be useful for any other device.

They are demonstrated by referring to the following basic sample format shown in figure 34.9.

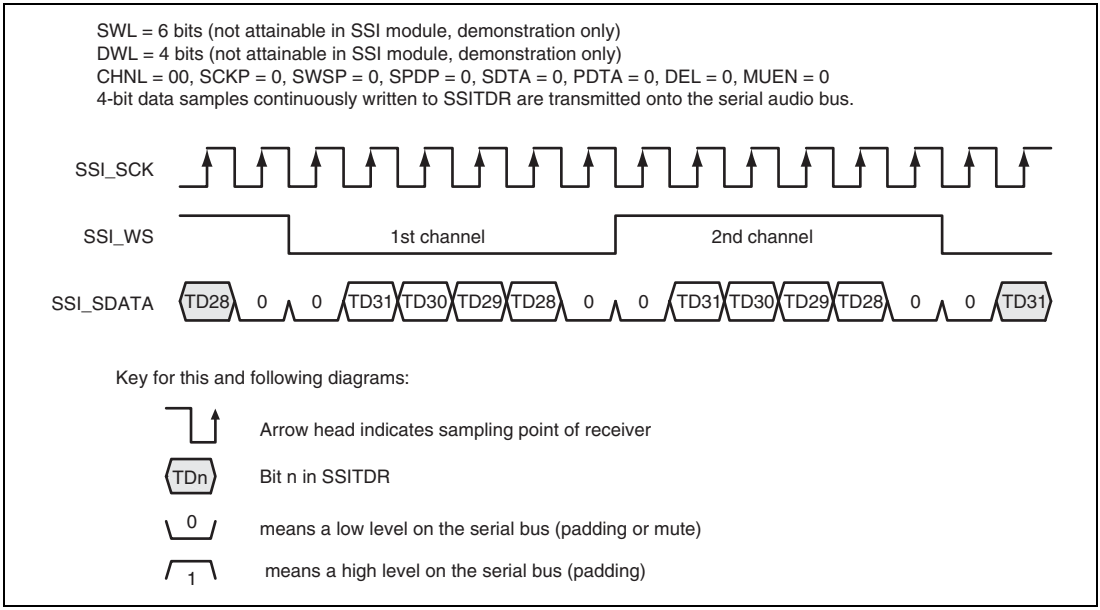


Figure 34.9 Basic Sample Format
(Transmit Mode with Example System/Data Word Length)

In figure 34.9, system word length of 6 bits and a data word length of 4 bits are used. Neither of these are possible with the SSI module but are used only for clarification of the other configuration bits.

1. Inverted Clock

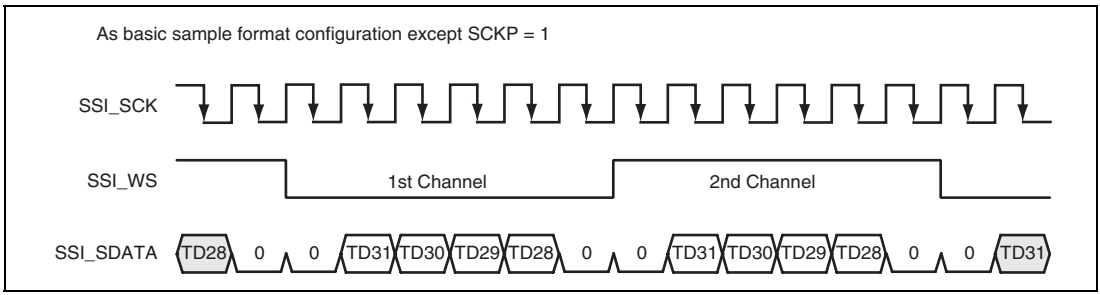


Figure 34.10 Inverted Clock

2. Inverted Word Select

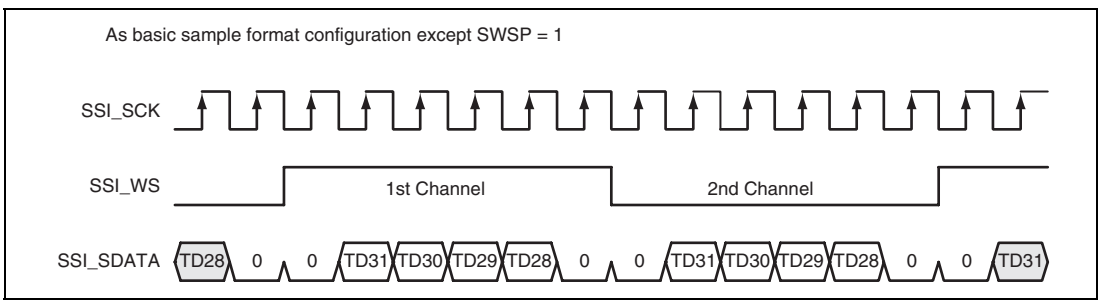


Figure 34.11 Inverted Word Select

3. Inverted Padding Polarity

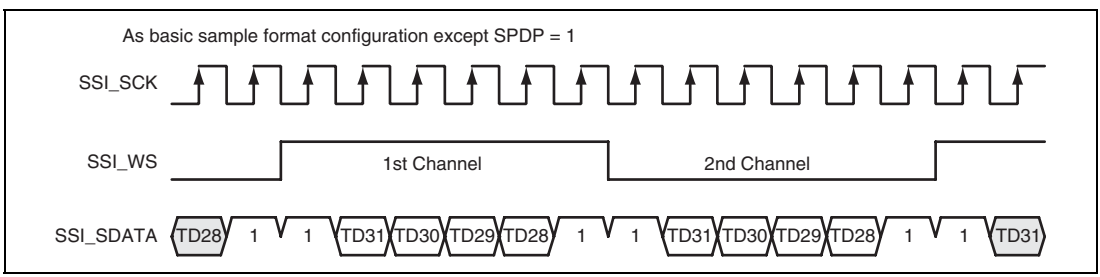


Figure 34.12 Inverted Padding Polarity

4. Padding Bits First, Followed by Serial Data, with Delay

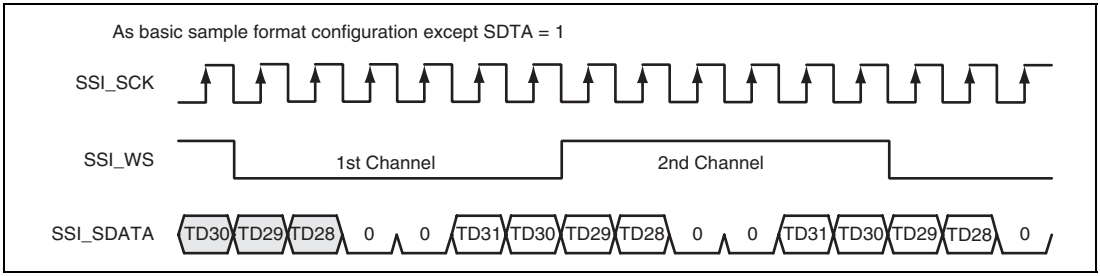


Figure 34.13 Padding Bits First, Followed by Serial Data, with Delay

5. Padding Bits First, Followed by Serial Data, without Delay

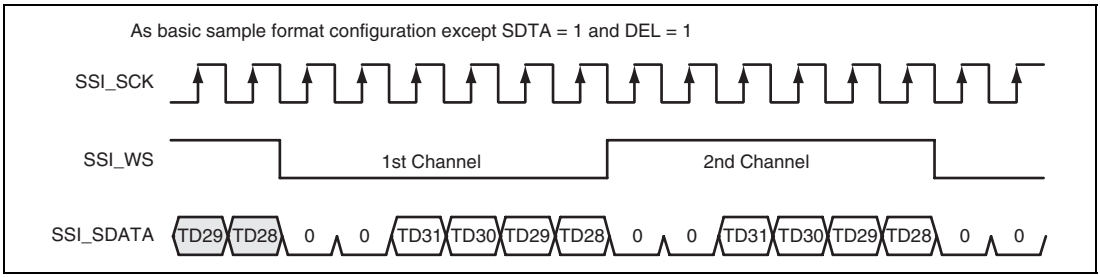


Figure 34.14 Padding Bits First, Followed by Serial Data, without Delay

6. Serial Data First, Followed by Padding Bits, without Delay

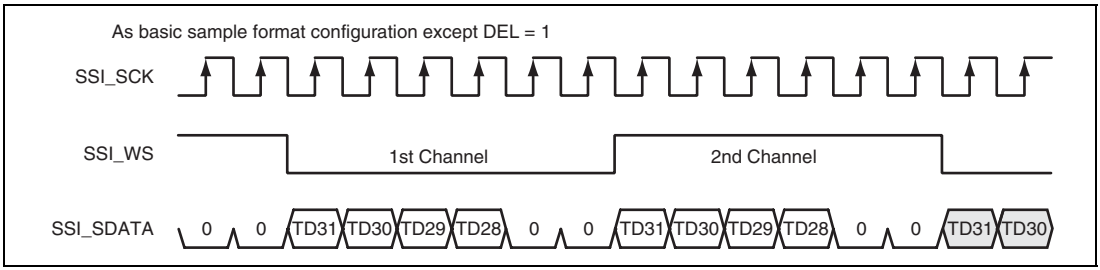


Figure 34.15 Serial Data First, Followed by Padding Bits, without Delay

7. Parallel Right Aligned with Delay

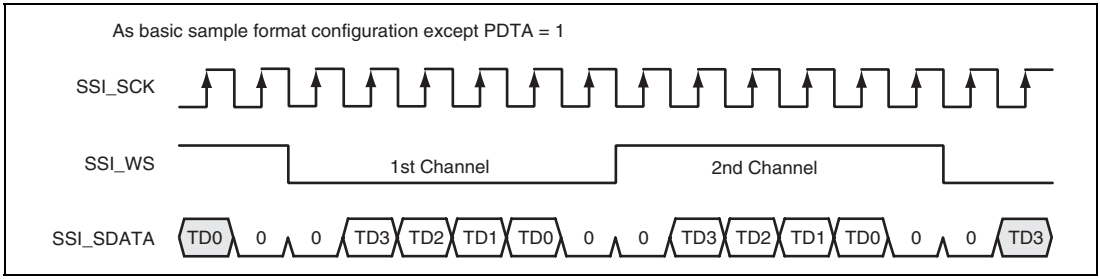


Figure 34.16 Parallel Right Aligned with Delay

8. Mute Enabled

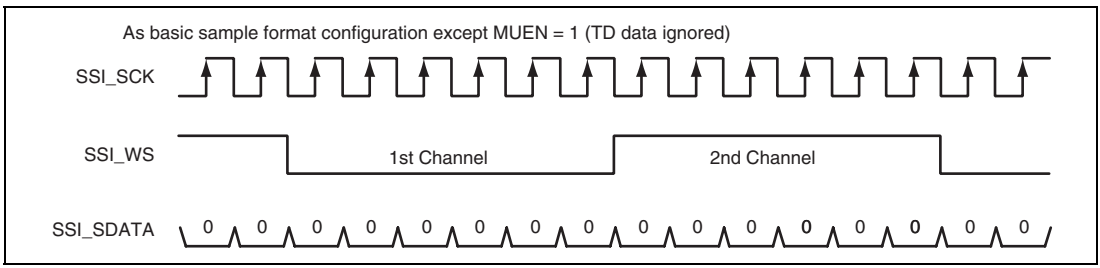


Figure 34.17 Mute Enabled

34.4.3 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 34.18 shows the transition diagram between these operation modes.

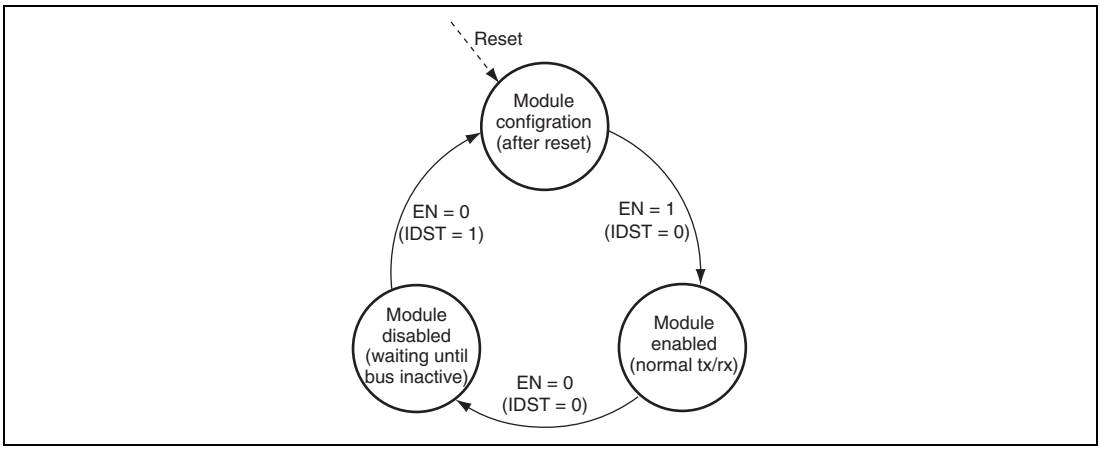


Figure 34.18 Transition Diagram between Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required settings in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

Setting the EN bit causes the SSI module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode depends on the selected operating mode. For details, see section 34.4.4, Transmit Operation and section 34.4.5, Receive Operation.

34.4.4 Transmit Operation

Transmission can be controlled in one of two ways: either DMA or an interrupt driven.

DMA driven is preferred to reduce the CPU load. In DMA control mode, an underflow or overflow of data or DMAC transfer end is notified by using an interrupt.

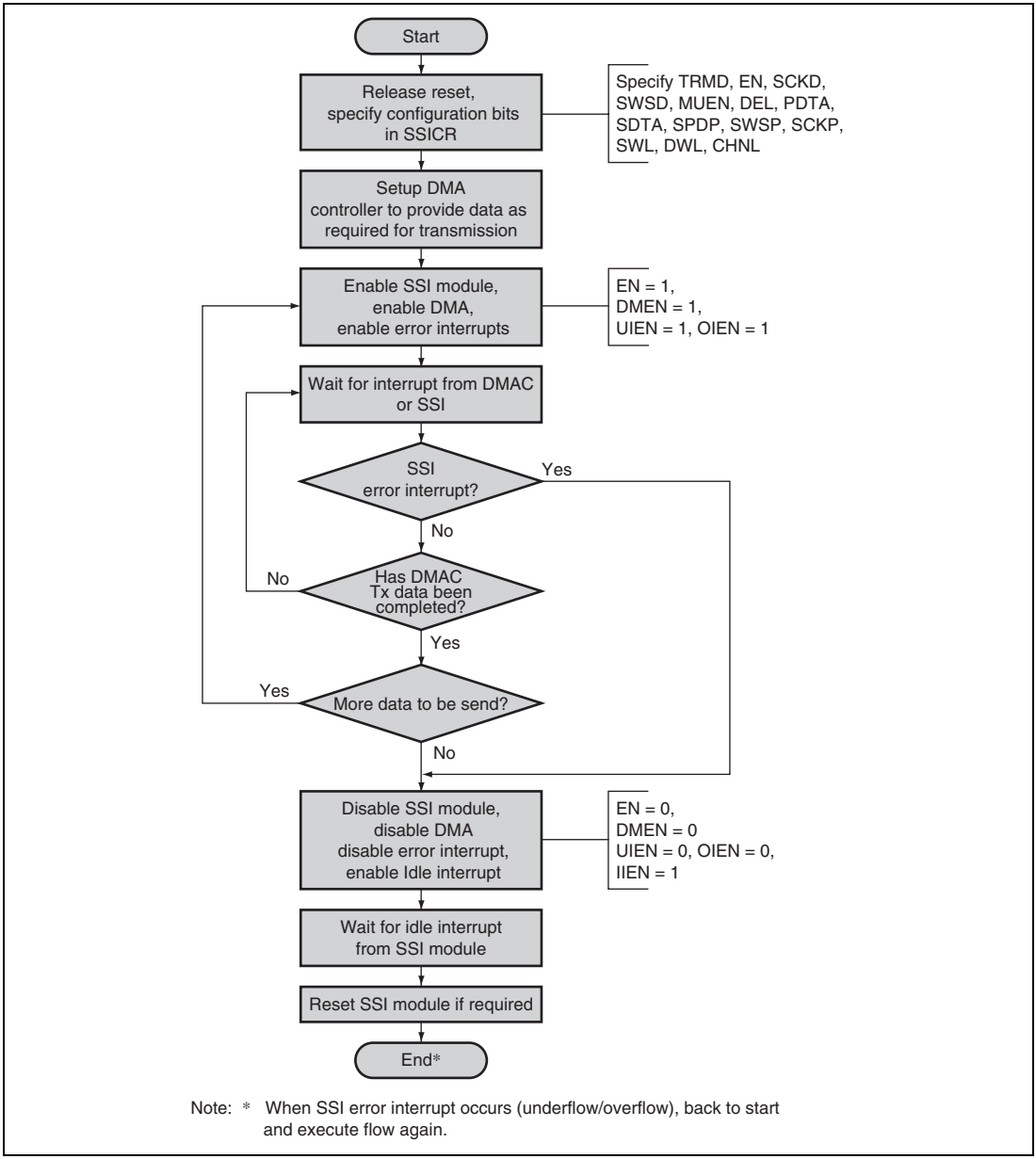
The alternative is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the SSI module is only double buffered and will require data to be written at least every system word period.

When disabling the SSI module, the SSI clock* must be supplied continuously until the module enters in the idle state, indicated by the IIRQ bit.

Figure 34.19 shows the transmit operation in the DMA controller mode. Figure 34.20 shows the transmit operation in the Interrupt controller mode.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin

(1) Transmission Using DMA Controller



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 34.19 Transmission Using DMA Controller

(2) Transmission using Interrupt Data Flow Control

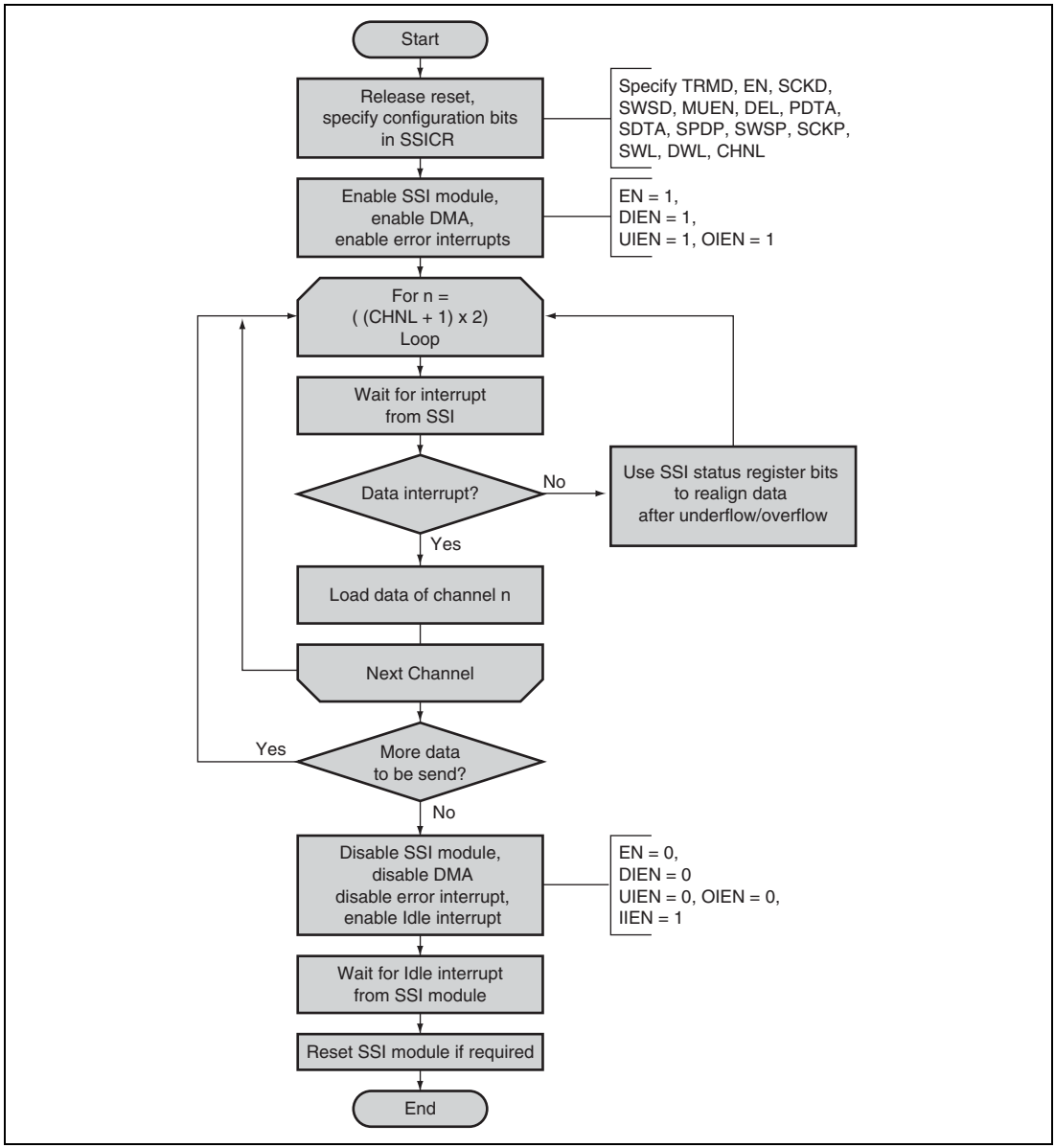


Figure 34.20 Transmission using Interrupt Data Flow Control

34.4.5 Receive Operation

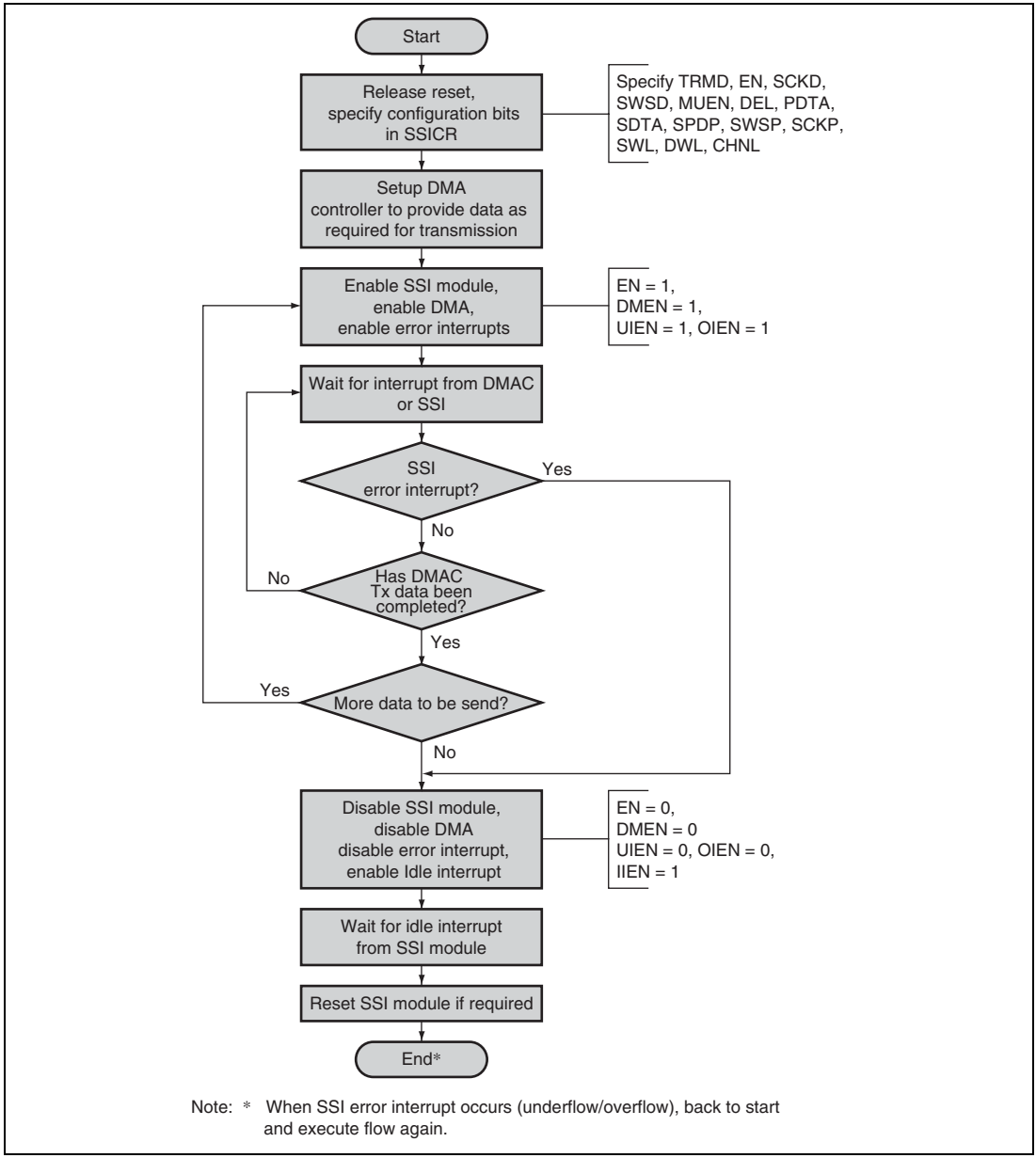
As with transmission the reception can be controlled in one of two ways: either DMA or an interrupt driven.

Figures 34.21 and 34.22 show the flow of operation.

When disabling the SSI module, the SSI clock must be supplied continuously until the module enters in the idle state, which is indicated by the IIRQ bit.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin

(1) Reception Using DMA Controller



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 34.21 Reception using DMA Controller

(2) Reception using Interrupt Data Flow Control

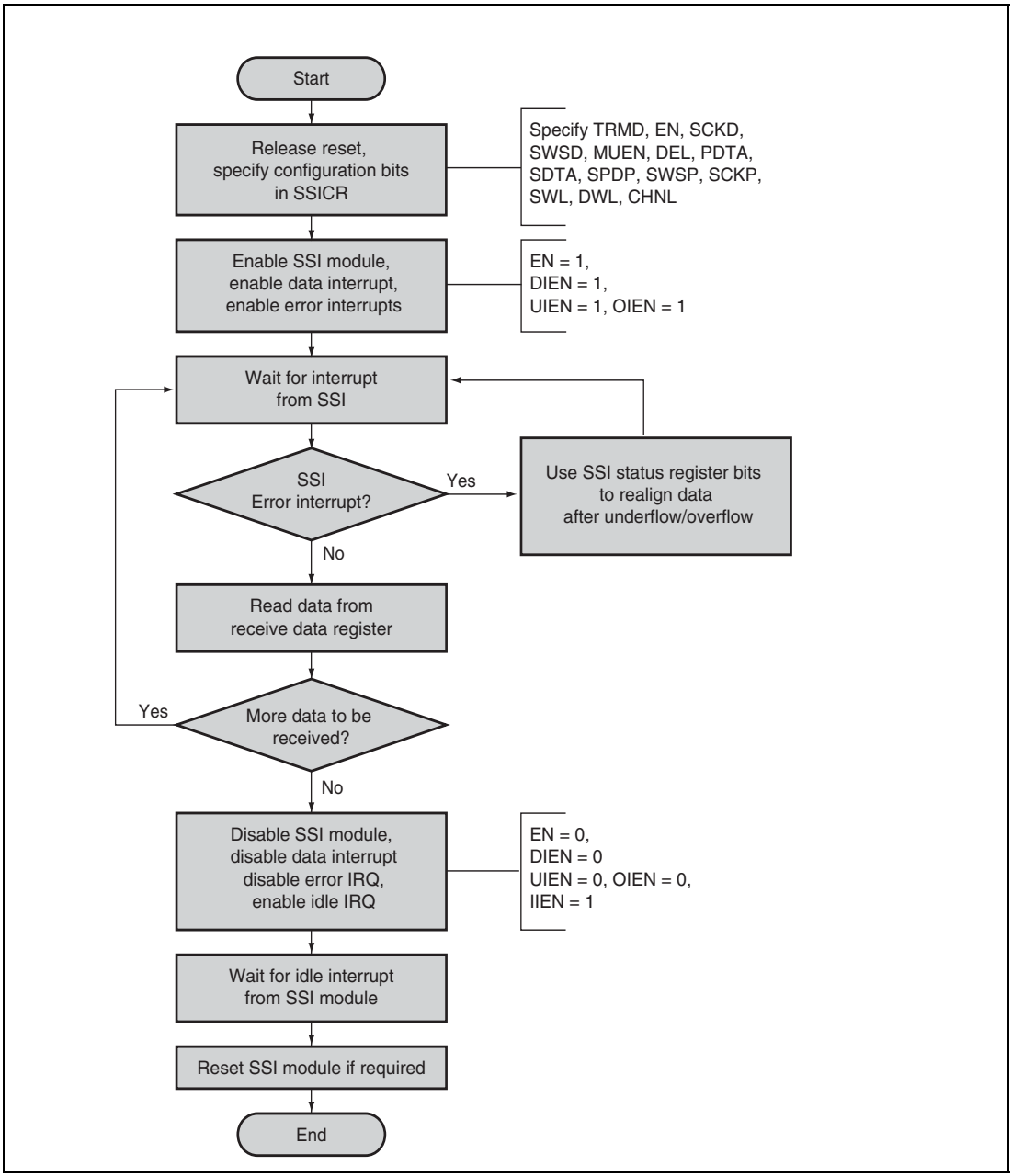


Figure 34.22 Reception using Interrupt Data Flow Control

When an underflow or overflow error condition is met, the CHNO[1:0] and SWNO bits can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the host CPU can read the number of channels and the number of system words to determine what point the serial audio stream has reached. In the transmitter case, the host CPU can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case, the host CPU can skip forward storing null sample data until it is ready to store the sample data that the SSI module is indicating that it will receive next to ensure consistency of the number of received data, and so resynchronize with the audio data stream.

34.4.6 Serial Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave mode, then the bit clock that is used in the shift register is derived from the SSI_SCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSI Module is in clock master mode, and the shift register uses the bit clock derived from the SSI_CLK input pin or its clock divided. This input clock is then divided by the ratio in the serial oversampling clock division ratio (CKDV) bit in SSICR and used as the bit clock in the shift register.

In either case, the SSI_SCK pin output is the same as the bit clock.

34.5 Usage Note

34.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation

If an overflow occurs during receive DMA operation, the module must be reactivated.

The receive buffer of SSI has 32-bit common register both left channel and right channel. If an overflow occurs under the condition of control register (SSICR) data-word length (DWL2 to DWL0) is 32-bit and system-word length (SWL2 to SWL0) is 32-bit, SSI has received the data at right channel that should be received at left channel.

If an overflow occurs through an overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR), disable the DMA transfer of the SSI to halt its operation by writing 0 to the EN bit and DMEN bit in SSICR (then terminate the DMA setting). And clear the overflow status flag by writing 0 to the OIRQ bit, set the DMA again and transfer restart.

34.5.2 Restrictions for Operation in Slave Mode

To terminate data transfer while this LSI is used in slave mode, clear the EN bit in SSICR to 0 to terminate data transfer before the word select signal supply is stopped.

In slave mode, data transfer is terminated if the EN bit (settings for terminating data transfer) is cleared and the falling edge of the word select signal (SSI_WS) is detected. If the word select signal supply is stopped before EN bit clear, the falling edge of the word select signal cannot be detected, and thereby data transfer is not terminated properly.

Section 35 USB Host Controller (USBH)

The USB host controller embedded in this LSI has a root hub and a one-port USB transceiver, and operates in Full speed mode. Open HCI interfaces and registers are also embedded in this LSI.

For the development of software, refer to the Open HCI specifications as well.

35.1 Features

- Support the Open HCI interface.
- Support the USB host interface.
- Root Hub function (although it supports only one port)
- Operate in Full speed mode (12Mbps) and low speed mode (1.5 Mbps)
- Support Overcurrent detection and Power source enable management.

A block diagram of the USBH is shown in figure 35.1.

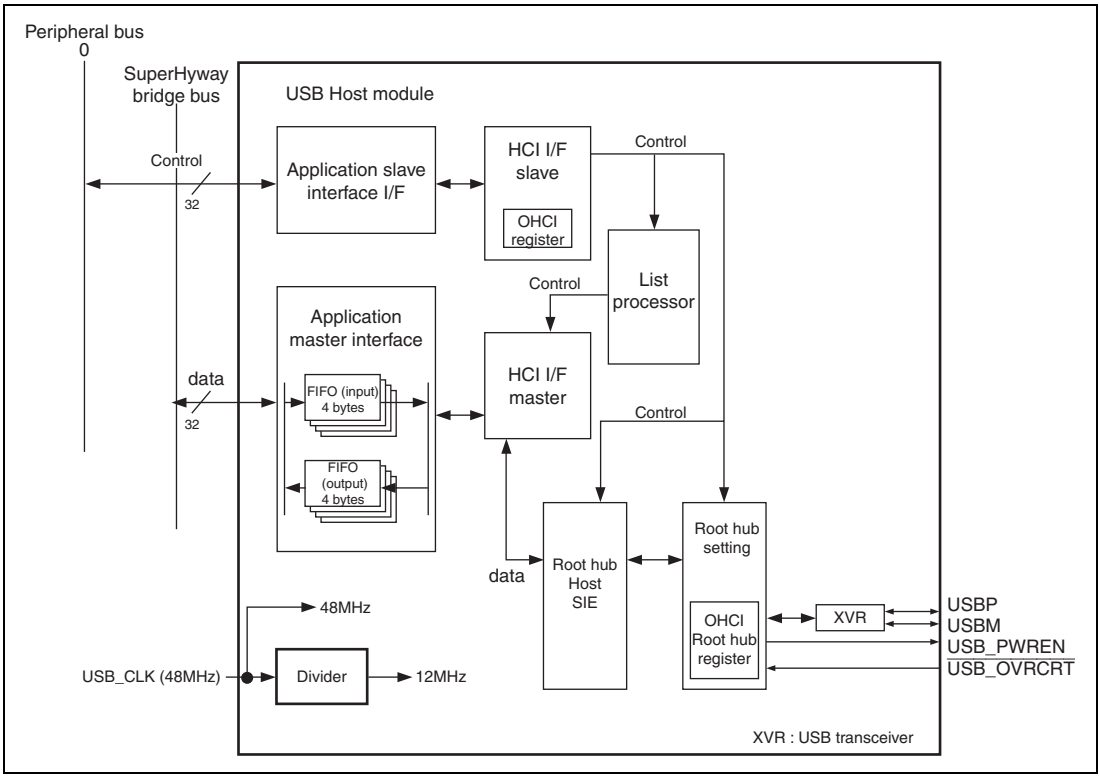


Figure 35.1 Block Diagram of USBH

35.2 Pin Description

Table 35.1 shows the pin configuration of the USB host module.

Table 35.1 USB Host Pin Assignment

Name	Function	I/O	Function
USBP	D+	I/O	USB port D+
USBM	D-	I/O	USB port D-
USB_PWREN	USB port power source enable	Output	USB port power source enable management
$\overline{\text{USB_OVRCRT}}$	USB port overcurrent detection	I/O	USB port overcurrent detection This is used to detect an overcurrent at low level and normal operation at high level.
USB_CLK	Clock pin	Input	USB clock input pin (48 MHz input)*

Note: * USB_CLK should be slower than Pck0.

35.3 Register Description

Table 35.2 shows the USBH register configuration. Table 35.3 shows the register state in each operating mode. Other than the ConfigulationControl register, all registers conform to the OpenHCIs specification. For details, refer to the OpenHCI Rev1.0. The Configuration Control register is only for this LSI.

Table 35.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
HcRevision register	USBHR	R	H'FFEC 8000	H'1FEC 8000	32
HcControl register	USBHC	R/W	H'FFEC 8004	H'1FEC 8004	32
HcCommandStatus register	USBHCS	R/W	H'FFEC 8008	H'1FEC 8008	32
HcInterruptStatus register	USBHIS	R/W	H'FFEC 800C	H'1FEC 800C	32
HcInterruptEnable register	USBHIE	R/W	H'FFEC 8010	H'1FEC 8010	32
HcInterruptDisable register	USBHID	R/W	H'FFEC 8014	H'1FEC 8014	32
HcHCCA register	USBHHCCA	R/W	H'FFEC 8018	H'1FEC 8018	32
HcPeriodCurrentED register	USBHPCED	R/W	H'FFEC 801C	H'1FEC 801C	32
HcControlHeadED register	USBHCHED	R/W	H'FFEC 8020	H'1FEC 8020	32
HcControlCurrentED register	USBHCCED	R/W	H'FFEC 8024	H'1FEC 8024	32
HcBulkHeadED register	USBHBHED	R/W	H'FFEC 8028	H'1FEC 8028	32
HcBulkCurrentED register	USBHBCED	R/W	H'FFEC 802C	H'1FEC 802C	32
HcDoneHead register	USBHDHED	R/W	H'FFEC 8030	H'1FEC 8030	32
HcFmInterval register	USBHFI	R/W	H'FFEC 8034	H'1FEC 8034	32
HcFmRemaining register	USBHFR	R	H'FFEC 8038	H'1FEC 8038	32
HcFmNumber register	USBHFN	R	H'FFEC 803C	H'1FEC 803C	32
HcPeriodicStart register	USBHPS	R/W	H'FFEC 8040	H'1FEC 8040	32
HcLSThreshold register	USBHLST	R/W	H'FFEC 8044	H'1FEC 8044	32
HcRhDescriptorA register	USBHRDA	R/W	H'FFEC 8048	H'1FEC 8048	32
HcRhDescriptorB register	USBHRDB	R/W	H'FFEC 804C	H'1FEC 804C	32

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
HcRhStatus register	USBHRS	R/W	H'FFEC 8050	H'1FEC 8050	32
HcRhPortStatus[2] register	USBHRPS2	R/W	H'FFEC 8058	H'1FEC 8058	32
ConfigurationControl register	USBHSC	R/W	H'FFEC 80F0	H'1FEC 80F0	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 35.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
HcRevision register	USBHR	H'00000010	H'00000010	Retained	Retained
HcControl register	USBHC	H'00000000	H'00000000	Retained	Retained
HcCommandStatus register	USBHCS	H'00000000	H'00000000	Retained	Retained
HcInterruptStatus register	USBHIS	H'00000000	H'00000000	Retained	Retained
HcInterruptEnable register	USBHIE	H'00000000	H'00000000	Retained	Retained
HcInterruptDisable register	USBHID	H'00000000	H'00000000	Retained	Retained
HcHCCA register	USBHHCCA	H'00000000	H'00000000	Retained	Retained
HcPeriodCurrentED register	USBHPCED	H'00000000	H'00000000	Retained	Retained
HcControlHeadED register	USBHCHED	H'00000000	H'00000000	Retained	Retained
HcControlCurrentED register	USBHCCED	H'00000000	H'00000000	Retained	Retained
HcBulkHeadED register	USBHBHED	H'00000000	H'00000000	Retained	Retained
HcBulkCurrentED register	USBHBCED	H'00000000	H'00000000	Retained	Retained
HcDoneHead register	USBHDHED	H'00000000	H'00000000	Retained	Retained
HcFmInterval register	USBHFI	H'00002EDF	H'00002EDF	Retained	Retained
HcFmRemaining register	USBHFR	H'00000000	H'00000000	Retained	Retained
HcFmNumber register	USBHFN	H'00000000	H'00000000	Retained	Retained
HcPeriodicStart register	USBHPS	H'00000000	H'00000000	Retained	Retained
HcLSThreshold register	USBHLST	H'00000628	H'00000628	Retained	Retained
HcRhDescriptorA register	USBHRDA	H'02001002	H'02001002	Retained	Retained
HcRhDescriptorB register	USBHRDB	H'00000000	H'00000000	Retained	Retained
HcRhStatus register	USBHRS	H'00000000	H'00000000	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
HcRhPortStatus[2] register	USBHRPS2	H'00000100	H'00000100	Retained	Retained
ConfigurationControl register	USBHSC	H'00000001	H'00000001	Retained	Retained

35.3.1 HcRevision Register (USBHR)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	REV[7:0]							
Initial value :	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W :	—	—	—	—	—	—	—	—	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0
7 to 0	REV[7:0]	H'10	R	Revision Indicates the OpenHCI Specification revision number implemented by the Hardware. (X.Y = H'XY) USB Host Controller supports the Open HCI1.0 specification.

35.3.2 HcControl Register (USBHC)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RWE	RWC	IR	HCFS[1:0]		BLE	CLE	IE	PLE	CBSR[1:0]	
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
10	RWE	0	R/W	RemoteWakeupConnectedEnable If a remote wakeup signal is supported, this bit enables that operation. Since remote wakeup signal is not supported, this bit is ignored.
9	RWC	0	R/W	RemoteWakeupConnected This bit indicates whether the Host Controller(HC) supports a remote wakeup signal.
8	IR	0	R/W	InterruptRouting This bit specifies interrupt routing: 0: Interrupts routed to normal interrupt processing unit (INT). 1: Interrupts routed to SMI.
7, 6	HCFS[1:0]	All 0	R/W	HostControllerFunctionalState These bits set the Host Controller state. The state encodings are: 00: UsbReset 01: UsbResume 10: UsbOperational 11: UsbSuspend The Host Controller may force a state change from UsbSuspend to UsbResume after detecting resume signaling from a downstream port.

Bit	Bit Name	Initial Value	R/W	Description
5	BLE	0	R/W	BulkListEnable When set, this bit enables processing of the Bulk list.
4	CLE	0	R/W	ControlListEnable This bit is set to enable the processing of the control list in the next frame. If cleared by HCD, the processing of the control list is not carried out after next SOF. The host controller must check this bit whenever the list will be processed. When disabling, HDC can correct the list. When USBHCCED indicates ED to be deleted, HCD should hasten the pointer by updating USBHCCED before re-enabling the list processing. 0: Control list processing is not carried out 1: Control list processing is carried out
3	IE	0	R/W	IsochronousEnable When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.
2	PLE	0	R/W	PeriodicListEnable When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
1, 0	CBSR[1:0]	00	R/W	ControlBulkServiceRatio Specify the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. '00' = 1 Control Endpoint; '11' = 4 Control Endpoints)

35.3.3 HcCommandStatus Register (USBHCS)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOC[1:0]	
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BLF	CLF	HCR
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
17, 16	SOC[1:0]	All 0	R	ScheduleOverrunCount These bits increment every time the SchedulingOverrun bit in HcInterruptStatus is set. The count wraps from 11 to 00.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
2	BLF	0	R/W	BulkListFilled When set, this bit indicates there is an active ED on the Bulk List. The bit can be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Bulk List.
1	CLF	0	R/W	ControlListFilled When set, this bit indicates there is an active ED on the Control List. The bit can be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Control List.

Bit	Bit Name	Initial Value	R/W	Description
0	HCR	0	R/W	HostControllerReset This bit is set to initiate a software reset. This bit is cleared by the Host Controller upon completion of the reset operation.

35.3.4 HcInterruptStatus Register (USBHIS)

All bits are set by hardware and cleared by software.

These bits in this register can be cleared by writing 1 to bit positions to be cleared.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	RHSC	FNO	UE	RD	SF	WDH	SO
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	RHSC	0	R/W	RootHubStatusChange This bit is set when the content of HcRhStatus register or the content of any HcRhPortStatus register has changed.
5	FNO	0	R/W	FrameNumberOverflow This bit is set when bit 15 of FrameNumber changes value from 0 to 1 or from 1 to 0.
4	UE	0	R/W	UnrecoverableError This bit is set when HC detects a system error that is not USB related.
3	RD	0	R/W	ResumeDetected This bit is set when the Host Controller detects resume signaling on a downstream port.

Bit	Bit Name	Initial Value	R/W	Description
2	SF	0	R/W	StartofFrame This bit is set when the Frame manager signals a Start of Frame's event.
1	WDH	0	R/W	WritebackDoneHead This bit is set after the Host Controller has written the value of HcDoneHead register to HccaDoneHead.
0	SO	0	R/W	SchedulingOverrun This bit is set when the List Processor determines a Schedule Overrun has occurred.

35.3.5 HcInterruptEnable Register (USBHIE)

Writing 1 to a bit in this register sets the corresponding bit, while writing a 0 to a bit leaves the bit unchanged.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIE	OC	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	RHSC	FNO	UE	RD	SF	WDH	SO
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MIE	0	R/W	MasterInterruptEnable This bit is a global interrupt enable. Writing 1 allows interrupts to be enabled via the specific enable bits listed below.
30	OC	0	R/W	OwnershipChangeEnable 0: Ignored 1: Interrupt due to Ownership Change is enabled.
29 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
6	RHCS	0	R/W	RootHubStatusChangeEnable 0: Ignored 1: Interrupt due to Root Hub Status Change is enabled.
5	FNO	0	R/W	FrameNumberOverflowEnable 0: Ignored 1: Interrupt due to Frame Number Overflow is enabled.
4	UE	0	R/W	UnrecoverableErrorEnable This function is not supported. Writing is ignored.
3	RD	0	R/W	ResumeDetectedEnable 0: Ignored 1: Interrupt due to Resume Detected is enabled.

Bit	Bit Name	Initial Value	R/W	Description
2	SF	0	R/W	StartOfFrameEnable 0: Ignored 1: Interrupt due to Start of Frame is enabled.
1	WDH	0	R/W	WritebackDoneHeadEnable 0: Ignored 1: Interrupt due to Writeback Done Head is enabled.
0	SO	0	R/W	SchedulingOverrunEnable 0: Ignored 1: Interrupt due to Scheduling Overrun is enabled.

35.3.6 HcInterruptDisable Register (USBHID)

Writing 1 to a bit in this register clears the corresponding bit, while writing 0 to a bit leaves the bit unchanged.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIE	OC	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	RHSC	FNO	UE	RD	SF	WDH	SO
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MIE	0	R/W	MasterInterruptDisable This bit is a global interrupt disable. Writing 1 disables all interrupts.
30	OC	0	R/W	OwnershipChangeDisable 0: Ignored 1: Interrupt due to Ownership Change is disabled.
29 to 7	—	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
6	RHSC	0	R/W	RootHubStatusChangeDisable 0: Ignored 1: Interrupt due to Root Hub Status Change is disabled.
5	FNO	0	R/W	FrameNumberOverflowDisable 0: Ignored 1: Interrupt due to Frame Number Overflow is disabled.
4	UE	0	R/W	UnrecoverableErrorDisable This function is not supported. Writing is ignored.
3	RD	0	R/W	ResumeDetectedDisable 0: Ignored 1: Interrupt due to Resume Detected is disabled.
2	SF	0	R/W	StartOfFrameDisable 0: Ignored 1: Interrupt due to Start of Frame is disabled.
1	WDH	0	R/W	WritebackDoneHeadDisable 0: Ignored 1: Interrupt due to Writeback Done Head is disabled.
0	SO	0	R/W	SchedulingOverrunDisable 0: Ignored 1: Interrupt generation due to Scheduling Overrun is disabled.

35.3.7 HcHCCA Register (USBHHCCA)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCCA[23:8]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HCCA[7:0]								—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	HCCA	All 0	R/W	HCCA Pointer to HCCA base address.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.8 HcPeriodCurrentED Register (USBHPCED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PCED[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCED[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	PCED	All 0	R	PeriodCurrentED Pointer to the current Periodic List ED. (Within Graphics Memory [Unified Memory] space)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.9 HcControlHeadED Register (USBHCHED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHED[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CHED[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	CHED	All 0	R/W	ControlHeadED Pointer to the Control List Head ED.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.10 HcControlCurrentED Register (USBHCCED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCED[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCED[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	CCED	All 0	R/W	ControlCurrentED Pointer to the current Control List ED.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.11 HcBulkHeadED Register (USBHBHED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BHED[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BHED[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	BHED	All 0	R/W	BulkHeadED Pointer to the Bulk List Head ED.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.12 HcBulkCurrentED Register (USBHBCED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BCED[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCED[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	BCED	All 0	R/W	BulkCurrentED Pointer to the current Bulk List ED.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

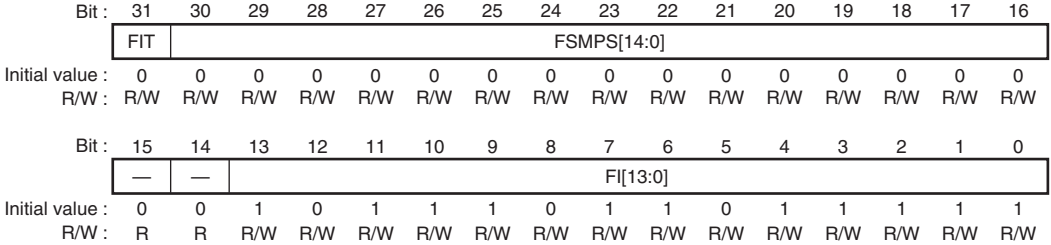
35.3.13 HcDoneHead Register (USBHDHED)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DH[27:12]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DH[11:0]												—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	DH	All 0	R	DoneHead Pointer to the current Done List Head ED.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0

35.3.14 HcFmInterval Register (USBHFI)



Bit	Bit Name	Initial Value	R/W	Description
31	FIT	0	R/W	FrameIntervalToggle This bit is toggled by Host Control Driver (HCD) whenever it loads a new value into FrameInterval bit.
30 to 16	FSMPS	All 0	R/W	FSLargestDataPacket These bits specify a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.
15, 14	—	00	R	Reserved These bits are always read as 0. The write value should always be 0
13 to 0	FI	H'2EDF	R/W	FrameInterval These bits specify the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is specified.

35.3.15 HcFrameRemaining Register (USBHFR)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	FR[13:0]													
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	FRT	0	R	FrameRemainingToggle This bit is loaded with FrameIntervalToggle when FrameRemaining is loaded.
30 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
13 to 0	FR	All0	R	FrameRemaining These bits are the 14-bit down counter used to time a frame. When the Host Controller is in the USB OPERATIONAL state, the counter decrements each 12 MHz clock period. When the count reaches 0, the end of a frame has been reached. The counter reloads with FrameInterval at that time. In addition, the counter reloads when the Host Controller transitions into USB OPERATIONAL.

35.3.16 HcFmNumber Register (USBHFN)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FN[15:0]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
15 to 0	FN	All 0	R/W	FrameNumber These bits are the 16-bit up counter. The count is incremented coincident with the loading of FrameRemaining bit. The count will roll over from H'FFFF to H'0000

35.3.17 HcPeriodicStart Register (USBHPS)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PS[13:0]													
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
13 to 0	PS	All 0	R/W	PeriodicStart These bits set a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

35.3.18 HcLSThreshold Register (USBHLST) (Not supporting LowSpeed mode)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	LST[11:0]											
Initial value :	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
R/W :	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0
11 to 0	LST	H'628	R/W	LSThreshold These bits are a value used by the Frame manager to determine whether or not a low speed transaction can be started in the current frame.

35.3.19 HcRhDescriptorA Register (USBHRDA) (Only one port is supported by this LSI.)

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. These bits should not be written during normal operation.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POTPGT[7:0]								—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	NOCP	OCPM	DT	NPS	PSM	NDP[7:0]							
Initial value :	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
R/W :	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	POTPGT	H'02	R/W	PowerOnToPowerGoodTime USB Host Controller power switching is effective within 2 ms. The bit value is represented as the number of 2 ms intervals. Only bits 25 and 24 can be written to. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is allowed. These bits should be written to support the system implementation. These bits should always be written to a non-zero value.
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	NOCP	1	R/W	NoOverCurrentProtection USB Host Controller implements global over-current reporting 0: Over-current status is reported 1: Over-current status is not reported This bit should be written to support the external system port over-current implementation.

Bit	Bit Name	Initial value	R/W	Description
11	OCPM	0	R/W	OverCurrentProtectionMode USB Host Controller implements global over-current reporting 0: Global Over-Current is reported. 1: Individual Over-Current is reported.This bit is only valid when NoOverCurrentProtection bit is cleared. Write 0 to this bit.
10	DT	0	R	DeviceType USB Host Controller is not a composite device.
9	NPS	0	R/W	NoPowerSwitching USB Host Controller implements global power switching. 0: Ports are power switched. 1: Ports are always powered on.This bit should be written to support the external system port power switching implementation.
8	PSM	0	R/W	PowerSwitchingMode USB Host Controller implements a global power switching mode. 0: Global Switching 1: Individual SwitchingThis bit is only valid when NoPowerSwitching is cleared. Write 0 to this bit.
7 to 0	NDP	H'02	R	NumberDownstreamPorts USB Host Controller supports one downstream port. (Only one port is supported although two ports are mentioned.)

35.3.20 HcRhDescriptorB Register (USBHRDB) (Only one port is supported by this LSI.)

This register is only reset by a power-on reset. It is written during system initialization to configure the Root Hub. These bits should not be written during normal operation.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PPCM[15:0]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
31 to 16	PPCM	All 0	R/W	PortPowerControlMask USB Host Controller implements global-power switching. These bits are only valid if NoPowerSwitching is cleared and PowerSwitchingMode bit is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/ClearPortPower). When cleared, the port only responds to global power switching commands (Set/ClearGlobalPower). 0: Device not removable 1: Global-power switching is masked Port Bit relationship 0: Reserved 1: Port 1 2: Port 2 : 15: Port 15 Unimplemented ports are reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
15 to 0	DR	All 0	R/W	DeviceRemovableUSB Host Controller ports default to removable devices. 0: Device removable 1: Device not removable Port Bit relationship 0: Reserved 1: Port 1 2: Port 2 : 15: Port 15 Unimplemented ports are reserved. These bits are always read as 0. The write value should always be 0.

35.3.21 HcRhStatus Register (USBHRS)

This register is reset by the UsbReset state.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCIC	LPSC
Initial value :	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCI	LPS
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial value	R/W	Description
31	CRWE	—	W	(write) ClearRemoteWakeupEnable Writing 1 to this bit clears DeviceRemoteWakeupEnable bit. Writing 0 has no effect
30 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17	OCIC	0	R/W	OverCurrentIndicatorChange This bit is set when OverCurrentIndicator changes. Writing 1 clears this bit. Writing 0 has no effect.
16	LPSC	0	R/W	(read) LocalPowerStatusChange Not supported by this LSI. The read value should always be 0. (write) SetGlobalPower Write 1 issues a SetGlobalPower command to the ports. Writing 0 has no effect.
15	DRWE	0	R/W	(read) DeviceRemoteWakeupEnable This bit enables ports' ConnectStatusChange as a remote wakeup event. 0: disabled 1: enabled (write) SetRemoteWakeupEnable Writing 1 sets DeviceRemoteWakeupEnable bit. Writing 0 has no effect.

Bit	Bit Name	Initial value	R/W	Description
14 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	OCI	0	R	OverCurrentIndicator This bit reflects the state of the OVRCUR pin. This bit is only valid if NoOverCurrentProtection and OverCurrentProtectionMode bits are cleared. 0: No over-current condition 1: Over-current condition
0	LPS	0	R/W	(read) LocalPowerStatus Not Supported by this LSI. The read value should be 0. (write) ClearGlobalPower Writing 1 issues a ClearGlobalPower command to the ports. Writing 0 has no effect.

35.3.22 HcRhPortStatus[2] Register (USBHRPS2)

This register is reset by the UsbReset state.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	PRSC	OCIC	PSSC	PESC	CSC
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	LSDA	PPS	—	—	—	PRS	POCI	PSS	PES	CCS
Initial value :	0	0	0	0	0	0	—	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

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	PRSC	0	R/W	PortResetStatusChange This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	OCIC	0	R/W	PortOverCurrentIndicatorChange This bit is set when OverCurrentIndicator changes. Writing 1 clears this bit. Writing 0 has no effect.
18	PSSC	0	R/W	PortSuspendStatusChange This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	PESC	0	R/W	PortEnableStatusChange This bit indicates that the port has been disabled due to a hardware event (PortEnableStatus bit is cleared). 0: Port has not been disabled. 1: PortEnableStatus bit has been cleared.

Bit	Bit Name	Initial value	R/W	Description
16	CSC	0	R/W	ConnectStatusChange This bit indicates a connection or disconnection event has been detected. Writing 1 clears this bit. Writing 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. Note: If DeviceRemoveable is set, this bit resets to 1.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	LSDA	Undefined*	R/W	(read) LowSpeedDeviceAttached This bit defines the speed (and bus idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full Speed device 1: Low Speed device (write) ClearPortPower Writing 1 clears PortPowerStatus bit. Writing 0 has no effect
8	PPS	0	R/W	(read) PortPowerStatus This bit reflects the power state of the port regardless of power switching mode. 0: Port power is off. 1: Port power is on. Note: If NoPowerSwitching bit is set, the read value should always be 0. (write) SetPortPower Writing 1 sets PortPowerStatus bit. Writing 0 has no effect.
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	PRS	0	R/W	(read) PortResetStatus 0: Port reset signal is not active. 1: Port reset signal is active. (write) SetPortReset Writing 1 sets PortResetStatus bit. Writing 0 has no effect.
3	POCI	0	R/W	(read) PortOverCurrentIndicator USB Host Controller supports global over-current reporting. This bit reflects the state of the OVRCUR pin dedicated to this port. This bit is only valid if NoOverCurrentProtection bit is cleared and OverCurrentProtectionMode is set. 0: No over-current condition 1: Over-current condition (write) ClearSuspendStatus Writing 1 initiates the selective resume sequence for the port. Writing 0 has no effect.
2	PSS	0	R/W	(read) PortSuspendStatus 0: Port is not suspended 1: Port is selectively suspended (write) SetPortSuspend Writing 1 sets PortSuspendStatus bit. Writing 0 has no effect.
1	PES	0	R/W	(read) PortEnableStatus 0: Port disabled. 1: Port enabled. (write) SetPortEnable Writing 1 sets PortEnableStatus bit. Writing 0 has no effect.

Bit	Bit Name	Initial value	R/W	Description
0	CCS	0	R/W	(read) CurrentConnectStatus 0: No device connected. 1: Device connected. Note: If DeviceRemoveable bit is set (not removable) this bit is always read as 1. (write) ClearPortEnable Writing 1 clears PortEnableStatus bit. Writing 0 has no effect.

Note: * Will have an effect on the status of the transceiver.

35.3.23 ConfigurationControl Register (USBHSC)

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PS
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W :	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PS	1	R/W	Port Switch 0: Host 1: Function

Note: When the function module and the host module are switched by Port Switch bit, if the USB_PWREN pin is used, use the procedure below.

Situation	Procedure
Switch to the function module (This module is selected in USB Function as default value after power-on sequence)	Check the status as follows:· • Set the PortSwitch bit to 1 after matching the polarity between the PULLUPE bit in the USB function module and the PortPowerStatus bit in the USB host module. • There is no need to match the polarity if the PULLUPE bit is already ON after the USB function module is started.

35.4 Functional Description

35.4.1 General Functionality

(1) USB Host Module

The USB Host module includes the integrated Root Hub with an external port as well as the List Processing (LP), the Serial Interface Engine (SIE) and USB clock divider. The interface combines the responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB. Application interface converts HCI interface to Peripheral bus interface and SuperHyway bridge bus interface. USB Host module supports OpenHCI registers. Data transfer is performed on SuperHyway bridge bus interface between External memory and USB host module. Registers in USB host module are controlled via Peripheral bus interface. Endpoint Descriptor(ED) and Transfer Descriptor(TD) need to be stored in External memory before the data transaction begins.

1. List Processor

   The List Processor consists of four main blocks. The four blocks are the List Control block, the ED block, the TD block, and the Request block. The first three blocks operate in a lock step fashion with the List Control block triggering the ED block, which in turn triggers the TD block. These blocks are responsible for issuing their own bus master requests to the Request block which interfaces to the Host Controller Bus Master.

2. Serial Interface Engine (SIE)

   The SIE is responsible for managing all transactions to the USB. It controls the bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding.

   All transactions on the USB are requested by the List Processor and Frame Manager. After the List Processor retrieves all information necessary to initiate communication to a USB device, it generates a request to the SIE accompanied by endpoint-specific control information required to generate proper protocol and packet formats to establish the desired communication pipe. The data buffer provides a data path for the data packets and controls the number of bytes transferred.

   The FM generates SOF events each millisecond for which the SIE generates an SOF token. The List Processor requests are ignored to allow the SOF to be serviced with the highest priority and without any delay.

3. Root Hub (Only one port is supported by this LSI.)

   The Root Hub is a collection of ports which are individually controlled and a hub which maintains control/status over functions common to all ports. The typical command request interface to the hub is emulated by the Host Controller Device(HCD) which communicates directly through the system bus (PCI) to the hub and port controls. The remainder of this section will divide the discussion into hub and port design responsibilities.

   The Root Hub descriptor registers, HcRhDescriptorA and HcRhDescriptorB, are implemented R/W to allow multiple configuration with minimal changes to the current implementation.

   Hub and port indicate the control and the status through the HcRhStatus and HcRhPortStatus Registers. Each port has its own HcRhPortStatus Registers. A command structure is defined through these registers which software uses to control the hub and ports. By writing 1 to bit locations specified in section 35.3, Register Description, the following commands can be executed. The command functions are discussed in the following sections.

- Hub Control

  The HC states also reflect the hub state. For example, when the HC is suspended, USB SUSPEND, the Root Hub is suspended. When the HC is in USB RESUME, the hub generates the appropriate bus signaling. USB RESET resets the Root Hub. The following sections describe hub and bus related controls and status.

- Port Control

  The Port is responsible for all activities associated with driving and monitoring bus states. The HCD controls this behavior through the register command interface.

- Clock Generation

  The USB interface is sourced by a 48 MHz clock which allows for a 4x data rate oversampling to maintain the receiver phase lock. This clock also sources all USB related clock rates (12 MHz).

- Static SOF Clock

  As the USB system host, the system frame counter is maintained at a constant 1 ms interval. This requires a static 12 MHz clock. This is created by dividing down the 48 MHz internal clock source. The clock is enabled when the HC is not in the USB SUSPEND state.

- Data Rate Clock

  The SIE requires that the transmit and receive clocks operate at 12 Hz. During FS transmissions, the data rate clock is equivalent to the static 12 MHz SOF clock.

  When receiving data, the data rate clock must match that of the source. Working in conjunction with the phase lock circuitry, the data rate clock is adjusted to maintain a 1 to 1 ratio of data bits and data clocks. This will result in periodic adjustment of the internal 48 MHz internal clock periods to maintain synchronization with the data source. When the packet is complete the data rate clock is re-linked to the static 12 MHz clock discussed above.

35.5 Connection Example of an External Circuit

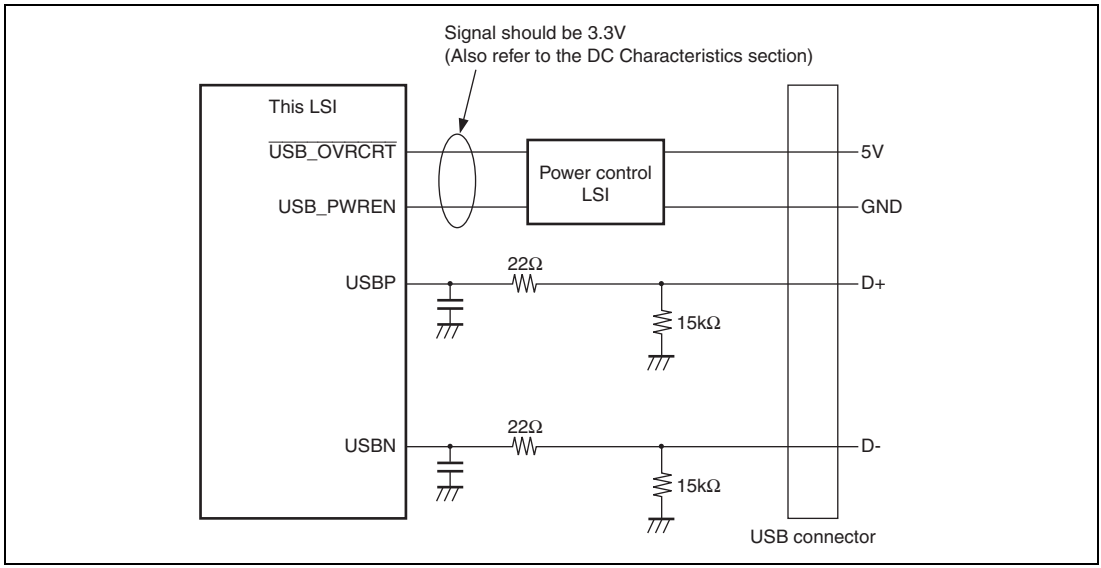


Figure 35.2 Connection Example of External Circuit

35.6 Usage Notes

35.6.1 External memory that USBH accesses

In access from the USBH to the external memory controlled by LBSC, some access size combinations are not supported by the LBSC. Therefore, when a local bus area such as SRAM area is specifies as the external memory to be accessed by the USBH, the memory access by the USBH may stop. To avoid this, specify a DDR-SDRAM area as the external memory accessed by the USBH.

35.6.2 Issuing USB Bus Reset

When connection of an USB device is detected, do not use the HCFS[1:0] bits in the HcControl register to issue a USB Reset. Be sure to use the PRS bit (Port Reset) in the HcRhPortStatus register to issue a USB Reset.

Section 36 USB Function Controller (USBF)

This LSI incorporates an USB function controller (USBF).

36.1 Features

- UDC (USB device controller) conforming to USB1.1 processes incorporated USB protocol automatically.

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed
- Endpoint configuration: An arbitrary endpoint configuration can be set

  The arbitrary endpoint can be configured by setting the correspondence between the endpoint (the endpoint number used by the USB host) and the EP FIFO number that is provided by this USB function controller (the transfer method and direction are fixed).

EP FIFO Number	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt	8	8	—
Endpoint 4	EP4	Isochronous-out	64	128	—
Endpoint 5	EP5	Isochronous-in	64	128	—

- Interrupt requests: generates various interrupt signals necessary for USB transmission/reception
- Clock: External input (48 MHz)
- Power-down mode

  Power consumption can be reduced by stopping UDC internal clock when USB cable is disconnected

  Automatic transition to/recovery from suspend state
- Supports self-powered mode

Figure 36.1 shows the block diagram of USBF.

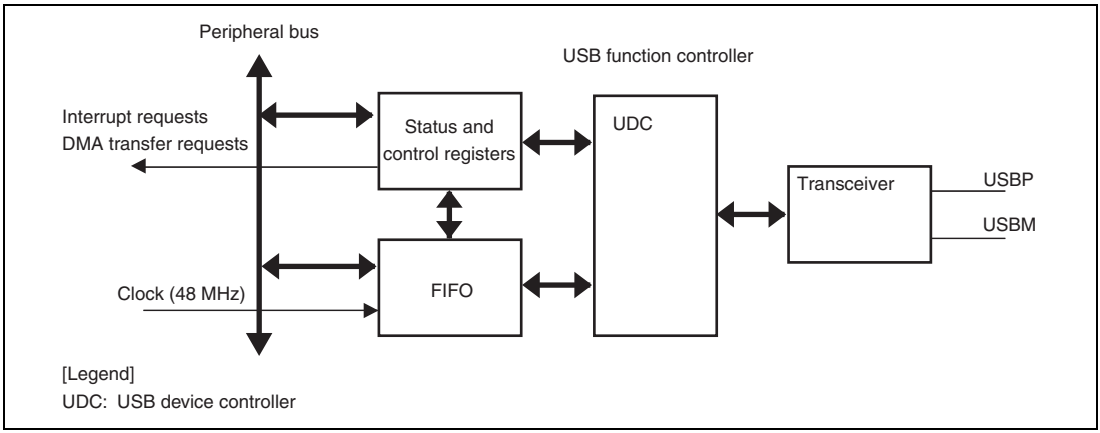


Figure 36.1 Block Diagram of USBF

36.2 Input/Output Pins

Table 36.1 lists the pin configuration of USBF.

Table 36.1 Pin Configuration and Functions

Name	Pin Name	I/O	Function
Overcurrent pin / VBUS pin	$\overline{\text{USB_OVRCRT}}$/VSBF_VBUS	Input	USB port over-current detection/ USB cable connection monitor pin
Clock pin	USB_CLK	Input	USB clock input pin (48 MHz input)*
Power enable pin	USB_PWREN/USBF_UPLUP	Output	USB port power enable control/ Pull-up control output pin
P pin	USBP	I/O	D+
M pin	USBM	I/O	D–

Note: * USB_CLK should be slower than Pck0.

36.3 Register Descriptions

Tables 36.2 (1) and (2) show the USBF register configuration when accessed in 8-bit units and 32-bit units, respectively Table 36.3 shows the register state in each operating mode.

Table 36.2 (1) Register Configuration (Access Size = 8 bits)

Register Name	Abbreviation	Area P4 Address*	Area 7 Address*	Access size
Interrupt flag register 0	IFR0	H'FFEC 0001	H'1FEC 0001	8
Interrupt flag register 1	IFR1	H'FFEC 0005	H'1FEC 0005	8
Interrupt flag register 2	IFR2	H'FFEC 0009	H'1FEC 0009	8
Interrupt flag register 3	IFR3	H'FFEC 000D	H'1FEC 000D	8
Interrupt enable register 0	IER0	H'FFEC 0011	H'1FEC 0011	8
Interrupt enable register 1	IER1	H'FFEC 0015	H'1FEC 0015	8
Interrupt enable register 2	IER2	H'FFEC 0019	H'1FEC 0019	8
Interrupt enable register 3	IER3	H'FFEC 001D	H'1FEC 009D	8
Interrupt select register 0	ISR0	H'FFEC 0021	H'1FEC 0021	8
Interrupt select register 1	ISR1	H'FFEC 0025	H'1FEC 0025	8
Interrupt select register 2	ISR2	H'FFEC 0029	H'1FEC 0029	8
Interrupt select register 3	ISR3	H'FFEC 002D	H'1FEC 002D	8
EP0i data register	EPDR0i	H'FFEC 0031	H'1FEC 0031	8
EP0o data register	EPDR0o	H'FFEC 0035	H'1FEC 0035	8
EP0s data register	EPDR0s	H'FFEC 0039	H'1FEC 0039	8
EP1 data register	EPDR1	H'FFEC 0041	H'1FEC 0041	8
EP2 data register	EPDR2	H'FFEC 0051	H'1FEC 0051	8
EP3 data register	EPDR3	H'FFEC 0061	H'1FEC 0061	8
EP4 data register	EPDR4	H'FFEC 0071	H'1FEC 0071	8
EP5 data register	EPDR5	H'FFEC 0081	H'1FEC 0081	8
EP0o receive data size register	EPSZ0o	H'FFEC 0091	H'1FEC 0091	8
EP1 receive data size register	EPSZ1	H'FFEC 0095	H'1FEC 0095	8
EP4 receive data size register	EPSZ4	H'FFEC 0099	H'1FEC 0099	8
Data status register	DASTS	H'FFEC 009D	H'1FEC 009D	8

Register Name	Abbreviation	Area P4 Address*	Area 7 Address*	Access size
FIFO clear register 0	FCLR0	H'FFEC 00A1	H'1FEC 00A1	8
FIFO clear register 1	FCLR1	H'FFEC 00A5	H'1FEC 00A5	8
Endpoint stall register 0	EPSTL0	H'FFEC 00A9	H'1FEC 00A9	8
Endpoint stall register 1	EPSTL1	H'FFEC 00AD	H'1FEC 00AD	8
Trigger register	TRG	H'FFEC 00B1	H'1FEC 00B1	8
DMA transfer setting register	DMA	H'FFEC 00B5	H'1FEC 00B5	8
Configuration value register	CVR	H'FFEC 00B9	H'1FEC 00B9	8
Control register 0	CTLR0	H'FFEC 00BD	H'1FEC 00BD	8
Time stamp register H	TSRH	H'FFEC 00C1	H'1FEC 00C1	8
Time stamp register L	TSRL	H'FFEC 00C5	H'1FEC 00C5	8
Endpoint information register	EPIR	H'FFEC 00C9	H'1FEC 00C9	8
Interrupt flag register 4	IFR4	H'FFEC 00D1	H'1FEC 00D1	8
Interrupt enable register 4	IER4	H'FFEC 00D5	H'1FEC 00D5	8
Interrupt select register 4	ISR4	H'FFEC 00D9	H'1FEC 00D9	8
Control register 1	CTLR1	H'FFEC 00DD	H'1FEC 00DD	8
Timer register H	TMRH	H'FFEC 00E1	H'1FEC 00E1	8
Timer register L	TMRL	H'FFEC 00E5	H'1FEC 00E5	8
Set time out register H	STOH	H'FFEC 00E9	H'1FEC 00E9	8
Set time out register L	STOL	H'FFEC 00ED	H'1FEC 00ED	8

Table 36.2 (2) Register Configuration (Access Size = 32 bits)

Register Name	Abbreviation	Area P4 Address*	Area 7 Address*	Access size
Interrupt flag register 0	IFR0	H'FFEC 0000	H'1FEC 0000	32
Interrupt flag register 1	IFR1	H'FFEC 0004	H'1FEC 0004	32
Interrupt flag register 2	IFR2	H'FFEC 0008	H'1FEC 0008	32
Interrupt flag register 3	IFR3	H'FFEC 000C	H'1FEC 000C	32
Interrupt enable register 0	IER0	H'FFEC 0010	H'1FEC 0010	32
Interrupt enable register 1	IER1	H'FFEC 0014	H'1FEC 0014	32
Interrupt enable register 2	IER2	H'FFEC 0018	H'1FEC 0018	32
Interrupt enable register 3	IER3	H'FFEC 001C	H'1FEC 001C	32
Interrupt select register 0	ISR0	H'FFEC 0020	H'1FEC 0020	32
Interrupt select register 1	ISR1	H'FFEC 0024	H'1FEC 0024	32
Interrupt select register 2	ISR2	H'FFEC 0028	H'1FEC 0028	32
Interrupt select register 3	ISR3	H'FFEC 002C	H'1FEC 002C	32
EP0i data register	EPDR0i	H'FFEC 0030	H'1FEC 0030	32
EP0o data register	EPDR0o	H'FFEC 0034	H'1FEC 0034	32
EP0s data register	EPDR0s	H'FFEC 0038	H'1FEC 0038	32
EP1 data register	EPDR1	H'FFEC 0040	H'1FEC 0040	32
EP2 data register	EPDR2	H'FFEC 0050	H'1FEC 0050	32
EP3 data register	EPDR3	H'FFEC 0060	H'1FEC 0060	32
EP4 data register	EPDR4	H'FFEC 0070	H'1FEC 0070	32
EP5 data register	EPDR5	H'FFEC 0080	H'1FEC 0080	32
EP0o receive data size register	EPSZ0o	H'FFEC 0090	H'1FEC 0090	32
EP1 receive data size register	EPSZ1	H'FFEC 0094	H'1FEC 0094	32
EP4 receive data size register	EPSZ4	H'FFEC 0098	H'1FEC 0098	32
Data status register	DASTS	H'FFEC 009C	H'1FEC 009C	32
FIFO clear register 0	FCLR0	H'FFEC 00A0	H'1FEC 00A0	32
FIFO clear register 1	FCLR1	H'FFEC 00A4	H'1FEC 00A4	32
Endpoint stall register 0	EPSTL0	H'FFEC 00A8	H'1FEC 00A8	32
Endpoint stall register 1	EPSTL1	H'FFEC 00AC	H'1FEC 00AC	32
Trigger register	TRG	H'FFEC 00B0	H'1FEC 00B0	32

Register Name	Abbreviation	Area P4 Address*	Area 7 Address*	Access size
DMA transfer setting register	DMA	H'FFEC 00B4	H'1FEC 00B4	32
Configuration value register	CVR	H'FFEC 00B8	H'1FEC 00B8	32
Control register 0	CTLR0	H'FFEC 00BC	H'1FEC 00BC	32
Time stamp register H	TSRH	H'FFEC 00C0	H'1FEC 00C0	32
Time stamp register L	TSRL	H'FFEC 00C4	H'1FEC 00C4	32
Endpoint information register	EPIR	H'FFEC 00C8	H'1FEC 00C8	32
Interrupt flag register 4	IFR4	H'FFEC 00D0	H'1FEC 00D0	32
Interrupt enable register 4	IER4	H'FFEC 00D4	H'1FEC 00D4	32
Interrupt select register 4	ISR4	H'FFEC 00D8	H'1FEC 00D8	32
Control register 1	CTLR1	H'FFEC 00DC	H'1FEC 00DC	32
Timer register H	TMRH	H'FFEC 00E0	H'1FEC 00E0	32
Timer register L	TMRL	H'FFEC 00E4	H'1FEC 00E4	32
Set time out register H	STOH	H'FFEC 00E8	H'1FEC 00E8	32
Set time out register L	STOL	H'FFEC 00EC	H'1FEC 00EC	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 36.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Interrupt flag register 0	IFR0	H'xxxx xx10	H'xxxx xx10	Retained	Retained
Interrupt flag register 1	IFR1	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt flag register 2	IFR2	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt flag register 3	IFR3	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt enable register 0	IER0	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt enable register 1	IER1	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt enable register 2	IER2	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt enable register 3	IER3	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt select register 0	ISR0	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt select register 1	ISR1	H'xxxx xx03	H'xxxx xx03	Retained	Retained
Interrupt select register 2	ISR2	H'xxxx xx1F	H'xxxx xx1F	Retained	Retained
Interrupt select register 3	ISR3	H'xxxx xx00	H'xxxx xx00	Retained	Retained
EP0i data register	EPDR0i	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP0o data register	EPDR0o	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP0s data register	EPDR0s	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP1 data register	EPDR1	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP2 data register	EPDR2	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP3 data register	EPDR3	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP4 data register	EPDR4	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP5 data register	EPDR5	H'xxxx xxxx	H'xxxx xxxx	Retained	Retained
EP0o receive data size register	EPSZ0o	H'xxxx xx00	H'xxxx xx00	Retained	Retained
EP1 receive data size register	EPSZ1	H'xxxx xx00	H'xxxx xx00	Retained	Retained
EP4 receive data size register	EPSZ4	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Data status register	DASTS	H'xxxx xx00	H'xxxx xx00	Retained	Retained
FIFO clear register 0	FCLR0	H'xxxx xx00	H'xxxx xx00	Retained	Retained
FIFO clear register 1	FCLR1	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Endpoint stall register 0	EPSTL0	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Endpoint stall register 1	EPSTL1	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Trigger register	TRG	H'xxxx xx00	H'xxxx xx00	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
DMA transfer setting register	DMA	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Configuration value register	CVR	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Control register 0	CTLR0	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Time stamp register H	TSRH	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Time stamp register L	TSRL	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Endpoint information register	EPIR	H'xxxx xx00	H'xxxx xxxx	Retained	Retained
Interrupt flag register 4	IFR4	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt enable register 4	IER4	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Interrupt select register 4	ISR4	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Control register 1	CTLR1	H'xxxx xx00	H'xxxx xx02	Retained	Retained
Timer register H	TMRH	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Timer register L	TMRL	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Set time out register H	STOH	H'xxxx xx00	H'xxxx xx00	Retained	Retained
Set time out register L	STOL	H'xxxx xx00	H'xxxx xx00	Retained	Retained

36.3.1 Interrupt Flag Register 0 (IFR0)

IFR0 is an interrupt flag register for EP0i, EP1, EP2, bus reset, and setup command reception. When each flag is set to 1 and interrupt is enabled in the corresponding bit of IER0, an interrupt request (USBF10 or USB11) specified by the corresponding bit in ISR0 is issued to INTC.

Clearing the flag is performed by writing 0. Writing 1 is not valid and nothing is changed. To clear bits, access the register so that 0 should be only to the bits for the interrupt sources to be cleared and that 1 should be written to the other bits. Do not use a bit field declaration of the C language to clear bits.

EP2 EMPTY and EP1 FULL are status bits that indicate the FIFO states of EP1 and EP2, respectively. Therefore, EP2 EMPTY and EP1 FULL cannot be cleared.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0O TS	EP0I TR	EP0I TS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7	BRST	0	R/W	Bus Reset [Setting condition] When a bus reset signal is detected on the USB bus. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
6	EP1 FULL	0	R	EP1 (Bulk-out) FIFO Full [Setting condition] The FIFO buffer of EP1 has a dual-buffer configuration, and this bit is set when at least one of the FIFO buffer is full. [Setting conditions] • When reset • When both FIFO buffers are empty. Note: EP1 FULL is a status bit, and cannot be cleared.
5	EP2 TR	0	R/W	EP2 (Bulk-in) Transfer Request [Setting condition] When an IN token is received from the host to EP2 and both of FIFO buffers are empty. [Clearing conditions] • When reset • When 0 is written to by CPU
4	EP2 EMPTY	1	R	EP2 (Bulk-in) FIFO Empty [Setting conditions] • When reset • The FIFO buffer of EP2 has a dual-buffer configuration, and this bit is set when at least one of the FIFO buffer is empty. [Clearing condition] When both of FIFO buffers are not empty. Note: EP2 EMPTY is a status bit, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
3	SETUP TS	0	R/W	Setup Command Receive Complete [Setting condition] When 8-byte data that decodes the command by the function is normally received from the host to EP0s and an ACK handshake is returned to the host from the function. [Clearing conditions] • When reset • When 0 is written to by CPU
2	EP0o TS	0	R/W	EP0o Receive Complete [Setting condition] When data is normally received from the host to EP0o and an ACK handshake is returned from the function to the host. [Clearing conditions] • When reset • When 0 is written to by CPU
1	EP0i TR	0	R/W	EP0i Transfer Request [Setting condition] When IN token is issued from the host to EP0i and the FIFO buffer is empty. [Clearing conditions] • When reset • When 0 is written to by CPU
0	EP0i TS	0	R/W	EP0i Transmit Complete [Setting condition] When data to be transmitted to the host is written to EP0i, then data is normally transferred from the function to the host, and an ACK handshake is returned. [Clearing conditions] • When reset • When 0 is written to by CPU

36.3.2 Interrupt Flag Register 1 (IFR1)

IFR1 is an interrupt flag register for VBUS and EP3. When each flag is set to 1 and an interrupt is enabled in the corresponding bit of IER1, an interrupt request (USBFI0 or USBFI1) specified by the corresponding bit in ISR1 is issued to INTC.

Clearing the flag is performed by writing 0. Writing 1 is not valid and nothing is changed. To clear bits, access the register so that 0 should be written only to the bits for the interrupt sources to be cleared and that 1 should be written to the other bits. Do not use a bit field declaration of the C language to clear bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	VBUS MN	EP3 TR	EP3 TS	VBUSF
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	VBUS MN	0	R	USB Connection Status Status bit to monitor the USBF_VBUS pin state. Reflects the state of the USBF_VBUS pin. 0: Disconnected 1: Connected

Bit	Bit Name	Initial Value	R/W	Description
2	EP3 TR	0	R/W	EP3 (Interrupt) Transfer Request [Setting condition] When an IN token is issued from the host to EP3 and the FIFO buffer is empty. [Clearing conditions] • When reset • When 0 is written to by CPU
1	EP3 TS	0	R/W	EP3 (Interrupt) Transmit Complete [Setting condition] When data to be transmitted to the host is written to EP3, then data is normally transferred from the host to the function, and an ACK handshake is returned. [Clearing conditions] • When reset • When 0 is written to by CPU
0	VBUSF	0	R/W	USB Disconnection Detection The USBF_VBUS pin of this module is used for detecting connection/disconnection. [Setting condition] When the function is connected to the USB bus or disconnected from it. [Clearing conditions] • When reset • When 0 is written to by CPU.

36.3.3 Interrupt Flag Register 2 (IFR2)

IFR2 is an interrupt flag register for SURSS, SURSF, CFDN, SOF SETC, and SETI. When each flag is set to 1 and an interrupt is enabled in the corresponding bit of IER2, an interrupt request (USBFI0 or USBFI1) specified by the corresponding bit in ISR2 is issued to INTC.

Clearing the flag is performed by writing 0. Writing 1 is not valid and nothing is changed. To clear bits, access the register so that 0 should be written only to the bits for the interrupt sources to be cleared and that 1 should be written to the other bits. Do not use a bit field declaration of the C language to clear bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SURSS	SURSF	CFDN	SOF	SETC	SETI
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SURSS	0	R	Suspend/Resume Status Status bit indicating the state of the bus 0: Normal state 1: Suspend state

Bit	Bit Name	Initial Value	R/W	Description
4	SURSF	0	R/W	Suspend/Resume Detection [Setting condition] When the bus transits from the normal state to the suspend state or from the suspend state to the normal state. [Clearing conditions] • When reset • When 0 is written to by CPU
3	CFDN	0	R/W	End Point Information Load Complete [Setting condition] When the end point information written in EPIR is completed to be set (loaded) in this controller. Note: This controller operates normally as USB after the setting of the end point information is completed. [Clearing conditions] • When reset • When 0 is written to by CPU
2	SOF	0	R/W	SOF Packet [Setting condition] When the valid SOF packet is detected. [Clearing conditions] • When reset • When 0 is written to by CPU
1	SETC	0	R/W	Set Configuration Command Detection [Setting condition] When the valid Set Configuration command is detected. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
0	SETI	0	R/W	Set Interface Command Detection [Setting condition] When the valid Set Interface command is detected. [Clearing conditions] • When reset • When 0 is written to by CPU

36.3.4 Interrupt Flag Register 3 (IFR3)

IFR3 is an interrupt flag register for EP4 TS, EP4 TF, EP5 TS, and EP5 TR. When each flag is set to 1 and an interrupt is enabled in the corresponding bit of IER3, an interrupt request (USBFI0 or USBFI1) specified by the corresponding bit in ISR3 is issued to INTC.

Clearing the flag is performed by writing 0. Writing 1 is not valid and nothing is changed. To clear bits, access the register so that 0 should be written only to the bits for the interrupt sources to be cleared and that 1 should be written to the other bits. Do not use a bit field declaration of the C language to clear bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	EP5TR	EP5TS	EP4TF	EP4TS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 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	EP5 TR	0	R/W	EP5 (Isochronous-in) Transmit Request Flag indicating the FIFO state of EP5. After the SOF packet is received, the FIFO buffer is switched automatically. The FIFO buffer which has transmitted data to the host in the previous frame (before SOF reception) can be written to by the CPU. This bit indicates the transmit state in the previous frame. [Setting condition] The FIFO buffer to be transmitted is empty when an IN token is issued from the host to EP5. [Clearing conditions] • When reset • When 0 is written to by CPU
2	EP5 TS	0	R/W	EP5 (Isochronous-in) Normal Transmission Flag indicating the FIFO state of EP5. After the SOF packet is received, the FIFO buffer is switched automatically. The FIFO buffer which has transmitted data to the host in the previous frame (before SOF reception) can be written to by the CPU. This bit indicates the transmit state in the previous frame. [Setting condition] When a transmission was carried out normally in the previous frame. [Clearing conditions] • When reset • When 0 is written to by CPU

Bit	Bit Name	Initial Value	R/W	Description
1	EP4 TF	0	R/W	EP4 (Isochronous-out) Abnormal Reception Flag indicating the FIFO state of EP4. Indicates the state of the FIFO buffer that was readable after the data reception is completed and the next SOF packet is received. [Setting condition] When the transfer data from the host is abnormally received (packet error) by EP4. [Clearing conditions] • When reset • When 0 is written to by CPU
0	EP4 TS	0	R/W	EP4 (Isochronous-out) Normal Reception Flag indicating the FIFO state of EP4. Indicates the state of the FIFO buffer that was readable after the data reception is completed and the next SOF packet is received. [Setting condition] When the transfer data from the host is normally received by EP4. [Clearing conditions] • When reset • When 0 is written to by CPU

36.3.5 Interrupt Flag Register 4 (IFR4)

IFR4 is an interrupt flag register for for TMOUT. When each flag is set to 1 and an interrupt is enabled in the corresponding bit of IER4, an interrupt request (USBFI0 or USBFI1) specified by the corresponding bit in ISR4 is issued to INTC.

Clearing the flag is performed by writing 0. Writing 1 is not valid and nothing is changed. To clear bits, access the register so that 0 should be written only to the bits for the interrupt sources to be cleared and that 1 should be written to the other bits. Do not use a bit field declaration of the C language to clear bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMOUT
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT	0	R/W	Time Out [Setting condition] When the value of the timer register (TMR) is reached to that of the set time out register (STO). [Clearing conditions] • When reset • When 0 is written to by CPU

36.3.6 Interrupt Select Register 0 (ISR0)

ISR0 selects the interrupt request source (USBFI1 or USBFI0) to issue the interrupt requests indicated in IFR0 to INTC. When the corresponding bit in ISR0 is cleared to 0, the USBFI0 interrupt request is issued to INTC. When the bit is set to 1, the USBFI1 interrupt request is issued to INTC. In the initial value, each of interrupt source of the IFR0 is USBFI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BRST IS	EP1 FULL IS	EP2 TR IS	EP2 EMPTY IS	SETUP TS IS	EP0o TR IS	EP0i TR IS	EP0i TS IS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7	BRST IS	0	R/W	BRST Interrupt Select
6	EP1 FULL IS	0	R/W	EP1 FULL Interrupt Select
5	EP2 TR IS	0	R/W	EP2 TR Interrupt Select
4	EP2 EMPTY IS	0	R/W	EP2 EMPTY Interrupt Select
3	SETUP TS IS	0	R/W	SETUP Interrupt Select
2	EP0o TS IS	0	R/W	EP0o TS Interrupt Select
1	EP0i TR IS	0	R/W	EP0i TR Interrupt Select
0	EP0i TS IS	0	R/W	EP0i TS Interrupt Select

36.3.7 Interrupt Select Register 1 (ISR1)

ISR1 selects the interrupt request source (USBFI1 or USBFI0) to issue the interrupt requests indicated in IFR1 to INTC. When the corresponding bit in ISR1 is cleared to 0, the USBFI0 interrupt request is issued to INTC. When the bit is set to 1, the USBFI1 interrupt request is issued to INTC. In the initial value, each of interrupt source of the IFR1 is USBFI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	EP3 TR IS	EP3 TS IS	VBUSF IS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	EP3 TR IS	1	R/W	EP3 TR Interrupt Select
1	EP3 TS IS	1	R/W	EP3 TS Interrupt Select
0	VBUSF IS	1	R/W	VBUSF Interrupt Select

36.3.8 Interrupt Select Register 2 (ISR2)

ISR2 selects the interrupt request source (USBFI1 or USBFI0) to issue the interrupt requests indicated in IFR2 to INTC. When the corresponding bit in ISR2 is cleared to 0, the USBFI0 interrupt request is issued to INTC. When the bit is set to 1, the USBFI1 interrupt request is issued to INTC. In the initial value, each of interrupt source of the IFR2 is USBFI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SURSE IS	CFDN IS	SOFE IS	SETC IS	SETIE IS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SURSE IS	1	R/W	SURSE Interrupt Select
3	CFDN IS	1	R/W	CFDN Interrupt Select
2	SOFE IS	1	R/W	SOFE Interrupt Select
1	SETCE IS	1	R/W	SETCE Interrupt Select
0	SETIE IS	1	R/W	SETIE Interrupt Select

36.3.9 Interrupt Select Register 3 (ISR3)

ISR3 selects the interrupt request source (USBFI1 or USBFI0) to issue the interrupt requests indicated in IFR3 to INTC. When the corresponding bit in ISR3 is cleared to 0, the USBFI0 interrupt request is issued to INTC. When the bit is set to 1, the USBFI1 interrupt request is issued to INTC. In the initial value, each of interrupt source of the IFR3 is USBFI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	EP5 TR IS	EP5 TS IS	EP4 TF IS	EP4 TS IS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	EP5 TR IS	0	R/W	EP5 TR Interrupt Select
2	EP5 TS IS	0	R/W	EP5 TS Interrupt Select
1	EP4 TF IS	0	R/W	EP4 TF Interrupt Select
0	EP4 TS IS	0	R/W	EP4 TS Interrupt Select

36.3.10 Interrupt Select Register 4 (ISR4)

ISR4 selects the interrupt request source (USBFI1 or USBFI0) to issue the interrupt requests indicated in IFR4 to INTC. When the corresponding bit in ISR4 is cleared to 0, the USBFI0 interrupt request is issued to INTC. When the bit is set to 1, the USBFI1 interrupt request is issued to INTC. In the initial value, each of interrupt source of the IFR4 is USBFI1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMOUT IS
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT IS	0	R/W	TMOUT Interrupt Select

36.3.11 Interrupt Enable Register 0 (IER0)

IER0 enables the interrupt requests of the interrupt flag register 0. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the INTN interrupt request (USBFI0 or USBFI1) set in the ISR0 is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BRST IE	EP1 FULL IE	EP2 TR IE	EP2 EMPTY IE	SETUP TS IE	EP0o TS IE	EP0i TR IE	EP0i TS IE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7	BRST IE	0	R/W	BRST Interrupt Enable
6	EP1 FULL IE	0	R/W	EP1 FULL Interrupt Enable
5	EP2 TR IE	0	R/W	EP2 TR Interrupt Enable
4	EP2 EMPTY IE	0	R/W	EP2 EMPTY Interrupt Enable
3	SETUP TS IE	0	R/W	SETUP TS Interrupt Enable
2	EP0o TS IE	0	R/W	EP0o TS Interrupt Enable
1	EP0i TR IE	0	R/W	EP0i TR Interrupt Enable
0	EP0i TS IE	0	R/W	EP0i TS Interrupt Enable

36.3.12 Interrupt Enable Register 1 (IER1)

IER1 enables the interrupt requests of the IFR1. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request (USBFI0 or USBFI1) set in the ISR1 is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	EP3 TR IE	EP3 TS IE	VBUSF IE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	EP3 TR IE	0	R/W	EP3 TR Interrupt Enable
1	EP3 TS IE	0	R/W	EP3 TS Interrupt Enable
0	VBUSF IE	0	R/W	VBUSF Interrupt Enable

36.3.13 Interrupt Enable Register 2 (IER2)

IER2 enables the interrupt requests of the IFR2. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request (USBFI0 or USBFI1) set in the ISR2 is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SURSE IE	CFDN IE	SOFE IE	SETCE IE	SETIE IE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SURSE IE	0	R/W	SURSE Interrupt Enable
3	CFDN IE	0	R/W	CFDN Interrupt Enable
2	SOFE IE	0	R/W	SOFE Interrupt Enable
1	SETCE IE	0	R/W	SETCE Interrupt Enable
0	SETIE IE	0	R/W	SETIE Interrupt Enable

36.3.14 Interrupt Enable Register 3 (IER3)

IER3 enables the interrupt requests of the IFR3. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request (USBFI0 or USBFI1) set in the ISR3 is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	EP5 TR IE	EP5 TS IE	EP4 TE IE	EP4 TS IE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	EP5 TR IE	0	R/W	EP5 TR Interrupt Enable
2	EP5 TS IE	0	R/W	EP5 TS Interrupt Enable
1	EP4 TF IE	0	R/W	EP4 TF Interrupt Enable
0	EP4 TS IE	0	R/W	EP4 TS Interrupt Enable

36.3.15 Interrupt Enable Register 4 (IER4)

IER4 enables the interrupt requests of the IFR4. When an interrupt flag is set to 1 while the corresponding bit of each interrupt is set to 1, the interrupt request (USBFI0 or USBFI1) set in the ISR4 is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMOUT IE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TMOUT IE	0	R/W	TMOUT Interrupt Enable

36.3.16 EP0i Data Register (EPDR0i)

EPDR0i is an 8-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of transmit data for control-in. Transmit data is fixed by writing one packet of data and setting EP0iPKTE in the trigger register. When an ACK handshake is returned from the host after the data has been transmitted, EP0iTS in interrupt flag register 0 is set. This FIFO buffer can be initialized by means of EP0iCLR in the FCLR0 register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Data register for control-in transfer

36.3.17 EP0o Data Register (EPDR0o)

EPDR0o is an 8-byte receive FIFO buffer for endpoint 0. EPDR0o holds endpoint 0 receive data other than setup commands. When data is received normally, EP0oTS in interrupt flag register 0 is set, and the number of receive bytes is indicated in the EP0o receive data size register. After the data has been read, setting EP0oRDFN in the trigger register enables the next packet to be received. This FIFO buffer can be initialized by means of BP0oCLR in the FCLR0 register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

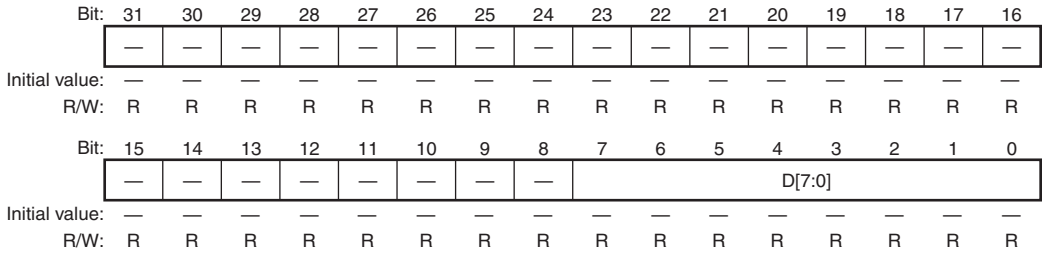
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	D[7:0]	Undefined	R	Data register for control-out transfer

36.3.18 EP0s Data Register (EPDR0s)

EPDR0s is a data register specifically for endpoint 0 setup command. EPDR0s holds 8-byte command data sent in the setup stage. However, only the command to be processed by a microprocessor (firmware) is received. The command data to be processed automatically by this module is not stored.

Since the setup command mast be received, previous data in the buffer is over written with new data. In other words, when the reception of data in the setup stage starts during read, reception has priority and read data is invalid.

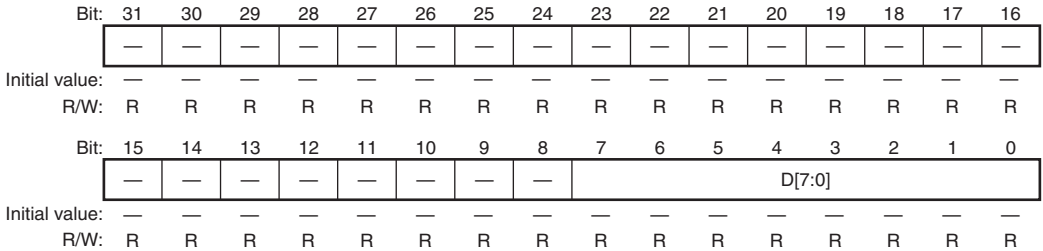


Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	D[7:0]	Undefined	R	Data register for storing the setup command at the control-out transfer

Note: The EPDR0s register should be read in longword units or eight consecutive times in byte units. If reading is stopped before it completes, data received in the subsequent setup stage is not read successfully.

36.3.19 EP1 Data Register (EPDR1)

EPDR1 is a 128-byte receive FIFO buffer for endpoint 1. EPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. The number of receive byte is displayed in the EP1 receive data size register. The buffer on read side can be received again by writing EP1RDFN in the trigger register to 1 after data is read. The receive data of this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP1CLR in the FCLR0 register.



Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	D[7:0]	Undefined	R	Data register for endpoint1 transfer

36.3.20 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of EP2CLR in the FCLR0 register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Data register for endpoint 2 transfer

36.3.21 EP3 Data Register (EPDR3)

EPDR3 is an 8-byte transmit FIFO buffer for endpoint 3. EPDR4 holds one packet of transmit data for the interrupt transfer of endpoint 3. Transmit data is fixed by writing one packet of data and setting EP3PKTE in the trigger register. When an ACK handshake is returned from the host after the data has been transmitted, EP3TS in interrupt flag register 1 is set. This FIFO buffer can be initialized by means of EP3CLR in the FCLR0 register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Data register for endpoint 3 transfer

36.3.22 EP4 Data Register (EPDR4)

EPDR4 is a 128-byte receive FIFO buffer for endpoint 4. EPDR4 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. The number of receive byte is displayed in the EP4 receive data size register. The receive data is fixed when an SOF packet is received. Accordingly, all receive data must be read until the next SOF packet is received. When the next SOF packet is received, the FIFO side is automatically switched over, and the previous data will not be possible to be read. This FIFO buffer can be initialized by means of EP4CLR in the FCLR1 register.

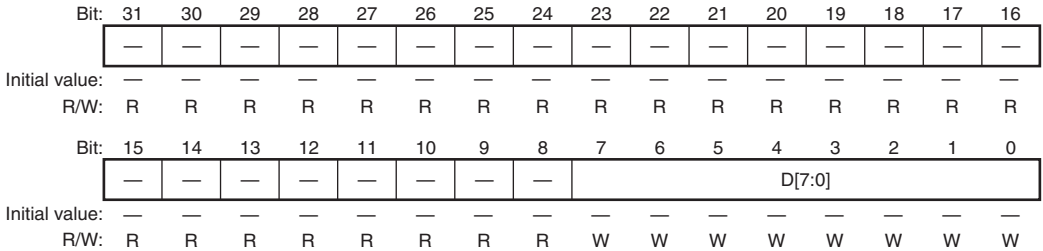
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	D[7:0]	Undefined	R	Data register for endpoint 4 transfer

36.3.23 EP5 Data Register (EPDR5)

EPDR5 is a 128-byte transmit FIFO buffer for endpoint 5. EPDR5 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and an SOF packet is received, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. This FIFO buffer can be initialized by means of EP5CLR and EP5CCLR in the FCLR1 register. (EP5CLR initializes both FIFOs and EP5CCLR initializes one FIFO which is connected to the CPU.)



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Data register for endpoint 5 transfer

36.3.24 EP0o Receive Data Size Register (EPSZ0o)

EPSZ0o is a receive data size resister for endpoint 0o. EPSZ0o indicates the number of bytes received from the host.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	—	All 0	R	Number of receive data for endpoint 0

36.3.25 EP1 Receive Data Size Register (EPSZ1)

EPSZ1 is a receive data size resister for endpoint 1. EPSZ1 indicates the number of bytes received from the host. FIFO of endpoint 1 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

36.3.26 EP4 Receive Data Size Register (EPSZ4)

EPSZ4 is a receive data size resister for endpoint 4. EPSZ4 indicates the number of bytes received from the host. FIFO of endpoint 4 has a dual-buffer configuration. The size of the received data indicated by this register is the size of the currently selected side (can be read by CPU).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	—	All 0	R	Number of received bytes for endpoint 4

36.3.27 Trigger Register (TRG)

TRG generates one-shot triggers FIFO for each endpoint of EP0s, EP0i, EP0o, EP1, EP2, and EP3. The packet enable trigger for the IN FIFO register and read complete trigger for the OUT FIFO register are triggers to be given.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP0i PKTE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7	—	0	W	Reserved The write value should always be 0.
6	EP3 PKTE	0	W	EP3 Packet Enable
5	EP1 RDFN	0	W	EP1 Read Complete
4	EP2 PKTE	0	W	EP2 Packet Enable
3	—	0	W	Reserved The write value should always be 0.
2	EP0s RDFN	0	W	EP0s Read Complete
1	EP0o RDFN	0	W	EP0o Read Complete
0	EP0i PKTE	0	W	EP0i Packet Enable

36.3.28 Data Status Register (DASTS)

DASTS indicates whether the IN FIFO data register contains valid data. DASTS is set to 1 when data written to IN FIFO is enabled by writing PKTE in TRG to 1, and cleared when all data has been transmitted to the host. In case of a dual-configuration FIFO for endpoint 2, this bit is cleared to 0 when both sides are empty.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	EP3 DE	EP2 DE	—	—	—	EP0i DE
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3 DE	0	R	EP3 Data Enable
4	EP2 DE	0	R	EP2 Data Enable
3 to 1	—	All 0	R	Reserved These bits are already read as 0.
0	EP0iDE	0	R	EP0i data enable

36.3.29 FIFO Clear Register 0 (FCLR0)

FCLR is a one shot register to clear the FIFO buffers for endpoints 0 to 3. Writing 1 to a bit clears the data in the corresponding FIFO buffer.

In case of reception FIFO, by writing data in the FIFO buffer, the data by which PKTE in TRG is not written to 1 and the data enabled by writing 1 can be cleared. In case of OUT FIFO, the data of which reception has not been completed can be cleared.

Both sides of the dual-configuration FIFO buffers (EP1 or EP3) can be cleared.

The corresponding interrupt flag is not cleared by this clear instruction. Do not clear a FIFO buffer during transmission and reception.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	EP3 CLR	EP1 CLR	EP2 CLR	—	—	EP0o CLR	EP0i CLR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7	—	Undefined	W	Reserved The write value should always be 0.
6	EP3 CLR	Undefined	W	EP3 Clear
5	EP1 CLR	Undefined	W	EP1 Clear
4	EP2 CLR	Undefined	W	EP2 Clear
3, 2	—	Undefined	W	Reserved The write value should always be 0.
1	EP0o CLR	Undefined	W	EP0o Clear
0	EP0i CLR	Undefined	W	EP0i Clear

36.3.30 FIFO Clear Register 1 (FCLR1)

FCLR is a one shot register to clear the FIFO buffers for endpoints 4 and 5. Writing 1 to a bit clears the data in the corresponding FIFO buffer.

The corresponding interrupt flag is not cleared by this clear instruction. Do not clear a FIFO buffer during transmission and reception.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	EP5 CCLR	—	—	EP5 CLR	EP4 CLR
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 5	—	Undefined	W	Reserved The write value should always be 0.
4	EP5 CCLR	Undefined	W	EP5 CPU Clear
3, 2	—	Undefined	W	Reserved The write value should always be 0.
1	EP5 CLR	Undefined	W	EP5 Clear
0	EP4 CLR	Undefined	W	EP4 Clear

36.3.31 DMA Transfer Setting Register (DMA)

DMA is set when the dual address transfer is used to the data register for endpoints 1 and 2 to which transfer is possible by DMA. The USB_PWREN/USBF_UPLUP pin level can be controlled by the bit 2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PULLUP E	EP2 DMAE	EP1 DMAC
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PULLUP E	0	R/W	Pull-up Enable Controls connection notification to USB host/hub. 0: USB_PWREN/USBF_UPLUP pin goes high 1: USB_PWREN/USBF_UPLUP pin goes low
1	EP2 DMAE	0	R/W	EP2DMA Enable Enables DMA transfer for EP2.
0	EP1 DMAE	0	R/W	EP1DMAE Enable Enables DMA transfer for EP1.

36.3.32 Endpoint Stall Register 0 (EPSTL0)

EPSTL stalls each endpoint. The endpoint in which the stall bit is set to 1 returns a stall handshake to the host from the next transfer when 1 is written to. The stall bit for endpoint 0 is cleared automatically on reception of 8 byte command data for which decoding is performed by the function and the EP0 STL bit is cleared. When the SETUPTS flag bit in the IFR0 register is set to 1, a write of the EP0 STL bit to 1 is ignored. For detailed operation, see section 36.8, Stall Operations.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	EP3 STL	EP2 STL	EP1 STL	EP0 STL
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 4	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	EP3 STL	0	R/W	EP3 Stall Sets EP3 stall
2	EP2 STL	0	R/W	EP2 Stall Sets EP2 stall
1	EP1 STL	0	R/W	EP1 Stall Sets EP1 stall
0	EP0 STL	0	R/W	EP0 Stall Sets EP0 stall

36.3.33 Endpoint Stall Register 1 (EPSTL1)

EPSTL stalls each endpoint. The endpoint in which the stall bit is set to 1 returns a stall handshake to the host from the next transfer when 1 is written to. For detailed operation, see section 36.8, Stall Operations.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EP5 STL	EP4 STL
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 2	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	EP5 STL	0	R/W	EP5 Stall Sets EP5 stall
0	EP4 STL	0	R/W	EP4 Stall Sets EP4 stall

36.3.34 Configuration Value Register (CVR)

CVR is a register to store the Configuration/Interface/value to be set when the Set Configuration/Set Interface command is normally received.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CNFV[1:0]		INTV[1:0]		—	ALTV[2:0]		
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

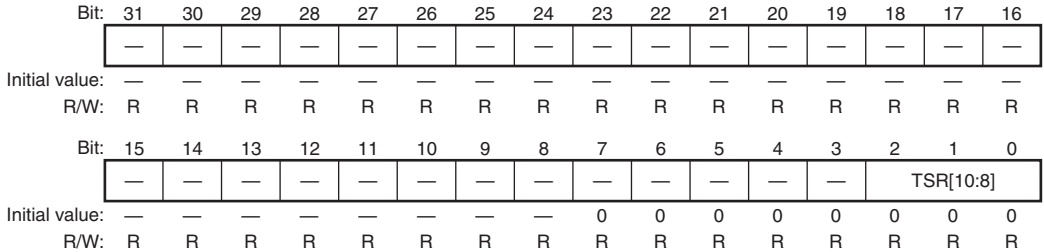
Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7, 6	CNFV[1:0]	00	R	Configuration Value The configuration setting value is stored when the Set Configuration command has been received. CNFV is updated when the SETC bit in the interrupt flag register is set to 1.
5, 4	INTV[1:0]	00	R	Interface Value The interface setting value is stored when the Set Interface command has been received. INTV is updated when the SETI bit in the interrupt flag register is set to 1.
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	ALTV[2 : 0]	000	R	Alternate Value The alternate setting value is stored when the Set interface command has been received. ALTV is updated when the SETI bit in the interrupt flag register is set to 1.

36.3.35 Time Stamp Register (TSRH/TSRL)

TSR is a register to store the current time stamp value. The time stamp is updated when the SOF bit in IFR0 is set to 1. The value of the time stamp when the SOF mark function is enabled and the SOF packet is broken remains as previous one.

TSR is handled as an 11-bit (TSR[10:0]) register in the USBF which consists of TSR[10:8] bits in TSRH and TSR[7:0] bits in TSRL. Although TSRH can be read directly, TSRL is read via an 8-bit temporary register. Therefore, the registers should be accessed in the order of TSRH and TSRL in byte units. TSRL cannot be read singly.

- TSRH



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TSR[10:8]		
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 3	—	All 0	R	Reserved These bits are always read as 0.
2 to 0	TSR[10:8]	All 0	R	Upper Three Bits of Time Stamp Data

- TSRL

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TSR[7:0]							
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value.
7 to 0	TSR[7:0]	All 0	R	Lower Eight Bits of Time Stamp Data

36.3.36 Control Register 0 (CTLR0)

CTLR0 sets functions of ASCE, PWMD, RSME, and RWUP.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RWUPS	RSME	PWMD	ASCE	—
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R

Bit	Bit name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RWUPS	0	R	Remote Wakeup Status Status bit to indicate that the remote wakeup from the host is enabled/disabled. Indicates 0 when the remote wakeup is disabled with Device Remote Wakeup by the Set Feature/Clear Feature request and indicates 1 when it is enabled.
3	RSME	0	R/W	Resume Enable Bit to clear the suspend state (performs the remote wakeup) When this bit is written to 1, a resume register is set. When this bit will be used, be sure to hold to 1 for one clock or more at 12 MHz in minimum and then clear to 0 again.
2	PWMD	0	R/W	Power Mode Sets USBF power mode. 0: Self-powered 1: Bus-powered

Bit	Bit name	Initial value	R/W	Description
1	ASCE	0	R/W	Automatic Stall Clear Enable When this bit is set to 1, the stall handshake is returned to the host and the stall setting bit (EPSTLR/EPXSTL) of the returned endpoint is automatically cleared. Control in a unit of endpoint is disabled as this bit is common for all endpoints. When this bit is set to 0, be sure to clear the stall setting bit of each endpoint by using software. This bit should be set to 1 before each stall bit in EPSTL is set to 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

36.3.37 Control Register 1 (CTLR1)

CTLR1 makes settings of internal timer which is used in the isochronous transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMR ACLR	TMR EN
Initial value:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TMRACLR	1	R/W	Timer Auto Clear Selects method to clear TMR (timer register). 0: Not cleared. When clearing TMR, write 0 to TMR by CPU. 1: Automatically cleared every time when SOF is received.
0	TMREN	0	R/W	Timer Enable TMREN is TMR (timer register) enable bit. 0: Timer operation is disabled 1: Timer operation is enabled

36.3.38 Endpoint Information Register (EPIR)

EPIR is a register to set the configuration information for each endpoint. 5 bytes of the information are required for one endpoint and the formats are listed in tables 36.4 and 36.5. Write the data in order from endpoint 0. Do not write more than 5 (bytes) × 10 (endpoints) = 50 bytes. Write this information once at power-on reset. Do not write it again afterwards.

Write data of one endpoint is described below. EPIR writes data in the same address in order. Therefore though there is only one EPIR register, write data for registration number N (N is from 0 to 9) is listed as EPIRN0 to EPIRN4 (EPIR [registration number] [write order]) for the purpose of explaining. Write data in order from EPIR00.

- EPIRN0:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:4]				D[3:2]		D[1:0]	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 4	D[7:4]	Undefined	W	Endpoint Number Settable range: 0 to 5
3, 2	D[3:2]	Undefined	W	Configuration Number to which Endpoint Belongs Settable range: 0 or 1
1, 0	D[1:0]	Undefined	W	Interface Number to which Endpoint Belongs Settable range: 0 to 3

- EPIRN1:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:6]		D[5:4]		D3	D[2:0]		
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7,6	D[7:6]	Undefined	W	Alternate Number to which Endpoint Belongs Settable range: 0 or 1
5, 4	D[5:4]	Undefined	W	Transfer Method of Endpoint Settable range: 0: Control 1: Isochronous 2: Bulk 3: Interrupt
3	D3	Undefined	W	Transfer Direction of Endpoint Settable range: 0: Out 1: In
2 to 0	D[2:0]	Undefined	W	Reserved The write value should always be 0.

- EPIRN2:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:1]							D0
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 1	D[7:1]	Undefined	W	Maximum Packet Size of Endpoint Settable range: 0 to 64
0	D0	Undefined	W	Reserved The write value should always be 0.

- EPIRN3:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Reserved The write value should always be 0.

- EPIRN4:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	D[7:0]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. Write value should always be 0.
7 to 0	D[7:0]	Undefined	W	Endpoint FIFO Number Settable range: 0 to 5

An endpoint number is an endpoint number used by the USB host. The endpoint FIFO number corresponds to the endpoint number which is described in this manual. When each endpoint number and endpoint FIFO number corresponds to each other, transfer can be performed between the USB host and the endpoint FIFO. Note that the setting values are limited as described below.

- Since each endpoint FIFO is optimized by a dedicated hardware corresponding to each transfer method, transfer direction, and maximum packet size, set the endpoint FIFO with a transfer method, transfer direction, and maximum packet size shown in the table below.

  Example: Endpoint FIFO number 1 cannot be set as other than bulk transfer, OUT, and maximum packet size (64 bytes). Although endpoint FIFO number 4 cannot be set as other than isochronous transfer and OUT, maximum packet size can be set in the range of 0 to 64 bytes.
- Endpoint 0 and endpoint FIFO number 0 must correspond.
- The maximum packet size of endpoint FIFO number 0 can be set to 8 bytes only.
- The setting value of endpoint FIFO number 0 can be set to the maximum packet size only and the rest data is all 0.
- The maximum packet size of endpoint FIFO numbers 1 and 2 can be set to 64 only.
- The maximum packet size of endpoint FIFO numbers 3 can be set to 8 only.
- The maximum packet size of endpoint FIFO numbers 4 and 5 can be set in the range of 0 to 64.

- When the isochronous transfer is set, Alternate can be used in the range of 0 and 1 for the same endpoint. Be sure to allocate the Alternate to the same endpoint FIFO number.
- Endpoint information can be set up to 10 in maximum.
- Endpoint information of 10 pieces must be written.
- All information of endpoints which are not used must be written as 0.

A list of restrictions of settable transfer method, transfer direction, and maximum packet size is described in table 36.4.

Table 36.4 Restrictions of Settable Values

Endpoint FIFO No.	Maximum Packet Size	Transfer Method	Transfer Direction
0	8 bytes	Control	—
1	64 bytes	Bulk	OUT
2	64 bytes	Bulk	IN
3	8 bytes	Interrupt	IN
4	0 to 64 bytes	Isochronous	OUT
5	0 to 64 bytes	Isochronous	IN

- Example of Setting

  This is an example when endpoint 4 and 5 used for the isochronous transfer are allocated with Alternate value.

Table 36.5 Example of Endpoint Configuration

EP No.	Conf.	Int.	Alt.	Transfer Method	Transfer Direction	Maximum Packet Size	EP FIFO No.
0	—	—	—	Control	IN/OUT	8 bytes	0
1	1	0	0	Bulk	OUT	64 bytes	1
2	1	0	0	Bulk	IN	64 bytes	2
3	1	0	0	Interrupt	IN	8 bytes	3
—	1	1	0	—	—	—	—
—	1	1	1	—	—	—	—
4	1	2	0	Isochronous	OUT	0 bytes	4
4	1	2	1	Isochronous	OUT	64 bytes	4
5	1	3	0	Isochronous	IN	0 bytes	5
5	1	3	1	Isochronous	IN	64 bytes	5

Table 36.6 Example of Setting of Endpoint Configuration Information

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[N]4
0	00	00	10	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	10	00	03
4	00	00	00	00	00
5	00	00	00	00	00
6	46	10	00	00	04
7	46	50	80	00	04
8	67	18	00	00	05
9	57	58	80	00	05

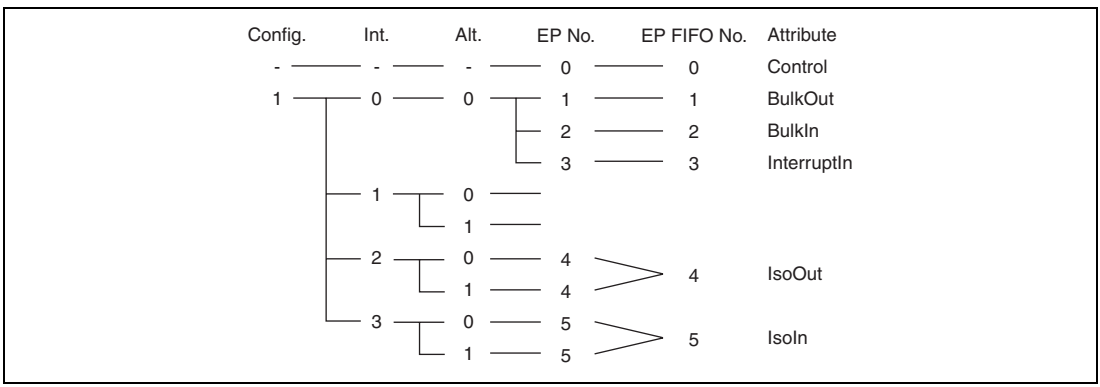


Figure 36.2 Example of Endpoint Configuration

36.3.39 Timer Register (TMRH/TMRL)

TMRH/TMRL is a 16-bit timer which is operated with a peripheral clock ϕ. Measuring the SOF packet reception interval enables the SOF packet break to be detected.

The timer is operated, stopped, and cleared according to the settings of the control register 1 (CTLR 1)

TMRH/TMRL is handled as a 16-bit (TMR[15:0]) register in the USBF which consists of TMR[15:8] bits in TMRH and TMR[7:0] bits in TMRL. Although TMRH can be read directly, TMRL is read via an 8-bit temporary register. Therefore, the registers should be read in the order of TMRH and TMRL in byte units. TMRL cannot be read singly.

- TMRH

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMR[15:8]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. The write value should always be 0.
7 to 0	TMR[15:8]	0	R/W	Upper Eight Bits of Count Value

- TMRL

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMR[7:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. The write value should always be 0.
7 to 0	TMR[7:0]	0	R/W	Lower Eight Bits of Count Value

36.3.40 Set Time Out Register (STOH/STOL)

STOH/STOL specifies the time out value of the timer register. When the count value of the timer register reaches the specified time out value, the time out interrupt flag in the interrupt flag register 4 is set.

STOH/STOL is handled as a 16-bit (STO[15:0]) register in the USBF which consists of STO[15:8] bits in STOH and STO[7:0] bits in STOL. Although STOH can be read directly STOL is read via an 8-bit temporary register. Therefore, the registers should be read in the order of STOH and STOL in byte units. STOL cannot be read singly.

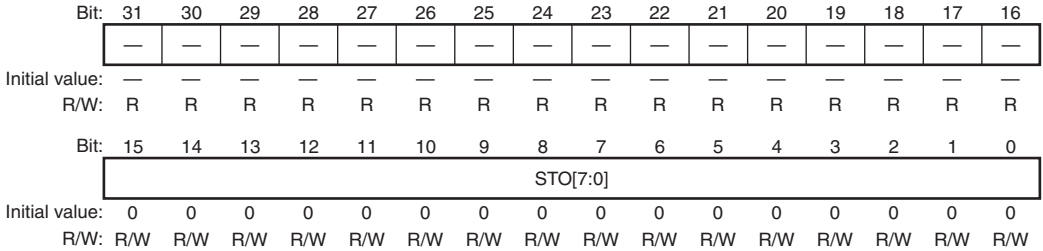
- STOH

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STO[15:8]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved These bits are always read as undefined value. The write value should always be 0.
7 to 0	STO[15:8]	0	R/W	Upper Eight Bits of Specified Time Out Value

- STOL



Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
7 to 0	STO[7:0]	0	R/W	Lower Eight Bits of Specified Time Out Value

36.4 Operation

36.4.1 Cable Connection

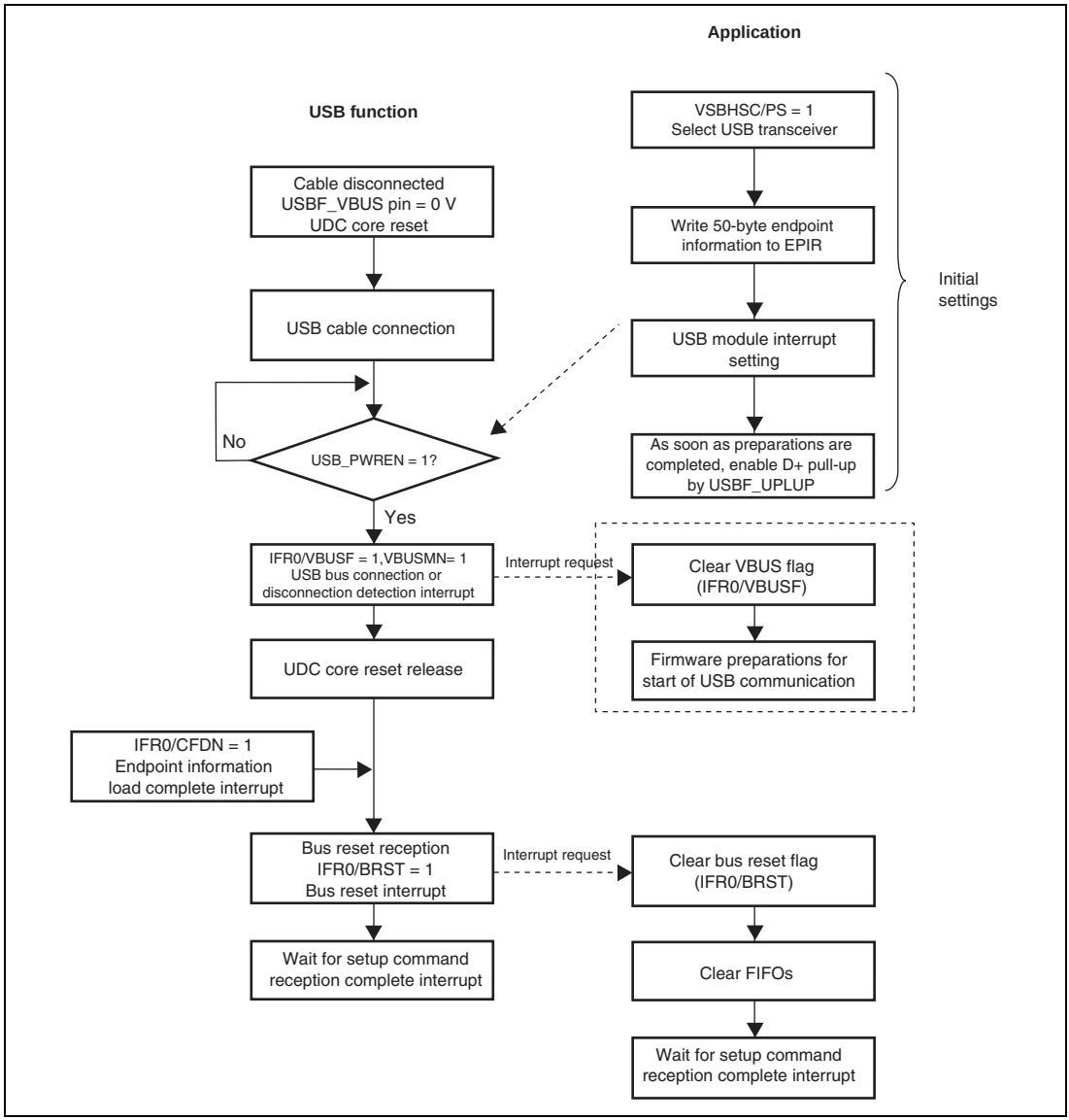


Figure 36.3 Cable Connection Operation

In applications that do not require USB cable connection to be detected, processing by the USB connection or disconnection detection interrupt is not necessary. Preparations should be made with the bus reset interrupt.

Also, in applications that require connection detection regardless of D+ pull-up control, detection should be carried out using IRQ or a general input port.

36.4.2 Cable Disconnection

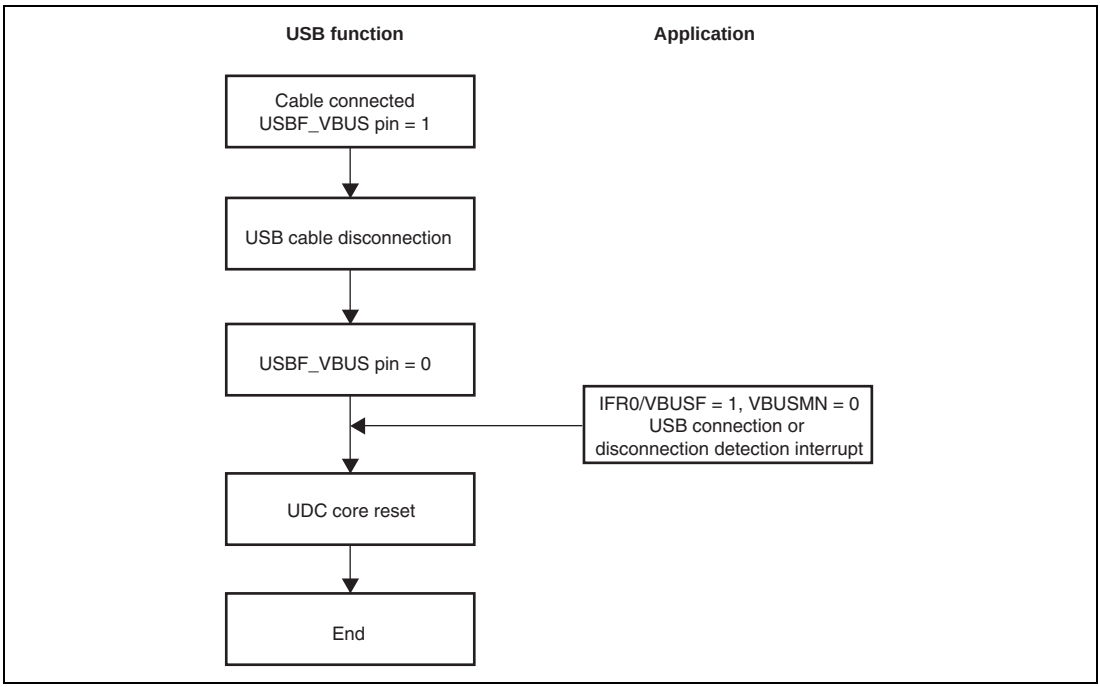
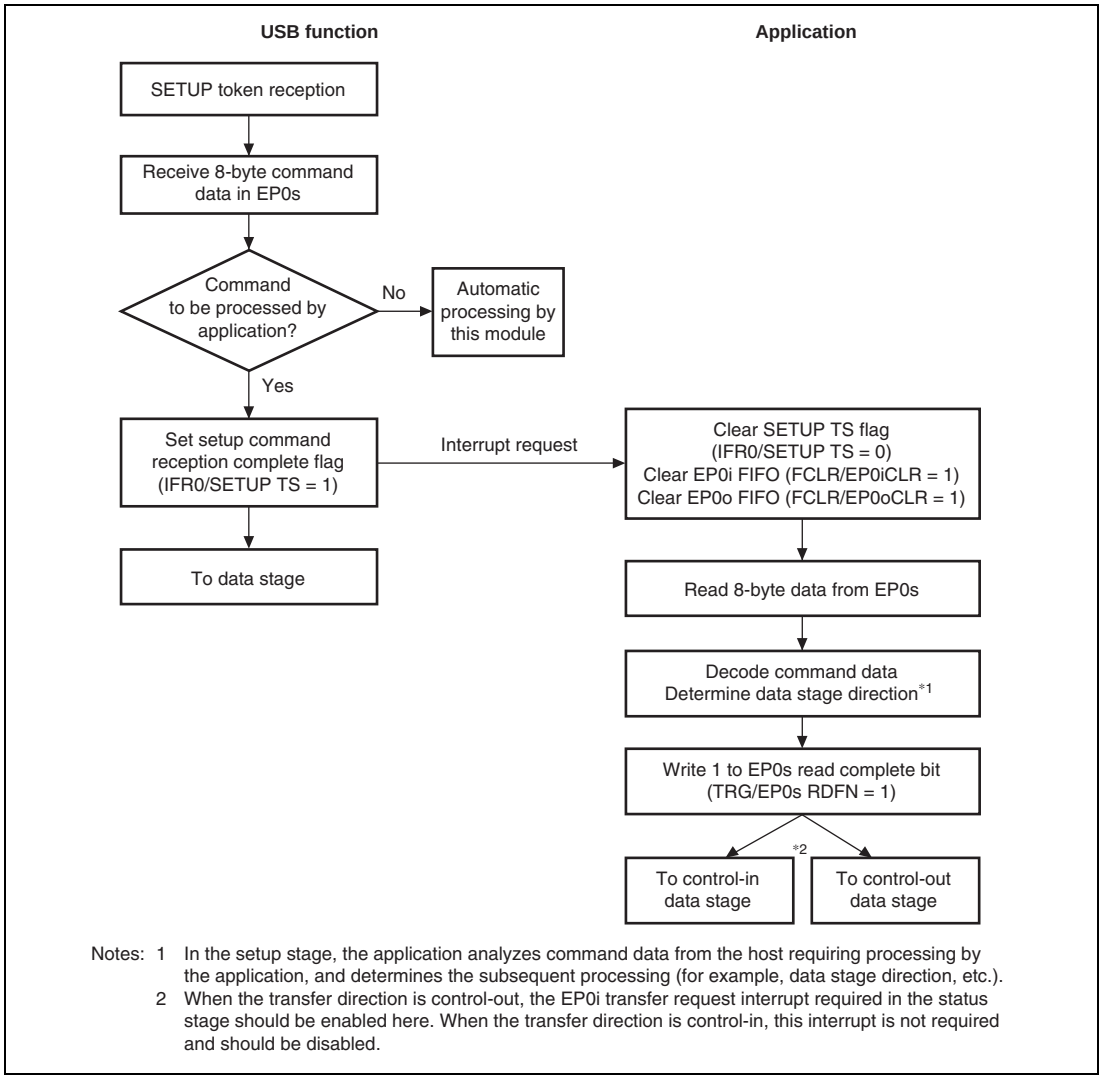


Figure 36.4 Cable Disconnection Operation

In applications that require connection/disconnection detection regardless of D+ pull-up control, detection should be carried out using IRQ or a general input port.

- Setup Stage



Notes: 1 In the setup stage, the application analyzes command data from the host requiring processing by the application, and determines the subsequent processing (for example, data stage direction, etc.).
2 When the transfer direction is control-out, the EP0i transfer request interrupt required in the status stage should be enabled here. When the transfer direction is control-in, this interrupt is not required and should be disabled.

Figure 36.5 Setup Stage Operation

- Data Stage (Control-In)

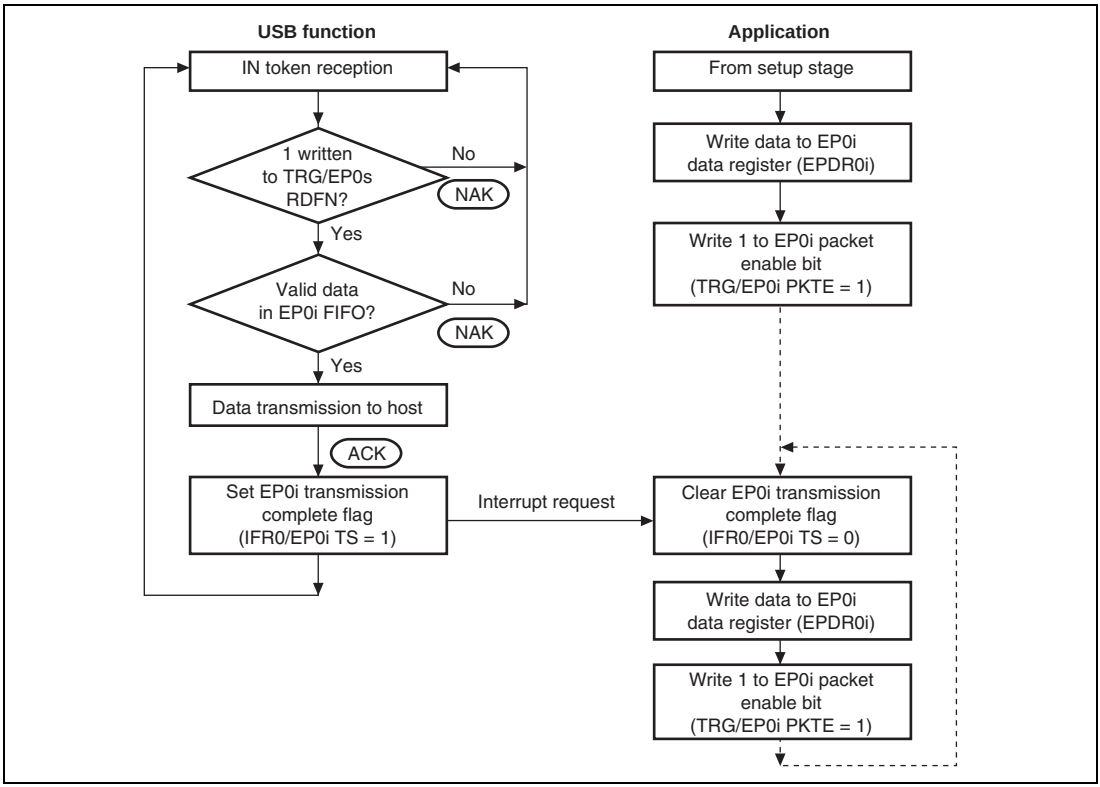


Figure 36.6 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (IFR0/EP0i TS = 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requested by the host, the function indicates the end of the data stage by returning to the host a packet shorter than the maximum packet size. If the size of the data transmitted by the function is an integral multiple of the maximum packet size, the function indicates the end of the data stage by transmitting a zero-length packet.

- Data Stage (Control-Out)

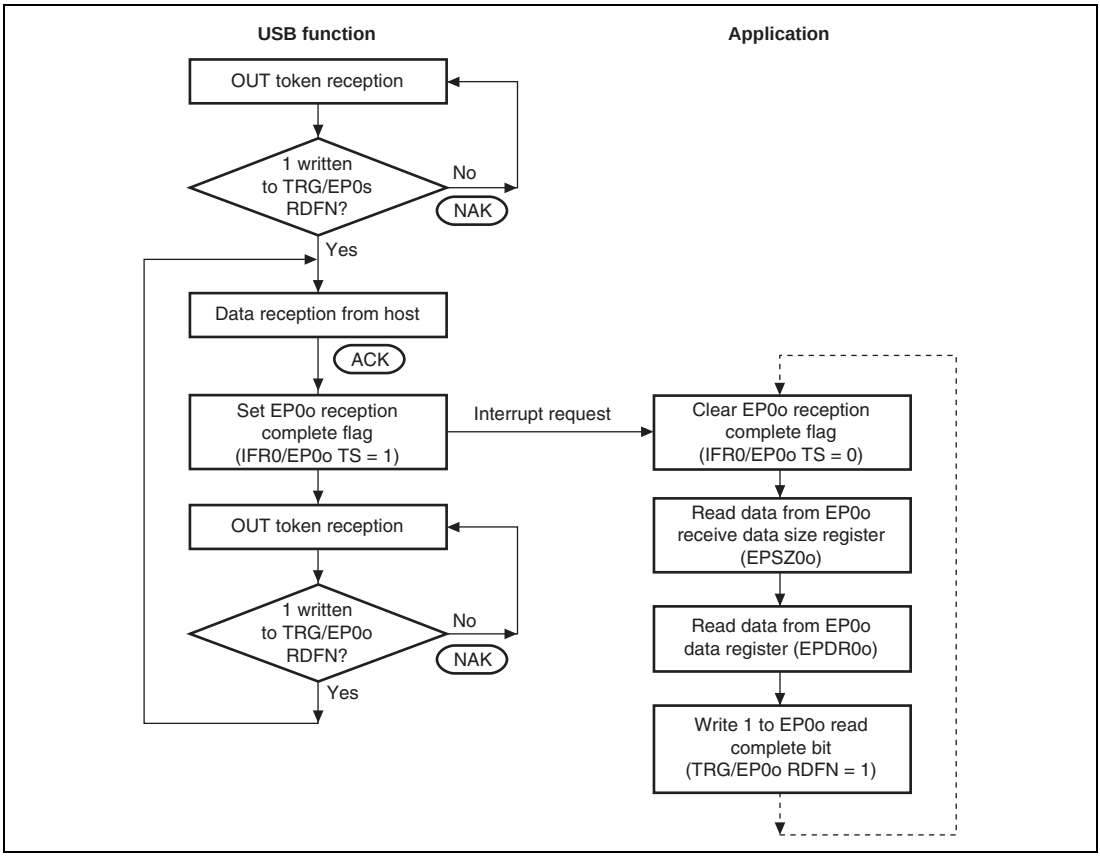


Figure 36.7 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is out-transfer, the application waits for data from the host, and after data is received (IFR0/EP0o TS = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read complete bit, empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status stage is entered.

- Status Stage (Control-In)

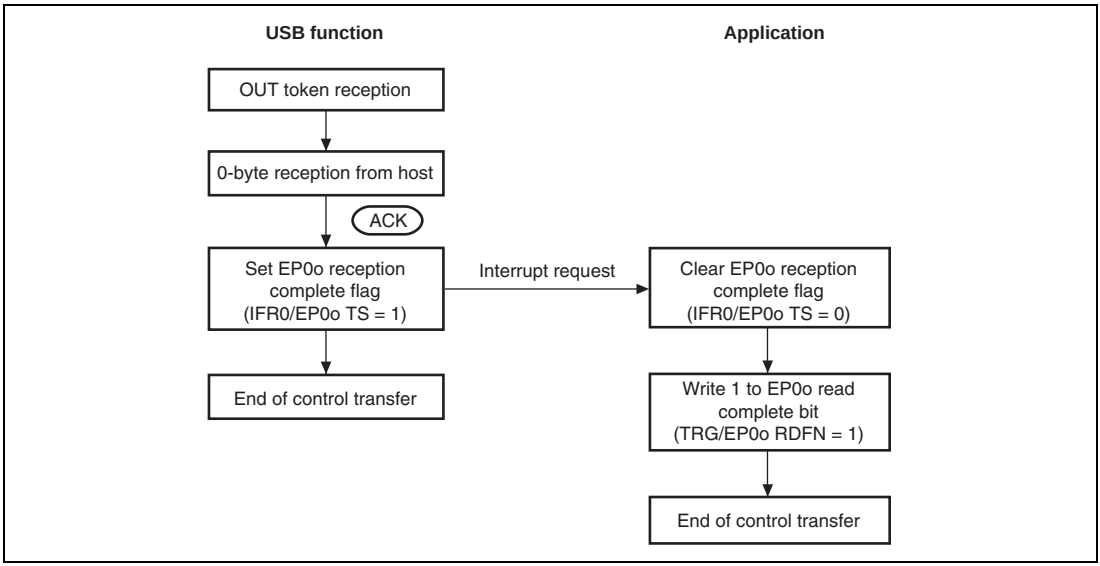


Figure 36.8 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application receives 0-byte data from the host, and ends control transfer.

- Status Stage (Control-Out)

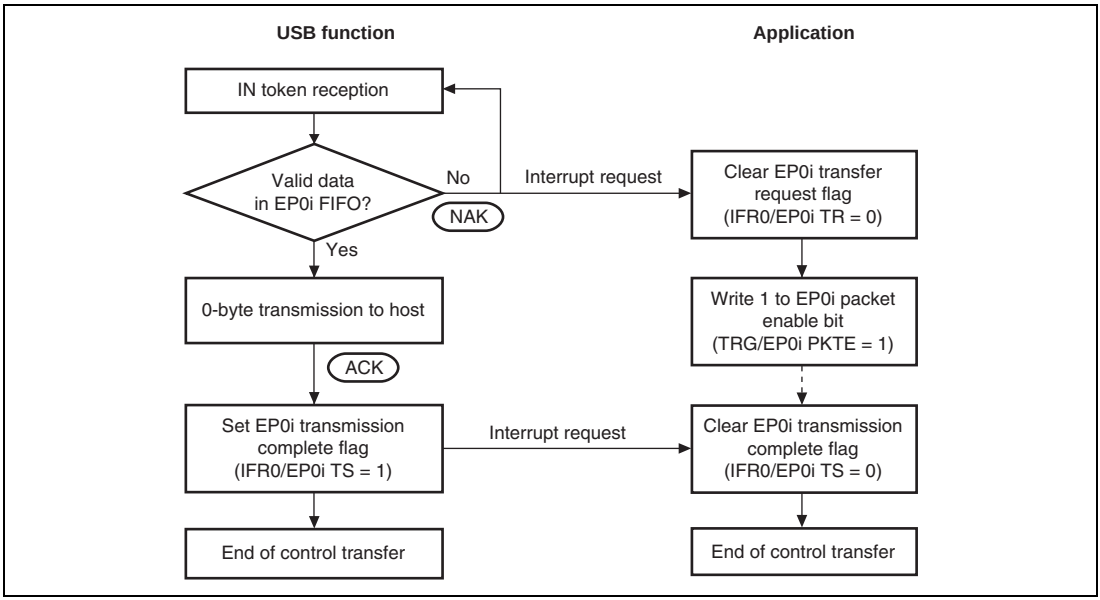


Figure 36.9 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is received at the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

36.4.3 EP1 Bulk-Out Transfer (Dual FIFOs)

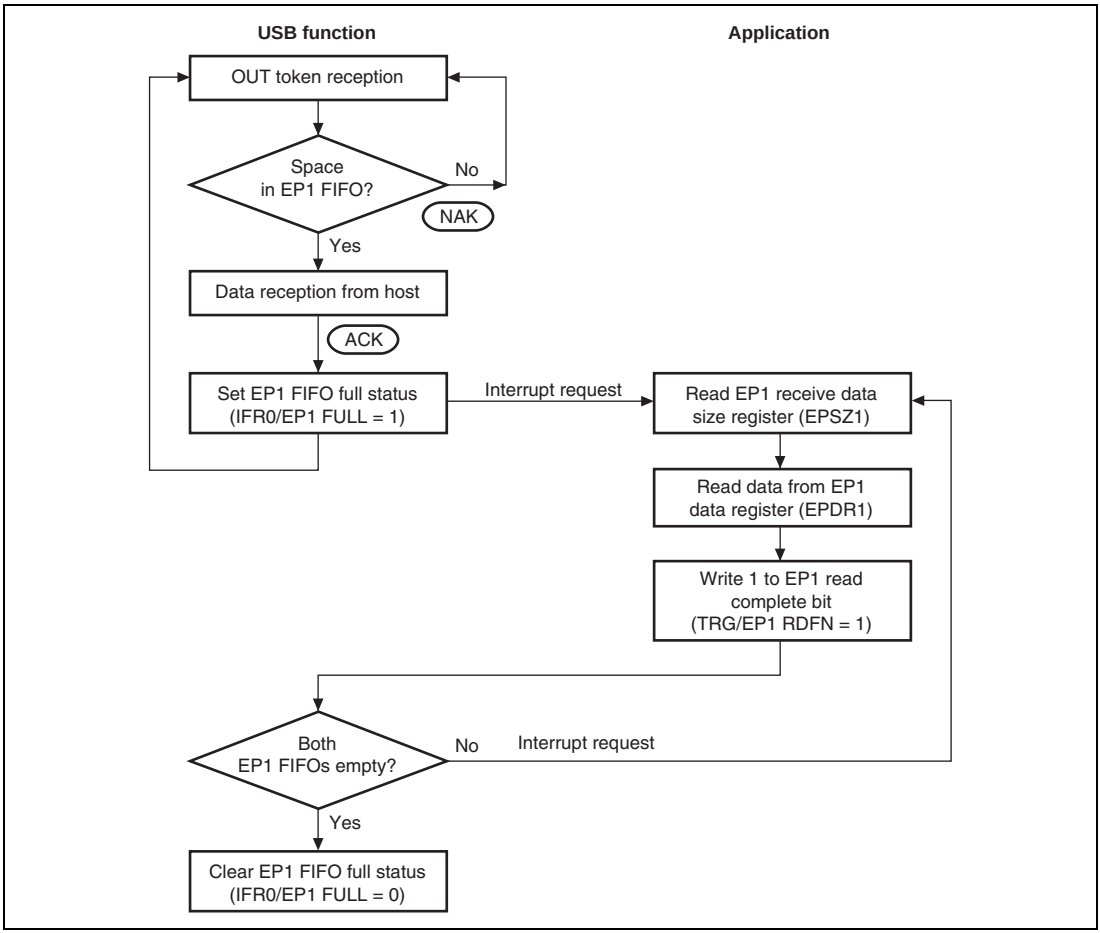


Figure 36.10 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can perform data reception and receive data reads without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the IFR0/EP1 FULL bit is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty, and so the next packet can be received immediately. When both FIFOs are full, NACK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the TRG/EP1 RDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

36.4.4 EP2 Bulk-In Transfer (Dual FIFOs)

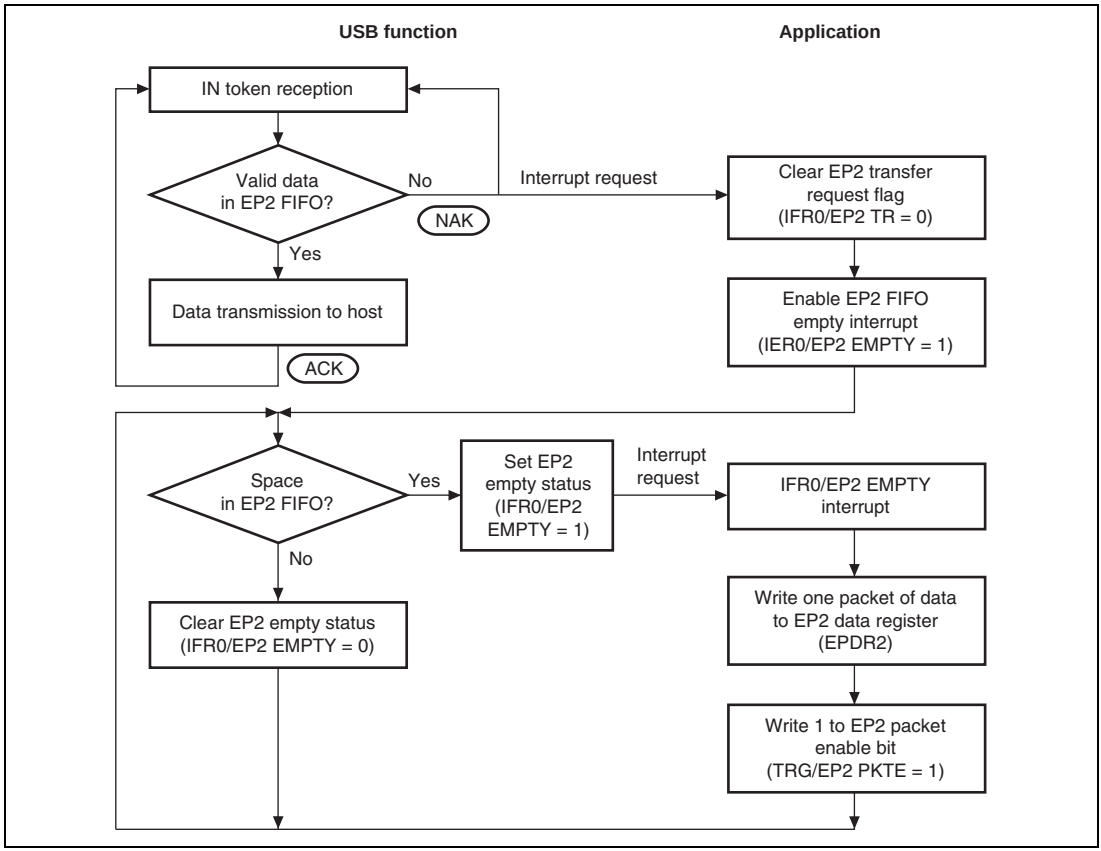


Figure 36.11 EP2 Bulk-In Transfer Operation

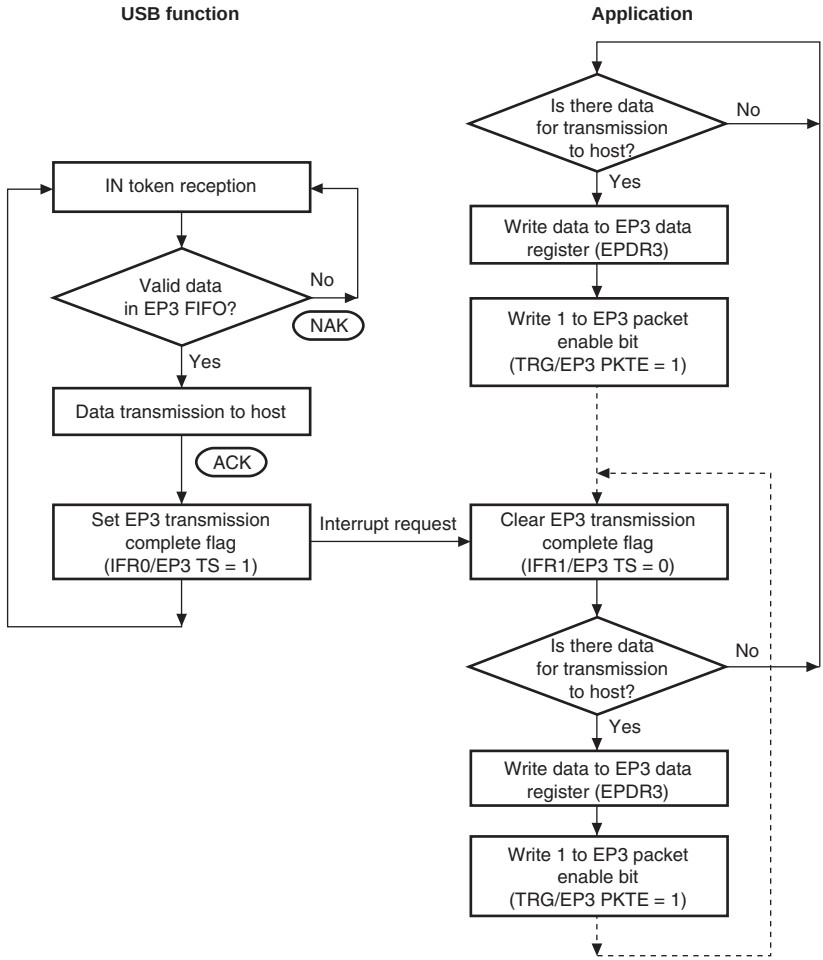
EP2 has two 64-byte FIFOs, but the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration. However, one data write is performed for one FIFO. For example, even if both FIFOs are empty, it is not possible to perform EP2/PKTE at one time after consecutively writing 128 bytes of data. EP2/PKTE must be performed for each 64-byte write.

When performing bulk-in transfer, as there is no valid data in the FIFOs on reception of the first IN token, an IFR0/EP2 TR interrupt is requested. With this interrupt, 1 is written to the IER0/EP2 EMPTY bit, and the EP2 FIFO empty interrupt is enabled. At first, both EP2 FIFOs are empty, and so an EP2 FIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write for one FIFO, the other FIFO is empty, and so the next transmit data can be written to the other FIFO immediately. When both FIFOs are full, EP2 EMPTY is cleared to 0. If at least one FIFO is empty, IFR0/EP2 EMPTY is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission can be continued.

When transmission of all data has been completed, write 0 to IER0/EP2 EMPTY and disable interrupt requests.

36.4.5 EP3 Interrupt-In Transfer



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include an operation flow in which, if there is data to be transferred, the EP3 DE bit in the USB data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 36.12 EP3 Interrupt-In Transfer Operation

36.5 EP4 Isochronous-Out Transfer

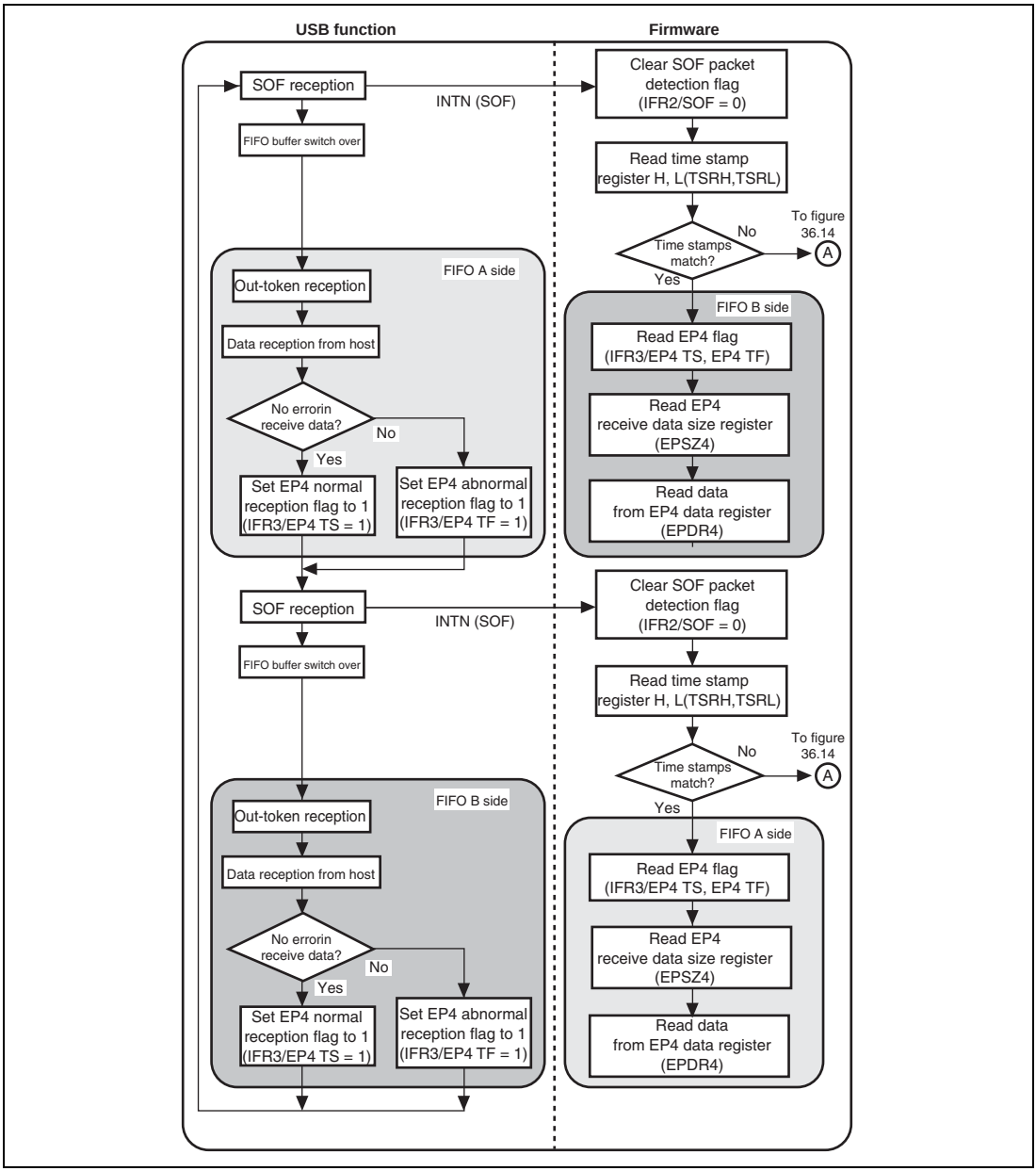


Figure 36.13 EP4 Isochronous-Out Transfer Operation (SOF is Normal)

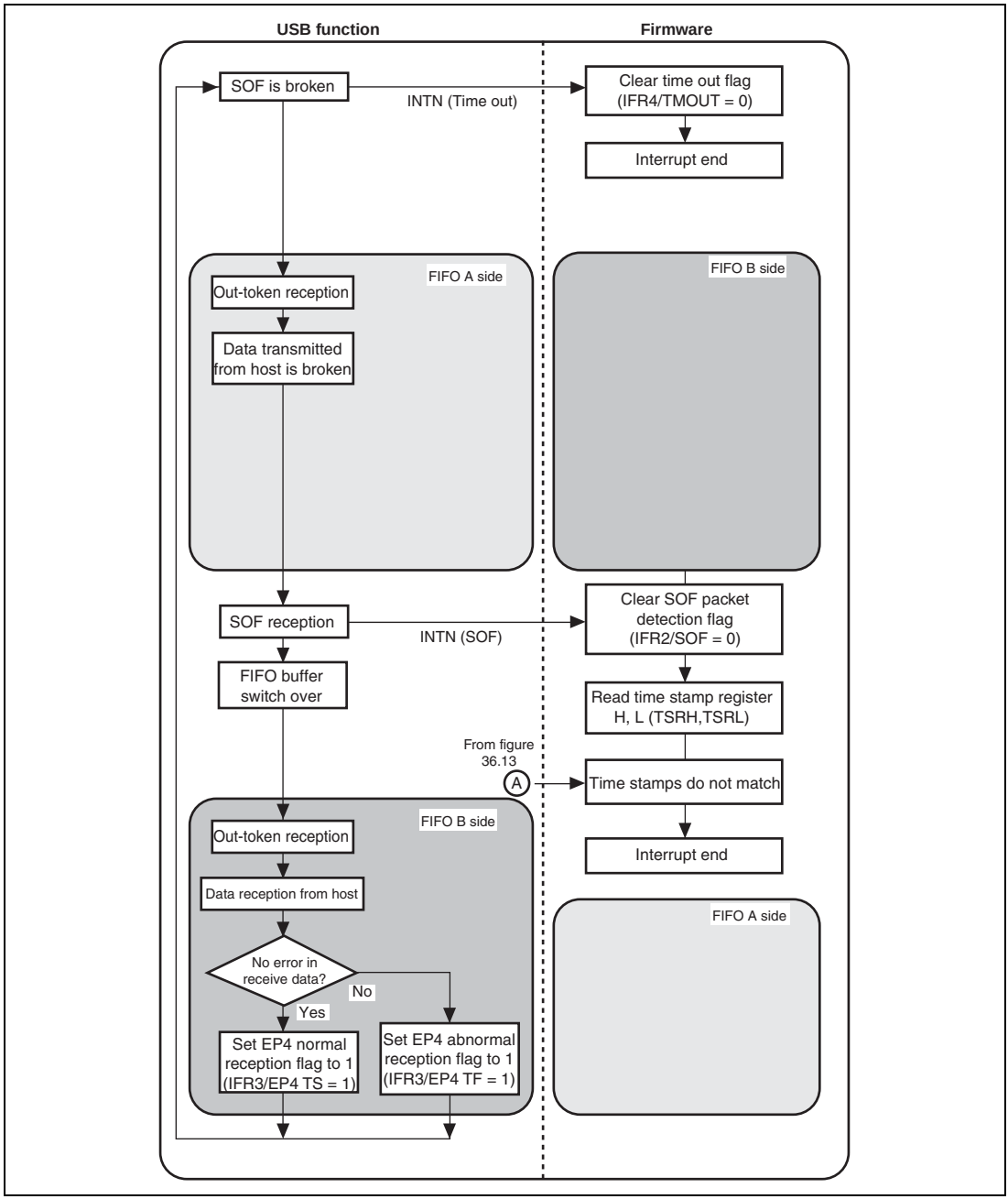


Figure 36.14 EP4 Isochronous-Out Transfer Operation (SOF is Broken)

Figure 36.13 shows the normal operation of the USB function and firmware in isochronous-out transfer.

EP4 has two up to 64-byte FIFOs, but the user can perform data transmission and read receive data without being aware of this dual-FIFO configuration.

In isochronous transfer, transfer is occurred only once per one frame (1 ms). So, when SOF is received, the FIFO buffer is switched automatically with hardware.

FIFO buffers are switched over by the SOF reception. Therefore, the FIFO buffer in which the USB function receives the data from the host and the FIFO buffer in which the firmware reads the receive data have different buffers, and a read and write of FIFO buffer are not competed. Accordingly, the data read by the firmware is the data received in one frame before. The buffers of FIFOs are switched over automatically by the SOF reception, so reading of data must be completed within the frame.

The USB function receives data from the host after an out-token is received. If there is an error in the data, set the internal TF flag to 1. If there is no error in the data, set the internal TS flag to 1.

In firmware, first, the processing routine of the isochronous transfer is called by SOF interrupt to check the time stamp. Then data is read from the FIFO buffer. The flag information (TS, TF) is read and decided if the data has an error. The flag information at this time represents the status of the currently readable FIFO buffer.

SOF happens to be broken because of external cause during transmission from the host. In this case, an operation flow is different from that in figure 36.13. As an example, figure 36.14 shows the operation flow of a broken frame and a subsequent frame when SOF is broken once. When SOF is broken, the FIFO buffer is not switched in current frame, and a time out interrupt is occurred after time set by user has been elapsed. The USB function controller discards the data which has been transmitted to the frame from the host.

The firmware detects the SOF break by the time out interrupt. In this case, the FIFO buffer connected to the CPU does not read data since data has already been read. When the SOF interrupt is occurred in the subsequent frame, the processing routine of the isochronous transfer is called and the time stamps are compared. The time stamps do not much since the SOF break occurred in the previous frame. Data is not read since the data in FIFO is not current one.

36.6 EP5 Isochronous-In Transfer

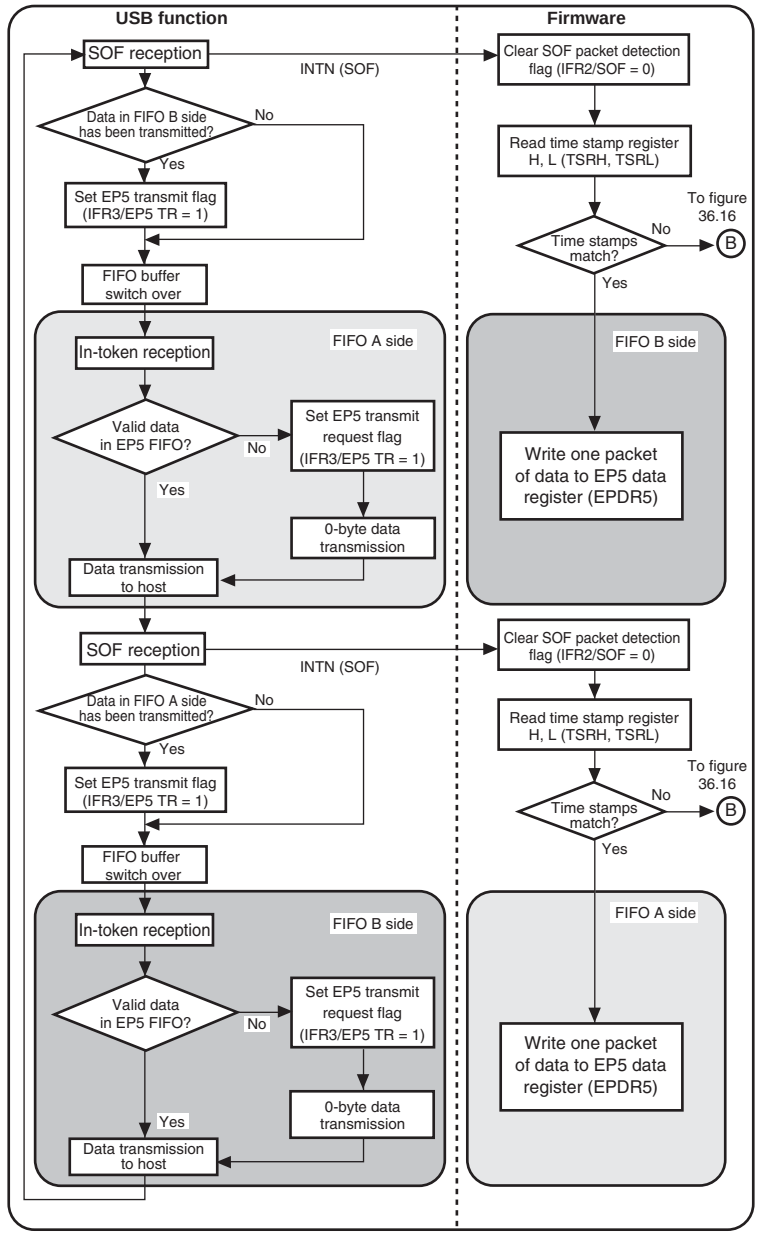


Figure 36.15 EP5 Isochronous-In Transfer Operation (SOF is Normal)

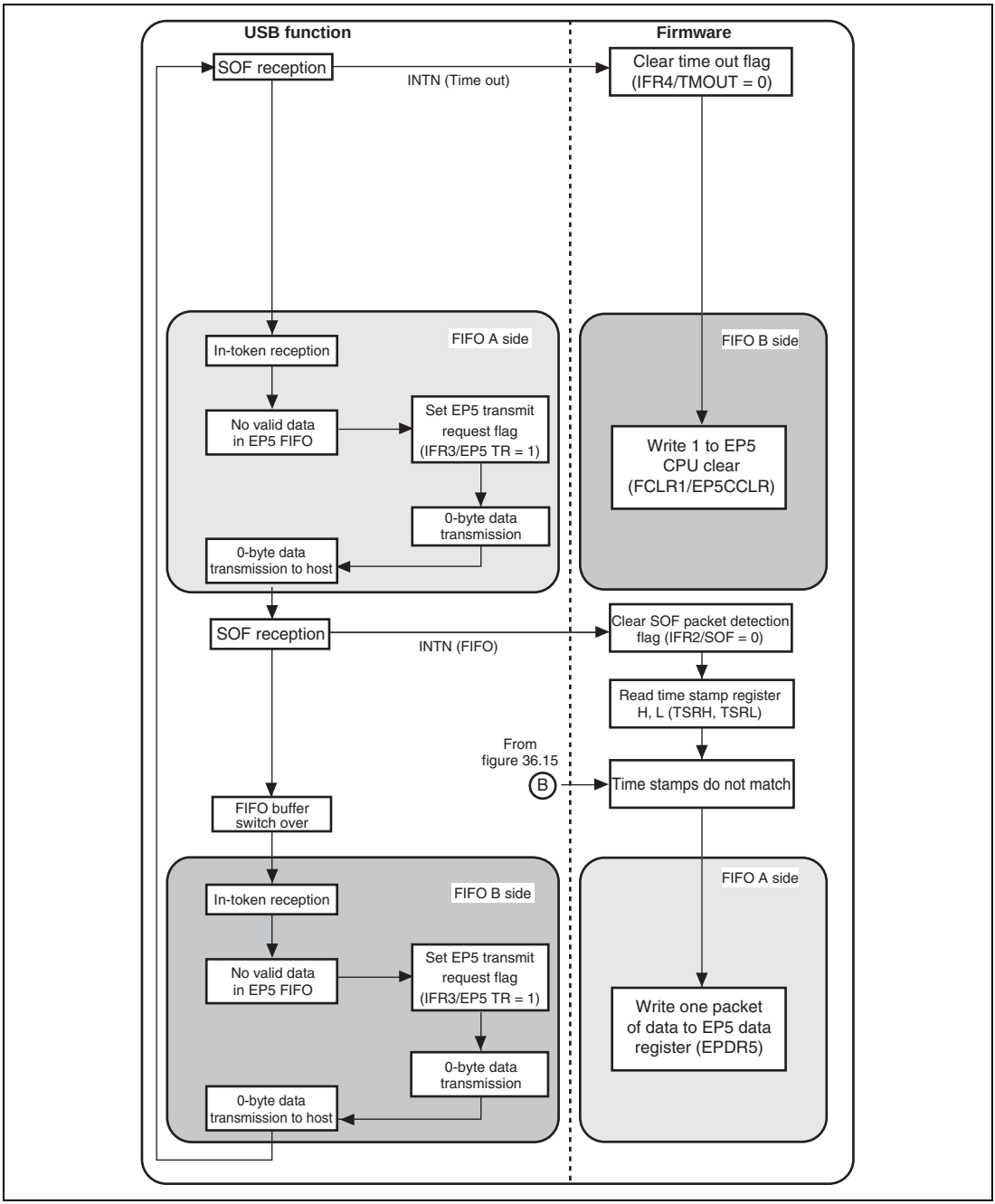


Figure 36.16 EP5 Isochronous-In Transfer Operation (SOF in Broken)

Figure 36.15 shows the normal operation of the USB function and firmware in isochronous-in transfer.

EP5 has two up to 64-byte FIFOs, but the user can perform data transmission and write transmit data without being aware of this dual-FIFO configuration.

In isochronous transfer, transfer is occurred only once per one frame (1 ms). So, when SOF is received, the FIFO buffer is switched automatically with hardware.

FIFO buffers are switched over by the SOF reception. Therefore, the FIFO buffer in which the USB function transmits the data and the FIFO buffer in which the firmware writes the transmit data have different buffers, and a read and write of FIFO buffer are not competed. Accordingly, the data written by the firmware is the data transmitted in one frame after. The buffers of FIFOs are switched over automatically by the SOF reception, so writing of data must be completed within the frame.

The USB function transmits data to the host, and the internal TR flag is set to 1, when data to be transmitted to the host exists in FIFO after an in-token is received. If there is no data in the FIFO buffer, set the internal TR flag to 1 and transmit 0-byte data to the host.

In firmware, first, the processing routine of the isochronous transfer is called by SOF interrupt to check the time stamp. Then one packet data is written to FIFO. This written data is transmitted to the host in the next frame.

SOF happens to be broken because of external cause during transmission from the host. In this case, an operation flow is different from that in figure 36.15. As an example, figure 36.16 shows the operation flow of a broken frame and a subsequent frame when SOF is broken once. When SOF is broken, the FIFO buffer is not switched in corresponding frame, and a time out interrupt is occurred after time set by user has been elapsed.

The firmware detects the SOF break by the time out interrupt. In this case, the FIFO buffer connected to the CPU has the data to be transmitted in the current frame. If this data is transmitted in the next frame, the data which is not current one is transmitted. Therefore, the firmware writes the EP5 CPU clear (FCLR1/EP5 CCLR) to 1. When the SOF interrupt is occurred in the subsequent frame, the processing routine of the isochronous transfer is called and the time stamps are compared. The time stamps do not much since the SOF break occurred in the previous frame. One packet of data is written by the firmware according to the transmitted time stamp.

In the frame in which the SOF is broken, the FIFO buffer is not switched and there in no data to be transmitted to the host. Therefore, USB function controller transmits 0-byte data to the host. Since the data to be transmitted is cleared by firmware, 0-byte data is transmitted to the host.

36.7 Processing of USB Standard Commands and Class/Vendor Commands

36.7.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Whether command decoding is required on the application side is indicated in table 36.7 below.

Table 36.7 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear feature	Get descriptor
Get configuration	Class/Vendor command
Get interface	Synch frame
Get status	Set descriptor
Set address	
Set configuration	
Set feature	
Set interface	

If decoding is not necessary on the application side, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the EP0s FIFO. After normal reception is completed, the IFR0/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, 8 bytes of data must be read from the EP0s data register (EPDR0S) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

36.8 Stall Operations

36.8.1 Overview

This section describes stall operations in this module. There are two cases in which the USB function controller stall function is used:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function controller due to a USB specification violation

The USB function controller has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit to EP0 is automatically cleared only when the setup command is received.

36.8.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function controller. When the application wishes to stall a specific endpoint, it sets the corresponding bit in EPSTL (1-1 in figure 36.17). The internal status bits are not changed at this time. When a transaction is sent from the host for the endpoint for which the EPSTL bit was set, the USB function controller references the internal status bit, and if this is not set, references the corresponding bit in EPSTL (1-2 in figure 36.17). If the corresponding bit in USBEPSTL is set, the USB function controller sets the internal status bit and returns a stall handshake to the host (1-3 in figure 36.17). In this time, if the CTLR/ASCE bit is set to 1, the corresponding bit in EPSTL is automatically cleared to 0 and a stall handshake is returned to the host (1-4 in figure 36.17). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 36.17), the USB function controller continues to return a stall handshake while the bit in EPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 36.17). To clear a stall, therefore, it is necessary for the corresponding bit in EPSTL to be cleared automatically when a stall is returned from the USB controller while the CTLR/ASCE bit is set to 1, or to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 36.17).

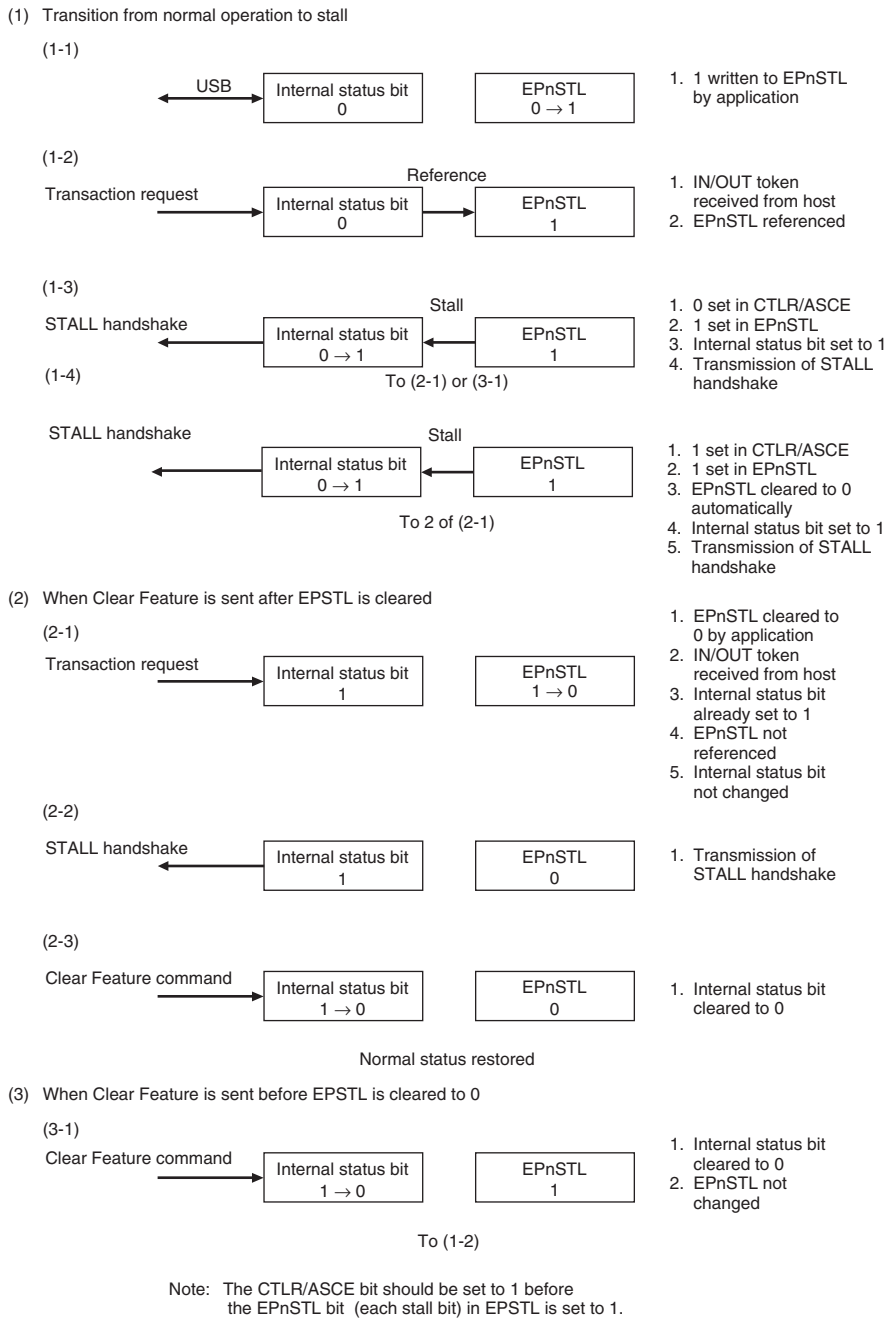


Figure 36.17 Forcible Stall by Application

36.8.3 Automatic Stall by USB Function Controller

When a stall setting is made with the Set Feature command, or in the event of a USB specification violation, the USB function controller automatically sets the internal status bit for the relevant endpoint without regard to the corresponding bit in EPSTL, and returns a stall handshake (1-1 in figure 36.18).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the corresponding bit in EPSTL. After a bit is cleared by the Clear Feature command, the corresponding bit in EPSTL is referenced (3-1 in figure 36.18). The USB function controller continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 36.18). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 36.18). In this time, if set by the application, the corresponding bit in EPSTL should also be cleared (2-1 in figure 36.18).

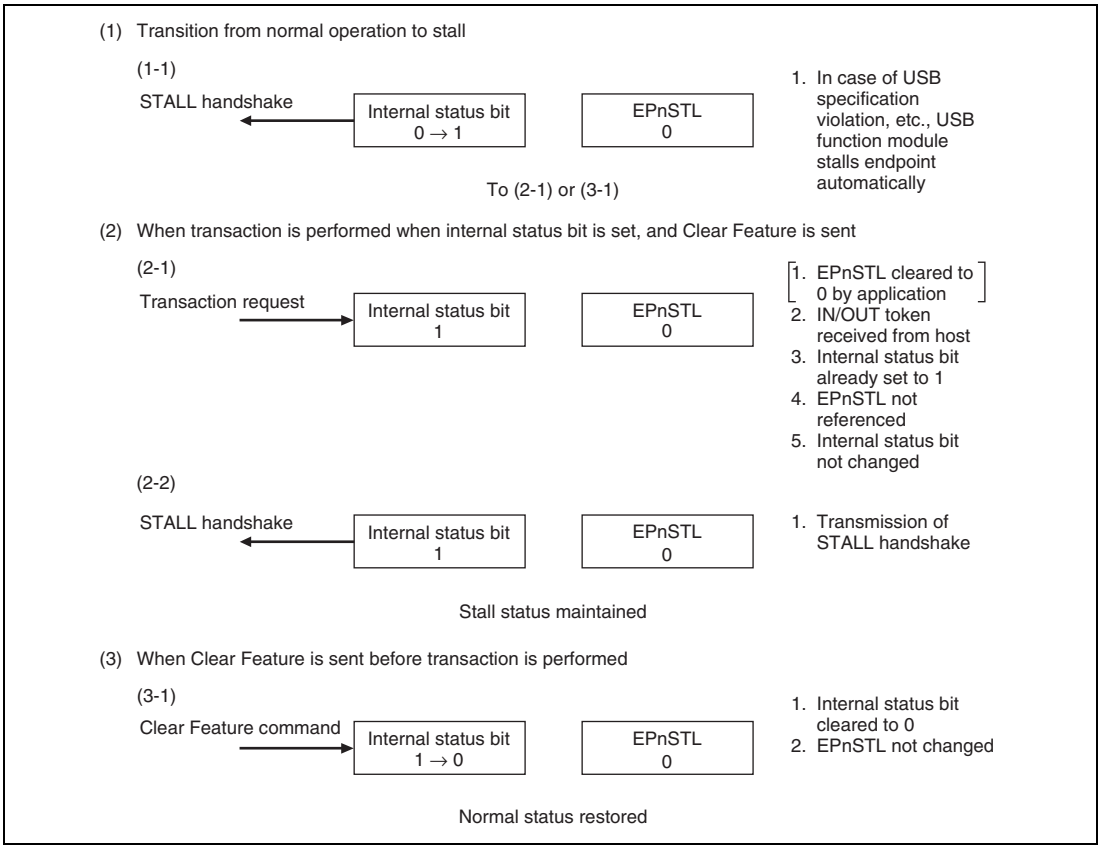


Figure 36.18 Automatic Stall by USB Function Controller

36.9 Examples of External Circuit

36.9.1 Example of the Connection between USB Function Controller

Figures 36.19 shows an example connection of USB function controller.
When using the USB function controller, the signals must be input to the cable connection monitor pin USBF_VBUS. The USBF_VBUS pin is multiplexed with the USB_PWREN pin. According to the status of the USBF_VBUS pin, the USB function controller recognizes whether the cable is connected/disconnected. Also, pin D+ must be pulled up in order to notify the USB host/hub that the connection is established. The sample circuit in figure 36.19 uses the USBF_UPLUP pin for pull-up control.

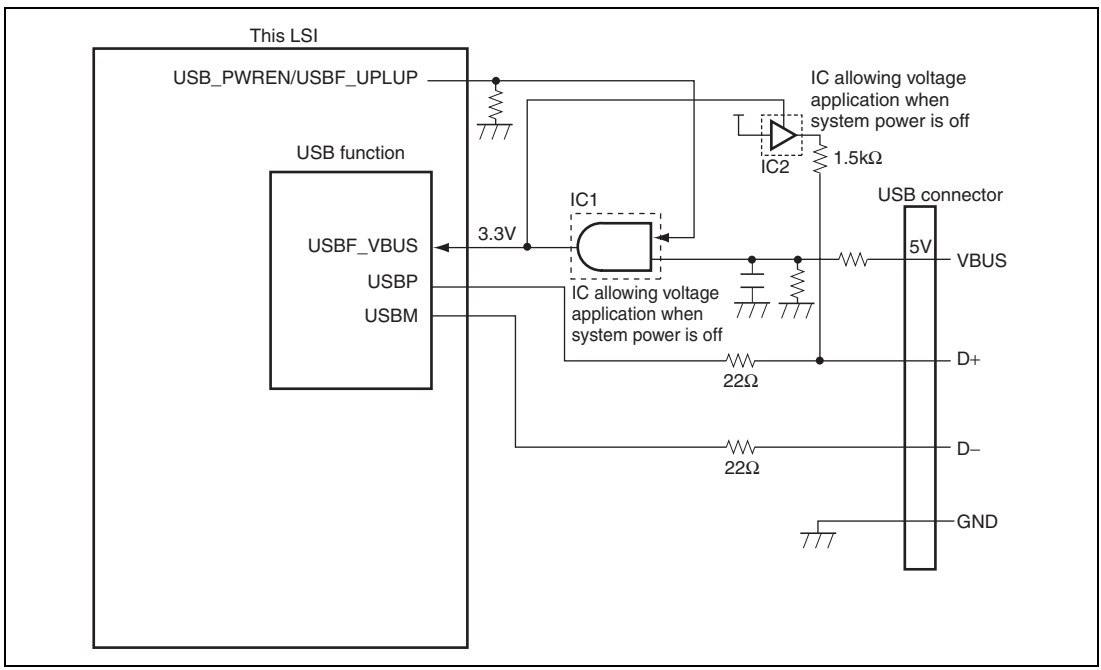


Figure 36.19 Example of Transceiver Connection for USB function Controller

36.10 Usage Notes

36.10.1 Setup Data Reception

The following points should be noted on the EP0s data register (EPDR0s) in which reception of 8-byte setup data is performed.

1. Since the setup command must be received in the USB, writing from the USB bus side is prior to reading from the CPU side. While the CPU reads data after completion of reception and reception of the next setup command is started, reading from the CPU side is forcibly invalid. Therefore a value to be read after starting reception is undefined.
2. EPDR0s must be read in 8-byte units. If reading is suspended while it is in progress, data received in the next setup cannot be read successfully.

36.10.2 FIFO Clear

When the USB cable is disconnected during communication, data which is receiving or transmitting may remain in the FIFO. Therefore the FIFO must be cleared immediately after connecting the USB cable again.

Note that the FIFO in which data is receiving from the host or transmitting to the host must not be cleared.

36.10.3 Overreading/Overwriting of Data Register

The following points should be noted when the data register of the USBF is read from or written to.

Receive Data Register: The receive data register must not read data which is more than valid receive data bytes. That is, data which is more than bytes indicated in the receive data size register must not be read. In case of the receive data register which has the dual FIFO buffer, the maximum number of data which can be read in a single time is maximum packet size. Write 1 to TRG after data in the current valid buffer is read. This writing switches the FIFO buffer. Then, the new number of bytes is reflected in the receive data size and the next data can be read.

Transmit Data Register: The transmit data register must not write data which is more than maximum packet size. In case of the transmit data register which has the dual FIFO buffer, the maximum number of data which can be written in a single time is maximum packet size. Write 1 to TRG/PKTE after data is written. This writing switches the FIFO buffer. Then, the next data can be written to another buffer. Therefore data must not be written in both buffers in a single time.

36.10.4 Assigning EP0 Interrupt Sources

The EP0 interrupt sources assigned to IFR0 (bits 0, 1, and 2) must be assigned to the same interrupt pins by ISR0. The other interrupt sources have no restrictions.

36.10.5 FIFO Clear when DMA Transfer is Set

When the DMA transfer is enabled in endpoint 1, the data register cannot be cleared. Cancel the DMA transfer before clearing the data register.

36.10.6 Note on Using TR Interrupt

The bulk-in transfer has a transfer request interrupt (TR interrupt). The following points should be noted when using a TR interrupt.

When the IN token is sent from the USB host and there is no data in the corresponding EP FIFO, the TR interrupt flag is set. However, the TR interrupt is generated continuously at the timing as shown in figure 36.20. In this case, note that erroneous operation should not occur.

Note: When the IN token is received and there is no data in the corresponding EP FIFO, an NAK is determined. However, the TR interrupt flag is set after an NAK handshake is transmitted. Therefore when the next IN token is received before TRG/PKTE is written, the TR interrupt flag is set again.

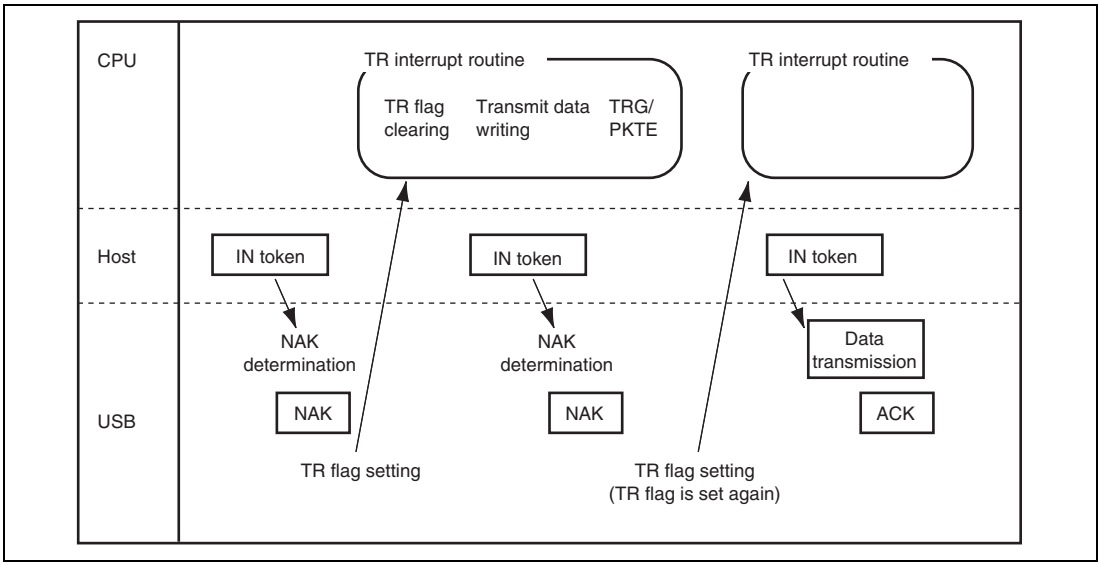


Figure 36.20 Set Timing of TR Interrupt Flag

Section 37 LCD Controller (LCDC)

A unified memory architecture is adopted for the LCD controller (LCDC) so that the image data for display is stored in system memory. The LCDC module reads data from system memory, uses the palette memory to determine the colors, then puts the display on the LCD panel. It is possible to connect the LCDC to the LCD module* other than microcomputer bus interface types and NTSC/PAL types and those that apply the LVDS interface.

Note: * LCD module can be connected to the LVDS interface by using the LSI with LVDS conversion LSI.

37.1 Features

The LCDC has the following features.

- Panel interface
  - Serial interface method
  - Supports data formats for STN/dual-STN/TFT panels (8/12/16/18-bit bus width)*[1]
- Supports 4/8/15/16-bpp (bits per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from 16×1 to 1024×1024*[2]
- 24-bit color palette memory (16 of the 24 bits are valid; R:5/G:6/B:5)
- STN/DSTN panels are prone to flicker and shadowing. The controller applies 65536-color control by 24-bit space-modulation FRC with 8-bit RGB values for reduced flicker.
- Dedicated display memory is unnecessary using part of the DDR_SDRAM (area 3) as the VRAM to store display data of the LCDC.
- The display is stable because of the large 2.4-kbyte line buffer
- Supports the inversion of the output signal to suit the LCD panel's signal polarity
- Supports the selection of data formats (the endian setting for bytes, backed pixel method) by register settings
- An interrupt can be generated at the user specified position (controlling the timing of VRAM update start prevents flicker)
- A hardware-rotation mode is included to support the use of landscape-format LCD panels as portrait-format LCD panels (the horizontal width of the panel before rotation must be within 320 pixels (see table 37.6.)

Notes: 1. When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower bit lines should be connected to GND or to the lowest bit from which data is output.
2. For details, see section 37.4.1, LCD Module Sizes which can be Displayed in this LCDC.

Figure 37.1 shows a block diagram of LCDC.

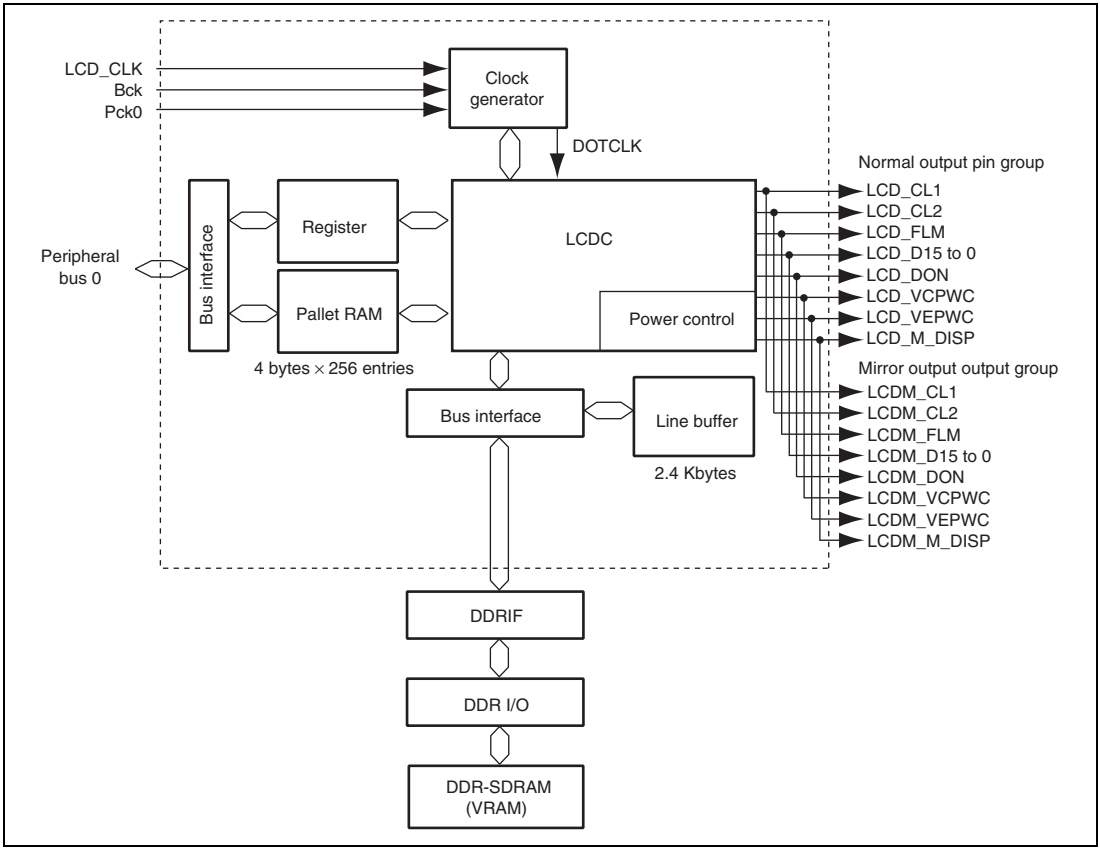


Figure 37.1 LCDC Block Diagram

37.2 Input/Output Pins

Table 37.1 summarizes the LCDC's pin configuration.

The LCDC output pins are divided in two groups: normal output group and mirror output group. The input/output operations of pins in two groups are always the same. The pin select register of the PFC is used to select the LCDC pins. As pins in two groups have different input/output timing, mixed use of pins in two groups is not allowed.

Table 37.1 Pin Configuration

Pin			
Normal Output	**Mirror Output**	**I/O**	**Function**
LCD_D15 to 0	LCDM_D15 to 0	Output	Data for LCD panel
LCD_DON	LCDM_DON	Output	Display-on signal (DON)
LCD_CL1	LCDM_CL1	Output	Shift-clock 1 (STN/DSTN)/horizontal sync signal (HSYNC) (TFT)
LCD_CL2	LCDM_CL2	Output	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLK) (TFT)
LCD_M_DISP	LCDM_M_DISP	Output	LCD current-alternating signal/DISP signal
LCD_FLM	LCDM_FLM	Output	First line marker/vertical sync signal (VSYNC) (TFT)
LCD_VCPWC	LCDM_VCPWC	Output	LCD-module power control (VCC)
LCD_VEPWC	LCDM_VEPWC	Output	LCD-module power control (VEE)
LCD_CLK*		Input	LCD clock-source input

Note: Check the LCD module specifications carefully in section 37.5, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

* Only this pin is available as the LCD_CLK pin in the LCDC module.

37.3 Register Configuration

Table 37.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
Palette data register 00 to FF	LDPR00 to LDPRFF	R/W	H'FFE8 0000 to H'FFE8 03FC	H'1FE8 0000 to H'1FE8 03FC	32
LCDC input clock register	LDICKR	R/W	H'FFE8 0400	H'1FE8 0400	16
LCDC module type register	LDMTR	R/W	H'FFE8 0402	H'1FE8 0402	16
LCDC data format register	LDDFR	R/W	H'FFE8 0404	H'1FE8 0404	16
LCDC scan mode register	LDSMR	R/W	H'FFE8 0406	H'1FE8 0406	16
LCDC data fetch start address register for upper display panel	LDSARU	R/W	H'FFE8 0408	H'1FE8 0408	32
LCDC data fetch start address register for lower display panel	LDSARL	R/W	H'FFE8 040C	H'1FE8 040C	32
LCDC fetch data line address offset register for display panel	LDLAOR	R/W	H'FFE8 0410	H'1FE8 0410	16
LCDC palette control register	LDPALCR	R/W	H'FFE8 0412	H'1FE8 0412	16
LCDC horizontal character number register	LDHCNR	R/W	H'FFE8 0414	H'1FE8 0414	16
LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'FFE8 0416	H'1FE8 0416	16
LCDC vertical displayed line number register	LDVDLNR	R/W	H'FFE8 0418	H'1FE8 0418	16
LCDC vertical total line number register	LDVTLNR	R/W	H'FFE8 041A	H'1FE8 041A	16
LCDC vertical synchronization signal register	LDVSYNR	R/W	H'FFE8 041C	H'1FE8 041C	16
LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'FFE8 041E	H'1FE8 041E	16
LCDC interrupt control register	LDINTR	R/W	H'FFE8 0420	H'1FE8 0420	16
LCDC power management mode register	LDPMMR	R/W	H'FFE8 0424	H'1FE8 0424	16
LCDC power supply sequence period register	LDPSPR	R/W	H'FFE8 0426	H'1FE8 0426	16
LCDC control register	LDCNTR	R/W	H'FFE8 0428	H'1FE8 0428	16

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
LCDC user specified interrupt control register	LDUINTR	R/W	H'FFE8 0434	H'1FE8 0434	16
LCDC user specified interrupt line number register	LDUINTLNR	R/W	H'FFE8 0436	H'1FE8 0436	16
LCDC memory access interval number register	LDLIRNR	R/W	H'FFE8 0440	H'1FE8 0440	16

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 37.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
Palette data register 00 to FF	LDPR00 to LDPRFF	Undefined	Undefined	Retained	Retained
LCDC input clock register	LDICKR	H'1101	H'1101	Retained	Retained
LCDC module type register	LDMTR	H'0109	H'0109	Retained	Retained
LCDC data format register	LDDFR	H'000C	H'000C	Retained	Retained
LCDC scan mode register	LDSMR	H'0000	H'0000	Retained	Retained
LCDC data fetch start address register for upper display panel	LDSARU	H'0C000000	H'0C000000	Retained	Retained
LCDC data fetch start address register for lower display panel	LDSARL	H'0C000000	H'0C000000	Retained	Retained
LCDC fetch data line address offset register for display panel	LDLAOR	H'0280	H'0280	Retained	Retained
LCDC palette control register	LDPALCR	H'0000	H'0000	Retained	Retained
LCDC horizontal character number register	LDHCNR	H'4F52	H'4F52	Retained	Retained
LCDC horizontal synchronization signal register	LDHSYNR	H'0050	H'0050	Retained	Retained
LCDC vertical displayed line number register	LDVDLNR	H'01DF	H'01DF	Retained	Retained
LCDC vertical total line number register	LDVTLNR	H'01DF	H'01DF	Retained	Retained

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
LCDC vertical synchronization signal register	LDVSYNR	H'01DF	H'01DF	Retained	Retained
LCDC AC modulation signal toggle line number register	LDACLNR	H'000C	H'000C	Retained	Retained
LCDC interrupt control register	LDINTR	H'0000	H'0000	Retained	Retained
LCDC power management mode register	LDPMMR	H'0010	H'0010	Retained	Retained
LCDC power supply sequence period register	LDPSPR	H'F60F	H'F60F	Retained	Retained
LCDC control register	LDCNTR	H'0000	H'0000	Retained	Retained
LCDC user specified interrupt control register	LDUINTR	H'0000	H'0000	Retained	Retained
LCDC user specified interrupt line number register	LDUINTLNR	H'004F	H'004F	Retained	Retained
LCDC memory access interval number register	LDLIRNR	H'0000	H'0000	Retained	Retained

37.3.1 LCDC Input Clock Register (LDICKR)

This LCDC can select bus clock, the peripheral clock, or the external clock as its operation clock source. The selected clock source can be divided using an internal divider into a clock of 1/1 to 1/32 and be used as the LCDC operating clock (DOTCLK). The clock output from the LCDC is used to generate the synchronous clock output (LCD_CL2) for the LCD panel from the operating clock selected in this register. For a TFT panel, LCD_CL2 = DOTCLK, and for an STN or DSTN panel, LCD_CL2 = a clock with a frequency of (DOTCLK/data bus width of output to LCD panel). The LDICKR must be set so that the clock input to the LCDC is 66 MHz or less regardless of the LCD_CL2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ICKSEL[1:0]		—	—	—	—	—	—	DCDR[5:0]					
Initial value:	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	ICKSEL[1:0]	00	R/W	Input Clock Select Set the clock source for DOTCLK. 00: Setting prohibited 01: Peripheral clock is selected 10: External clock is selected 11: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	DCDR[5:0]	000001	R/W	Clock Division Ratio Set the input clock division ratio. For details on the setting, refer to table 37.4.

Table 37.4 I/O Clock Frequency and Clock Division Ratio

DCDR[5:0]	Clock Division Ratio	I/O Clock Frequency (MHz)		
		50.000	60.000	66.000
000001	1/1	50.000	60.000	66.000
000010	1/2	25.000	30.000	33.000
000011	1/3	16.667	20.000	22.000
000100	1/4	12.500	15.000	16.500
000110	1/6	8.333	10.000	11.000
001000	1/8	6.250	7.500	8.250
001100	1/12	4.167	5.000	5.500
010000	1/16	3.125	3.750	4.125
011000	1/24	2.083	2.500	2.750
100000	1/32	1.563	1.875	2.063

Note: Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

37.3.2 LCDC Module Type Register (LDMTR)

LDMTR sets the control signals output from this LCDC and the polarity of the data signals, according to the polarity of the signals for the LCD module connected to the LCDC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	—	MCNT	CL1CNT	CL2CNT	—	—	MIFTYP[5:0]					
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	FLMPOL	0	R/W	FLM (Vertical Sync Signal) Polarity Select Selects the polarity of the LCD_FLM (vertical sync signal, first line marker) for the LCD module. 0: LCD_FLM pulse is high active 1: LCD_FLM pulse is low active
14	CL1POL	0	R/W	CL1 (Horizontal Sync Signal) Polarity Select Selects the polarity of the LCD_CL1 (horizontal sync signal) for the LCD module. 0: LCD_CL1 pulse is high active 1: LCD_CL1 pulse is low active
13	DISPPOL	0	R/W	DISP (Display Enable) Polarity Select Selects the polarity of the LCD_M_DISP (display enable) for the LCD module. 0: LCD_M_DISP is high active 1: LCD_M_DISP is low active

Bit	Bit Name	Initial Value	R/W	Description
12	DPOL	0	R/W	Display Data Polarity Select Selects the polarity of the LCD_D (display data) for the LCD module. This bit supports inversion of the LCD module. 0: LCD_D is high active, transparent-type LCD panel 1: LCD_D is low active, reflective-type LCD panel
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MCNT	0	R/W	M Signal Control Sets whether or not to output the LCD's current-alternating signal of the LCD module. 0: M (AC line modulation) signal is output 1: M signal is not output
9	CL1CNT	0	R/W	CL1 (Horizontal Sync Signal) Control Sets whether or not to enable CL1 output during the vertical retrace period. 0: CL1 is output during vertical retrace period 1: CL1 is not output during vertical retrace period
8	CL2CNT	1	R/W	CL2 (Dot Clock of LCD Module) Control Sets whether or not to enable CL2 output during the vertical and horizontal retrace period. 0: CL2 is output during vertical and horizontal retrace period 1: CL2 is not output during vertical and horizontal retrace period
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 to 0	MIFTYP [5:0]	001001	R/W	Module Interface Type Select Set the LCD panel type and data bus width to be output to the LCD panel. There are three LCD panel types: STN, DSTN, and TFT. There are four data bus widths for output to the LCD panel: 4, 8, 12, and 16 bits. When the required data bus width for a TFT panel is 16 bits or more, connect the LCDC and LCD panel according to the data bus size of the LCD panel. Unlike in a TFT panel, in an STN or DSTN panel, the data bus width setting does not have a 1:1 correspondence with the number of display colors and display resolution, e.g., an 8-bit data bus can be used for 16 bpp, and a 12-bit data bus can be used for 4 bpp. This is because the number of display colors in an STN or DSTN panel is determined by how data is placed on the bus, and not by the number of bits. For data specifications for an STN or DSTN panel, see the specifications of the LCD panel used. The output data bus width should be set according to the mechanical interface specifications of the LCD panel. If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC consisting of the 8-bit R, G, and B included in the LCDC, regardless of the color and gradation settings. Accordingly, the color and gradation specified by DSPCOLOR is selected from 16 million colors in an STN or DSTN panel. If a palette is used, the color specified in the palette is displayed. 000000: STN monochrome 4-bit data bus module 000001: STN monochrome 8-bit data bus module 001000: STN color 4-bit data bus module 001001: STN color 8-bit data bus module 001010: STN color 12-bit data bus module 001011: STN color 16-bit data bus module 010001: DSTN monochrome 8-bit data bus module 010011: DSTN monochrome 16-bit data bus module 011001: DSTN color 8-bit data bus module 011010: DSTN color 12-bit data bus module 011011: DSTN color 16-bit data bus module 101011: TFT color 16-bit data bus module Settings other than above: Setting prohibited

37.3.3 LCDC Data Format Register (LDDFR)

LDDFR sets the bit alignment for pixel data in one byte and selects the data type and number of colors used for display so as to match the display driver software specifications.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PABD	—	DSPCOLOR[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PABD	0	R/W	Byte Data Pixel Alignment Sets the pixel data alignment type in one byte of data. The contents of aligned data per pixel are the same regardless of this bit's setting. For example, data H'05 should be expressed as B'0101 which is the normal style handled by a MOV instruction of the this CPU, and should not be selected between B'0101 and B'1010. 0: Big endian for byte data 1: Little endian for byte data
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	DSPCOLOR [6:0]	0001100	R/W	Display Color Select Set the number of display colors for the display (0 is written to upper bits of 4 to 6 bpp). For display colors to which the description (via palette) is added below, the color set by the color palette is actually selected by the display data and displayed. The number of colors that can be selected in rotation mode is restricted by the display resolution. For details, see table 37.5. 0000000: Monochrome, 2 grayscales, 1 bpp (via palette) 0000001: Monochrome, 4 grayscales, 2 bpp (via palette) 0000010: Monochrome, 16 grayscales, 4 bpp (via palette) 0000100: Monochrome, 64 grayscales, 6 bpp (via palette) 0001010: Color, 16 colors, 4 bpp (via palette) 0001100: Color, 256 colors, 8 bpp (via palette) 0011101: Color, 32K colors (RGB: 555), 15 bpp 0101101: Color, 64K colors (RGB: 565), 16 bpp Settings other than above: Setting prohibited

37.3.4 LCDC Scan Mode Register (LDSMR)

LDSMR selects whether or not to enable the hardware rotation function that is used to rotate the LCD panel, and sets the burst length for the VRAM (synchronous DRAM in area 3) used for display.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ROT	—	—	—	AU[1:0]		—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	ROT	0	R/W	Rotation Module Select Selects whether or not to rotate the display by hardware. Note that the following restrictions are applied to rotation. • An STN or TFT panel must be used. A DSTN panel is not allowed. • The maximum horizontal (internal scan direction of the LCD panel) width of the LCD panel is 320. • Set a binary exponential that exceeds the display size in LDLAOR. (For example, 256 must be selected when a 320 × 240 panel is rotated to be used as a 240 × 320 panel and the horizontal width of the image is 240 bytes.) 0: Not rotated 1: Rotated 90 degrees rightwards (left side of image is displayed on the upper side of the LCD module)
12 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, 8	AU[1:0]	00	R/W	Access Unit Select Select access unit of VRAM. This bit is enabled when ROT = 1 (rotate the display). When ROT = 0, 16-burst memory read operation is carried out whatever the AU setting is. 00: 4-burst 01: 8-burst 10: 16-burst 11: 32-burst Notes: 1. Above burst lengths are used for 32-bit bus. For 16-bit bus, the burst lengths are twice the lengths of 32-bit bus. 2. When displaying a rotated image, the burst length is limited depending on the number of column address bits and bus width of connected SDRAM. For details, see table 37.5.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

37.3.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU)

LDSARU sets the start address from which data is fetched by the LCDC for display of the LCDC panel. When a DSTN panel is used, this register specifies the fetch start address for the upper side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SAU[25:16]									
Initial value:	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAU[15:4]												—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAU[25:4]	All 0	R/W	Start Address for Upper Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. The minimum alignment unit of LDSARU is 512 bytes when the hardware rotation function is not used. Write 0 to the lower nine bits. When using the hardware rotation function, set the LDSARU value so that the upper-left address of the image is aligned with the 512-byte boundary.

2. When the hardware rotation function is used (ROT = 1), set the upper-left address of the image which can be calculated from the display image size in this register. The equation below shows how to calculate the LDSARU value when the image size is 240 × 340 and LDLAOR = 256. The LDSARU value is obtained not from the panel size but from the memory size of the image to be displayed. Note that LDLAOR must be a binary exponential at least as large as the horizontal width of the image. Calculate backwards using the LDSARU value (LDSARU − 256 (LDLAOR value) × (320 − 1)) to ensure that the upper-left address of the image is aligned with the 512-byte boundary.
LDSARU = (upper-left address of image) + 256 (LDLAOR value) × 319 (line)

37.3.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL)

When a DSTN panel is used, LDSARL specifies the fetch start address for the lower side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SAL[25:16]									
Initial value:	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL[15:4]												—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAL[25:4]	All 0	R/W	Start Address for Lower Panel Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3. STN and TFT: Cannot be used DSTN: Start address for fetching display data corresponding to the lower panel
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

37.3.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR)

LDLAOR sets the address width of the Y-coordinates increment used for LCDC to read the image recognized by the graphics driver. This register specifies how many bytes the address from which data is to be read should be moved when the Y coordinates have been incremented by 1. This register does not have to be equal to the horizontal width of the LCD panel. When the memory address of a point (X, Y) in the two-dimensional image is calculated by Ax + By+ C, this register becomes equal to B in this equation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAO[15:0]															
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	LAO [15:0]	H'0280	R/W	Line Address Offset The minimum alignment unit of LDLAOR is 16 bytes. Because the LCDC handles these values as 16-byte data, the values written to the lower four bits of the register are always treated as 0. The lower four bits of the register are always read as 0. The initial values (× resolution = 640) will continuously and accurately place the VGA (640 × 480 dots) display data without skipping an address between lines. For details, see table 37.5. A binary exponential at least as large as the horizontal width of the image is recommended for the LDLAOR value, taking into consideration the software operation speed. When the hardware rotation function is used, the LDLAOR value should be a binary exponential (in this example, 256) at least as large as the horizontal width of the image (after rotation, it becomes 240 in a 240 × 320 panel) instead of the horizontal width of the LCD panel (320 in a 320 × 240 panel).

37.3.8 LCDC Palette Control Register (LDPALCR)

LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, color-palette setting mode should be selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PALS	—	—	—	PALEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Normal display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

37.3.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF)

LDPR registers are for accessing palette data directly allocated (4 bytes x 256 addresses) to the memory space. To access the palette memory, access the corresponding register among this register group (LDPR00 to LDPRFF). Each palette register is a 32-bit register including three 8-bit areas for R, G, and B. For details on the color palette specifications, see section 37.4.3, Color Palette Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PALDnn[23:16]							
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PALDnn[15:0]															
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	PALDnn[23:0]	—	R/W	Palette Data Bits 18 to 16, 9, 8, and 2 to 0 are reserved within each RGB palette and cannot be set. However, these bits can be extended according to the upper bits.

Note: nn = H'00 to H'FF

37.3.10 LCDC Horizontal Character Number Register (LDHCNR)

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the entire scan width including the horizontal retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDCN[7:0]								HTCN[7:0]							
Initial value:	0	1	0	0	1	1	1	1	0	1	0	1	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	HDCN [7:0]	01001111	R/W	Horizontal Display Character Number Set the number of horizontal display characters (unit: character = 8 dots). Specify to the value of (the number of display characters) -1. Example: For a LCD module with a width of 640 pixels. HDCN = (640/8) -1 = 79 = H'4F
7 to 0	HTCN [7:0]	01010010	R/W	Horizontal Total Character Number Set the number of total horizontal characters (unit: character = 8 dots). Specify to the value of (the number of total characters) -1. However, the minimum horizontal retrace period is three characters (24 dots). Example: For a LCD module with a width of 640 pixels. HTCN = [(640/8)-1] +3 = 82 = H'52 In this case, the number of total horizontal dots is 664 dots and the horizontal retrace period is 24 dots.

Notes: 1. The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN. Also, the total number of characters of HTCN must be an even number. (The set value will be an odd number, as it is one less than the actual number.)

2. Set HDCN according to the display resolution as follows:
1 bpp: (multiplex of 16) – 1 [1 line is multiplex of 128 pixel]
2 bpp: (multiplex of 8) – 1 [1 line is multiplex of 64 pixel]
4 bpp: (multiplex of 4) – 1 [1 line is multiplex of 32 pixel]
6 bpp/8 bpp: (multiplex of 2) – 1 [1 line is multiplex of 16 pixel]

37.3.11 LCDC Horizontal Sync Signal Register (LDHSYNR)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals for the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSYNW[3:0]				—	—	—	—	HSYNP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	HSYNW [3:0]	0000	R/W	Horizontal Sync Signal Width Set the width of the horizontal sync signals (CL1 and Hsync) (unit: character = 8 dots). Specify to the value of (the number of horizontal sync signal width) -1. Example: For a horizontal sync signal width of 8 dots. HSYNW = (8 dots/8 dots/character) -1 = 0 = H'0
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	HSYNP [7:0]	01010000	R/W	Horizontal Sync Signal Output Position Set the output position of the horizontal sync signals (unit: character = 8 dots). Specify to the value of (the number of horizontal sync signal output position) -1. Example: For a LCD module with a width of 640 pixels. HSYNP = [(640/8) +1] -1 = 80 = H'50 In this case, the horizontal sync signal is active from the 648th through the 655th dot.

Note: The following conditions must be satisfied:
HTCN ≥ HSYNP+HSYNW+1
HSYNP ≥ HDCN+1

37.3.12 LCDC Vertical Display Line Number Register (LDVDLNR)

LDVDLNR specifies the LCD module's vertical size (for both scan direction and vertical direction). For a DSTN panel, specify an even number at least as large as the LCD panel's vertical size regardless of the size of the upper and lower panels, e.g. 480 for a 640 x 480 panel.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VDLN[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	VDLN[10:0]	00111011111	R/W	Vertical Display Line Number Set the number of vertical display lines (unit: line). Specify to the value of (the number of display line) -1. Example: For an 480-line LCD module VDLN = 480-1 = 479 = H'1DF

37.3.13 LCDC Vertical Total Line Number Register (LDVTLNR)

LDVTLNR specifies the LCD panel's entire vertical size including the vertical retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VTLN[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	VTLN[10:0]	00111011111	R/W	Vertical Total Line Number Set the total number of vertical display lines (unit: line). Specify to the value of (the number of total line) -1. The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN>=VDLN, VTLN>=1. Example: For an 480-line LCD module and a vertical period of 0 lines. VTLN = (480+0) –1 = 479 = H'1DF

37.3.14 LCDC Vertical Sync Signal Register (LDVSYNR)

LDVSYNR specifies the vertical (scan direction and vertical direction) sync signal timing of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSYNW[3:0]				—	VSYNP[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	VSYNW[3:0]	0000	R/W	Vertical Sync Signal Width Set the width of the vertical sync signals (FLM and Vsync) (unit: line). Specify to the value of (the vertical sync signal width) -1. Example: For a vertical sync signal width of 1 line. VSYNW = (1-1) = 0 = H'0
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 0	VSYNP[10:0]	00111011111	R/W	Vertical Sync Signal Output Position Set the output position of the vertical sync signals (FLM and Vsync) (unit: line). Specify to the value of (the number of vertical sync signal output position) -2. DSTN should be set to an odd number value. It is handled as (setting value+1)/2. Example: For an 480-line LCD module and a vertical retrace period of 0 lines (in other words, VTLN=479 and the vertical sync signal is active for the first line): • Single display VSYNP = [(1-1)+VTLN]mod(VTLN+1) = [(1-1)+479]mod(479+1) = 479mod480 = 479 =H'1DF • Dual displays VSYNP = [(1-1)×2+VTLN]mod(VTLN+1) = [(1-1)×2+479]mod(479+1) = 479mod480 = 479 =H'1DF

37.3.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR)

LDACLNR specifies the timing to toggle the AC modulation signal (LCD current-alternating signal) of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ACLN[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	ACLN[4:0]	01100	R/W	AC Line Number Set the number of lines where the LCD current-alternating signal of the LCD module is toggled (unit: line). Specify to the value of (the number of toggle line) - 1. Example: For toggling every 13 lines. ACLN = 13-1 = 12= H'0C

Note: When the total line number of the LCD panel is even, set an even number so that toggling is performed at an odd line.

37.3.16 LCDC Interrupt Control Register (LDINTR)

LDINTR specifies where to control the Vsync interrupt of the LCD module. See also section 37.3.20, LCDC User Specified Interrupt Control Register (LDUINTR) and section 37.3.21, LCDC User Specified Interrupt Line Number Register (LDUINTLNR) for interrupts. Note that operations by this register setting and LCDC user specified interrupt control register (LDUINTR) setting are independent.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINTEN	FINTEN	VSINTEN	VEINTEN	MINTS	FINTS	VSINTS	VEINTS	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	MINTEN	0	R/W	Memory Access Interrupt Enable Enables or disables an interrupt generation at the start point of each vertical retrace line period for VRAM access by LCDC. 0: Disables an interrupt generation at the start point of each vertical retrace line period for VRAM access 1: Enables an interrupt generation at the start point of each vertical retrace line period for VRAM access
14	FINTEN	0	R/W	Frame End Interrupt Enable Enables or disables the generation of an interrupt after the last pixel of a frame is output to LDC panel. 0: Disables an interrupt generation when the last pixel of the frame is output 1: Enables an interrupt generation when the last pixel of the frame is output
13	VSINTEN	0	R/W	Vsync Starting Point Interrupt Enable Enables or disables the generation of an interrupt at the start point of LCDC's Vsync. 0: Interrupt at the start point of the Vsync is disabled 1: Interrupt at the start point of the Vsync is enabled

Bit	Bit Name	Initial Value	R/W	Description
12	VEINTEN	0	R/W	Vsync Ending Point Interrupt Enable Enables or disables the generation of an interrupt at the end point of LCDC's Vsync. 0: Interrupt at the end point of the Vsync signal is disabled 1: Interrupt at the end point of the Vsync signal is enabled
11	MINTS	0	R/W	Memory Access Interrupt State Indicates the memory access interrupt handling state. This bit indicates 1 when the LCDC memory access interrupt is generated (set state). During the memory access interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a memory access interrupt or has been informed that the generated memory access interrupt has completed 1: LCDC has generated a memory access end interrupt and not yet been informed that the generated memory access interrupt has completed
10	FINTS	0	R/W	Flame End Interrupt State Indicates the flame end interrupt handling state. This bit indicates 1 at the time when the LCDC flame end interrupt is generated (set state). During the flame end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a flame end interrupt or has been informed that the generated flame end interrupt has completed 1: LCDC has generated a flame end interrupt and not yet been informed that the generated flame end interrupt has completed

Bit	Bit Name	Initial Value	R/W	Description
9	VSINTS	0	R/W	Vsync Start Interrupt State Indicates the LCDC's Vsync start interrupt handling state. This bit is set to 1 at the time a Vsync start interrupt is generated. During the Vsync start interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a Vsync start interrupt or has been informed that the generated Vsync start interrupt has completed 1: LCDC has generated a Vsync start interrupt and has not yet been informed that the generated Vsync start interrupt has completed
8	VEINTS	0	R/W	Vsync End Interrupt State Indicates the LCDC's Vsync end interrupt handling state. This bit is set to 1 at the time a Vsync end interrupt is generated. During the Vsync end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a Vsync end interrupt or has been informed that the generated Vsync end interrupt has completed 1: LCDC has generated a Vsync end interrupt and has not yet been informed that the generated Vsync interrupt has completed
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

37.3.17 LCDC Power Management Mode Register (LDPMMR)

LDPMMR controls the power supply circuit that provides power to the LCD module. The usage of two types of power-supply control pins, LCD_VCPWC and LCD_VEPWC, and turning on or off the power supply function are selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC[3:0]				OFFD[3:0]				—	VCPE	VEPE	DONE	—	—	LPS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	ONC[3:0]	0000	R/W	LCDC Power-On Sequence Period Set the period from LCD_VEPWC assertion to LCD_DON assertion in the power-on sequence of the LCD module in frame units. Specify to the value of (the period) -1. This period is the (c) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module. For details on setting this register, see table 37.6, Available Power-Supply Control-Sequence Periods at Typical Frame Rates. (The setting method is common for ONA, ONB, OFFD, OFFE, and OFFF.)
11 to 8	OFFD[3:0]	0000	R/W	LCDC Power-Off Sequence Period Set the period from LCD_DON negation to LCD_VEPWC negation in the power-off sequence of the LCD module in frame units. Specify to the value of (the period) -1. This period is the (d) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module.
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	VCPE	0	R/W	LCD_VCPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_VCPWC pin. 0: Disabled: LCD_VCPWC pin is masked and fixed low 1: Enabled: LCD_VCPWC pin output is asserted and negated according to the power-on or power-off sequence
5	VEPE	0	R/W	LCD_VEPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_VEPWC pin. 0: Disabled: LCD_VEPWC pin is masked and fixed low 1: Enabled: LCD_VEPWC pin output is asserted and negated according to the power-on or power-off sequence
4	DONE	1	R/W	LCD_DON Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_DON pin. 0: Disabled: LCD_DON pin is masked and fixed low 1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	LPS[1:0]	00	R	LCD Module Power-Supply Input State Indicates the power-supply input state of the LCD module when using the power-supply control function. 0: LCD module power off 1: LCD module power on

37.3.18 LCDC Power-Supply Sequence Period Register (LDPSPR)

LDPSPR controls the power supply circuit that provides power to the LCD module. The timing to start outputting the timing signals to the LCD_VEPWC and LCD_VCPWC pins is specified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONA[3:0]				ONB[3:0]				OFFE[3:0]				OFFF[3:0]			
Initial value:	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	ONA[3:0]	1111	R/W	LCDC Power-On Sequence Period Set the period from LCD_VCPWC assertion to starting output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-on sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (a) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module.
11 to 8	ONB[3:0]	0110	R/W	LCDC Power-On Sequence Period Set the period from starting output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to the LCD_VEPWC assertion in the power-on sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (b) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	OFFE[3:0]	0000	R/W	LCDC Power-Off Sequence Period Set the period from LCD_VEPWC negation to stopping output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-off sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (e) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module.
3 to 0	OFFF[3:0]	1111	R/W	LCDC Power-Off Sequence Period Set the period from stopping output of the display data (LCD_D) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to LCD_VCPWC negation to in the power-off sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (f) period in figures 37.4 to 37.7, Power-Supply Control Sequence and States of the LCD Module.

37.3.19 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LCDC.

When 1s are written to the DON2 bit and the DON bit, the LCDC starts display. Turn on the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'00 to B'11. Do not make any action to the DON bit until the sequence ends.

When 0 is written to the DON bit, the LCDC stops display. Turn off the LCD module following the sequence set in the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'11 to B'00. Do not make any action to the DON bit until the sequence ends.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DON2	—	—	—	DON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

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	DON2	0	R/W	Display On 2 Specifies the start of the LCDC display operation. 0: LCDC is being operated or stopped 1: LCDC starts operation When this bit is read, always read as 0. Write 1 to this bit only when starting display. If a value other than 0 is written when starting display, the operation is not guaranteed. When 1 is written to, it resumes automatically to 0. Accordingly, this bit does not need to be cleared by writing 0.
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	DON	0	R/W	Display On Specifies the start and stop of the LCDC display operation. The control sequence state can be checked by referencing the LPS[1:0] of LDPMMR. 0: Display-off mode: LCDC is stopped 1: Display-on mode: LCDC operates

Notes: 1. Write H'0011 to LDCNTR when starting display and H'0000 when completing display. Data other than H'0011 and H'0000 must not be written to.
2. Setting bit DON2 to 1 makes the contents of the palette RAM undefined. Before writing to the palette RAM, set bit DON2 to 1.

37.3.20 LCDC User Specified Interrupt Control Register (LDUINTR)

LDUINTR sets whether the user specified interrupt is generated, and indicates its processing state. This interrupt is generated at the time when image data which is set by the line number register (LDUINTLNR) in LCDC is read from VRAM.

This LCDC issues the interrupts (LCDCI): user specified interrupt by this register, memory access interrupt by the LCDC interrupt control register (LDINTR), and OR of Vsync interrupt output. This register and LCDC interrupt control register (LDINTR) settings affect the interrupt operation independently.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UINTEN	—	—	—	—	—	—	—	UINTS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

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	UINTEN	0	R/W	User Specified Interrupt Enable Sets whether generate an LCDC user specified interrupt. 0: LCDC user specified interrupt is not generated 1: LCDC user specified interrupt is generated

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	UINTS	0	R/W	User Specified Interrupt State This bit is set to 1 at the time an LCDC user specified interrupt is generated (set state). During the user specified interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a user specified interrupt or has been informed that the generated user specified interrupt has completed 1: LCDC has generated a user specified interrupt and has not yet been notified that the generated user specified interrupt has completed

Note: Interrupt processing flow:

1. Interrupt signal is input
2. LDINTR is read
3. If MINTS, FINTS, VSINTS, or VEINTS is 1, a generated interrupt is memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt. Processing for each interrupt is performed.
4. If MINTS, FINTS, VSINTS, or VEINTS is 0, a generated interrupt is not memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt.
5. UINTS is read.
6. If UINTS is 1, a generated interrupt is a user specified interrupt. Process for user specified interrupt is carried out.
7. If UINTS is 0, a generated interrupt is not a user specified interrupt. Other processing is performed.

37.3.21 LCDC User Specified Interrupt Line Number Register (LDUINTLNR)

LDUINTLNR sets the point where the user specified interrupt is generated. Setting is done in horizontal line units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	UINTLN[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 0	UINTLN [10:0]	00001001111	R/W	User Specified Interrupt Generation Line Number Specifies the line in which the user specified interrupt is generated (line units). Set (the number of lines in which interrupts are generated) −1 Example: Generate the user specified interrupt in the 80th line. UINTLN = 160/2 − 1 = 79 = H'04F

Notes: 1. When using the LCD module with STN/TFT display, the setting value of this register should be equal to lower than the vertical display line number (VDLN) in LDVDLNR.
2. When using the LCD module with DSTN display, the setting value of this register should be equal to or lower than half the vertical display line number (VDLN) in LDVDLNR. The user specified interrupt is generated at the point when the LCDC read the specified piece of image data in lower display from VRAM.

37.3.22 LCDC Memory Access Interval Number Register (LDLIRNR)

LDLIRNR controls the bus cycle interval when the LCDC reads VRAM. When LDLIRNR is set to a value other than H'00, the LCDC does not access VRAM until clock count of the DDR-SDRAM matches the value set in LDLIRNR. When LDLIRNR is set to H'00 (initial value), the LCDC accesses VRAM one clock after the LCDC accessed VRAM.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	LIRN[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	LIRN[7:0]	All 0	R/W	VRAM Read Bus Cycle Interval Specifies the number of the DDR-SDRAM clock cycles which can be performed during burst cycles to read VRAM by LCDC. H'00: one clock cycle H'01: one clock cycle H'02: two clock cycles : H'FE: 254 clock cycles H'FF: 255 clock cycles

37.4 Operation

37.4.1 LCD Module Sizes which can be Displayed in this LCDC

This LCDC is capable of controlling displays with up to 1024 × 1024 dots and 16 bpp (bits per pixel). The image data for display is stored in VRAM, which is shared with the CPU. This LCDC should read the data from VRAM before display.

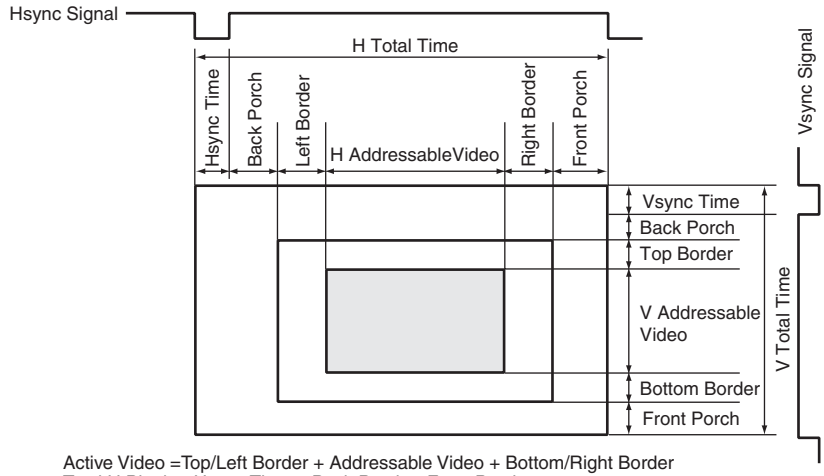
This LSI has a maximum 32-burst memory read operation and a 2.4-Kbyte line buffer, so although a complete breakdown of the display is unlikely, there may be some problems with the display depending on the combination. A recommended size at the frame rate of 60 Hz is 320 × 240 dots in 16 bpp or 640 × 480 dots in 8 bpp.

As a rough standard, the bus occupation ratio shown below should not exceed 40%.

$$\text{Bus occupation ratio (\%)} = \frac{\text{Overhead coefficient x Total number of display pixels ((HDCN + 1) x 8 x (VDLN + 1)) x Frame rate (Hz) x Number of colors (bpp)}}{\text{CLKOUT (Hz) x Bus width (bit)}} \times 100$$

The overhead coefficient becomes 1.375 when the CL2.5 DDR-SDRAM is connected to a 32-bit data bus.

Figure 37.2 shows the valid display and the retrace period.



Active Video =Top/Left Border + Addressable Video + Bottom/Right Border
Total H Blank = Hsync Time + Back Porch + Front Porch
Total V Blank = Vsync Time + Back Porch + Front Porch

HTCN = H Total Time
HDCN = H Addressable Video
HSYNP = H Addressable Video + Right Border + Front Porch
HSYNW = Hsync Time

VTLN = V Total Time
CDLN = V Addressable Video
VSYNP = V Addressable Video + Bottom Border + Front porch
VSYNW = Vsync Time

Figure 37.2 Valid Display and the Retrace Period

37.4.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM)

This LCDC is capable of displaying a landscape-format image on a LCD module by rotating a portrait format image for display by 90 degrees. Only the numbers of colors for each resolution are supported as shown in table 37.5. The size of the SDRAM (the number of column address bits) and its burst length are limited to read the SDRAM continuously.

The number of colors for display, SDRAM column addresses, and LCDC burst length are shown table 37.5.

A monochromatic LCD module is necessary for the display of images in the above monochromatic formats. A color LCD module is necessary for the display of images in the above color formats.

Table 37.5 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	8 bpp	9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240		16 bpp	9 bits	4 bursts
				10 bits	Not more than 8 bursts
234 × 320	320 × 234	Monochrome	6 bpp	9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	16 bpp	9 bits	4 bursts
				10 bits	Not more than 8 bursts
80 × 160	160 × 80	Monochrome	2 bpp	9 bits	—
				10 bits	—
			4 bpp (packed)	9 bits	—
				10 bits	—
			4 bpp (unpacked)	9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	9 bits	—
				10 bits	—
			4 bpp (unpacked)	9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	9 bits	Not more than 16 bursts
				10 bits	—
			16 bpp	9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	9 bits	—
				10 bits	—
			2 bpp	9 bits	—
				10 bits	—
			4 bpp (packed)	9 bits	—
				10 bits	—
			4 bpp (unpacked)	9 bits	—
				10 bits	—
			6 bpp	9 bits	—
				10 bits	—
		Color	4 bpp (packed)	9 bits	—
				10 bits	—
			4 bpp (unpacked)	9 bits	—
				10 bits	—
			8 bpp	9 bits	—
				10 bits	—

Note: * Specify the data of the number of line specified as burst length can be stored in address of SDRAM same as that of ROM.

37.4.3 Color Palette Specification

(1) Color Palette Register

This LCDC has a color palette which outputs 24 bits of data per entry and is able to simultaneously hold 256 entries. The color palette thus allows the simultaneous display of 256 colors chosen from among 16-M colors.

The procedure below may be used to set up color palettes at any time.

1. The PALEN bit in the LDPALCR is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode after three cycles of peripheral clock.
3. Access LDPALCR and confirm that the PALS bit is 1.
4. Access LDPR00 to LDPRFF and write the required values to the PALD00 to PALDFF bits.
5. Access LDPALCR and clear the PALEN bit to 0; return to normal display mode after a cycle of peripheral clock.

A 0 is output on the LCDC display data output (LCD_D) while the PALS bit in LDPALCR is set to 1.

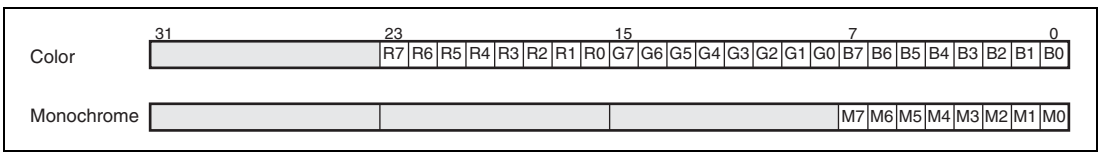


Figure 37.3 Color-Palette Data Format

PALDnn color and gradation data should be set as above.

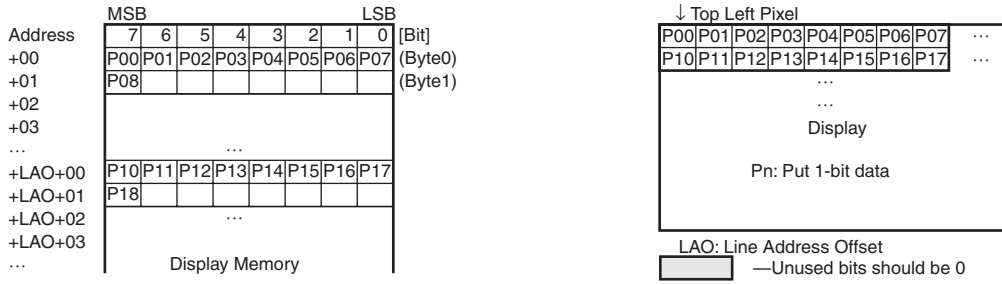
For a color display, PALDnn[23:16], PALDnn[15:8], and PALDnn[7:0] respectively hold the R, G, and B data. Although the bits PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] exist, no memory is associated with these bits. PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5. A 24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should, however, be written to the palette-data registers. When the values for PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are not 0, 1 or 0 should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. When the values of PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are 0, 0s should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. Then 24 bits are extended.

Grayscale data for a monochromatic display should be set in PALDnn[7:3]. PALDnn[23:8] are all "don't care". When the value in PALDnn[7:3] is not 0, 1s should be written to PALDnn[2:0].

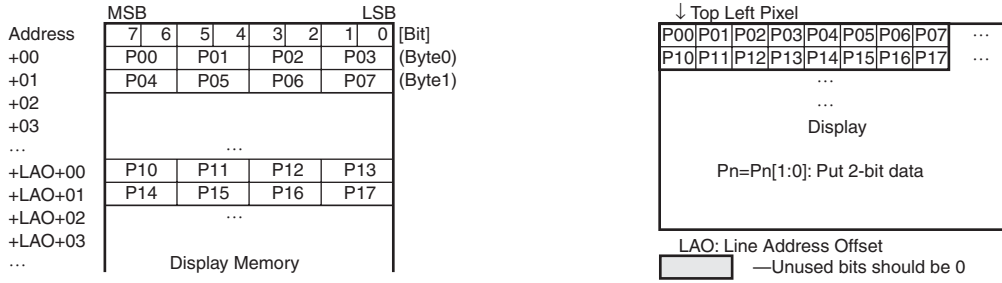
When the value in PALDnn[7:3] is 0, 0s should be written to PALDnn[2:0]. Then 8 bits are extended.

37.4.4 Data Format

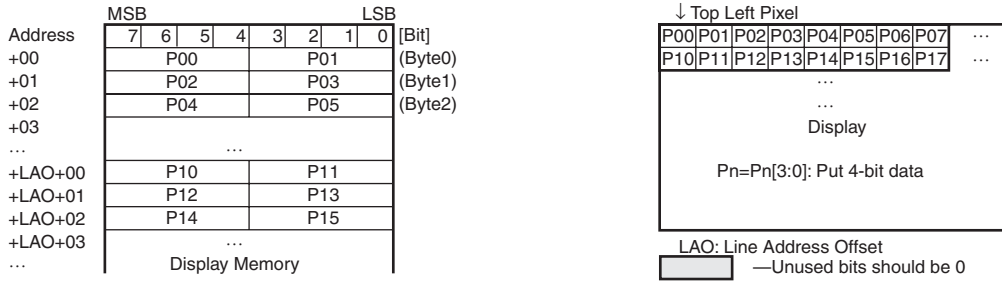
1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



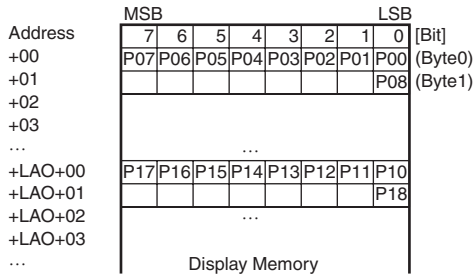
2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

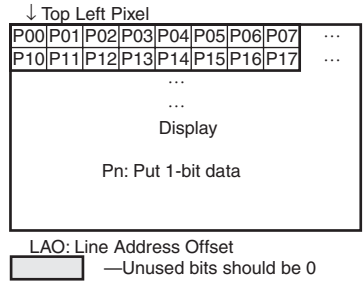


3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

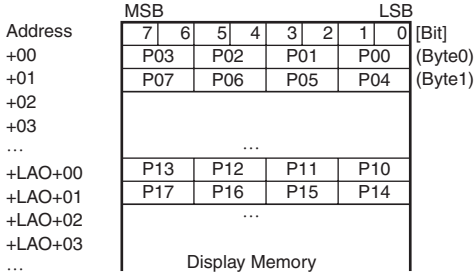


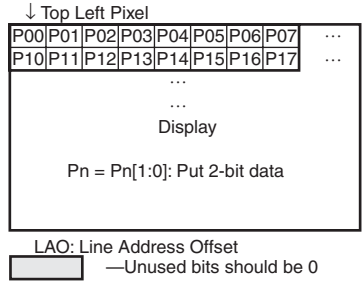
4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)



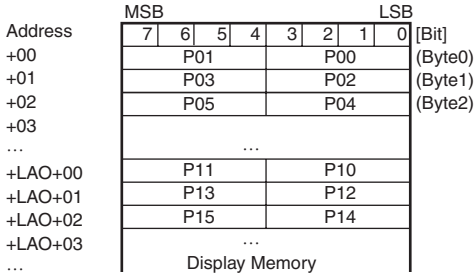


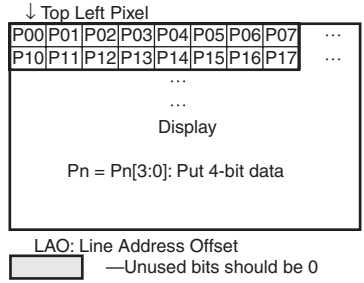
5. Packed 2bpp (Pixel Alignment in Byte is Little Endian)



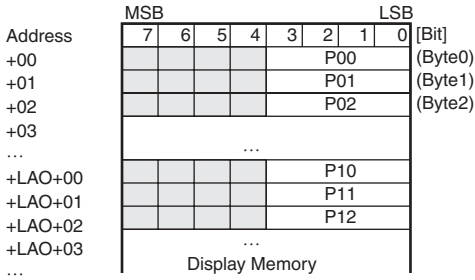


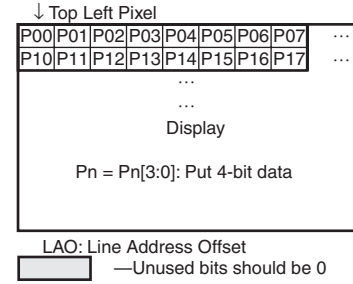
6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)



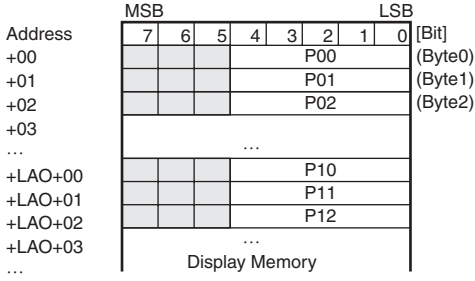


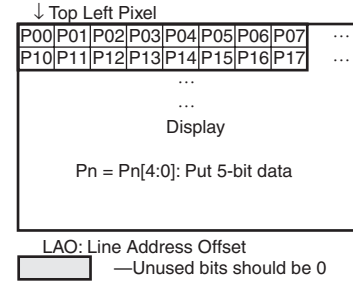
7. Unpacked 4bpp [Windows CE Recommended Format]



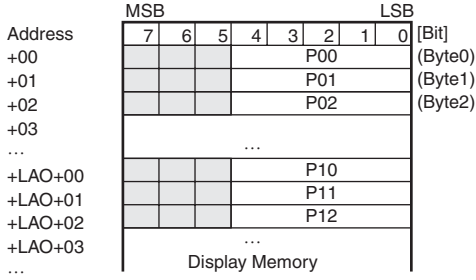


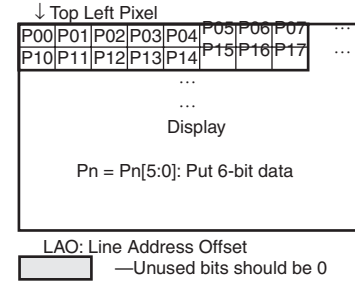
8. Unpacked 5bpp [Windows CE Recommended Format]



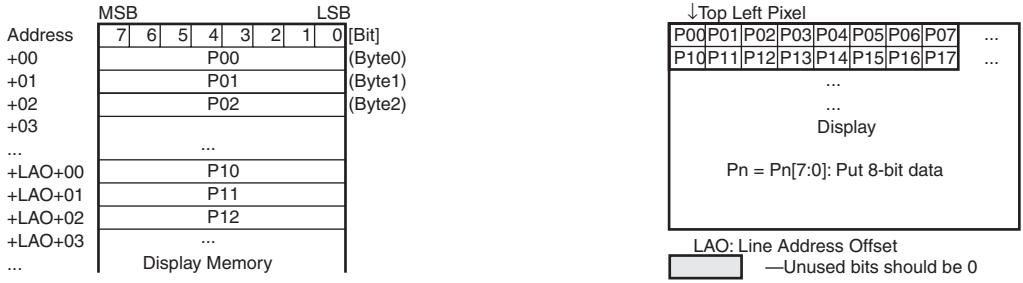


9. Unpacked 6bpp [Windows CE Recommended Format]

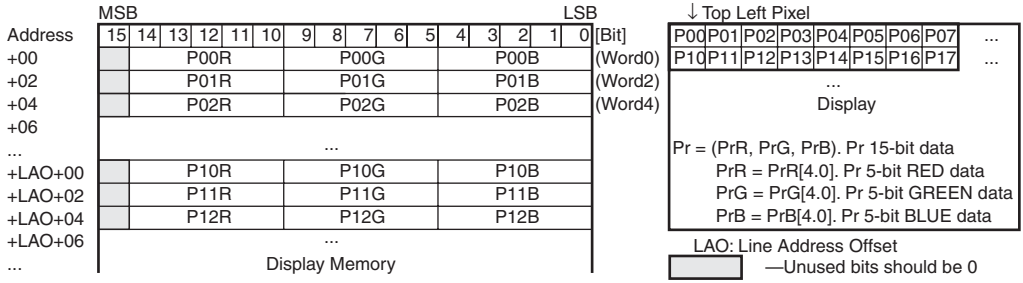




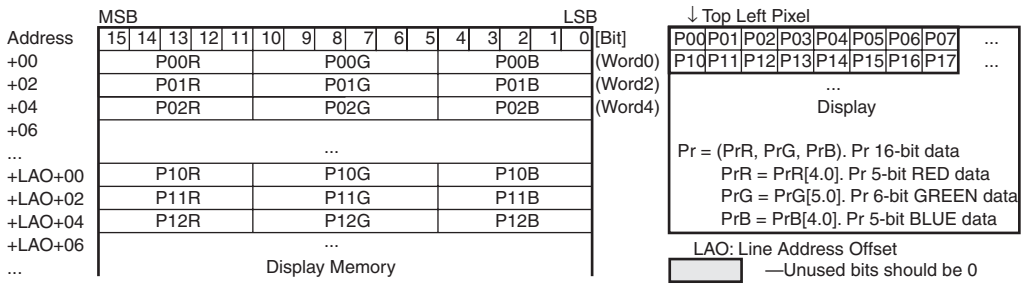
10. Packed 8bpp [Windows CE Recommended Format]



11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]



12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]



37.4.5 Setting the Display Resolution

The display resolution is set up in LDHCNR, LDHSYNR, LDVDLNR, LDVTLNR, and LDVSYNR. The LCD current-alternating period for an STN or DSTN display is set by using the LDACLNR. The initial values in these registers are typical settings for VGA (640 × 480 dots) on an STN or DSTN display.

The clock to be used is set with the LDICKR. The LCD module frame rate is determined by the display interval + retrace line interval (non-display interval) for one screen set in a size related register and the frequency of the clock used.

This LCDC has a Vsync interrupt function so that it is possible to issue an interrupt at the beginning of each vertical retrace line period (to be exact, at the beginning of the line after the last line of the display). This function is set up by using the LDINTR.

37.4.6 Power-Supply Control Sequence

An LCD module normally requires a specific sequence for processing to do with the cutoff of the input power supply. Settings in LDPMMR, LDPSPR, and LDCNTR, in conjunction with the LCD power-supply control pins (LCD_VCPWC, LCD_VEPWC, and LCD_DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figures 37.4 to 37.7 are timing charts that show outlines of power-supply control sequences and table 37.6 is a summary of available power-supply control sequence periods. Figures 37.4 to 37.7 show operations of the normal output pins (LCD_***). The mirror pins (LCDM_***) have the same timing as the normal output pins.

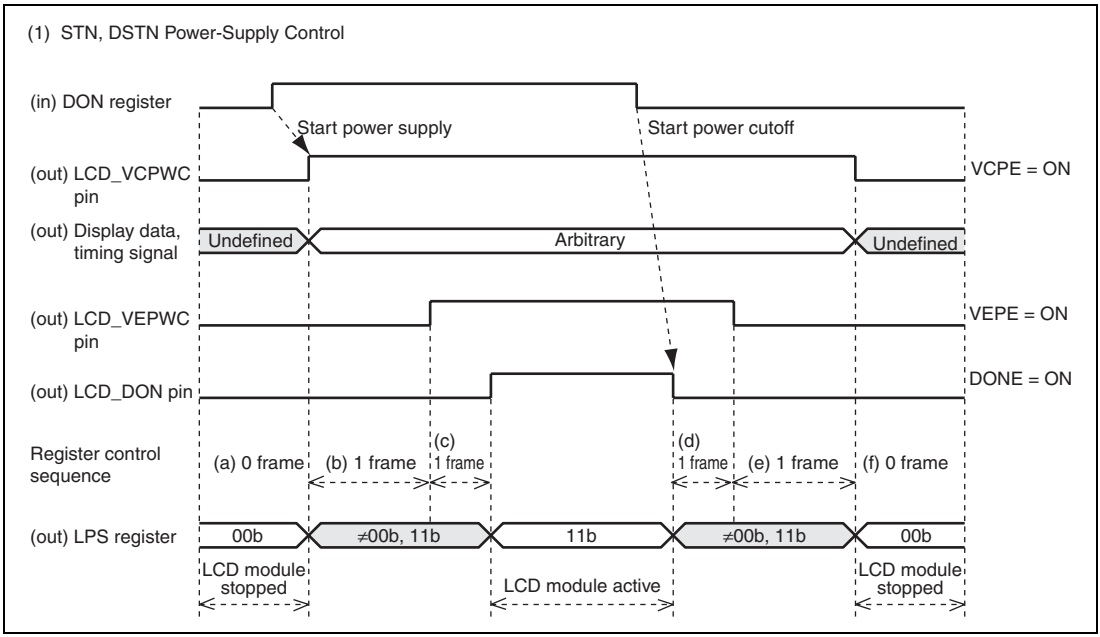


Figure 37.4 Power-Supply Control Sequence and States of the LCD Module

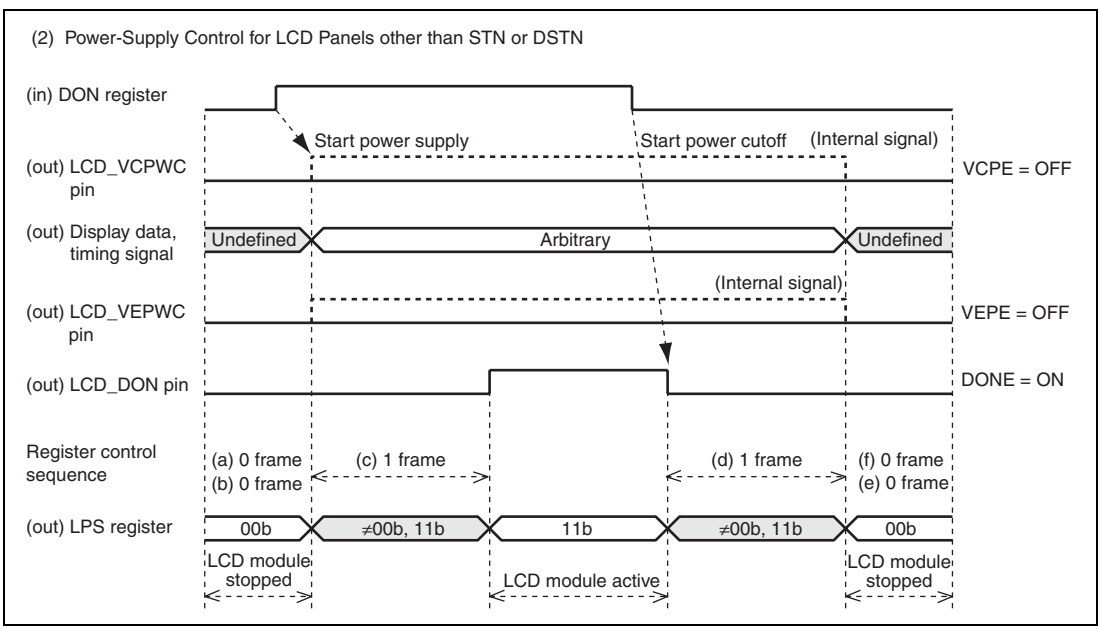


Figure 37.5 Power-Supply Control Sequence and States of the LCD Module

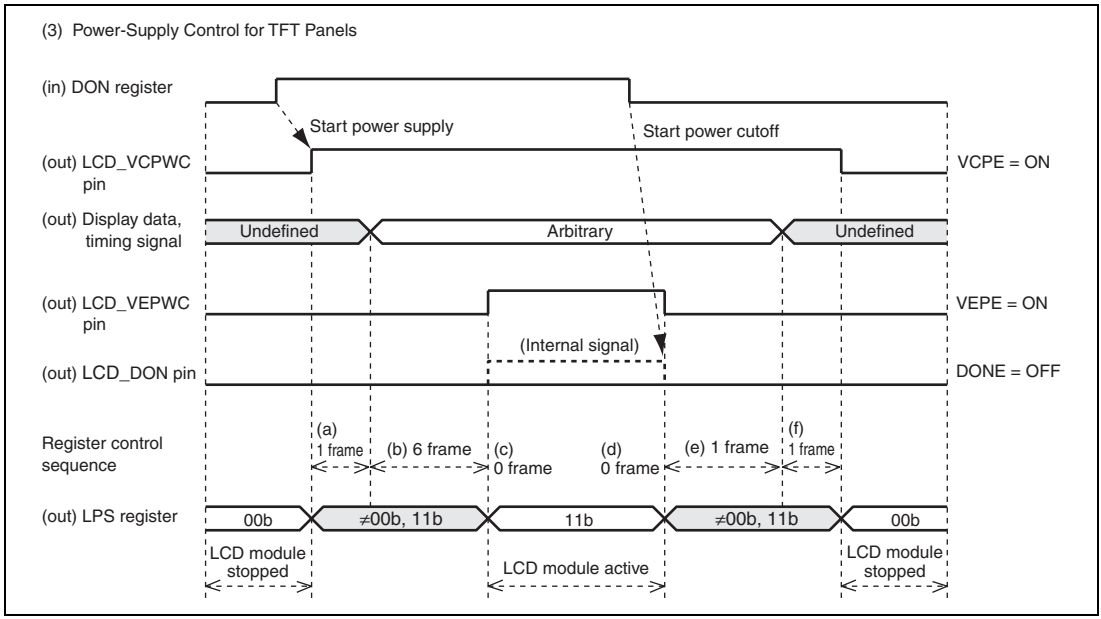


Figure 37.6 Power-Supply Control Sequence and States of the LCD Module

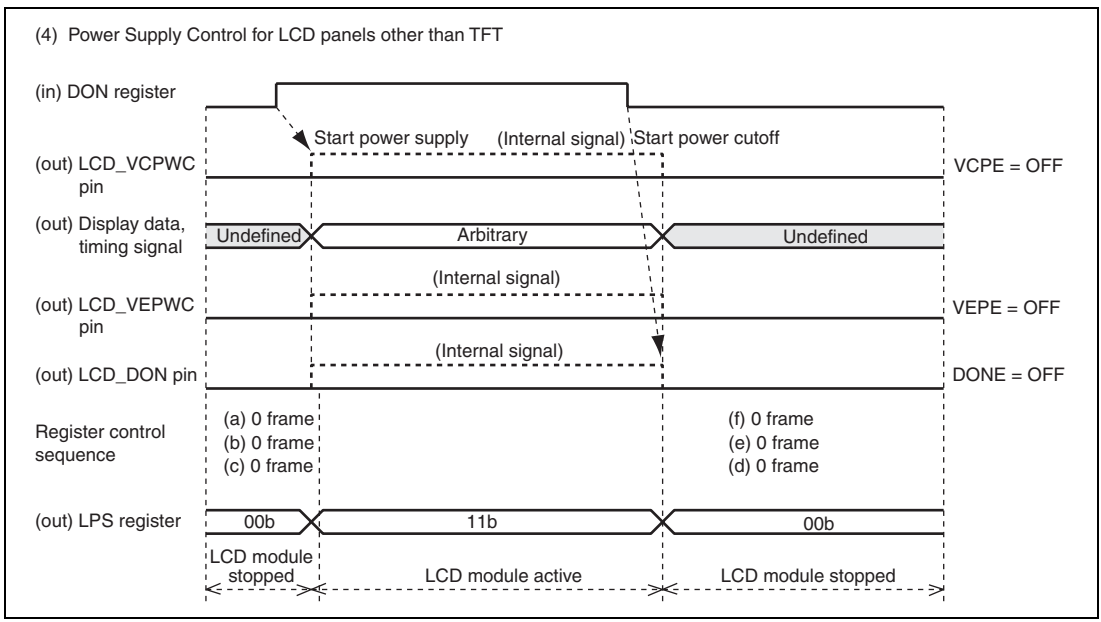


Figure 37.7 Power-Supply Control Sequence and States of the LCD Module

Table 37.6 Available Power-Supply Control-Sequence Periods at Typical Frame Rates

ONX, OFFX Register Value	Frame Rate	
	120 Hz	60 Hz
H'F	(–1+1)/120 = 0.00 (ms)	(–1+1)/60 = 0.00 (ms)
H'0	(0+1)/120 = 8.33 (ms)	(0+1)/60 = 16.67 (ms)
H'1	(1+1)/120 = 16.67 (ms)	(1+1)/60 = 33.33 (ms)
H'2	(2+1)/120 = 25.00 (ms)	(2+1)/60 = 50.00 (ms)
H'3	(3+1)/120 = 33.33 (ms)	(3+1)/60 = 66.67 (ms)
H'4	(4+1)/120 = 41.67 (ms)	(4+1)/60 = 83.33 (ms)
H'5	(5+1)/120 = 50.00 (ms)	(5+1)/60 = 100.00 (ms)
H'6	(6+1)/120 = 58.33 (ms)	(6+1)/60 = 116.67 (ms)
H'7	(7+1)/120 = 66.67 (ms)	(7+1)/60 = 133.33 (ms)
H'8	(8+1)/120 = 75.00 (ms)	(8+1)/60 = 150.00 (ms)
H'9	(9+1)/120 = 83.33 (ms)	(9+1)/60 = 166.67 (ms)
H'A	(10+1)/120 = 91.67 (ms)	(10+1)/60 = 183.33 (ms)
H'B	(11+1)/120 = 100.00 (ms)	(11+1)/60 = 200.00 (ms)
H'C	(12+1)/120 = 108.33 (ms)	(12+1)/60 = 216.67 (ms)
H'D	(13+1)/120 = 116.67 (ms)	(13+1)/60 = 233.33 (ms)
H'E	(14+1)/120 = 125.00 (ms)	(14+1)/60 = 250.00 (ms)

ONA, ONB, ONC, OFFD, OFFE, and OFFF are used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. 1 is subtracted from each register. H'0 to H'E settings select from 1 to15 frames. The setting H'F selects 0 frames.

Actual sequence periods depend on the register values and the frame frequency of the display. The following table gives power-supply control-sequence periods for display frame frequencies used by typical LCD modules.

- When ONB is set to H'6 and display's frame frequency is 120 Hz
  The display's frame frequency is 120 Hz. 1 frame period is thus 8.33 (ms) = 1/120 (sec).
  The power-supply input sequence period is 7 frames because ONB setting is subtracted by 1.
  As a result, the sequence period is 58.33 (ms) = 8.33 (ms) × 7.

Table 37.7 LCDC Operating Modes

Mode		Function
Display on (LCDC active)	Register setting: DON = 1	Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0	Register access is enabled. Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are not output to the LCD module.

Table 37.8 LCD Module Power-Supply States

(STN, DSTN module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems	DON Signal
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_D	LCD_VEPWC	LCD_DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

(TFT module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_D	LCD_VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The table above shows the states of the power supply, display data, and timing signals for the typical LCD module in its active and stopped states. Some of the supply voltages described may not be necessary, because some modules internally generate the power supply required for high-voltage systems from the logic-level power-supply voltage.

Notes on display-off mode (LCDC stopped):

If LCD module power-supply control-sequence processing is in use by the LCDC or the supply of power is cut off while the LCDC is in its display-on mode, normal operation is not guaranteed. In the worst case, the connected LCD module may be damaged.

37.4.7 Operation for Hardware Rotation

Operation in hardware-rotation mode is described below. Hardware-rotation mode can be thought of as using a landscape-format LCD panel instead of a portrait-format LCD panel by placing the landscape-format LCD panel as if it were a portrait-format panel. Whether the panel is intended for use in landscape or portrait format is thus no problem. The panel must, however, be within 320 pixels wide.

When making settings for hardware rotation, the following five differences from the setting for no hardware rotation must be noted. (The following example is for a display at 8 bpp. At 16 bpp, the amount of memory per dot will be doubled. The image size and register values used for rotation will thus be different.)

1. The image data must be prepared for display in the rotated panel. (If 240 × 320 pixels will be required after rotation, 240 × 320 pixel image data must be prepared.)
2. The register settings for the address of the image data must be changed (LDSARU and LDLAOR).

3. LDLAOR should be power of 2 (when the horizontal width after rotation is 240 pixels, LDLAOR should be set to 256).
4. Graphics software should be set up for the number 3 setting.
5. LDSARU should be changed to represent the address of the data for the lower-left pixel of the image rather than of the data for the upper-left pixel of the image.

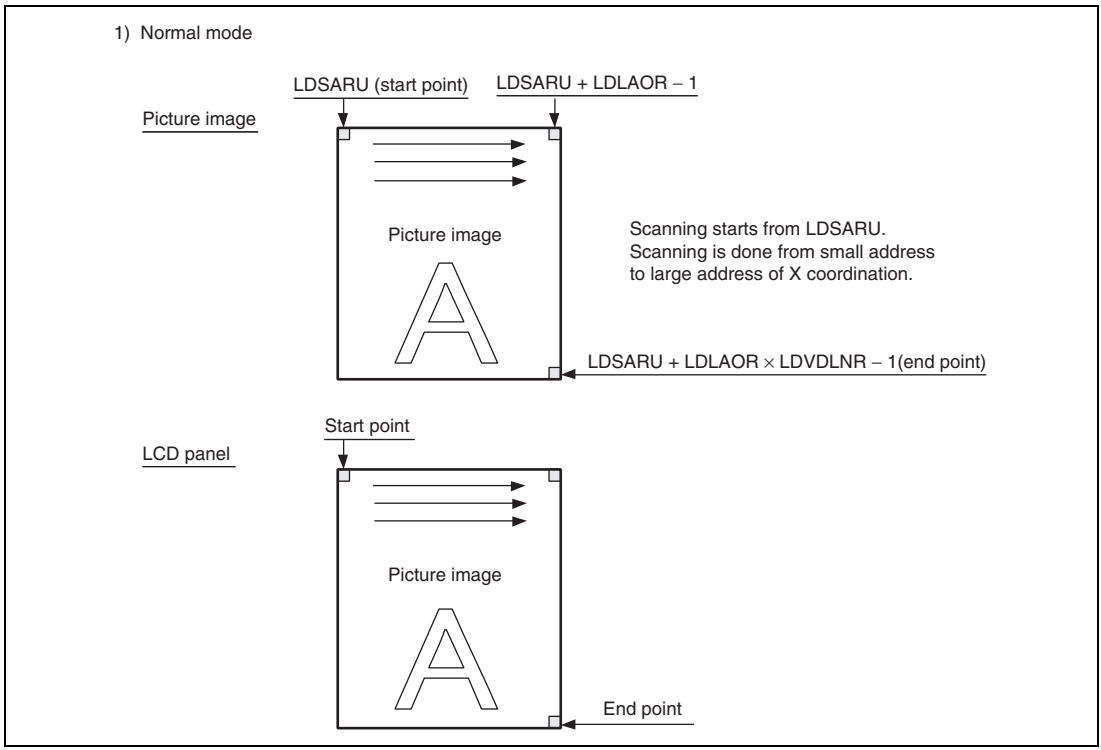


Figure 37.8 Operation for Hardware Rotation (Normal Mode)

For example, the registers have been set up for the display of image data in landscape format (320 × 240), which starts from LDSARU = 0x0c001000, on a 320 × 240 LCD panel. The graphics driver software is complete. Some changes are required to apply hardware rotation and use the panel as a 240 × 320 display. If LDLAOR is 512, the graphics driver software uses this power of 2 as the offset for the calculation of the addresses of Y coordinates in the image data. Before setting ROT to 1, the image data must be redrawn to suit the 240 × 320 LCD panel. LDLAOR will then be 256 because the size has changed and the graphics driver software must be altered accordingly. The point that corresponds to LDSARU moves from the upper left to the lower left of the display, so LDSARU should be changed to 0x0c001000 + 256 * 319.

Note: Hardware rotation allows the use of an LCD panel that has been rotated by 90 degrees. The settings in relation to the LCD panel should match the settings for the LCD panel before rotation. Rotation is possible regardless of the drawing processing carried out by the graphics driver software. However, the sizes in the image data and address offset values which are managed by the graphics driver software must be altered.

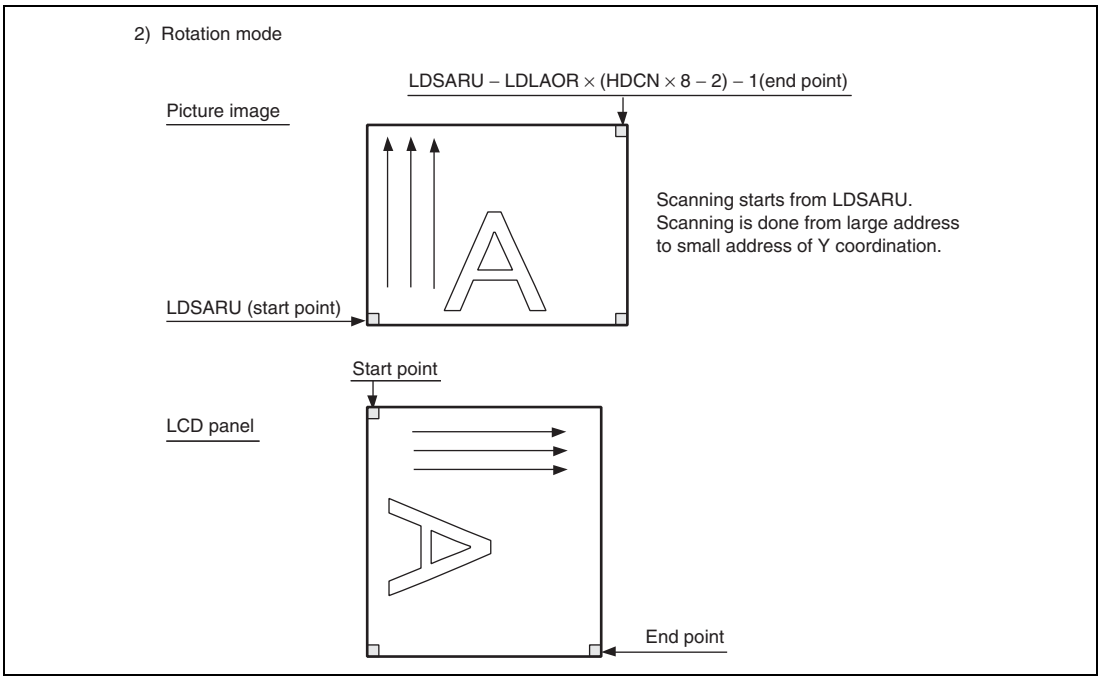


Figure 37.9 Operation for Hardware Rotation (Rotation Mode)

37.5 Clock and LCD Data Signal Examples

The following timing charts show timing of the normal output pins (LCD_***). The mirror pins (LCDM_***) have the same timing as the normal output pins.

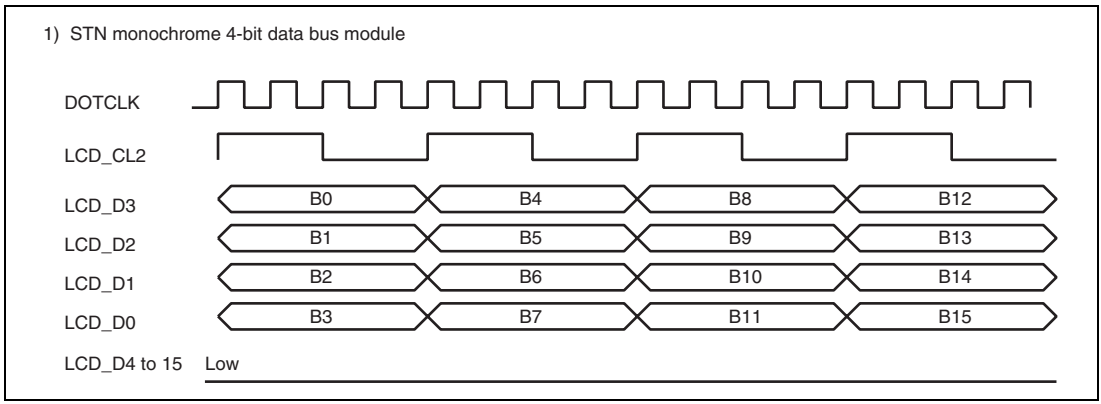


Figure 37.10 Clock and LCD Data Signal Example

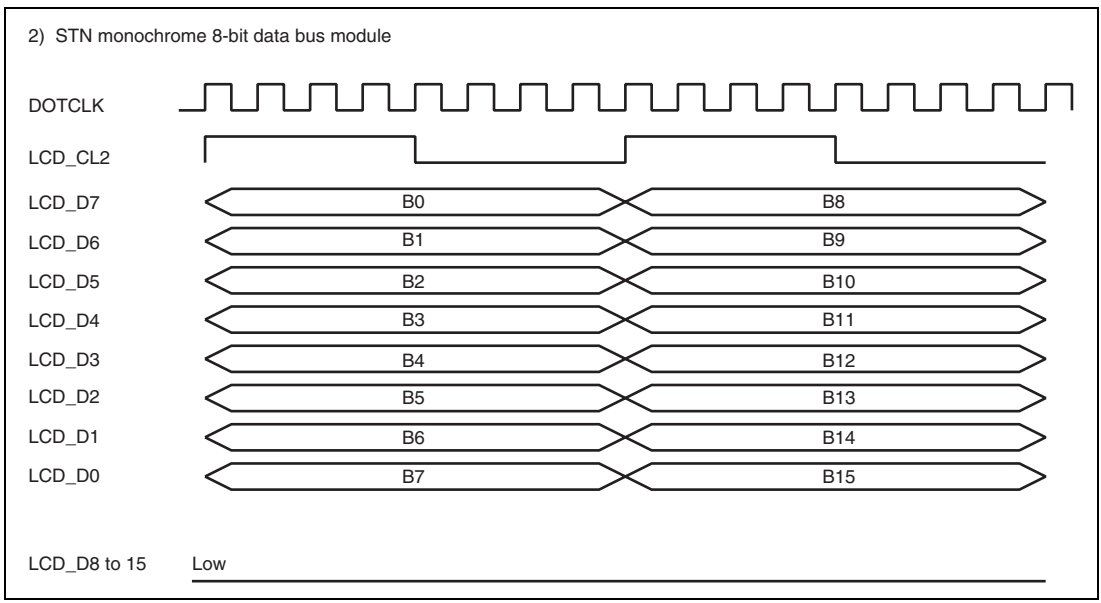


Figure 37.11 Clock and LCD Data Signal Example (STN Monochrome 8-Bit Data Bus Module)

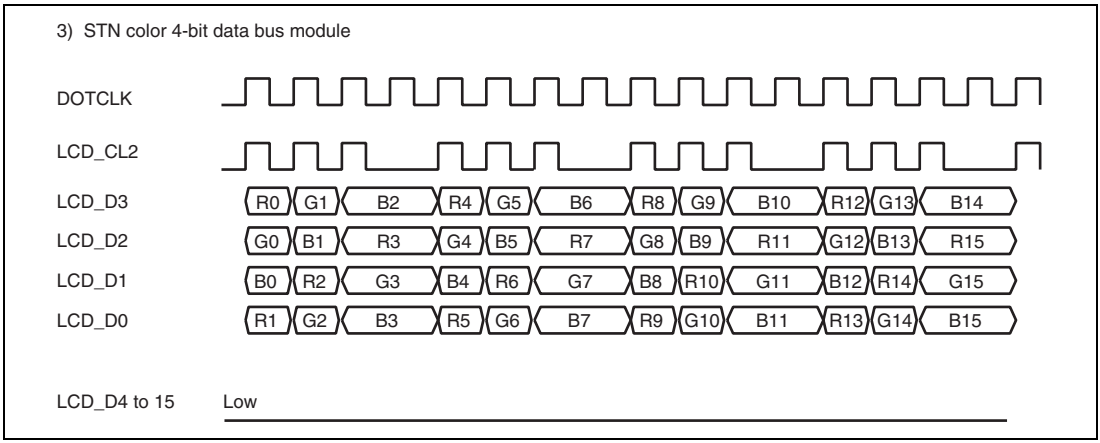


Figure 37.12 Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module)

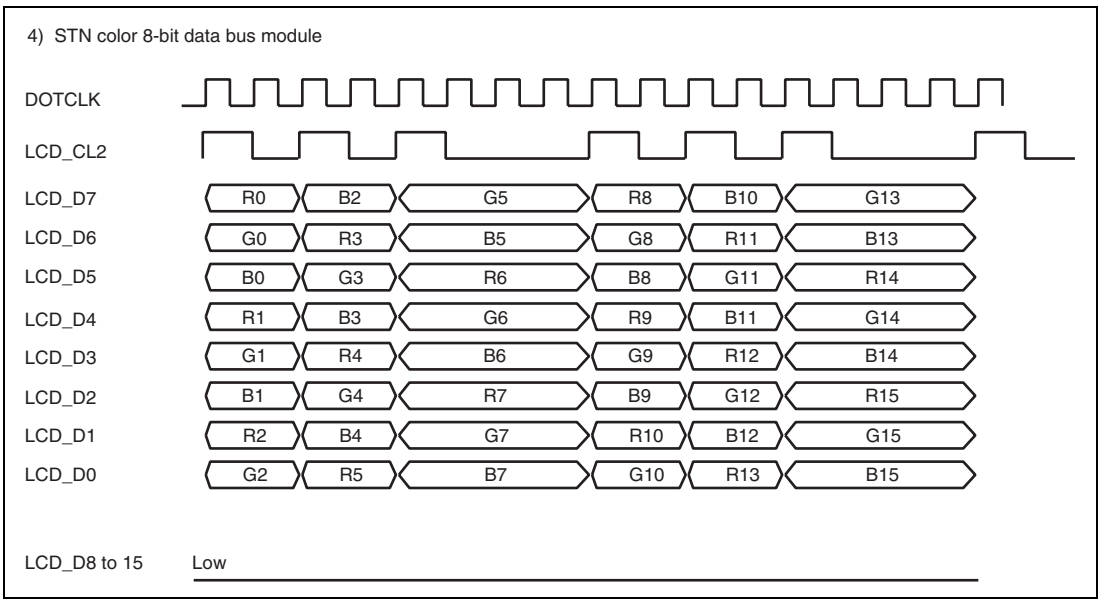


Figure 37.13 Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module)

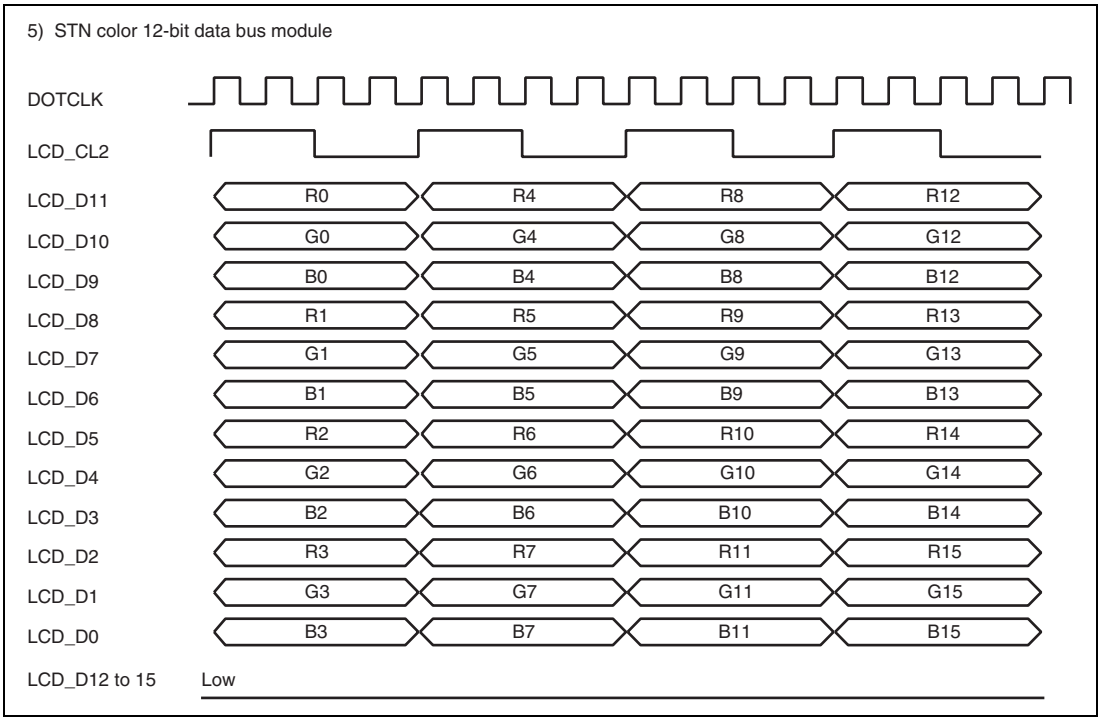


Figure 37.14 Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module)

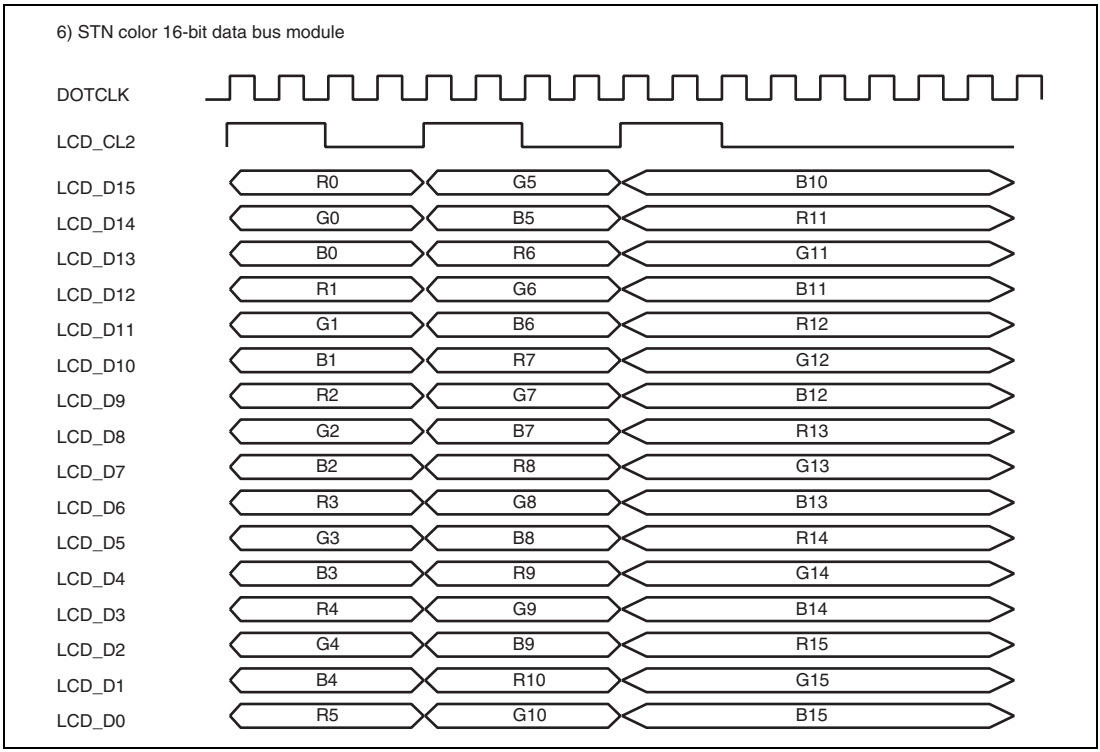


Figure 37.15 Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module)

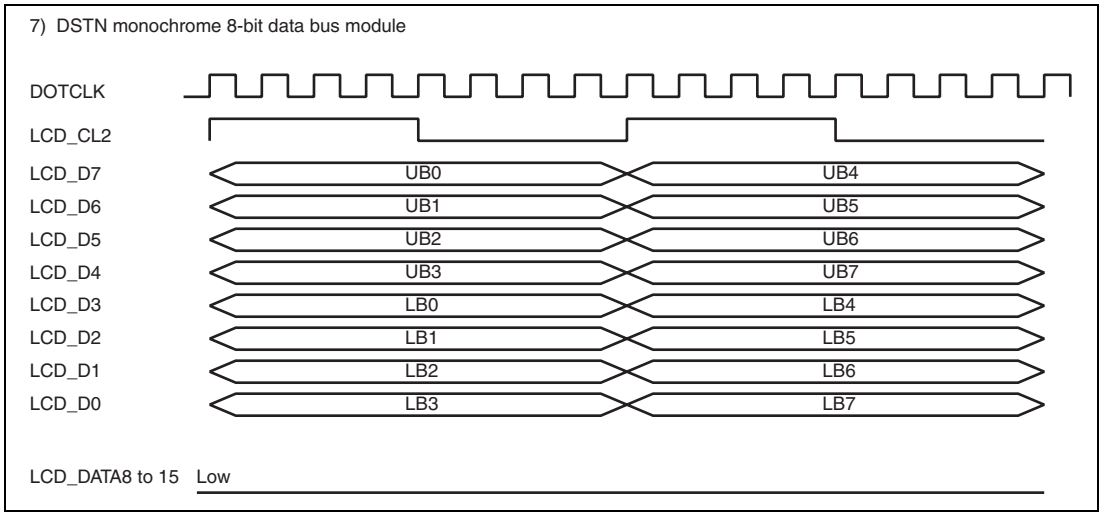


Figure 37.16 Clock and LCD Data Signal Example (DSTN Monochrome 8-Bit Data Bus Module)

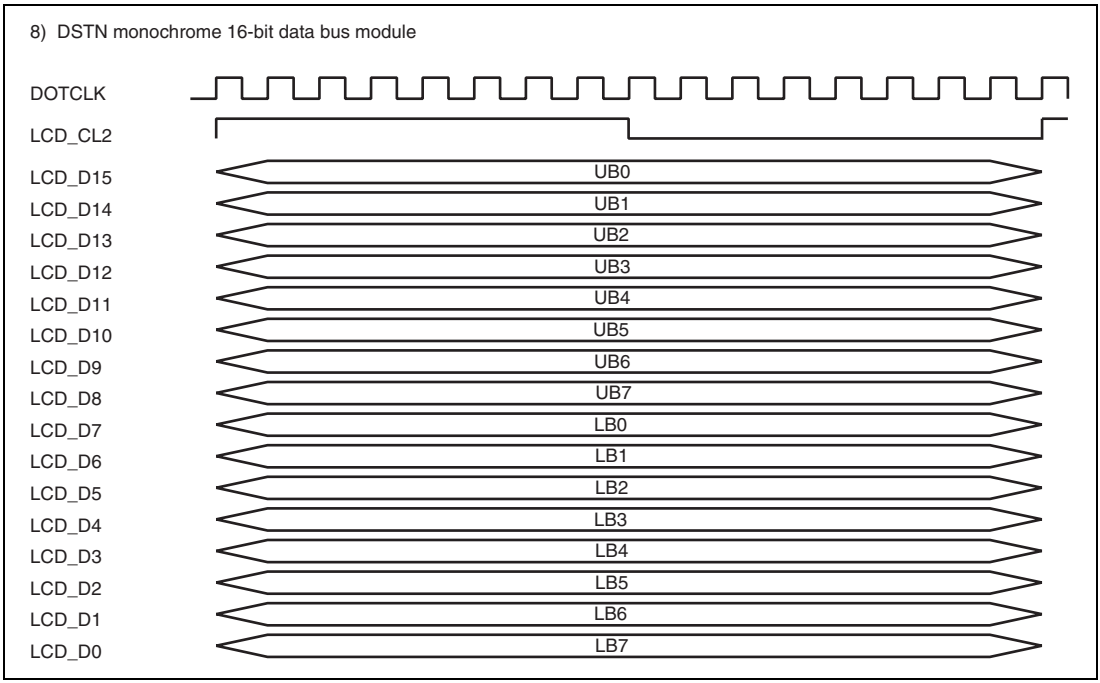


Figure 37.17 Clock and LCD Data Signal Example (DSTN Monochrome 16-Bit Data Bus Module)

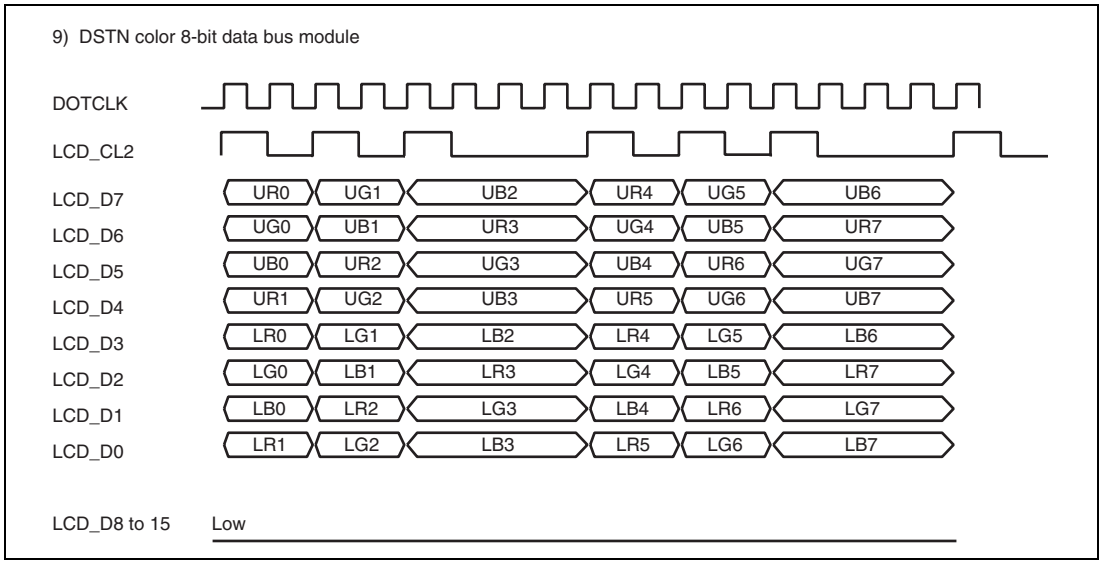


Figure 37.18 Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module)

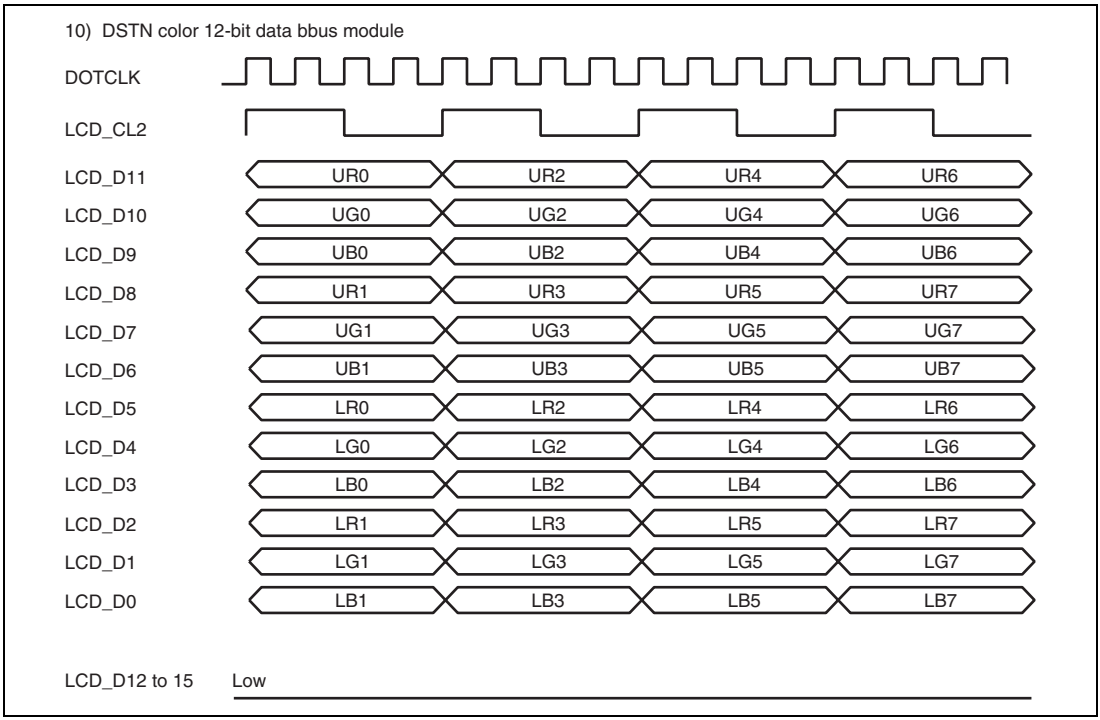


Figure 37.19 Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module)

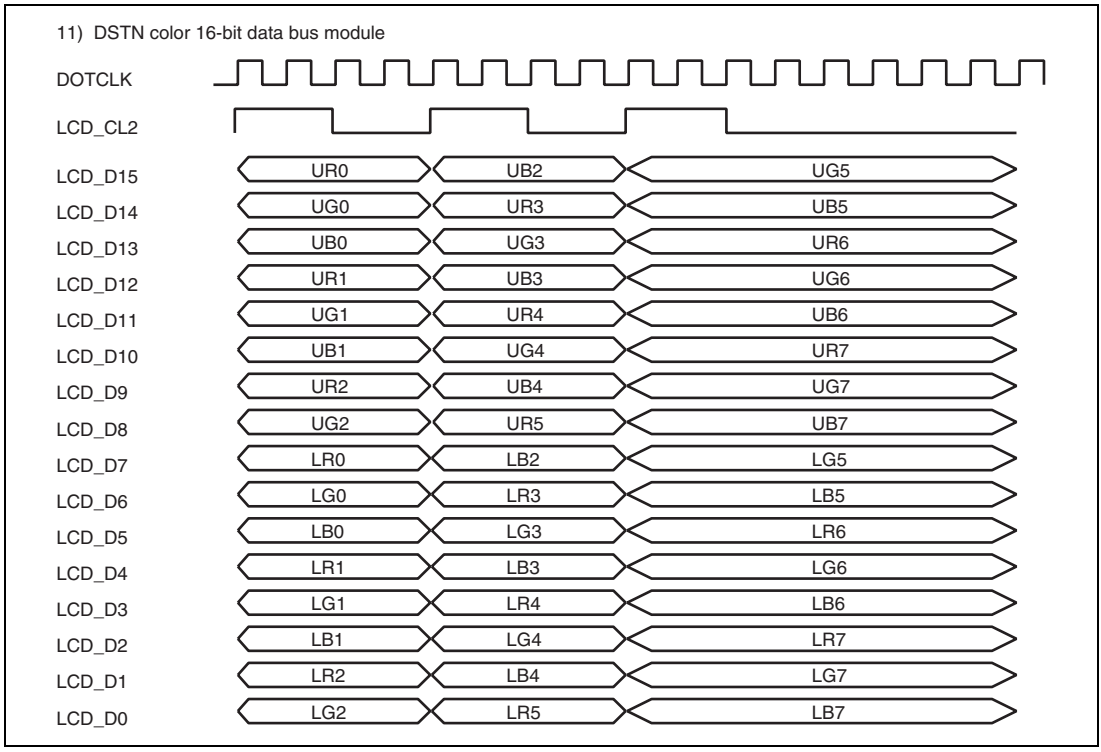


Figure 37.20 Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module)

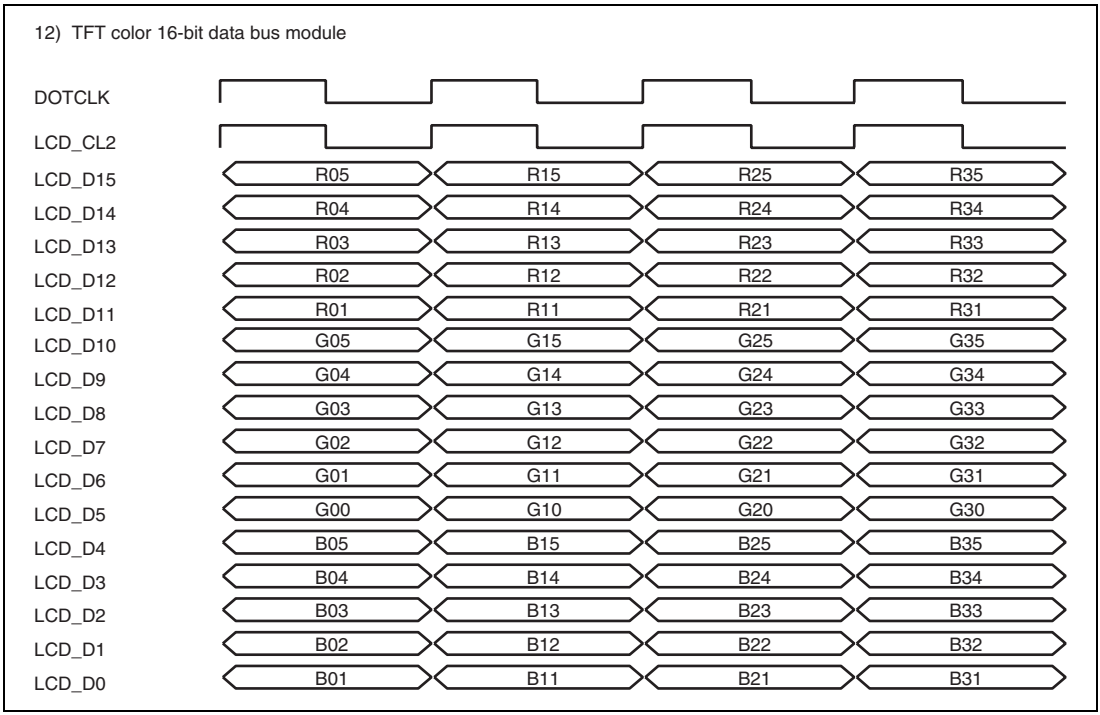


Figure 37.21 Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module)

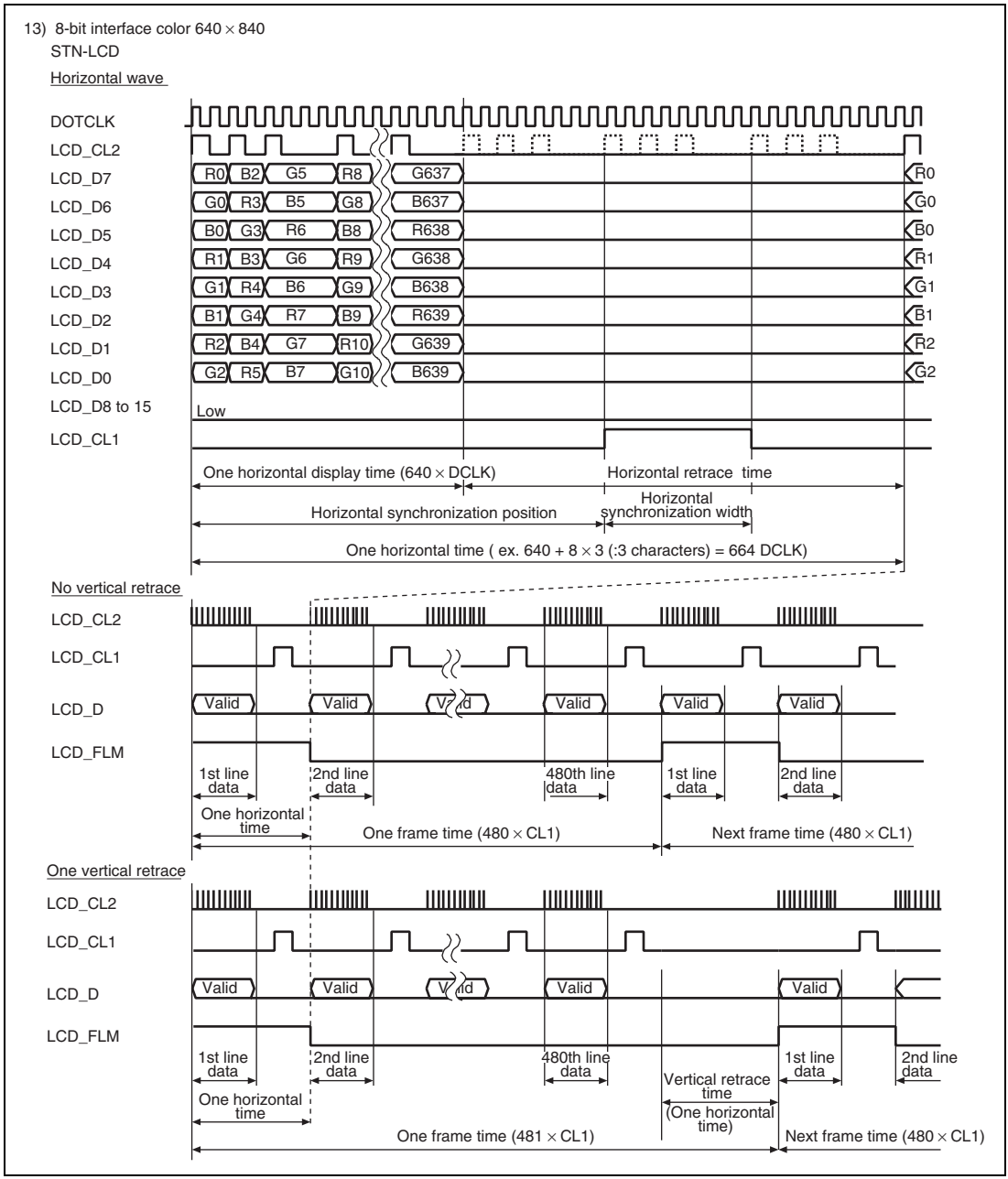


Figure 37.22 Clock and LCD Data Signal Example (8-Bit Interface Color 640 × 480)

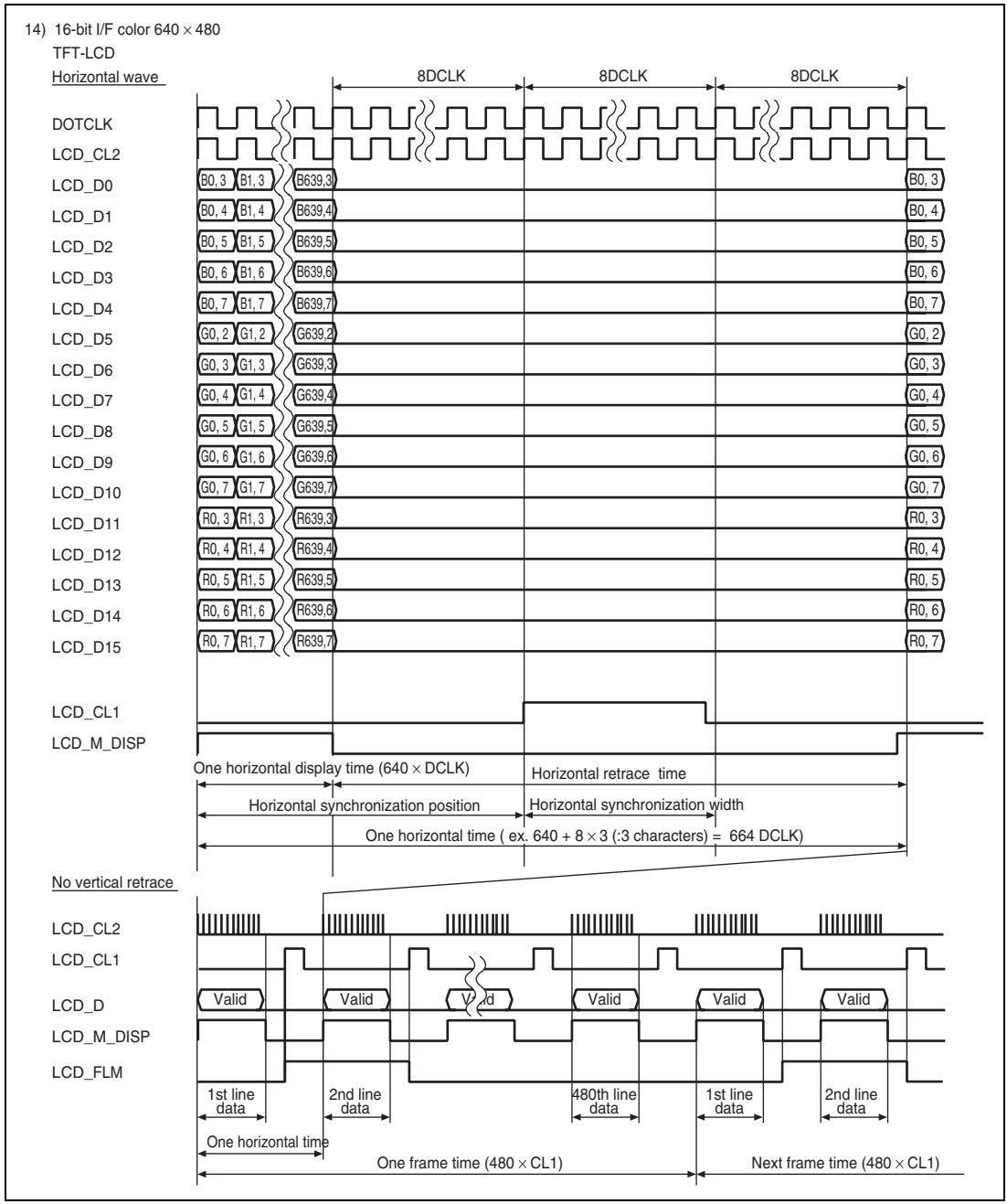


Figure 37.23 Clock and LCD Data Signal Example (16-Bit Interface Color 640 × 480)

37.6 Usage Notes

37.6.1 Procedure for Halting Access to Display Data Storage VRAM (DDR-SDRAM in Area 3)

Follow the procedure below to halt access to VRAM for storing display data (DDR-SDRAM in area 3).

Procedure for Halting Access to Display Data Storage VRAM:

1. Confirm that the LPS1 and LPS0 bits in LDPMMR are currently set to 1.
2. Clear the DON bit in LDCNTR to 0 (display-off mode).
3. Confirm that the LPS1 and LPS0 bits in LDPMMR have changed to 0.
4. Wait for the display time for a single frame to elapse.

This halting procedure is required before selecting self-refreshing for the display data storage VRAM (DDR-SDRAM in area 3) or making a transition to standby mode or module standby mode.

37.6.2 Notes on Using NMI Interrupt

If the NMIFL bit in the NMIFCR register is set to 1 by an NMI interrupt while the LCDC is used, the LCDC cannot access the VRAM that is used for the display data storage (DDR_SDRAM in area 3).
As the LCDC continues to output data stored in the lime buffer to the LCD panel data pin, the LCD display will be stopped if the line buffer becomes empty. Accordingly, NMI interrupts should be disabled and he NMIFL bit should be cleared to 0 before the line buffer becomes empty.

Section 38 A/D Converter

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to four analog input channels.

38.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Four input channels
- High-speed conversion
  - Conversion time: maximum 8.5 µs per channel (Pck0 = 33 MHz operation)
- Three conversion modes
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
- Sample-and-hold function
- A/D interrupt requested at the end of conversion
  - At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

Figure 38.1 shows a block diagram of the A/D converter.

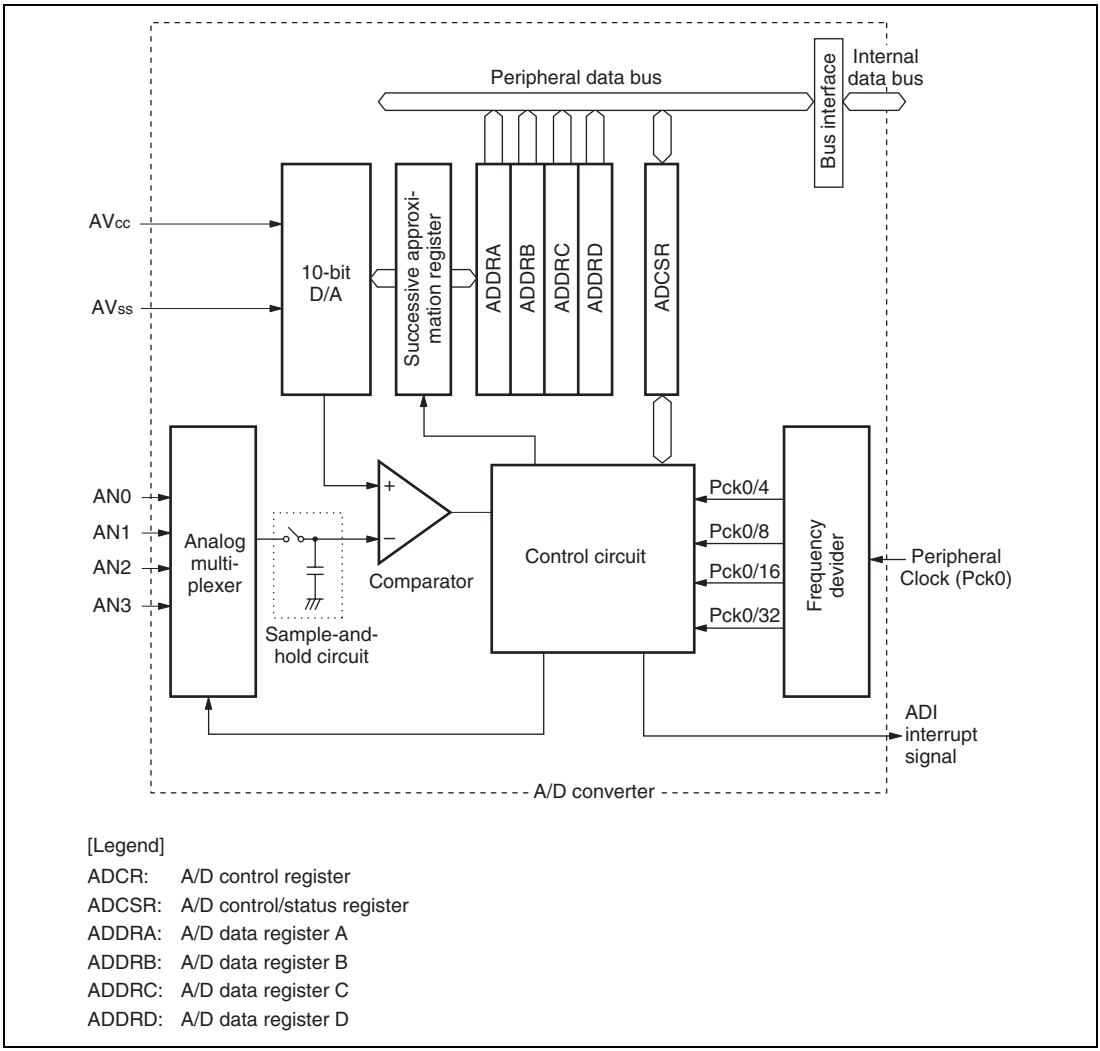


Figure 38.1 Block Diagram of A/D Converter

38.2 Input Pins

Table 38.1 summarizes the A/D converter's input pins. AVcc and AVss are the power supply inputs for the analog circuits in the A/D converter. AVcc also functions as the A/D converter reference voltage pin.

Table 38.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply
Analog ground pin	AVss	Input	Analog ground and reference voltage for A/D conversion
Analog input pin 0	AN0	Input	Analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	

38.3 Register Descriptions

Table 38.2 shows the ADC register configuration. Table 38.3 shows the register state in each operating mode.

Table 38.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
A/D data register A	ADDRA	R	H'FFEA 0000	H'1FEA 0000	16
A/D data register B	ADDRB	R	H'FFEA 0002	H'1FEA 0002	16
A/D data register C	ADDRC	R	H'FFEA 0004	H'1FEA 0004	16
A/D data register D	ADDRD	R	H'FFEA 0006	H'1FEA 0006	16
A/D control/status register	ADCSR	R/W	H'FFEA 0010	H'1FEA 0010	16

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 38.3 Register State in Each Operating Mode

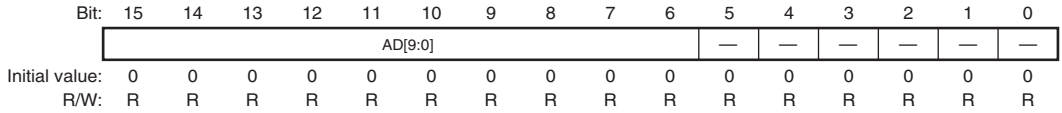
Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
A/D data register A	ADDRA	H'0000	H'0000	Retained	Retained
A/D data register B	ADDRB	H'0000	H'0000	Retained	Retained
A/D data register C	ADDRC	H'0000	H'0000	Retained	Retained
A/D data register D	ADDRD	H'0000	H'0000	Retained	Retained
A/D control/status register	ADCSR	H'0000	H'0000	Retained	Retained

38.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte (bits 15 to 6) of the A/D data register. Bits 5 to 0 of an A/D data register are always read as 0. Table 38.4 indicates the pairings of analog input channels and A/D data registers.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.



Bit	Bit Name	Initial Value	R/W	Description
15 to 6	AD[9:0]	0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 38.4 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

38.3.2 A/D Control/Status Registers (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'0080 by a reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	—	—	—	—	—	CKS[1:0]		MDS[1:0]		—	CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Only 0 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag Indicates the end of A/D conversion. [Clearing conditions] (1) Cleared by reading ADF while ADF = 1, then writing 0 to ADF [Setting conditions] Single mode: A/D conversion ends Multi mode: A/D conversion has cycled through the selected channels (A/D conversion cycles through the selected channels) Scan mode: A/D conversion has cycled through the selected channels (A/D conversion is continuously repeated for the selected channels) Note: When clearing the ADST bit to 0 to stop A/D conversion in scan mode or in multi mode (ADF = 0), after clearing the ADST bit to 0, read the ADST bit in ADCSR and confirm that it is 0. Then, after at least the time for A/D conversion on one channel has elapsed, set the ADST bit to 1 again. Note that the A/D conversion time differs according to the A/D conversion clock division ratio.
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is cleared to 0 automatically when conversion on the specified channel ends. Even when the ADST bit is cleared to 0 (by software), A/D conversion does not stop (0 cannot be written to this bit during A/D conversion). Multi mode: A/D conversion starts. This bit is cleared to 0 automatically when conversion on the specified channels has been performed for one cycle. When the ADST bit is cleared to 0 (by software), A/D conversion stops when the currently executed channel ends. Scan mode: A/D conversion starts. A/D conversion continues until the ADST bit is cleared to 0 by software, a reset, or a transition to standby mode.
12 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CKS[1:0]	01	R/W	Clock Select Selects the A/D conversion clock divition ratio. 00: Pck0/4 01: Pck0/8 10: Pck0/16 11: Pck0/32
5, 4	MDS[1:0]	00	R/W	conversion mode select Selects single mode, multi mode, or scan mode. 00: Single mode 01: Reserved (setting prohibited) 10: Multi mode 11: Scan mode
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection. Single mode / Multi mode or scan mode 000: AN0 / AN0 001: AN1 / AN0, AN1 010: AN2 / AN0 to AN2 011: AN3 / AN0 to AN3 100: Reserved (setting prohibited) 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

Note: * Only 0 can be written to clear the flag.

38.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has three operating modes: single mode, multi mode, and scan mode.

38.4.1 Single Mode (MDS1 = 0, MDS0 = 0)

Single mode should be selected when A/D conversion on only one channel is required. A/D conversion starts when the ADST bit (bit 13) of the A/D control/status register (ADCSR) is set to 1 by software. The ADST bit holds 1 during A/D conversion and is automatically cleared to 0 when A/D conversion ends.

When A/D conversion ends, the ADF bit (bit 15) of ADCSR is set to 1. If the ADIE bit (bit 14) in ADCSR is also set to 1, an A/D conversion end interrupt (ADI) is requested at this time.

Writing 0 to the ADF bit after reading ADF = 1 clears the ADF bit.

When setting the A/D control/status register (ADCSR) or switching the analog input channel during A/D conversion, first clear the ADST bit to 0 to halt A/D conversion in order to avoid malfunction. After the change has been made, setting the ADST bit to 1 resumes A/D conversion.

Typical operations when channel 1 (AN1) is selected in single mode are described below. Figure 38.2 shows a timing diagram for this example.

1. Select single mode as the operating mode (MDS[1:0] = 00), AN1 as the input channel (CH[2:0] = 001), and enable A/D interrupt requests (ADIE = 1). Then start A/D conversion (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. At the same time, the ADF bit is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt request is generated.
4. The A/D interrupt processing routine starts.
5. The A/D interrupt processing routine reads and processes the A/D conversion result (ADDRB).
6. After reading ADF = 1, write 0 in the ADF bit.
7. Execution of the A/D interrupt processing routine ends. After this, when the ADST bit is set to 1, A/D conversion starts and steps 2 to 7 are repeated.

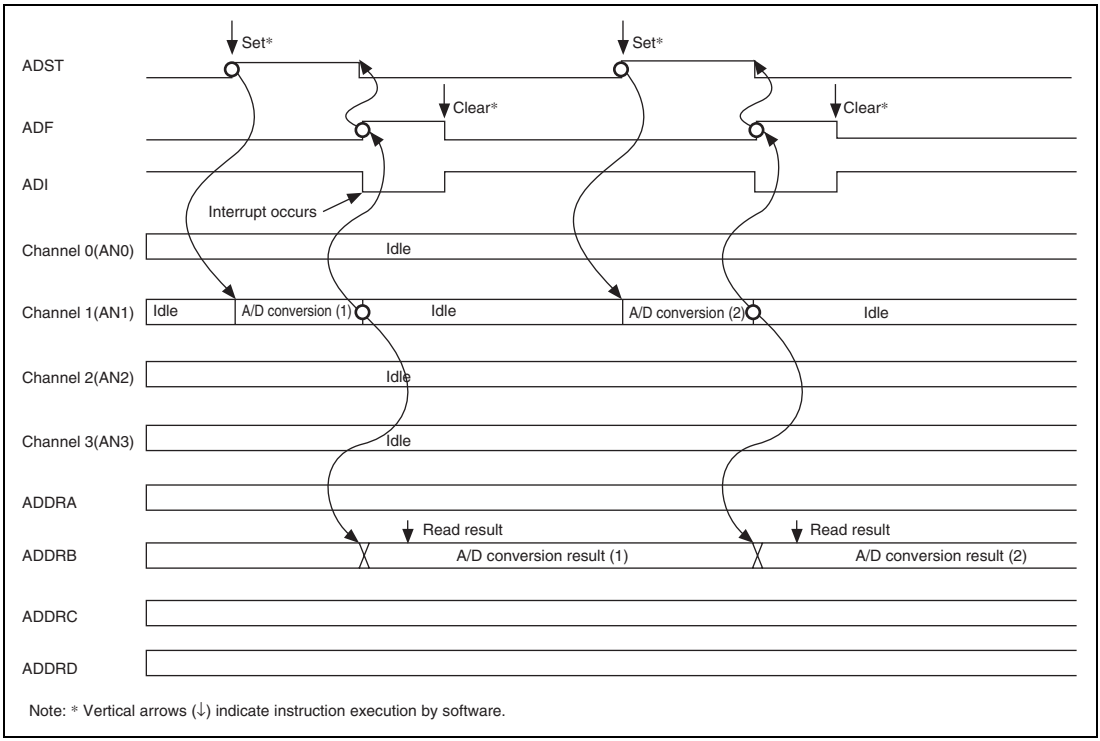


Figure 38.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

38.4.2 Multi Mode (MDS[1:0] = 10)

In multi mode, analog inputs for the specified channels (one or more) are converted once each. A/D conversion starts with the first channel (AN0) when the ADST bit (bit 13) of the A/D control/status register (ADCSR) is set to 1 by software.

When multiple channels are selected, A/D conversion for the second channel (AN1) starts immediately after A/D conversion for the first channel ends.

A/D conversion on the specified channels is performed for one cycle. The conversion results are transferred for storage to the ADDR that corresponds to the channel.

When setting the A/D control/status register (ADCSR) or switching the analog input channel during A/D conversion, first clear the ADST bit to 0 to halt A/D conversion in order to avoid malfunction. After the change has been made, setting the ADST bit to 1 selects the first channel and A/D conversion is resumed.

Typical operations when three channels (AN0 to AN2) are selected in multi mode are described below. Figure 38.3 shows a timing diagram for this example.

1. Select multi mode as the operating mode (MDS[1:0] = 10) and AN0 to AN2 as the analog input channels (CH[2:0] = 010). Then start A/D conversion (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion ends, the result is transferred into ADDRA. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
3. A/D conversion proceeds in the same way up to the third channel (AN2).
4. When A/D conversion of all selected channels (AN0 to AN2) is completed, the ADF bit is set to 1, the ADST bit is cleared to 0, and A/D conversion stops.

   If the ADIE bit is set to 1 at this time, an ADI interrupt is generated after A/D conversion ends.

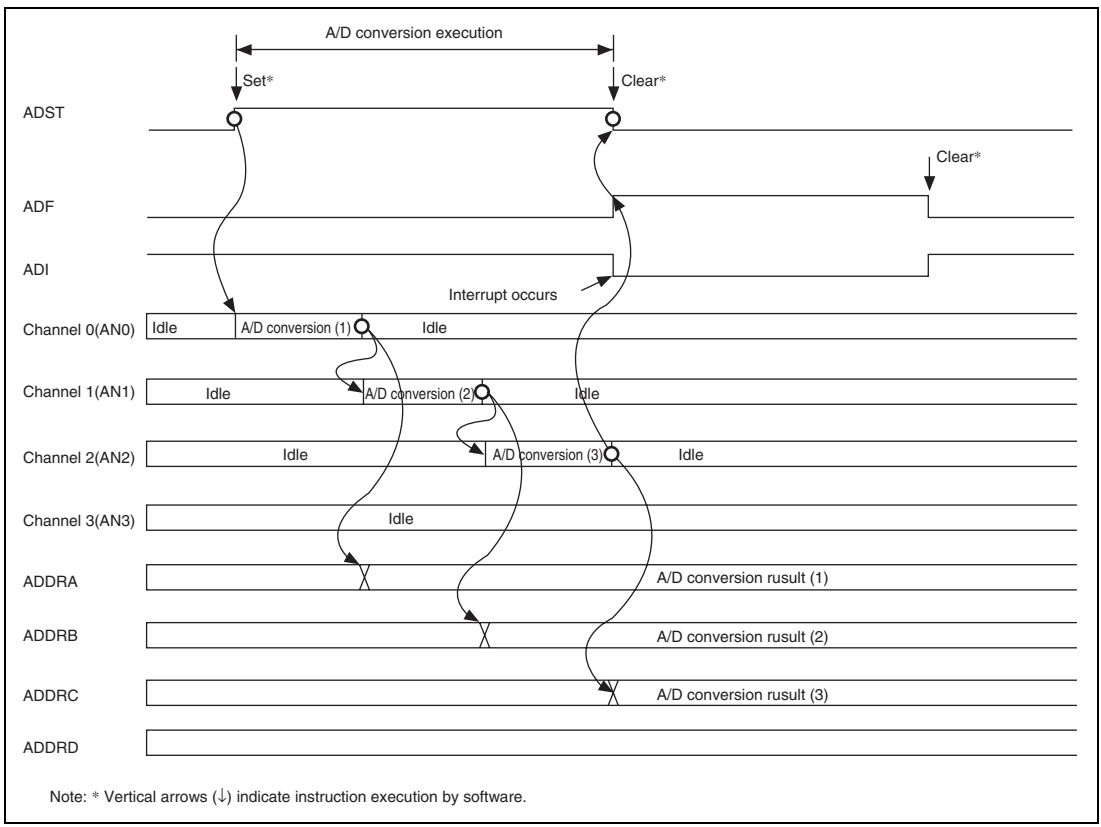


Figure 38.3 Example of A/D Converter Operation (Multi Mode, Three Channels AN0 to AN2 Selected)

38.4.3 Scan Mode (MDS1 = 1, MDS0 = 1)

In scan mode, A/D conversion is continuously repeated for the selected channels until the ADST bit (bit 13) is cleared to 0. The A/D conversion results are transferred for storage to the ADDR that corresponds to the channel. This mode is suitable for systems that continuously monitor analog inputs to multiple channels (or a single channel). A/D conversion starts with the first channel (AN0) when the ADST bit of the A/D control/status register (ADCSR) is set to 1 by software.

When multiple channels are selected, after A/D conversion for channel n ends, A/D conversion for channel (n + 1) starts immediately.

A/D conversion is continuously repeated for the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage to the ADDR that corresponds to the channel.

When setting the A/D control/status register (ADCSR) or switching the analog input channel during A/D conversion, first clear the ADST bit to 0 to halt A/D conversion in order to avoid malfunction. After the change has been made, setting the ADST bit to 1 selects the first channel and A/D conversion is resumed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described below. Figure 38.4 shows a timing diagram for this example.

1. Select scan mode as the operating mode (MDS[1:0] = 11) and AN0 to AN2 as the input channels (CH[2:0] = 010). Then start A/D conversion (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion ends, the result is transferred into ADDRA. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
3. A/D conversion proceeds in the same way up to the third channel (AN2).
4. When A/D conversion of all selected channels (AN0 to AN2) is completed, the ADF bit is set to 1, the first channel (AN0) is selected again, and A/D conversion is consecutively performed. (In multi mode, A/D conversion ends when the selected channels have been cycled through. However, in scan mode, after the selected channels have been cycled through, A/D conversion starts again from the first channel and is consecutively repeated.)

   If the ADIE bit is set to 1 at this time, an ADI interrupt is generated after A/D conversion ends.
5. While the ADST bit is set to 1, steps 2 to 4 above are repeated.

   When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).

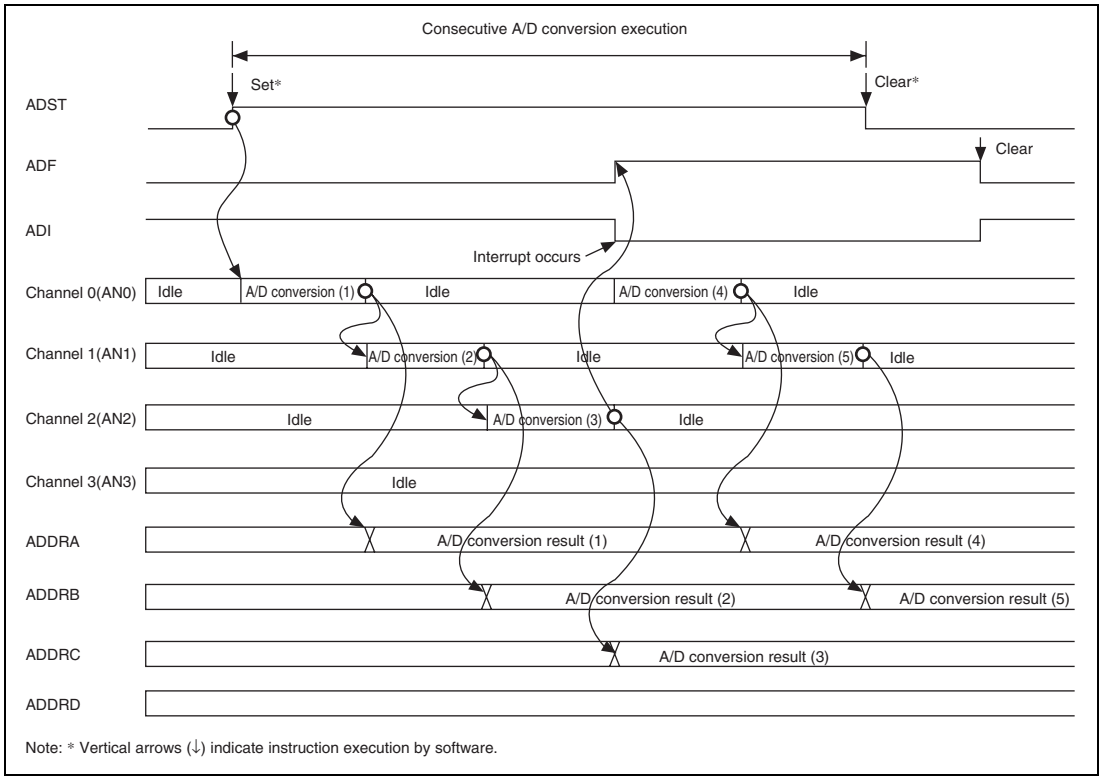


Figure 38.4 Example of A/D Converter Operation (Scan Mode, Three Channels AN0 to AN2 Selected)

38.4.4 A/D Conversion Time

Table 38.5 indicates the A/D conversion time.

Table 38.5 A/D Conversion Time

	Pck0/4		Pck0/8		Pck0/16		Pck0/32	
Conversion Time Type	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
A/D conversion time for the first conversion (single mode)*	136	139	268	275	532	547	1060	1091
A/D conversion time for the second and subsequent conversions (multi mode or scan mode)	—	128	—	256	—	512	—	1024

Notes: Values in the table are the numbers of states (one state is one peripheral clock (IO-Bus) Pck0 cycle).

* Period starting from when the ADST bit is set to 1 and until data is stored in the register.

38.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion.

The ADI interrupt request is enabled/disabled by specifying the ADIE bit in ADCSR.

38.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an input for an analog channel to its analog reference voltage and converts it into 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

1. Offset error (figure 38.5 (1))

   Deviation between actual A/D conversion characteristics and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in figure 38.5) to 0000000001 (001 in figure 38.5)
2. Full-scale error (figure 38.5 (2))

   Deviation between actual A/D conversion characteristics and ideal A/D conversion characteristics when the digital output value changes from 1111111110 (110 in figure 38.5) to the maximum 1111111111 (111 in figure 38.5).
3. Quantization error (figure 38.5 (3))

   Intrinsic error of the A/D converter and is expressed as 1/2 LSB.
4. Nonlinearity error (figure 38.5 (4))

   Deviation between actual A/D conversion characteristics and ideal A/D conversion characteristics between zero voltage and full-scale voltage. Note that it does not include offset, full-scale, and quantization errors.

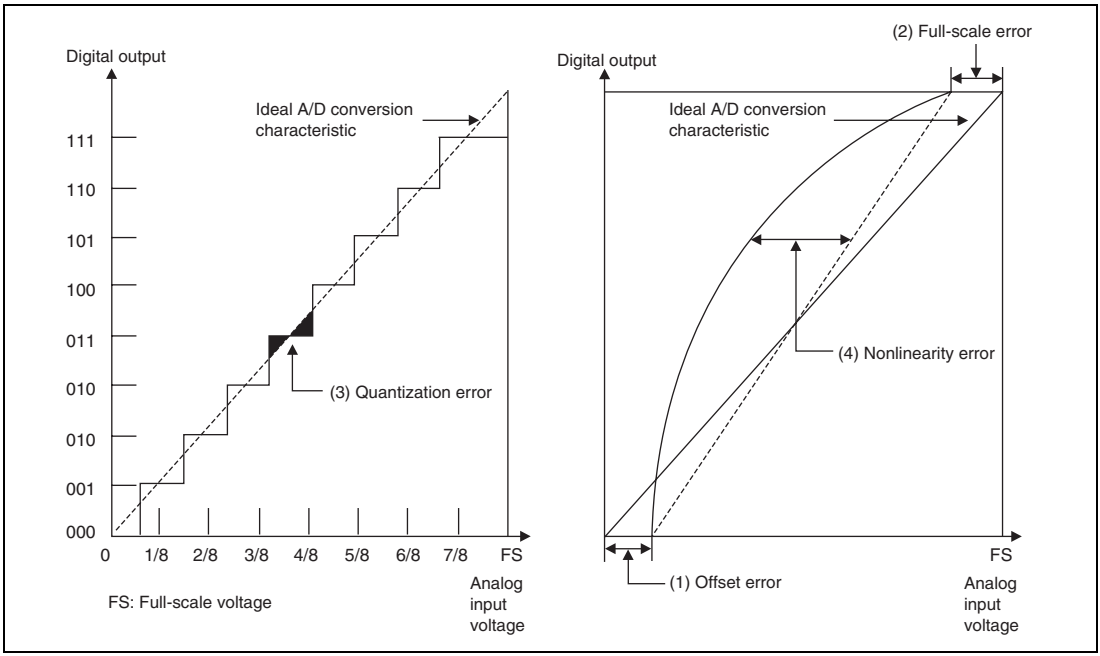


Figure 38.5 Definitions of A/D Conversion Accuracy

38.7 Usage Notes

When using the A/D converter, note the points listed below.

38.7.1 Setting Analog Input Voltage

1. Analog input voltage range

   During A/D conversion, the voltages input to the analog input pins ANn should be in the range AVss ≤ ANn ≤ AVcc (n = 0 to 3).

2. AVcc and AVss input voltages

   The AVcc and AVss input voltages should be as follows: AVcc = 3.3 V ± 0.3 V and AVss = Vss. (AVcc = Analog power supply, AVss = Analog ground, Vss = Internal digital power supply)

38.7.2 Processing of Analog Input Pins

To prevent damage from abnormal voltage such as voltage surges at the analog input pins (AN0 to AN3), connect a protection circuit like the one shown in figure 38.6. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be determined according to actual application conditions.

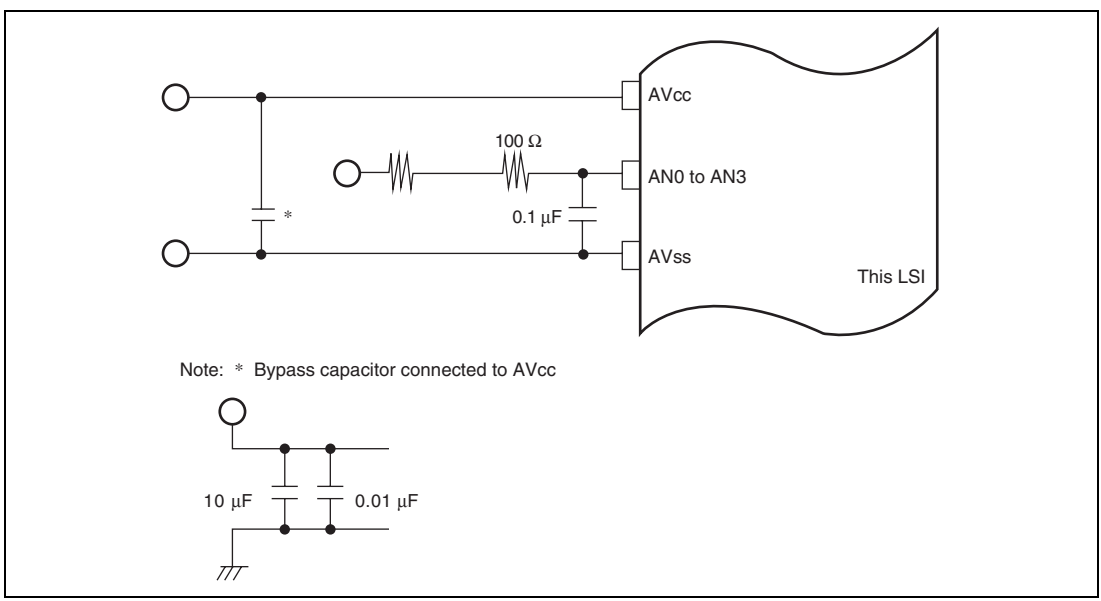


Figure 38.6 Example of Analog Input Pin Protection Circuit

38.7.3 Pck0 Clock and Clock Division Ratio Settings

Four types of frequency divided clocks can be used as the clock for A/D conversion.

Since the internal circuit configuration affects the limits of the interface between the analog and digital sections, be sure to see table 38.6 when setting the Pck0 clock and clock division ratio.

Table 38.6 Relationship between Clock Division Ratio and Usable Pck0 Clock Frequency

Clock Division Ratio	Pck0 Clock
Pck0/4	18 MHz or lower
Pck0/8	34 MHz or lower
Pck0/16	67 MHz or lower
Pck0/32	67 MHz or lower

38.7.4 A/D Conversion Stop

In multi mode or scan mode, A/D conversion does not stop as soon as the setting to halt A/D conversion has been made. A/D conversion stops as soon as A/D conversion of the data in the relevant channel has finished.

Section 39 D/A Converter (DAC)

This LSI incorporates a two-channel D/A converter (DAC) with the following features.

39.1 Features

- 8-bit resolution
- Two output channels
- Conversion time: Max. 10 μs (when load capacitance is 20 pF)
- Output voltage: 0 V to AVcc (analog power supply)

Figure 28.1 shows the block diagram for the DAC.

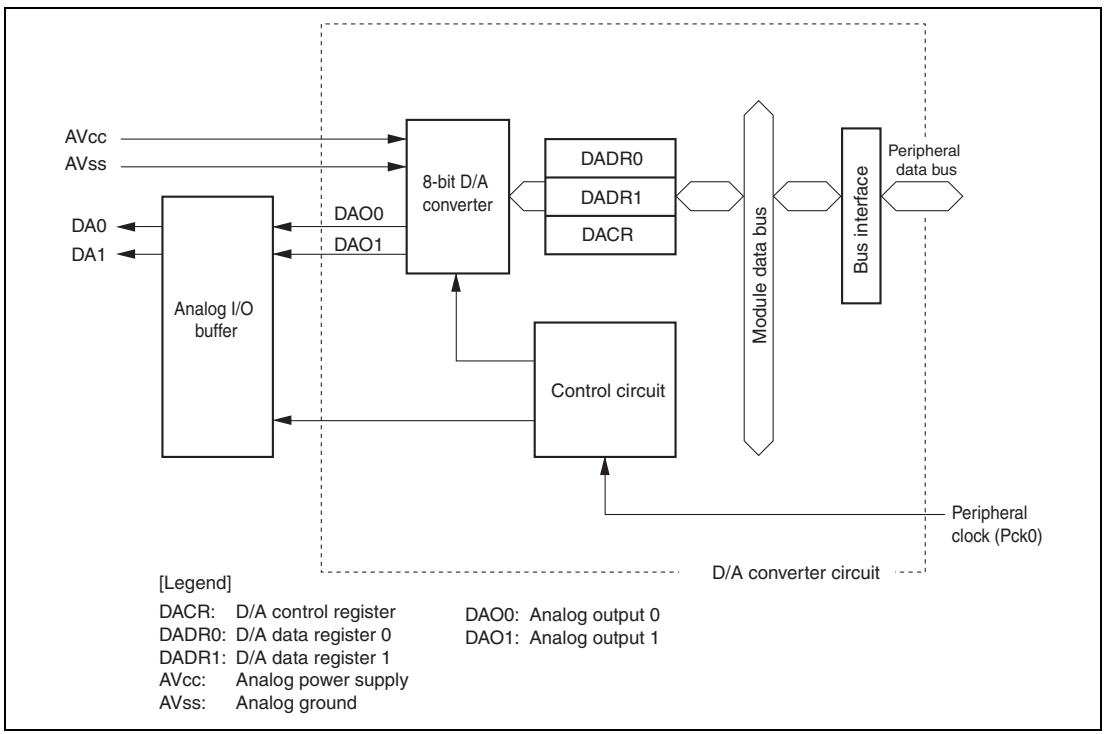


Figure 39.1 Block Diagram of D/A Converter

39.2 Input/Output Pins

Table 39.1 summarizes the input/output pins used by the D/A converter.

Table 39.1 Pin Configuration

Pin Name	I/O	Function
AVcc	—	Analog block power supply and D/A conversion reference voltage
AVss	—	Analog block ground
DA0	Output	Channel 0 analog output
DA1	Output	Channel 1 analog output

39.3 Register Descriptions

Table 39.2 shows the ADC register configuration. Table 39.3 shows the register state in each operating mode.

Table 39.2 Register Configuration

Register Name	Abbreviation	R/W	Area P4 Address*	Area 7 Address*	Access Size
D/A data register 0	DADR0	R/W	H'FFEA 8000	H'1FEA 8000	8
D/A data register 1	DADR1	R/W	H'FFEA 8002	H'1FEA 8002	8
D/A control register	DACR	R/W	H'FFEA 8004	H'1FEA 8004	8

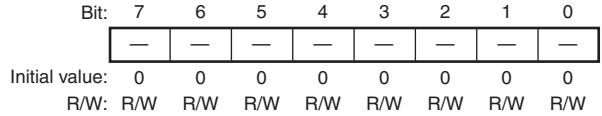
Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 39.3 Register State in Each Operating Mode

Register Name	Abbreviation	Power-On Reset	Manual Reset	Sleep	Standby
D/A data register 0	DADR0	H'00	H'00	Retained	Retained
D/A data register 1	DADR1	H'00	H'00	Retained	Retained
D/A control register	ADCR	H'3F	H'3F	Retained	Retained

39.3.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data for D/A conversion. When the D/A output enable bits (DAOE1, DAOE0) of the DA control register (DACR) are set to 1, the contents of the D/A data register are converted and output to analog output pins (DA0, DA1). The D/A data register is initialized to H'00 at reset. Note that the D/A data register is not initialized upon entering the software standby, module standby, or hardware standby mode.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	H'00	R/W	8-bit registers that store data for D/A conversion.

39.3.2 D/A Control Register (DACR)

The DACR register is an 8-bit readable/writable register that controls D/A converter operation. The DACR is initialized to H'3F at reset. Note that the DACR is not initialized in software standby, module standby, or hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	—	—	—	—	—	—
Initial value:	0	0	1	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	Controls D/A conversion for channel 1 and analog output. 0: D/A conversion for channel 1 and analog output (DA1) are disabled 1: D/A conversion for channel 1 and analog output (DA1) are enabled
6	DAOE0	0	R/W	Controls D/A conversion for channel 0 and analog output. 0: D/A conversion for channel 0 and analog output (DA0) are disabled 1: D/A conversion for channel 0 and analog output (DA0) are enabled
5 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1. If 0 is written to these bits, correct operation cannot be guaranteed.

39.4 Operation

The D/A converter incorporates two D/A channels that can operate individually.

The D/A converter executes D/A conversion while analog output is enabled by the D/A control register (DACR). If the D/A data registers (DADR0 and DADR1) are modified, the D/A converter immediately initiates the new data conversion. When the DAOE1 and DAOE0 bits in the DACR register are set to 1, D/A conversion results are output.

An example of D/A conversion for channel 0 is shown below. The operation timing is shown in figure 39.2.

1. Write conversion data to DADR0.
2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. The results are output after the conversion has ended. The output value will be (DADR0 contents/256) × AVcc.

   The conversion results are output continuously until DADR0 is modified or the DAOE0 bit is cleared to 0.
3. When D/A data register 0 (DMDR0) is modified, the conversion starts again.

   The results are output after the conversion has ended.
4. When the DAOE0 bit is cleared to 0, analog output is disabled (high-impedance state).

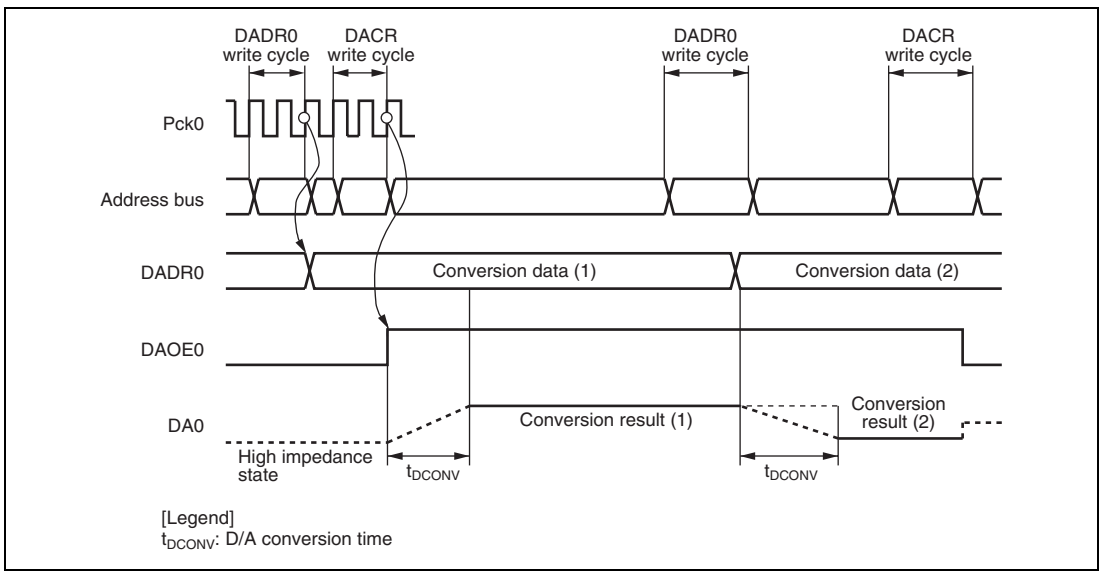


Figure 39.2 D/A Converter Operation Example

39.5 Usage Notes

The input voltages AVcc and AVss of the analog power supply should be as follows:

AVcc = 3.3 ± 0.3 V
AVss = Vss

Section 40 General Purpose I/O (GPIO)

40.1 Features

This LSI has 16 general ports (A to P), which provide 118 input/output pins in total.

Each of the general port pins is multiplexed with the pins of peripheral modules, and its function is selected as either a General Purpose I/O (GPIO) pin or a peripheral module pin.

The GPIO has the following features.

- Each port pin is a multiplexed pin, for which the port control register can set the pin function and MOS pull-up control individually.
- Each port has a data register that stores data for its pins.
- GPIO interrupts are supported.

Table 40.1 lists the multiplexed pins controlled by the GPIO registers.

Each port pin is pulled up in the initial state. When using peripheral modules, the relevant pins should be released from the pulled up state for use.

Table 40.1 Multiplexed Pins Controlled by Port Control Registers

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
A	PTA6 Input/output (Port)	AD1 TRI (PCIC)	MMC_VDDON Output (MMC)	—	—	—	—
	PTA5 Input/output (Port)	AD12 TRI (PCIC)	—	—	—	—	—
	PTA4 Input/output (Port)	AD13 TRI (PCIC)	—	$\overline{\text{SCIF1_RTS}}$ Input/output (SCIF1)	—	—	—
	PTA3 Input/output (Port)	AD15 TRI (PCIC)	—	$\overline{\text{SCIF1_CTS}}$ Input/output (SCIF1)	—	—	—
	PTA2 Input/output (Port)	$\overline{\text{LOCK}}$ STRI (PCIC)	—	SCIF1_TXD Output (SCIF1)	—	—	—
	PTA1 Input/output (Port)	$\overline{\text{DEVSEL}}$ STRI (PCIC)	—	SCIF1_RXD Input (SCIF1)	—	—	—
	PTA0 Input/output (Port)	PAR TRI (PCIC)	—	SCIF1_SCK Input/output (SCIF1)	—	—	—
B	PTB7 Input/output (Port)	AD6 TRI (PCIC)	—	—	—	LCDM_D2 Output (LCDCM)	PINT15 Input (INT)
	PTB6 Input/output (Port)	CBE0 TRI (PCIC)	—	—	—	LCDM_D3 Output (LCDCM)	PINT14 Input (INT)
	PTB5 Input/output (Port)	AD14 TRI (PCIC)	—	—	—	LCDM_M_DISP Output (LCDCM)	PINT13 Input (INT)
	PTB4 Input/output (Port)	CBE1 TRI (PCIC)	—	—	—	LCDM_D8 Output (LCDCM)	PINT12 Input (INT)

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
B	PTB3 Input/output (Port)	AD9 TRI (PCIC)	—	—	—	LCDM_D6 Output (LCDCM)	PINT11 Input (INT)
	PTB2 Input/output (Port)	AD11 TRI (PCIC)	—	—	—	LCDM_D7 Output (LCDCM)	PINT10 Input (INT)
	PTB1 Input/output (Port)	$\overline{\text{SERR}}$ O/D (PCIC)	—	—	—	LCDM_D9 Output (LCDCM)	PINT9 Input (INT)
	PTB0 Input/output (Port)	$\overline{\text{PERR}}$ STRI (PCIC)	—	—	—	LCDM_D10 Output (LCDCM)	PINT8 Input (INT)
C	PTC7 Input/output (Port)	AD3 TRI (PCIC)	MMC_CLK Output (MMC)	—	—	—	—
	PTC6 Input/output (Port)	AD5 TRI (PCIC)	—	—	—	LCDM_CL1 Output (LCDCM)	—
	PTC5 Input/output (Port)	AD0 TRI (PCIC)	MMC_CD Input (MMC)	—	—	LCDM_FLM Output (LCDCM)	—
	PTC4 Input/output (Port)	AD7 TRI (PCIC)	MMC_CMD Input/output (MMC)	—	—	LCDM_CL2 Output (LCDCM)	—
	PTC3 Input/output (Port)	AD8 TRI (PCIC)	$\overline{\text{MMC_ODMOD}}$ Output (MMC)	—	—	LCDM_D4 Output (LCDCM)	—
	PTC2 Input/output (Port)	AD2 TRI (PCIC)	—	—	—	LCDM_D0 Output (LCDCM)	—
	PTC1 Input/output (Port)	AD4 TRI (PCIC)	—	—	—	LCDM_D1 Output (LCDCM)	—
	PTC0 Input/output (Port)	AD10 TRI (PCIC)	MMC_DAT Input/output (MMC)	—	—	LCDM_D5 Output (LCDCM)	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
D	PTD7 Output (Port)	$\overline{\text{PCIRESET}}$ Output (PCIC)	PCC_RESET Output (PCC)	GET1_ETXD7 Output (GMII1)	—	LCDM_VEPWC Output (LCDCM)	—
	PTD6 Input/output (Port)	$\overline{\text{REQ2}}$ Input (PCIC)	PCC_BVD1 Input (PCC)	GET1_ETXD5 Output (GMII1)	SSI1_SCK Input/output (SSI1)	LCDM_VCPWC Output (LCDCM)	—
	PTD5 Input/output (Port)	AD18 TRI (PCIC)	$\overline{\text{PCC_CD2}}$ Input (PCC)	GET1_ERXD6 Input (GMII1)	SSI1_SDATA Input/output (SSI1)	LCDM_D14 Output (LCDCM)	—
	PTD4 Input/output (Port)	$\overline{\text{STOP}}$ STRI (PCIC)	$\overline{\text{PCC_CD1}}$ Input (PCC)	SIOF0_MCLK Input (SIOF0)	SSI1_WS Input/output (SSI1)	LCDM_DON Output (LCDCM)	—
	PTD3 Input/output (Port)	$\overline{\text{PCIFRAME}}$ STRI (PCIC)	PCC_BVD2 Input (PCC)	SIOF0_SCK Input/output (SIOF0)	$\overline{\text{HAC_RES}}$ Output (HAC)	LCDM_D12 Output (LCDCM)	—
	PTD2 Input/output (Port)	$\overline{\text{TRDY}}$ STRI (PCIC)	PCC_RDY Input (PCC)	SIOF0_RXD Input (SIOF0)	HAC_SYNC Output (HAC)	LCDM_D11 Output (LCDCM)	—
	PTD1 Input/output (Port)	CBE2 TRI (PCIC)	$\overline{\text{PCC_VS2}}$ Input (PCC)	SIOF0_TXD Output (SIOF0)	HAC_SD_OUT Output (HAC)	LCDM_D15 Output (LCDCM)	—
	PTD0 Input/output (Port)	$\overline{\text{IRDY}}$ STRI (PCIC)	$\overline{\text{PCC_VS1}}$ Input (PCC)	SIOF0_SYNC Input/output (SIOF0)	HAC_SD_IN Input (HAC)	LCDM_D13 Output (LCDCM)	—
E	PTE5 Input/output (Port)	AD29 TRI (PCIC)	SCIF2_TXD Output (SCIF2)	GET1_ GTX–CLK Output (GMII1)	SSI0_SCK Input/output (SSI0)	—	—
	PTE4 Input/output (Port)	AD22 TRI (PCIC)	SCIF2_RXD Input (SCIF2)	GET1_ERXD4 Input (GMII1)	SSI0_SDATA Input/output (SSI0)	—	—
	PTE3 Input/output (Port)	AD20 TRI (PCIC)	SCIF2_SCK Input/output (SCIF2)	GET1_ERXD5 Input (GMII1)	SSI0_WS Input/output (SSI0)	—	—
	PTE2 Input/output (Port)	AD16 TRI (PCIC)	$\overline{\text{PCC_IOIS16}}$ Input (PCC)	GET1_ERXD7 Input (GMII1)	$\overline{\text{TEND2}}$ Output (DMAC2)	—	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
E	PTE1 Input/output (Port)	PCICLK Input (PCIC)	—	GET1_ETXD4 Output (GMII1)	$\overline{\text{DACK2}}$ Output (DMAC2)	—	—
	PTE0 Input/output (Port)	$\overline{\text{INTA}}$ O/D (PCIC)	$\overline{\text{PCC_DRV}}$ Output (PCC)	GET1_ETXD6 Output (GMII1)	$\overline{\text{DREQ2}}$ Input (DMAC2)	—	—
F	PTF3 Input/output (Port)	CBE3 TRI (PCIC)	—	ET1_TX–CLK Input (MII1)	—	—	—
	PTF2 Input/output (Port)	AD31 TRI (PCIC)	SIM_RST Output (SIM)	ET1_MDIO Input/output (MII1)	$\overline{\text{TEND3}}$ Output (DMAC3)	—	—
	PTF1 Input/output (Port)	$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$ TRI (PCIC)	SIM_CLK Output (SIM)	ET1_MDC Output (MII1)	$\overline{\text{DACK3}}$ Output (DMAC3)	—	—
	PTF0 Input/output (Port)	$\overline{\text{GNT0/GNTIN}}$ TRI (PCIC)	SIM_D Input/output (SIM)	ET1_ETXD3 Output (MII1)	$\overline{\text{DREQ3}}$ Input (DMAC3)	—	—
G	PTG7 Input/output (Port)	AD28 TRI (PCIC)	—	ET1_TX–EN Output (MII1)	—	—	—
	PTG6 Input/output (Port)	AD26 TRI (PCIC)	—	ET1_TX–ER Output (MII1)	—	—	—
	PTG5 Input/output (Port)	$\overline{\text{GNT3}}$ TRI (PCIC)	—	ET1_RX–CLK Input (MII1)	—	—	—
	PTG4 Input/output (Port)	AD30 TRI (PCIC)	—	ET1_LINKST A Input (MII1)	—	—	—
	PTG3 Input/output (Port)	$\overline{\text{REQ3}}$ Input (PCIC)	—	ET1_ETXD2 Output (MII1)	—	—	—
	PTG2 Input/output (Port)	$\overline{\text{REQ1}}$ Input (PCIC)	—	ET1_ETXD1 Output (MII1)	—	—	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
G	PTG1 Input/output (Port)	$\overline{\text{GNT2}}$ TRI (PCIC)	—	ET1_ETXD0 Output (MII1)	—	—	—
	PTG0 Input/output (Port)	$\overline{\text{GNT1}}$ TRI (PCIC)	—	ET1_WOL Output (MII1)	—	—	—
H	PTH7 Input/output (Port)	AD17 TRI (PCIC)	TPU_TO3 Output (TPU)	ET1_RX–DV Input (MII1)	—	—	—
	PTH6 Input/output (Port)	AD27 TRI (PCIC)	TPU_TO2 Output (TPU)	ET1_CRS Input (MII1)	RMII1M_TXD_ EN Output (RMII1M)	—	—
	PTH5 Input/output (Port)	AD23 TRI (PCIC)	TPU_TO1 Output (TPU)	ET1_ERXD1 Input (MII1)	RMII1M_TXD0 Output (RMII1M)	—	—
	PTH4 Input/output (Port)	AD19 TRI (PCIC)	TPU_TO0 Output (TPU)	ET1_ERXD3 Input (MII1)	RMII1M_RXD0 Input (RMII1M)	—	—
	PTH3 Input/output (Port)	AD21 TRI (PCIC)	TPU_TI2B Input (TPU)	ET1_ERXD2 Input (MII1)	RMII1M_RXD1 Input (RMII1M)	—	—
	PTH2 Input/output (Port)	AD24 TRI (PCIC)	TPU_TI2A Input (TPU)	ET1_ERXD0 Input (MII1)	RMII1M_TXD1 Output (RMII1M)	—	—
	PTH1 Input/output (Port)	IDSEL Input (PCIC)	TPU_TI3B Input (TPU)	ET1_RX—ER Input (MII1)	RMII1M_CRS_ DV Input (RMII1M)	—	—
	PTH0 Input/output (Port)	AD25 TRI (PCIC)	TPU_TI3A Input (TPU)	ET1_COL Input (MII1)	RMII1M_RX_ ER Input (RMII1M)	—	—
I	PTI7 Input (Port)	IRQ3/$\overline{\text{IRL3}}$ Input (INT)	ST0M_D7I Input (STIF0M)	IIC1_SDA Input/output (IIC1)	—	—	—
	PTI6 Input (Port)	IRQ2/$\overline{\text{IRL2}}$ Input (INT)	ST0M_D6I Input (STIF0M)	IIC1_SCL Input/output (IIC1)	—	—	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
I	PTI5 Input/output (Port)	MD10*[1] Input (EXCPU)	ST1_VALID Input/output (STIF1)	—	—	LCD_D1 Output (LCDC)	—
	PTI4 Input/output (Port)	MD8*[1] Input (CPG)	ST1_START Input/output (STIF1)	ET1_PHY–INT Input (ETC1)	RMII0M0_ MDC Output (RMII0M0)	USB_PWREN/ USBF_UPLUP Output (USBH/F)	—
	PTI3 Input (Port)	—	ST0M_VALIDI Input (STIF0M)	IIC0_SDA Input/output (IIC0)	SIOF1_MCLK Input (SIOF1)	USB_CLK Input (USBH/F)	—
	PTI2 Input (Port)	—	ST0M_STARTI Input (STIF0M)	IIC0_SCL Input/output (IIC0)	SIOF1_RXD Input (SIOF1)	$\overline{\text{USB_}}$ $\overline{\text{OVERCRT}}$/ USBF_VBUS Input (USBH/F)	—
	PTI1 Input/output (Port)	STATUS1 Output (–)	ST1_REQ Input/output (STIF1)	RMII0_MDIO Input/output (RMII0)	—	—	—
	PTI0 Input/output (Port)	STATUS0 Output (–)	ST1_CLK/ ST1_STRB Input/output (STIF1)	RMII0_MDC Output (RMII0)	—	—	—
J	PTJ7 Input/output (Port)	$\overline{\text{INTB}}$ Input (PCIC)	ST0M_D5I Input (STIF0M)	$\overline{\text{IRQOUT}}$ Output (INT)	RMII1_TXD0 Output (RMII1)	LCD_D0 Output (LCDC)	—
	PTJ6 Input/output (Port)	—	ST0M_D4I Input (STIF0M)	ET0_CRS Input (MII0)	RMII1_TXD_ EN Output (RMII1)	LCD_FLM Output (LCDC)	—
	PTJ5 Input/output (Port)	—	ST0M_D3I Input (STIF0M)	ET0_ERXD3 Input (MII0)	RMII1_RXD0 Input (RMII1)	LCD_DON Output (LCDC)	—
	PTJ4 Input/output (Port)	—	ST0M_D2I Input (STIF0M)	ET0_ERXD2 Input (MII0)	RMII1_RXD1 Input (RMII1)	LCD_CL2 Output (LCDC)	—
	PTJ3 Input/output (Port)	—	ST0M_D1I Input (STIF0M)	ET0_ERXD1 Input (MII0)	RMII1_CRS_ DV Input (RMII1)	LCD_CL1 Output (LCDC)	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
J	PTJ2 Input/output (Port)	—	ST0M_D0I Input (STIF0M)	ET0_ERXD0 Input (MII0)	RMII1_TXD1 Output (RMII1)	LCD_M_DISP Output (LCDC)	—
	PTJ1 Input/output (Port)	—	ST0M_CLKIO/ ST0M_STRBI Input/output (STIF0M)	—	RMII1_RX_ER Input (RMII1)	LCD_CLK Input (LCDC)	—
	PTJ0 Input/output (Port)	—	ST0M_REQO Output (STIF0M)	GET0_GTX_ CLK Output (GMII0)	REF50CK Input (RMII0/1)	—	—
K	PTK7 Input/output (Port)	—	ST1_D7 Input/output (STIF1)	GET0_ERXD7 Input (GMII0)	SIOF2_MCLK Input (SIOF2)	LCD_VCPWC Output (LCDC)	—
	PTK6 Input/output (Port)	—	ST1_D6 Input/output (STIF1)	GET0_ERXD6 Input (GMII0)	SIOF2_SCK Input/output (SIOF2)	LCD_VEPWC Output (LCDC)	—
	PTK5 Input/output (Port)	—	ST1_D5 Input/output (STIF1)	GET0_ERXD5 Input (GMII0)	SIOF2_RXD Input (SIOF2)	LCD_D7 Output (LCDC)	—
	PTK4 Input/output (Port)	—	ST1_D4 Input/output (STIF1)	GET0_ERXD4 Input (GMII0)	SIOF2_TXD Output (SIOF2)	LCD_D6 Output (LCDC)	—
	PTK3 Input/output (Port)	—	ST1_D3 Input/output (STIF1)	GET0_ETXD7 Output (GMII0)	SIOF2_SYNC Input/output (SIOF2)	LCD_D5 Output (LCDC)	—
	PTK2 Input/output (Port)	—	ST1_D2 Input/output (STIF1)	GET0_ETXD6 Output (GMII0)	SIOF1_SCK Input/output (SIOF1)	LCD_D4 Output (LCDC)	—
	PTK1 Input/output (Port)	—	ST1_D1 Input/output (STIF1)	GET0_ETXD5 Output (GMII0)	SIOF1_TXD Output (SIOF1)	LCD_D3 Output (LCDC)	—
	PTK0 Input/output (Port)	—	ST1_D0 Input/output (STIF1)	GET0_ETXD4 Output (GMII0)	SIOF1_SYNC Input/output (SIOF1)	LCD_D2 Output (LCDC)	—
L	PTL7 Input/output (Port)	D23/EX_AD23 Input/output (LBSC*2/ EXCPU)	ST0_VALID Input/output (STIF0)	ET0_TX–EN Output (MII0)	$\overline{\text{TEND1}}$ Output (DMAC1)	LCD_D15 Output (LCDC)	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
L	PTL6 Input/output (Port)	D22/EX_AD22 Input/output (LBSC*2/EXCPU)	ST0_START Input/output (STIF0)	ET0_ETXD2 Output (MII0)	$\overline{\text{DACK1}}$ Output (DMAC1)	LCD_D14 Output (LCDC)	—
	PTL5 Input/output (Port)	D21/EX_AD21 Input/output (LBSC*2/EXCPU)	ST0_CLK/ ST0_STRB Input/output (STIF0)	ET0_ETXD1 Output (MII0)	$\overline{\text{DREQ1}}$ Input (DMAC1)	LCD_D13 Output (LCDC)	—
	PTL4 Input/output (Port)	D20/EX_AD20 Input/output (LBSC*2/EXCPU)	ST0_REQ Input/output (STIF0)	ET0_ETXD0 Output (MII0)	$\overline{\text{INTD}}$ Input (PCIC)	LCD_D12 Output (LCDC)	—
	PTL3 Input/output (Port)	D19/EX_AD19 Input/output (LBSC*2/EXCPU)	IRQ7/$\overline{\text{IRL7}}$ Input (INT)	ET0_MDIO Input/output (MII0)	$\overline{\text{INTC}}$ Input (PCIC)	LCD_D11 Output (LCDC)	—
	PTL2 Input/output (Port)	D18/EX_AD18 Input/output (LBSC*2/EXCPU)	IRQ6/$\overline{\text{IRL6}}$ Input (INT)	ET0_ETXD3 Output (MII0)	$\overline{\text{TEND0}}$ Output (DMAC0)	LCD_D10 Output (LCDC)	—
	PTL1 Input/output (Port)	D17/EX_AD17 Input/output (LBSC*2/EXCPU)	IRQ5/$\overline{\text{IRL5}}$ Input (INT)	ET0_MDC Output (MII0)	$\overline{\text{DACK0}}$ Output (DMAC0)	LCD_D9 Output (LCDC)	—
	PTL0 Input/output (Port)	D16/EX_AD16 Input/output (LBSC*2/EXCPU)	IRQ4/$\overline{\text{IRL4}}$ Input (INT)	ET0_COL Input (MII0)	$\overline{\text{DREQ0}}$ Input (DMAC0)	LCD_D8 Output (LCDC)	—
M	PTM7 Input/output (Port)	D31/EX_AD31 Input/output (LBSC*2/EXCPU)	ST0_D7 Input/output (STIF0)	ET0_RX–DV Input (MII0)	RMII0_TXD0 Output (RMII0)	—	PINT7 Input (INT)
	PTM6 Input/output (Port)	D30/EX_AD30 Input/output (LBSC*2/EXCPU)	ST0_D6 Input/output (STIF0)	ET0_RX–CLK Input (MII0)	RMII0_TXD1 Output (RMII0)	—	PINT6 Input (INT)
	PTM5 Input/output (Port)	D29/EX_AD29 Input/output (LBSC*2/EXCPU)	ST0_D5 Input/output (STIF0)	ET0_RX–ER Input (MII0)	RMII0_TXD_ EN Output (RMII0)	—	PINT5 Input (INT)
	PTM4 Input/output (Port)	D28/EX_AD28 Input/output (LBSC*2/EXCPU)	ST0_D4 Input/output (STIF0)	ET0_PHY–INT Input (ETC0)	RMII0_RXD0 Input (RMII0)	—	PINT4 Input (INT)
	PTM3 Input/output (Port)	D27/EX_AD27 Input/output (LBSC*2/EXCPU)	ST0_D3 Input/output (STIF0)	ET0_LINKSTA Input (MII0)	RMII0_RXD1 Input (RMII0)	—	PINT3 Input (INT)

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
M	PTM2 Input/output (Port)	D26/EX_AD26 Input/output (LBSC*2/EXCPU)	ST0_D2 Input/output (STIF0)	ET0_WOL Output (MII0)	RMII0_CRS_ DV Input (RMII0)	—	PINT2 Input (INT)
	PTM1 Input/output (Port)	D25/EX_AD25 Input/output (LBSC*2/EXCPU)	ST0_D1 Input/output (STIF0)	ET0_TX–CLK Input (MII0)	RMII0_RX_ ER Input (RMII0)	—	PINT1 Input (INT)
	PTM0 Input/output (Port)	D24/EX_AD24 Input/output (LBSC*2/EXCPU)	ST0_D0 Input/output (STIF0)	ET0_TX–ER Output (MII0)	—	RMII0M0_ MDIO Input/output (RMII0M0)	PINT0 Input (INT)
N	PTN5 Input/output (Port)	NMI Input (INT)	—	—	—	—	—
	PTN4 Input/output (Port)	$\overline{\text{SCIF0_RTS}}$ Input/output (SCIF0)	MD2*1 Input (CPG)	—	—	—	—
	PTN3 Input/output (Port)	$\overline{\text{SCIF0_CTS}}$ Input/output (SCIF0)	MD4*1 Input (LBSC)	—	—	—	—
	PTN2 Input/output (Port)	SCIF0_TXD Output (SCIF0)	MD1*1 Input (CPG)	—	—	—	—
	PTN1 Input/output (Port)	SCIF0_RXD Input (SCIF0)	MD3*1 Input (LBSC)	—	—	—	—
	PTN0 Input/output (Port)	SCIF0_SCK Input/output (SCIF0)	MD0*1 Input (CPG)	—	—	—	—
O	PTO7 Input/output (Port)	IRQ1/$\overline{\text{IRL1}}$ Input (INT)	$\overline{\text{TEND1M}}$ Output (DMAC1M)	SSI3_SCK Input/output (SSI3)	MD6*1 Input (PCIC)	—	—
	PTO6 Input/output (Port)	IRQ0/$\overline{\text{IRL0}}$ Input (INT)	$\overline{\text{DACK1M}}$ Output (DMAC1M)	—	MD5*1 Input (LBSC)	—	—

Port	Port Function (Related module)	Other Function 1 (Related module)	Other Function 2 (Related module)	Other Function 3 (Related module)	Other Function 4 (Related module)	Other Function 5 (Related module)	GPIO Interrupt
O	PTO5 Input/output (Port)	AUDCK Output (AUD)	$\overline{\text{DREQ1M}}$ Input (DMAC1M)	SSI3_SDATA Input/output (SSI3)	—	—	—
	PTO4 Input/output (Port)	AUDATA3 Output (AUD)	$\overline{\text{EX_INT}}$ Output (EXCPU)	SSI3_WS Input/output (SSI3)	—	—	—
	PTO3 Input/output (Port)	AUDATA2 Output (AUD)	RMII0M1_ MDIO Input/output (RMII0M1)	SSI2_SCK Input/output (SSI2)	—	—	—
	PTO2 Input/output (Port)	AUDATA1 Output (AUD)	RMII0M1_ MDC Output (RMII0M1)	—	—	—	—
	PTO1 Input/output (Port)	AUDATA0 Output (AUD)	RMII1_MDIO Input/output (RMII1)	SSI2_SDATA Input/output (SSI2)	—	—	—
	PTO0 Input/output (Port)	AUDSYNC Output (AUD)	RMII1_MDC Output (RMII1)	SSI2_WS Input/output (SSI2)	—	—	—

[Legend] TRI: Tri-state
STRI: Sustained tri-state
O/D: Open drain

Note: 1. Hatched areas in the table indicate the pin functions that are ready for use immediately after a reset.
2. When using on-chip modules. select functions for the relevant pins before initializing each module.
3. MD0 to MD6, MD8, and MD10 are valid only during power-on reset.
4. When 32-bit is selected as the data bus width by the LBSC, select this pin function.

40.2 Register Descriptions

Table 40.2 shows the GPIO register configuration. Table 40.3 shows the register states in each operating mode.

Table 40.2 Register Configuration (1)

Register Name	Abbreviation	R/W	Area P4 Address*[1]	Area 7 Address*[1]	Access Size*[2]
Port A control register	PACR	R/W	H'FFEF 0000	H'1FEF 0000	16
Port B control register	PBCR	R/W	H'FFEF 0002	H'1FEF 0002	16
Port C control register	PCCR	R/W	H'FFEF 0004	H'1FEF 0004	16
Port D control register	PDCR	R/W	H'FFEF 0006	H'1FEF 0006	16
Port E control register	PECR	R/W	H'FFEF 0008	H'1FEF 0008	16
Port F control register	PFCR	R/W	H'FFEF 000A	H'1FEF 000A	16
Port G control register	PGCR	R/W	H'FFEF 000C	H'1FEF 000C	16
Port H control register	PHCR	R/W	H'FFEF 000E	H'1FEF 000E	16
Port I control register	PICR	R/W	H'FFEF 0010	H'1FEF 0010	16
Port J control register	PJCR	R/W	H'FFEF 0012	H'1FEF 0012	16
Port K control register	PKCR	R/W	H'FFEF 0014	H'1FEF 0014	16
Port L control register	PLCR	R/W	H'FFEF 0016	H'1FEF 0016	16
Port M control register	PMCR	R/W	H'FFEF 0018	H'1FEF 0018	16
Port N control register	PNCR	R/W	H'FFEF 001A	H'1FEF 001A	16
Port O control register	POCR	R/W	H'FFEF 001C	H'1FEF 001C	16
Port A data register	PADR	R/W	H'FFEF 0020	H'1FEF 0020	8
Port B data register	PBDR	R/W	H'FFEF 0022	H'1FEF 0022	8
Port C data register	PCDR	R/W	H'FFEF 0024	H'1FEF 0024	8
Port D data register	PDDR	R/W	H'FFEF 0026	H'1FEF 0026	8
Port E data register	PEDR	R/W	H'FFEF 0028	H'1FEF 0028	8
Port F data register	PFDR	R/W	H'FFEF 002A	H'1FEF 002A	8
Port G data register	PGDR	R/W	H'FFEF 002C	H'1FEF 002C	8
Port H data register	PHDR	R/W	H'FFEF 002E	H'1FEF 002E	8
Port I data register	PIDR	R/W	H'FFEF 0030	H'1FEF 0030	8
Port J data register	PJDR	R/W	H'FFEF 0032	H'1FEF 0032	8

Register Name	Abbreviation	R/W	Area P4 Address*[1]	Area 7 Address*[1]	Access Size*[2]
Port K data register	PKDR	R/W	H'FFEF 0034	H'1FEF 0034	8
Port L data register	PLDR	R/W	H'FFEF 0036	H'1FEF 0036	8
Port M data register	PMDR	R/W	H'FFEF 0038	H'1FEF 0038	8
Port N data register	PNDR	R/W	H'FFEF 003A	H'1FEF 003A	8
Port O data register	PODR	R/W	H'FFEF 003C	H'1FEF 003C	8
Port I pull-up control register	PIPUPR	R/W	H'FFEF 0050	H'1FEF 0050	8
Port J pull-up control register	PJPUPR	R/W	H'FFEF 0052	H'1FEF 0052	8
Port K pull-up control register	PKPUPR	R/W	H'FFEF 0054	H'1FEF 0054	8
Port L pull-up control register	PLPUPR	R/W	H'FFEF 0056	H'1FEF 0056	8
Port M pull-up control register	PMPUPR	R/W	H'FFEF 0058	H'1FEF 0058	8
Port N pull-up control register	PNPUPR	R/W	H'FFEF 005A	H'1FEF 005A	8
Port O pull-up control register	POPUPR	R/W	H'FFEF 005C	H'1FEF 005C	8
Input pin pull-up control register	PPUPR	R/W	H'FFEF 0060	H'1FEF 0060	8
Pin select register 0	PSEL0	R/W	H'FFEF 0070	H'1FEF 0070	16
Pin select register 1	PSEL1	R/W	H'FFEF 0072	H'1FEF 0072	16
Pin select register 2	PSEL2	R/W	H'FFEF 0074	H'1FEF 0074	16
Pin select register 3	PSEL3	R/W	H'FFEF 0076	H'1FEF 0076	16
Pin select register 4	PSEL4	R/W	H'FFEF 0078	H'1FEF 0078	16

Notes: 1. Area P4 address is the address when the P4 area of virtual address space is used. Area 7 address is the address when the register is accessed through area 7 of physical address space by using the TLB.

2. There are 16-bit access registers and 8-bit access registers. These registers should be read or written in their specified access size.

Table 40.3 Register States in Each Operating Mode

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Port A control register	PACR	H'0000	Retained	Retained	Retained
Port B control register	PBCR	H'0000	Retained	Retained	Retained
Port C control register	PCCR	H'0000	Retained	Retained	Retained
Port D control register	PDCR	H'0000	Retained	Retained	Retained
Port E control register	PECR	H'0000	Retained	Retained	Retained
Port F control register	PFCR	H'0000	Retained	Retained	Retained
Port G control register	PGCR	H'0000	Retained	Retained	Retained
Port H control register	PHCR	H'0000	Retained	Retained	Retained
Port I control register	PICR	H'0AA0	Retained	Retained	Retained
Port J control register	PJCR	H'FFFF	Retained	Retained	Retained
Port K control register	PKCR	H'FFFF	Retained	Retained	Retained
Port L control register	PLCR	H'0000	Retained	Retained	Retained
Port M control register	PMCR	H'0000	Retained	Retained	Retained
Port N control register	PNCR	H'02AA	Retained	Retained	Retained
Port O control register	POCR	H'0FFF	Retained	Retained	Retained
Port A data register	PADR	H'00	Retained	Retained	Retained
Port B data register	PBDR	H'00	Retained	Retained	Retained
Port C data register	PCDR	H'00	Retained	Retained	Retained
Port D data register	PDDR	H'00	Retained	Retained	Retained
Port E data register	PEDR	H'00	Retained	Retained	Retained
Port F data register	PFDR	H'00	Retained	Retained	Retained
Port G data register	PGDR	H'00	Retained	Retained	Retained
Port H data register	PHDR	H'00	Retained	Retained	Retained
Port I data register	PIDR	H'xx	Retained	Retained	Retained
Port J data register	PJDR	H'xx	Retained	Retained	Retained
Port K data register	PKDR	H'xx	Retained	Retained	Retained
Port L data register	PLDR	H'00	Retained	Retained	Retained
Port M data register	PMDR	H'00	Retained	Retained	Retained
Port N data register	PNDR	H'xx	Retained	Retained	Retained
Port O data register	PODR	H'xx	Retained	Retained	Retained

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Port I pull-up control register	PIPUPR	H'FF	Retained	Retained	Retained
Port J pull-up control register	PJPUPR	H'FF	Retained	Retained	Retained
Port K pull-up control register	PKPUPR	H'FF	Retained	Retained	Retained
Port L pull-up control register	PLPUPR	H'FF	Retained	Retained	Retained
Port M pull-up control register	PMPUPR	H'FF	Retained	Retained	Retained
Port N pull-up control register	PNPUPR	H'FF	Retained	Retained	Retained
Port O pull-up control register	POPUPR	H'FF	Retained	Retained	Retained
Input pin pull-up control register	PPUPR	H'FF	Retained	Retained	Retained
Pin select register 0	PSEL0	H'0008	Retained	Retained	Retained
Pin select register 1	PSEL1	H'4888	Retained	Retained	Retained
Pin select register 2	PSEL2	H'0000	Retained	Retained	Retained
Pin select register 3	PSEL3	H'4444	Retained	Retained	Retained
Pin select register 4	PSEL4	H'0000	Retained	Retained	Retained

40.2.1 Port A Control Register (PACR)

PACR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PA6MD[1:0]		PA5MD[1:0]		PA4MD[1:0]		PA3MD[1:0]		PA2MD[1:0]		PA1MD[1:0]		PA0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PA6MD[1:0]	00	R/W	PTA6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
11, 10	PA5MD[1:0]	00	R/W	PTA5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PA4MD[1:0]	00	R/W	PTA4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PA3MD[1:0]	00	R/W	PTA3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PA2MD[1:0]	00	R/W	PTA2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PA1MD[1:0]	00	R/W	PTA1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PA0MD[1:0]	00	R/W	PTA0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7MD[1:0]		PB6MD[1:0]		PB5MD[1:0]		PB4MD[1:0]		PB3MD[1:0]		PB2MD[1:0]		PB1MD[1:0]		PB0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PB7MD[1:0]	00	R/W	PTB7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
13, 12	PB6MD[1:0]	00	R/W	PTB6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PB5MD[1:0]	00	R/W	PTB5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PB4MD[1:0]	00	R/W	PTB4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PB3MD[1:0]	00	R/W	PTB3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
5, 4	PB2MD[1:0]	00	R/W	PTB2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PB1MD[1:0]	00	R/W	PTB1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PB0MD[1:0]	00	R/W	PTB0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7MD[1:0]		PC6MD[1:0]		PC5MD[1:0]		PC4MD[1:0]		PC3MD[1:0]		PC2MD[1:0]		PC1MD[1:0]		PC0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PC7MD[1:0]	00	R/W	PTC7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
13, 12	PC6MD[1:0]	00	R/W	PTC6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PC5MD[1:0]	00	R/W	PTC5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PC4MD[1:0]	00	R/W	PTC4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PC3MD[1:0]	00	R/W	PTC3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PC2MD[1:0]	00	R/W	PTC2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PC1MD[1:0]	00	R/W	PTC1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
1, 0	PC0MD[1:0]	00	R/W	PTC0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7MD[1:0]		PD6MD[1:0]		PD5MD[1:0]		PD4MD[1:0]		PD3MD[1:0]		PD2MD[1:0]		PD1MD[1:0]		PD0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PD7MD[1:0]	00	R/W	PTD7 Mode 00: Other function 01: Port output 10: Setting prohibited 11: Setting prohibited
13, 12	PD6MD[1:0]	00	R/W	PTD6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PD5MD[1:0]	00	R/W	PTD5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
9, 8	PD4MD[1:0]	00	R/W	PTD4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PD3MD[1:0]	00	R/W	PTD3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PD2MD[1:0]	00	R/W	PTD2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PD1MD[1:0]	00	R/W	PTD1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PD0MD[1:0]	00	R/W	PTD0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.5 Port E Control Register (PECR)

PECR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PE5MD[1:0]		PE4MD[1:0]		PE3MD[1:0]		PE2MD[1:0]		PE1MD[1:0]		PE0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 12	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
11, 10	PE5MD[1:0]	00	R/W	PTE5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PE4MD[1:0]	00	R/W	PTE4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PE3MD[1:0]	00	R/W	PTE3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PE2MD[1:0]	00	R/W	PTE2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
3, 2	PE1MD[1:0]	00	R/W	PTE1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PE0MD[1:0]	00	R/W	PTE0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PF3MD[1:0]		PF2MD[1:0]		PF1MD[1:0]		PF0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7, 6	PF3MD[1:0]	00	R/W	PTF3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
5, 4	PF2MD[1:0]	00	R/W	PTF2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PF1MD[1:0]	00	R/W	PTF1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PF0MD[1:0]	00	R/W	PTF0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7MD[1:0]		PG6MD[1:0]		PG5MD[1:0]		PG4MD[1:0]		PG3MD[1:0]		PG2MD[1:0]		PG1MD[1:0]		PG0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PG7MD[1:0]	00	R/W	PTG7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
13, 12	PG6MD[1:0]	00	R/W	PTG6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PG5MD[1:0]	00	R/W	PTG5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PG4MD[1:0]	00	R/W	PTG4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
7, 6	PG3MD[1:0]	00	R/W	PTG3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PG2MD[1:0]	00	R/W	PTG2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PG1MD[1:0]	00	R/W	PTG1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PG0MD[1:0]	00	R/W	PTG0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7MD[1:0]		PH6MD[1:0]		PH5MD[1:0]		PH4MD[1:0]		PH3MD[1:0]		PH2MD[1:0]		PH1MD[1:0]		PH0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15,14	PH7MD[1:0]	00	R/W	PH7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
13, 12	PH6MD[1:0]	00	R/W	PH6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PH5MD[1:0]	00	R/W	PH5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PH4MD[1:0]	00	R/W	PTH4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
7, 6	PH3MD[1:0]	00	R/W	PTH3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
5, 4	PH2MD[1:0]	00	R/W	PTH2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PH1MD[1:0]	00	R/W	PTH1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PH0MD[1:0]	00	R/W	PTH0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.9 Port I Control Register (PICR)

PICR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PI7MD[1:0]		PI6MD[1:0]		PI5MD[1:0]		PI4MD[1:0]		PI3MD[1:0]		PI2MD[1:0]		PI1MD[1:0]		PI0MD[1:0]	
Initial value:	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PI7MD[1:0]	00	R/W	PTI7 Mode 00: Other function 01: Setting prohibited 10: Port input 11: Setting prohibited
13, 12	PI6MD[1:0]	00	R/W	PTI6 Mode 00: Other function 01: Setting prohibited 10: Port input 11: Setting prohibited
11, 10	PI5MD[1:0]	10	R/W	PTI5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
9, 8	PI4MD[1:0]	10	R/W	PTI4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
7, 6	PI3MD[1:0]	10	R/W	PTI3 Mode 00: Other function 01: Setting prohibited 10: Port input 11: Setting prohibited
5, 4	PI2MD[1:0]	10	R/W	PTI2 Mode 00: Other function 01: Setting prohibited 10: Port input 11: Setting prohibited
3, 2	PI1MD[1:0]	00	R/W	PTI1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PI0MD[1:0]	00	R/W	PTI0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.10 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7MD[1:0]		PJ6MD[1:0]		PJ5MD[1:0]		PJ4MD[1:0]		PJ3MD[1:0]		PJ2MD[1:0]		PJ1MD[1:0]		PJ0MD[1:0]	
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PJ7MD[1:0]	11	R/W	PTJ7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
13, 12	PJ6MD[1:0]	11	R/W	PTJ6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
11, 10	PJ5MD[1:0]	11	R/W	PTJ5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PJ4MD[1:0]	11	R/W	PTJ4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

Bit	Bit Name	Initial value	R/W	Description
7, 6	PJ3MD[1:0]	11	R/W	PTJ3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
5, 4	PJ2MD[1:0]	11	R/W	PTJ2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
3, 2	PJ1MD[1:0]	11	R/W	PTJ1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PJ0MD[1:0]	11	R/W	PTJ0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.11 Port K Control Register (PKCR)

PKCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PK7MD[1:0]		PK6MD[1:0]		PK5MD[1:0]		PK4MD[1:0]		PK3MD[1:0]		PK2MD[1:0]		PK1MD[1:0]		PK0MD[1:0]	
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PK7MD[1:0]	11	R/W	PTK7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
13, 12	PK6MD[1:0]	11	R/W	PTK6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
11, 10	PK5MD[1:0]	11	R/W	PTK5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PK4MD[1:0]	11	R/W	PTK4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

Bit	Bit Name	Initial value	R/W	Description
7, 6	PK3MD[1:0]	11	R/W	PTK3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
5, 4	PK2MD[1:0]	11	R/W	PTK2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
3, 2	PK1MD[1:0]	11	R/W	PTK1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PK0MD[1:0]	11	R/W	PTK0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.12 Port L Control Register (PLCR)

PLCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PL7MD[1:0]		PL6MD[1:0]		PL5MD[1:0]		PL4MD[1:0]		PL3MD[1:0]		PL2MD[1:0]		PL1MD[1:0]		PL0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PL7MD[1:0]	00	R/W	PTL7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
13, 12	PL6MD[1:0]	00	R/W	PTL6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
11, 10	PL5MD[1:0]	00	R/W	PTL5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PL4MD[1:0]	00	R/W	PTL4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

Bit	Bit Name	Initial value	R/W	Description
7, 6	PL3MD[1:0]	00	R/W	PTL3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
5, 4	PL2MD[1:0]	00	R/W	PTL2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
3, 2	PL1MD[1:0]	00	R/W	PTL1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PL0MD[1:0]	00	R/W	PTL0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.13 Port M Control Register (PMCR)

PMCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PM7MD[1:0]		PM6MD[1:0]		PM5MD[1:0]		PM4MD[1:0]		PM3MD[1:0]		PM2MD[1:0]		PM1MD[1:0]		PM0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PM7MD[1:0]	00	R/W	PTM7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
13, 12	PM6MD[1:0]	00	R/W	PTM6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
11, 10	PM5MD[1:0]	00	R/W	PTM5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PM4MD[1:0]	00	R/W	PTM4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

Bit	Bit Name	Initial value	R/W	Description
7, 6	PM3MD[1:0]	00	R/W	PTM3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
5, 4	PM2MD[1:0]	00	R/W	PTM2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
3, 2	PM1MD[1:0]	00	R/W	PTM1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PM0MD[1:0]	00	R/W	PTM0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.14 Port N Control Register (PNCR)

PNCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PN5MD[1:0]		PN4MD[1:0]		PN3MD[1:0]		PN2MD[1:0]		PN1MD[1:0]		PN0MD[1:0]	
Initial value:	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 12	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
11, 10	PN5MD[1:0]	00	R/W	PTN5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PN4MD[1:0]	10	R/W	PTN4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
7, 6	PN3MD[1:0]	10	R/W	PTN3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

Bit	Bit Name	Initial value	R/W	Description
5, 4	PN2MD[1:0]	10	R/W	PTN2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
3, 2	PN1MD[1:0]	10	R/W	PTN1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
1, 0	PN0MD[1:0]	10	R/W	PTN0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited

40.2.15 Port O Control Register (POCR)

POCR is a 16-bit readable/writable register that selects the pin function and MOS input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PO7MD[1:0]		PO6MD[1:0]		PO5MD[1:0]		PO4MD[1:0]		PO3MD[1:0]		PO2MD[1:0]		PO1MD[1:0]		PO0MD[1:0]	
Initial value:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15, 14	PO7MD[1:0]	00	R/W	PTO7 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
13, 12	PO6MD[1:0]	00	R/W	PTO6 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Setting prohibited
11, 10	PO5MD[1:0]	11	R/W	PTO5 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
9, 8	PO4MD[1:0]	11	R/W	PTO4 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

Bit	Bit Name	Initial value	R/W	Description
7, 6	PO3MD[1:0]	11	R/W	PTO3 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
5, 4	PO2MD[1:0]	11	R/W	PTO2 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
3, 2	PO1MD[1:0]	11	R/W	PTO1 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)
1, 0	PO0MD[1:0]	11	R/W	PTO0 Mode 00: Other function 01: Port output 10: Port input (MOS pull-up: Off) 11: Port input (MOS pull-up: On)

40.2.16 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores port A data.

Bit:	7	6	5	4	3	2	1	0
	—	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PA6DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

40.2.17 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores port B data.

Bit:	7	6	5	4	3	2	1	0
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PB7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

40.2.18 Port C Data Register (PCDR)

PCDR is an 8-bit readable/writable register that stores port C data.

Bit:	7	6	5	4	3	2	1	0
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PC7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

40.2.19 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores port D data.

Bit:	7	6	5	4	3	2	1	0
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PD7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

40.2.20 Port E Data Register (PEDR)

PEDR is an 8-bit readable/writable register that stores port E data.

Bit:	7	6	5	4	3	2	1	0
	—	—	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	All0	R	Reserved
6				These bits are always read as 0, and the write value should always be 0.
5	PE5DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

40.2.21 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores port F data.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7 to 4	—	All0	R	Reserved These bits are always read as 0, and the write value should always be 0.
3	PF3DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

40.2.22 Port G Data Register (PGDR)

PGDR is an 8-bit readable/writable register that stores port G data.

Bit:	7	6	5	4	3	2	1	0
	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PG7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PG6DT	0	R/W	
5	PG5DT	0	R/W	
4	PG4DT	0	R/W	
3	PG3DT	0	R/W	
2	PG2DT	0	R/W	
1	PG1DT	0	R/W	
0	PG0DT	0	R/W	

40.2.23 Port H Data Register (PHDR)

PHDR is an 8-bit readable/writable register that stores port H data.

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PH7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PH6DT	0	R/W	
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

40.2.24 Port I Data Register (PIDR)

PIDR is an 8-bit readable/writable register that stores port I data.

Bit:	7	6	5	4	3	2	1	0
	PI7DT	PI6DT	PI5DT	PI4DT	PI3DT	PI2DT	PI1DT	PI0DT
Initial value:	0	0	Pin state	Pin state	Pin state	Pin state	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PI7DT	0	R	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PI6DT	0	R	
5	PI5DT	Pin state	R/W	
4	PI4DT	Pin state	R/W	
3	PI3DT	Pin state	R	
2	PI2DT	Pin state	R	
1	PI1DT	0	R/W	
0	PI0DT	0	R/W	

40.2.25 Port J Data Register (PJDR)

PJDR is an 8-bit readable/writable register that stores port J data.

Bit:	7	6	5	4	3	2	1	0
	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT
Initial value:	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PJ7DT	Pin state	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PJ6DT	Pin state	R/W	
5	PJ5DT	Pin state	R/W	
4	PJ4DT	Pin state	R/W	
3	PJ3DT	Pin state	R/W	
2	PJ2DT	Pin state	R/W	
1	PJ1DT	Pin state	R/W	
0	PJ0DT	Pin state	R/W	

40.2.26 Port K Data Register (PKDR)

PKDR is an 8-bit readable/writable register that stores port K data.

Bit:	7	6	5	4	3	2	1	0
	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT
Initial value:	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PK7DT	Pin state	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PK6DT	Pin state	R/W	
5	PK5DT	Pin state	R/W	
4	PK4DT	Pin state	R/W	
3	PK3DT	Pin state	R/W	
2	PK2DT	Pin state	R/W	
1	PK1DT	Pin state	R/W	
0	PK0DT	Pin state	R/W	

40.2.27 Port L Data Register (PLDR)

PLDR is an 8-bit readable/writable register that stores port L data.

Bit:	7	6	5	4	3	2	1	0
	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	PL2DT	PL1DT	PL0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PL7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PL6DT	0	R/W	
5	PL5DT	0	R/W	
4	PL4DT	0	R/W	
3	PL3DT	0	R/W	
2	PL2DT	0	R/W	
1	PL1DT	0	R/W	
0	PL0DT	0	R/W	

40.2.28 Port M Data Register (PMDR)

PMDR is an 8-bit readable/writable register that stores port M data.

Bit:	7	6	5	4	3	2	1	0
	PM7DT	PM6DT	PM5DT	PM4DT	PM3DT	PM2DT	PM1DT	PM0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PM7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PM6DT	0	R/W	
5	PM5DT	0	R/W	
4	PM4DT	0	R/W	
3	PM3DT	0	R/W	
2	PM2DT	0	R/W	
1	PM1DT	0	R/W	
0	PM0DT	0	R/W	

40.2.29 Port N Data Register (PNDR)

PNDR is an 8-bit readable/writable register that stores port N data.

Bit:	7	6	5	4	3	2	1	0
	—	—	PN5DT	PN4DT	PN3DT	PN2DT	PN1DT	PN0DT
Initial value:	0	0	0	Pin state	Pin state	Pin state	Pin state	Pin state
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	All0	R	Reserved
6				These bits are always read as 0, and the write value should always be 0.
5	PN5DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
4	PN4DT	Pin state	R/W	
3	PN3DT	Pin state	R/W	
2	PN2DT	Pin state	R/W	
1	PN1DT	Pin state	R/W	
0	PN0DT	Pin state	R/W	

40.2.30 Port O Data Register (PODR)

PODR is an 8-bit readable/writable register that stores port O data.

Bit:	7	6	5	4	3	2	1	0
	PO7DT	PO6DT	PO5DT	PO4DT	PO3DT	PO2DT	PO1DT	PO0DT
Initial value:	0	0	Pin state	Pin state	Pin state	Pin state	Pin state	Pin state
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PO7DT	0	R/W	Each of these bits stores output data for the corresponding pin that is used as a general output port. If the port is read, the value of the corresponding bit in this register will be read for a pin configured as a general output port, while the state of the corresponding pin will be read for a pin configured as a general input port.
6	PO6DT	0	R/W	
5	PO5DT	Pin state	R/W	
4	PO4DT	Pin state	R/W	
3	PO3DT	Pin state	R/W	
2	PO2DT	Pin state	R/W	
1	PO1DT	Pin state	R/W	
0	PO0DT	Pin state	R/W	

40.2.31 Port I Pull-Up Control Register (PIPUPR)

PIPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTI7 to PTI0, and when the pins of Port I are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PICR.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PI1PUPR	PI0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.
1	PI1PUPR	1	R/W	Controls pull-up of the PTI1 pin 0: PTI1 pin pull-up off 1: PTI1 pin pull-up on
0	PI0PUPR	1	R/W	Controls pull-up of the PTI0 pin 0: PTI0 pin pull-up off 1: PTI0 pin pull-up on

40.2.32 Port J Pull-Up Control Register (PJPUPR)

PJPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTJ7 to PTJ0, and when the pins of Port J are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PJCR.

Bit:	7	6	5	4	3	2	1	0
	PJ7PUPR	PJ6PUPR	PJ5PUPR	PJ4PUPR	PJ3PUPR	PJ2PUPR	PJ1PUPR	PJ0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PJ7PUPR	1	R/W	Controls pull-up of the PTJ7 pin 0: PTJ7 pin pull-up off 1: PTJ7 pin pull-up on
6	PJ6PUPR	1	R/W	Controls pull-up of the PTJ6 pin 0: PTJ6 pin pull-up off 1: PTJ6 pin pull-up on
5	PJ5PUPR	1	R/W	Controls pull-up of the PTJ5 pin 0: PTJ5 pin pull-up off 1: PTJ5 pin pull-up on
4	PJ4PUPR	1	R/W	Controls pull-up of the PTJ4 pin 0: PTJ4 pin pull-up off 1: PTJ4 pin pull-up on
3	PJ3PUPR	1	R/W	Controls pull-up of the PTJ3 pin 0: PTJ3 pin pull-up off 1: PTJ3 pin pull-up on
2	PJ2PUPR	1	R/W	Controls pull-up of the PTJ2 pin 0: PTJ2 pin pull-up off 1: PTJ2 pin pull-up on
1	PJ1PUPR	1	R/W	Controls pull-up of the PTJ1 pin 0: PTJ1 pin pull-up off 1: PTJ1 pin pull-up on

Bit	Bit Name	Initial value	R/W	Description
0	PJ0PUPR	1	R/W	Controls pull-up of the PTJ0 pin 0: PTJ0 pin pull-up off 1: PTJ0 pin pull-up on

40.2.33 Port K Pull-Up Control Register (PKPUPR)

PKPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTK7 to PTK0, and when the pins of Port K are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PKCR.

Bit:	7	6	5	4	3	2	1	0
	PK7PUPR	PK6PUPR	PK5PUPR	PK4PUPR	PK3PUPR	PK2PUPR	PK1PUPR	PK0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PK7PUPR	1	R/W	Controls pull-up of the PTK7 pin 0: PTK7 pin pull-up off 1: PTK7 pin pull-up on
6	PK6PUPR	1	R/W	Controls pull-up of the PTK6 pin 0: PTK6 pin pull-up off 1: PTK6 pin pull-up on
5	PK5PUPR	1	R/W	Controls pull-up of the PTK5 pin 0: PTK5 pin pull-up off 1: PTK5 pin pull-up on
4	PK4PUPR	1	R/W	Controls pull-up of the PTK4 pin 0: PTK4 pin pull-up off 1: PTK4 pin pull-up on
3	PK3PUPR	1	R/W	Controls pull-up of the PTK3 pin 0: PTK3 pin pull-up off 1: PTK3 pin pull-up on

Bit	Bit Name	Initial value	R/W	Description
2	PK2PUPR	1	R/W	Controls pull-up of the PTK2 pin 0: PTK2 pin pull-up off 1: PTK2 pin pull-up on
1	PK1PUPR	1	R/W	Controls pull-up of the PTK1 pin 0: PTK1 pin pull-up off 1: PTK1 pin pull-up on
0	PK0PUPR	1	R/W	Controls pull-up of the PTK0 pin 0: PTK0 pin pull-up off 1: PTK0 pin pull-up on

40.2.34 Port L Pull-Up Control Register (PLPUPR)

PLPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTL7 to PTL0, and when the pins of Port L are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PLCR.

Bit:	7	6	5	4	3	2	1	0
	PL7PUPR	PL6PUPR	PL5PUPR	PL4PUPR	PL3PUPR	PL2PUPR	PL1PUPR	PL0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PL7PUPR	1	R/W	Controls pull-up of the PTL7 pin 0: PTL7 pin pull-up off 1: PTL7 pin pull-up on
6	PL6PUPR	1	R/W	Controls pull-up of the PTL6 pin 0: PTL6 pin pull-up off 1: PTL6 pin pull-up on
5	PL5PUPR	1	R/W	Controls pull-up of the PTL5 pin 0: PTL5 pin pull-up off 1: PTL5 pin pull-up on

Bit	Bit Name	Initial value	R/W	Description
4	PL4PUPR	1	R/W	Controls pull-up of the PTL4 pin 0: PTL4 pin pull-up off 1: PTL4 pin pull-up on
3	PL3PUPR	1	R/W	Controls pull-up of the PTL3 pin 0: PTL3 pin pull-up off 1: PTL3 pin pull-up on
2	PL2PUPR	1	R/W	Controls pull-up of the PTL2 pin 0: PTL2 pin pull-up off 1: PTL2 pin pull-up on
1	PL1PUPR	1	R/W	Controls pull-up of the PTL1 pin 0: PTL1 pin pull-up off 1: PTL1 pin pull-up on
0	PL0PUPR	1	R/W	Controls pull-up of the PTL0 pin 0: PTL0 pin pull-up off 1: PTL0 pin pull-up on

40.2.35 Port M Pull-Up Control Register (PMPUPR)

PMPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTM7 to PTM0, and when the pins of Port M are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PMCR.

Bit:	7	6	5	4	3	2	1	0
	PM7PUPR	PM6PUPR	PM5PUPR	PM4PUPR	PM3PUPR	PM2PUPR	PM1PUPR	PM0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PM7PUPR	1	R/W	Controls pull-up of the PTM7 pin 0: PTM7 pin pull-up off 1: PTM7 pin pull-up on
6	PM6PUPR	1	R/W	Controls pull-up of the PTM6 pin 0: PTM6 pin pull-up off 1: PTM6 pin pull-up on
5	PM5PUPR	1	R/W	Controls pull-up of the PTM5 pin 0: PTM5 pin pull-up off 1: PTM5 pin pull-up on
4	PM4PUPR	1	R/W	Controls pull-up of the PTM4 pin 0: PTM4 pin pull-up off 1: PTM4 pin pull-up on
3	PM3PUPR	1	R/W	Controls pull-up of the PTM3 pin 0: PTM3 pin pull-up off 1: PTM3 pin pull-up on
2	PM2PUPR	1	R/W	Controls pull-up of the PTM2 pin 0: PTM2 pin pull-up off 1: PTM2 pin pull-up on
1	PM1PUPR	1	R/W	Controls pull-up of the PTM1 pin 0: PTM1 pin pull-up off 1: PTM1 pin pull-up on
0	PM0PUPR	1	R/W	Controls pull-up of the PTM0 pin 0: PTM0 pin pull-up off 1: PTM0 pin pull-up on

40.2.36 Port N Pull-Up Control Register (PNPUPR)

PNPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTN7 to PTN0, and when the pins of Port N are used by “other function”, pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by PNCR.

Bit:	7	6	5	4	3	2	1	0
	—	—	PN5PUPR	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R/W	R	R	R	R	R

Bit	Bit Name	Initial value	R/W	Description
7	—	All 1	R	Reserved
6				These bits are always read as 1, and the write value should always be 1.
5	PN5PUPR	1	R/W	Controls pull-up of the PTN5 pin
				0: PTN5 pin pull-up off
				1: PTN5 pin pull-up on
4 to 0	—	All 1	R	Reserved
				These bits are always read as 1, and the write value should always be 1.

40.2.37 Port O Pull-Up Control Register (POPUPR)

POPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTO7 to PTO0, and when the pins of Port O are used by "other function", pull-up control is performed for the individual pins. The settings in this register are invalid for the pins specified to function as port pins by POCR.

Bit:	7	6	5	4	3	2	1	0
	—	—	PO5PUPR	PO4PUPR	PO3PUPR	PO2PUPR	PO1PUPR	PO0PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	All 1	R	Reserved
6				These bits are always read as 1, and the write value should always be 1.
5	PO5PUPR	1	R/W	Controls pull-up of the PTO5 pin 0: PTO5 pin pull-up off 1: PTO5 pin pull-up on
4	PO4PUPR	1	R/W	Controls pull-up of the PTO4 pin 0: PTO4 pin pull-up off 1: PTO4 pin pull-up on
3	PO3PUPR	1	R/W	Controls pull-up of the PTO3 pin 0: PTO3 pin pull-up off 1: PTO3 pin pull-up on
2	PO2PUPR	1	R/W	Controls pull-up of the PTO2 pin 0: PTO2 pin pull-up off 1: PTO2 pin pull-up on
1	PO1PUPR	1	R/W	Controls pull-up of the PTO1 pin 0: PTO1 pin pull-up off 1: PTO1 pin pull-up on
0	PO0PUPR	1	R/W	Controls pull-up of the PTO0 pin 0: PTO0 pin pull-up off 1: PTO0 pin pull-up on

40.2.38 Input-Pin Pull-Up Control Register (PPUPR)

PPUPR is an 8-bit readable/writable register that individually controls the pull-up for the pin connected to each bit.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	IOIS16UP	BREQPUP	RDYPUP
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	All 1	R	Reserved These bits are always read as 1, and the write value should always be 1.
2	$\overline{\text{IOIS16}}$UP	1	R/W	Controls pull-up of the $\overline{\text{IOIS16}}$ pin 0: $\overline{\text{IOIS16}}$ pin pull-up off 1: $\overline{\text{IOIS16}}$ pin pull-up on
1	BREQPUP	1	R/W	Controls pull-up of the $\overline{\text{BREQ}}$ pin 0: $\overline{\text{BREQ}}$ pin pull-up off 1: $\overline{\text{BREQ}}$ pin pull-up on
0	RDYPUP	1	R/W	Controls pull-up of the $\overline{\text{RDY}}$/$\overline{\text{PCC_WAIT}}$ pin 0: $\overline{\text{RDY}}$/$\overline{\text{PCC_WAIT}}$ pin pull-up off 1: $\overline{\text{RDY}}$/$\overline{\text{PCC_WAIT}}$ pin pull-up on

40.2.39 Pin Select Register 0 (PSEL0)

PSEL0 is a 16-bit readable/writable register that selects the functions of the Port A (PTA), Port B (PTB), and Port C (PTC) pins multiplexed with “other function”.

When using the pins with “other function” assigned, set PSEL0 and then set the corresponding port control register to select “other function”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PTSEL0[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
4 to 0	PTSEL0 [4:0]	01000	R/W	These bits select the functions of Port A (PTA), Port B (PTB), and Port C (PTC).

Bit setting	Selected function			
PTSEL0[4:0]	PTA		PTB	PTC
01xxx	PCIC*		PCIC*	PCIC*
00001	SCIF1	MMC	—	MMC
10xxx	SCIF1	—	LCDCM	LCDCM
Other than above	Setting prohibited			

[Legend]

x: Don't care

Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

40.2.40 Pin Select Register 1 (PSEL1)

PSEL1 is a 16-bit readable/writable register that selects the functions of the Port D (PTD), Port E (PTE), Port F (PTF), Port G (PTG), and Port H (PTH) pins multiplexed with “other function”.

When using the pins with “other function” assigned, set PSEL1 and then set the corresponding port control register to select “other function”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL1[14:12]			PTSEL1[11:8]				PTSEL1[7:4]				PTSEL1[3:0]			
Initial value:	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
14 to 12	PTSEL1 [14:12]	100	R/W	These bits select the function of Port H (PTH). **Bit setting / Selected function** **PTSEL1[14:12] / PTH** 1xx / PCIC* 01x / MII1 001 / RMII1M 000 / TPU [Legend] x: Don't care Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

Bit	Bit Name	Initial value	R/W	Description
11 to 8	PTSEL1 [11:8]	1000	R/W	These bits select the functions of Port F (PTF) and Port G (PTG).

Bit setting	Selected function	
PTSEL1[11:8]	PTF	PTG
1xxx	PCIC*	PCIC*
01xx	MII1	MII1
0011	SIM	—
0010	DMAC3	—
Other than above	Setting prohibited	

[Legend]

x: Don't care

Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

7 to 4	PTSEL1 [7:4]	1000	R/W	These bits select the functions of Port E (PTE).

Bit setting	Selected function	
PTSEL1[7:4]	PTE	
1xxx	PCIC*	
0000	GMII1	
0010	DMAC2	SCIF2
0011	DMAC2	SSI0
0100	PCC	SCIF2
0101	PCC	SSI0
Other than above	Setting prohibited	

[Legend]

x: Don't care

Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

Bit	Bit Name	Initial value	R/W	Description
3 to 0	PTSEL1 [3:0]	1000	R/W	These bits select the functions of Port D (PTD).

Bit setting	Selected function	
PTSEL1[3:0]	**PTD**	
1xxx	PCIC*	
0000	GMII1	SIOF0
0001	LCDCM	
0010	PCC	
0100	HAC	SSI1
0101	HAC	GMII1
Other than above	Setting prohibited	

[Legend]

x: Don't care

Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

40.2.41 Pin Select Register 2 (PSEL2)

PSEL2 is a 16-bit readable/writable register that selects the functions of the Port I (PTI), Port J (PTJ), and Port K (PTK) pins multiplexed with “other function”.

When using the pins with “other function” assigned, set PSEL2 and then set the corresponding port control register to select “other function”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL2[14:12]			PTSEL2[11:8]				PTSEL2[7:6]		PTSEL2[5:4]		PTSEL2[3:2]		PTSEL2[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
14 to 12	PTSEL2 [14:12]	All 0	R/W	These bits select the functions of Port K (PTK).

Bit setting	Selected function	
PTSEL2[14:12]	**PTK**	
000	LCDC	—
001	GMII0	—
010	SIOF1	SIOF2
100	STIF1	—
Other than above	Setting prohibited	

Bit	Bit Name	Initial value	R/W	Description
11 to 8	PTSEL2 [11:8]	All 0	R/W	These bits select the functions of Port J (PTJ7 to PTJ1).

Bit setting	Selected function		
PTSEL2 [11:8]	PTJ1	PTJ7 to PTJ2	
0000	LCDC	LCDC	—
0010	LCDC	MII0	PCIC*
0011	LCDC	MII0	INT
0110	RMII1	RMII1	—
1000	STIF0M	STIF0M	—
Other than above	Setting prohibited		

Note: * When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

7 6	PTSEL2 [7:6]	All 0	R/W	These bits select the functions of Port J (PTJ0).

Bit setting	Selected function
PTSEL2[7:6]	PTJ0
00	REF50CK
01	GMII0
10	STIF0M
Other than above	Setting prohibited

5 4	PTSEL2 [5:4]	All 0	R/W	These bits select the functions of Port I (PTI5, PTI4).

Bit setting	Selected function	
PTSEL2[5:4]	PTI5, PTI4	
00	LCDC	USBH/F
01	LCDC	ETC1
10	LCDC	RMII0M0
11	STIF1	—

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial value</th><th>R/W</th><th>Description</th></tr>
<tr><td>3
2</td><td>PTSEL2 [3:2]</td><td>All 0</td><td>R/W</td><td>These bits select the functions of Port I (PTI3, PTI2).
<table><tr><th>Bit setting</th><th>Selected function</th></tr><tr><td>PTSEL2[3:2]</td><td>PTI3, PTI2</td></tr><tr><td>00</td><td>USB</td></tr><tr><td>01</td><td>STIF0M</td></tr><tr><td>10</td><td>SIOF1</td></tr><tr><td>11</td><td>IIC0</td></tr></table></td></tr>
<tr><td>1
0</td><td>PTSEL2 [1:0]</td><td>All 0</td><td>R/W</td><td>These bits select the functions of Port I (PTI1, PTI0).
<table><tr><th>Bit setting</th><th>Selected function</th></tr><tr><td>PTSEL2[1:0]</td><td>PTI1, PTI0</td></tr><tr><td>00</td><td>SYS</td></tr><tr><td>01</td><td>STIF1</td></tr><tr><td>1x</td><td>RMII0</td></tr></table>[Legend]
x: Don't care</td></tr>
</table>

40.2.42 Pin Select Register 3 (PSEL3)

PSEL3 is a 16-bit readable/writable register that selects the functions of the Port L (PTL), and Port M (PTM) pins multiplexed with “other function”.

When using the pins with “other function” assigned, set PSEL3 and then set the corresponding port control register to select “other function”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL3[14:12]			—	PTSEL3[10:8]			—	PTSEL3[6:4]			—	PTSEL3[2:0]		
Initial value:	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
14 to 12	PTSEL3 [14:12]	100	R/W	These bits select the functions of Port L (PTL7 to PTL4) and Port M (PTM7 to PTM1).
11	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

Bit setting	Selected function		
PTSEL3[14:12]	PTL7 to PTL4		PTM7 to PTM1
1xx	LBSC*1/ EXCPU	—	LBSC*1/ EXCPU
000	LCDC	—	RMII0
001	MII0	—	MII0
010	DMAC1	PCIC*2	RMII0
011	STIF0	—	STIF0

[Legend]

x: Don't care

Notes: 1. When 32-bit is selected as the data bus width in the LBSC, select this pin function.

2. When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.

Bit	Bit Name	Initial value	R/W	Description
10 to 8	PTSEL3 [10:8]	100	R/W	These bits select the function of Port M (PTM0). **Bit setting / Selected function** **PTSEL3[10:8] / PTM0** 1xx / LBSC*/EXCPU 000 / STIF0 001 / RMII0M0 011 / MII0 [Legend] x: Don't care Note: * When 32-bit is selected as the data bus width in the LBSC, select this pin function.
7	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
6 to 4	PTSEL3 [6:4]	100	R/W	These bits select the function of Port L (PTL3). **Bit setting / Selected function** **PTSEL3[6:4] / PTL3** 1xx / LBSC*1/EXCPU 000 / LCDC 001 / PCIC*2 010 / IRQ7 011 / MII0 [Legend] x: Don't care Notes: 1. When 32-bit is selected as the data bus width in the LBSC, select this pin function. 2. When clearing interrupt mask of the interrupt controller (INTC) with the PCIC function selected, make sure to select the PCIC function with this register in advance.
3	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
2 to 0	PTSEL3 [2:0]	100	R/W	These bits select the functions of Port L (PTL2 to PTL0).

Bit setting	Selected function
PTSEL3[2:0]	**PTL2 to PTL0**
1xx	LBSC*/EXCPU
000	LCDC
001	DMAC0
010	INT
011	MII0

[Legend]

x: Don't care

Note: * When 32-bit is selected as the data bus width in the LBSC, select this pin function.

40.2.43 Pin Select Register 4 (PSEL4)

PSEL4 is a 16-bit readable/writable register that selects the functions of the Port I (PTI) and Port O (PTO) pins multiplexed with “other function”, and also selects the function of the $\overline{\text{IOIS16}}$/TCLK pin and the REF125CK/HAC_BITCLK pin.

When using the pins with “other function” assigned, set PSEL4 and then set the corresponding port control register to select “other function”.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL4F	PTSEL4E	PTSEL4D	—	PTSEL4B	PTSEL4[10:9]		PTSEL48	—	PTSEL46	PTSEL4[5:4]		PTSEL4[3:2]		PTSEL4[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15	PTSEL4F	0	R/W	Selects the function of the $\overline{\text{IOIS16}}$/TCLK pin. 0: $\overline{\text{IOIS16}}$ function is selected 1: TCLK function is selected When the $\overline{\text{IOIS16}}$ function is selected, this bit should be set to the initial value. Do not select the $\overline{\text{IOIS16}}$ function after the TCLK function is selected.
14	PTSEL4E	0	R/W	Selects the function of the REF125CK/HAC_BITCLK pin. 0: REF125CK function is selected 1: HAC_BITCLK function is selected
13	PTSEL4D	0	R/W	Selects the pin for $\overline{\text{DREQ1}}$ 0: Input from PTL5 is used as $\overline{\text{DREQ1}}$ 1: Input from PTO5 is used as $\overline{\text{DREQ1}}$
12	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
11	PTSEL4B	0	R/W	Selects the pins for ST0_D0 to ST0_D07, ST0_START, ST0_VALID, and ST0_CLK. 0: Inputs from PTL5 to PTL7 and PTM0 to PTM7 are used as ST0_D0 to ST0_D07, ST0_START, ST0_VALID, and ST0_CLK, respectively 1: Inputs from PTJ1 to PTJ7, PTI2, PTI3, PTI6, and PTI7 are used as ST0_D0 to ST0_D07, ST0_START, ST0_VALID, and ST0_CLK, respectively
10, 9	PTSEL4 [10:9]	All 0	R/W	These bits select the pin for MDIO0. 00: Input from PTL3 is used as ET0_ MDIO (only PTL3 can be used when using GMII0 and MII0) 01: Input from PTI1 is used as RMII0_ MDIO 10: Input from PTM0 is used as RMII0_ MDIO 11: Input from PTO3 is used as RMII0_ MDIO
8	PTSEL48	0	R/W	Selects the pin for MDIO1. 0: Input from PTF2 is used as ET1_ MDIO (only PTF2 can be used when using GMII1 and MII1) 1: Input from PTO1 is used as ET1_ MDIO
7	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
6	PTSEL46	0	R/W	Selects the pins for RMII1_RX_ER, RMII1_CRS_DV, RMII1_RXD1, and RMII1_RXD0. 0: Inputs from PTJ1 and PTJ3 to PTJ5 are used as RMII1_RX_ER, RMII1_CRS_DV, RMII1_RXD1, and RMII1_RXD0, respectively 1: Inputs from PTH0, PTH1, PTH3, and PTH4 are used as RMII1_RX_ER, RMII1_CRS_DV, RMII1_RXD1, and RMII1_RXD0, respectively

Bit	Bit Name	Initial value	R/W	Description
5 4	PTSEL4 [5:4]	00	R/W	These bits select the functions of Port I (PTI7 and PTI6).

Bit setting	Selected function
PTSEL4[5:4]	**PTI7, PTI6**
00	INT
01	IIC1
11	STIF0M
Other than above	Setting prohibited

3 2	PTSEL4 [3:2]	00	R/W	These bits select the function of Port O (PTO3 to PTO0).

Bit setting	Selected function	
PTSEL4[3:2]	**PTO3 to PTO0**	
00	AUD	—
01	SSI2	—
1x	RMII0M1	RMII1M

[Legend]

x: Don't care

1 0	PTSEL4 [1:0]	00	R/W	These bits select the function of Port O (PTO7 to PTO4).

Bit setting	Selected function	
PTSEL4 [1:0]	**PTO7 to PTO4**	
00	AUD	INT
01	DMAC1M	EXCPU
1x	SSI3	—

[Legend]

x: Don't care

40.3 Usage Examples

Example procedures for configuring general-purpose input/output ports (GPIO) are shown below.

40.3.1 Port Output Function

To set up a pin for the port output function, write B'01 to the corresponding two bits in the port control register (PACR to PPCR). This allows the data of the corresponding bit in the port data register (PADR to PPDR) to be output from that pin.

Note that settings in the pull-up control register (PAPUPR to PPPUPR) and pin select register (PSEL0 to PSEL4) are invalid for the pins configured for the port output function.

40.3.2 Port Input Function

To set up a pin for the port input function, write B'10 (when not using MOS pull-up) or B'11 (when using MOS pull-up) to the corresponding two bits in the port control register (PACR to PPCR). This allows the value of that pin to be read from the corresponding bit in the port data register (PADR to PPDR).

Note that settings in the pull-up control register (PAPUPR to PPPUPR) and pin select register (PSEL0 to PSEL4) are invalid for the pins configured for the port output function.

40.3.3 Peripheral Module Function

To set up a pin for use by peripheral modules, first set the pin select register (PSEL0 to PSEL4) to select the module that uses that pin.

Then, if the pin is to be used as an input or input/output pin, set the pull-up control register (PAPUPR to PPPUPR) to create the MOS pull-up setting: to the corresponding bit, write 0 when not using the MOS pull-up or write 1 when using the MOS pull-up. For a pin used for output, the MOS pull-up is always turned off for any setting of the pull-up control register.

Finally, write B'00 to the corresponding two bits in the port control register (PACR to PPCR).

Section 41 User Break Controller (UBC)

The user break controller (UBC) provides versatile functions to facilitate program debugging. These functions help to ease creation of a self-monitor/debugger, which allows easy program debugging using this LSI alone, without using the in-circuit emulator. Various break conditions can be set in the UBC: instruction fetch or read/write access of an operand, operand size, data contents, address value, and program stop timing for instruction fetch.

41.1 Features

1. The following break conditions can be set.

   Break channels: Two (channels 0 and 1)

   User break conditions can be set independently for channels 0 and 1, and can also be set as a single sequential condition for the two channels, that is, a sequential break. (Sequential break involves two cases such that the channel 0 break condition is satisfied in a certain bus cycle and then the channel 1 break condition is satisfied in a different bus cycle, and vice versa.)

- Address

  When 40 bits containing ASID and 32-bit address are compared with the specified value, all the ASID bits can be compared or masked.

  32-bit address can be masked bit by bit, allowing the user to mask the address in desired page sizes such as lower 12 bits (4-Kbyte page) and lower 10 bits (1-Kbyte page).

- Data

  32 bits can be masked only for channel 1.

- Bus cycle

  The program can break either for instruction fetch (PC break) or operand access.

- Read or write access
- Operand sizes

  Byte, word, longword, and quadword are supported.

2. The user-designated exception handling routine for the user break condition can be executed.
3. Pre-instruction-execution or post-instruction-execution can be selected as the PC break timing.
4. A maximum of $2^{12} - 1$ repetition counts can be specified as the break condition (available only for channel 1).

Figure 41.1 shows the UBC block diagram.

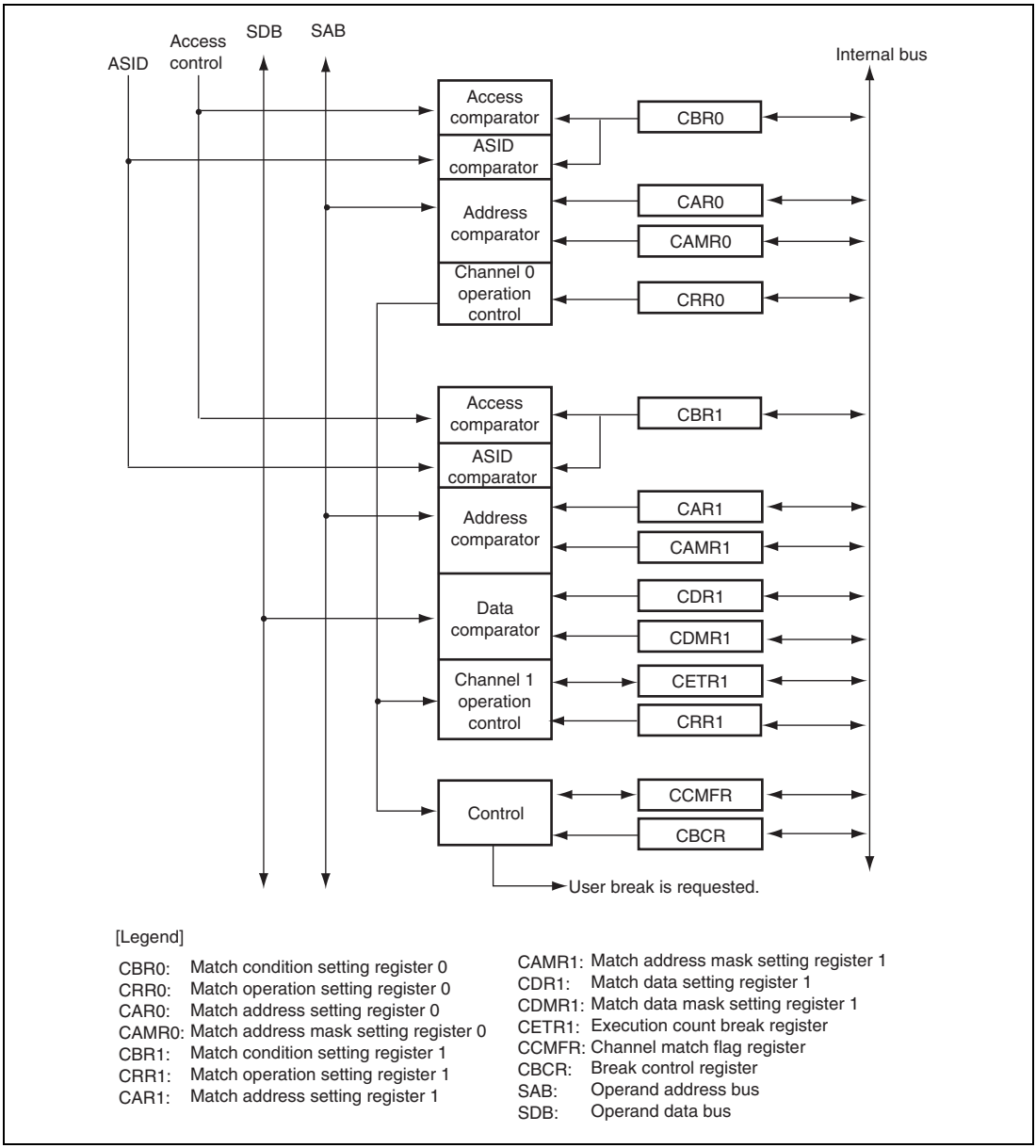


Figure 41.1 Block Diagram of UBC

41.2 Register Descriptions

The UBC has the following registers.

Table 41.1 Register Configuration

Name	Abbreviation	R/W	P4 Address*	Area 7 Address*	Access Size
Match condition setting register 0	CBR0	R/W	H'FF200000	H'1F200000	32
Match operation setting register 0	CRR0	R/W	H'FF200004	H'1F200004	32
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008	32
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C	32
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020	32
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024	32
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028	32
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C	32
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030	32
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034	32
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038	32
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600	32
Break control register	CBCR	R/W	H'FF200620	H'1F200620	32

Note: * P4 addresses are used when area P4 in the virtual address space is used, and area 7 addresses are used when accessing the register through area 7 in the physical address space using the TLB.

Table 41.2 Register Status in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sleep	Standby
Match condition setting register 0	CBR0	H'20000000	Retained	Retained	Retained
Match operation setting register 0	CRR0	H'00002000	Retained	Retained	Retained
Match address setting register 0	CAR0	Undefined	Retained	Retained	Retained
Match address mask setting register 0	CAMR0	Undefined	Retained	Retained	Retained
Match condition setting register 1	CBR1	H'20000000	Retained	Retained	Retained
Match operation setting register 1	CRR1	H'00002000	Retained	Retained	Retained
Match address setting register 1	CAR1	Undefined	Retained	Retained	Retained
Match address mask setting register 1	CAMR1	Undefined	Retained	Retained	Retained
Match data setting register 1	CDR1	Undefined	Retained	Retained	Retained
Match data mask setting register 1	CDMR1	Undefined	Retained	Retained	Retained
Execution count break register 1	CETR1	Undefined	Retained	Retained	Retained
Channel match flag register	CCMFR	H'00000000	Retained	Retained	Retained
Break control register	CBCR	H'00000000	Retained	Retained	Retained

The access size must be the same as the control register size. If the size is different, the register is not written to if attempted, and reading the register returns the undefined value. A desired break may not occur between the time when the instruction for rewriting the control register is executed and the time when the written value is actually reflected on the register. In order to confirm the exact timing when the control register is updated, read the data which has been written most recently. The subsequent instructions are valid for the most recently written register value.

41.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)

CBR0 and CBR1 are readable/writable 32-bit registers which specify the break conditions for channels 0 and 1, respectively. The following break conditions can be set in the CBR0 and CBR1: (1) whether or not to include the match flag in the conditions, (2) whether or not to include the ASID, and the ASID value when included, (3) whether or not to include the data value, (4) operand size, (5) whether or not to include the execution count, (6) bus type, (7) instruction fetch cycle or operand access cycle, and (8) read or write access cycle.

- CBR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SZ			—	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; thus, not checked. 1: The ASID is included in the match conditions.

Bit	Bit Name	Initial Value	R/W	Description
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: MF0 bit of the CCMFR register 000001: MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR0[0], MFI must be set to 000000 or 000001. And note that the channel 0 is not hit when MFE bit of this register is 1 and MFI bits are 000000 in the condition of CCRMF.MF0 = 0.
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match conditions; thus, not checked (any operand size specifies the match condition).*[1] 001: Byte access 010: Word access 011: Longword access 100: Quadword access*[2] Others: Reserved (setting prohibited)
11 to 8	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits of this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.

2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

- CBR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBE	SZ			ETBE	—	—	—	CD		ID		—	RW		CE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match flag value specified by the MFI bit of this register in the match conditions. When the specified match flag value is 1, the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match conditions.
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID specified by the AIV bit of this register in the match conditions. 0: The ASID is not included in the match conditions; thus, not checked. 1: The ASID is included in the match conditions.
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the match conditions. 000000: The MF0 bit of the CCMFR register 000001: The MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, but when 1 is written into CBR1[0], MFI must be set to 000000 or 000001. And note that the channel 1 is not hit when MFE bit of this register is 1 and MFI bits are 000001 in the condition of CCRMF.MF1 = 0.

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the match conditions.
15	DBE	0	R/W	Data Value Enable*3 Specifies whether or not to include the data value in the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 0: The data value is not included in the match conditions; thus, not checked. 1: The data value is included in the match conditions.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 000: The operand size is not included in the match condition; thus, not checked (any operand size specifies the match condition). *1 001: Byte access 010: Word access 011: Longword access 100: Quadword access*2 Others: Reserved (setting prohibited)
11	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execution count value in the match conditions. If this bit is 1 and the match condition satisfaction count matches the value specified by the CETR1 register, the operation specified by the CRR1 register is performed. 0: The execution count value is not included in the match conditions; thus, not checked. 1: The execution count value is included in the match conditions.
10 to 8	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD	All 0	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the operand access cycle is specified as a match condition. 00: Operand bus for operand access Others: Reserved (setting prohibited)
5, 4	ID	All 0	R/W	Instruction Fetch/Operand Access Select Specifies the instruction fetch cycle or operand access cycle as the match condition. 00: Instruction fetch cycle or operand access cycle 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access cycle
3	—	0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match condition. This bit is valid only when the operand access cycle is specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is 0, all the other bits in this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.

2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and the match data mask setting register.

3. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

41.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)

CRR0 and CRR1 are readable/writable 32-bit registers which specify the operation to be executed when channels 0 and 1 satisfy the match condition, respectively. The following operations can be set in the CRR0 and CRR1 registers: (1) breaking at a desired timing for the instruction fetch cycle and (2) requesting a break.

- CRR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than the ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.

Bit	Bit Name	Initial Value	R/W	Description
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

- CRR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

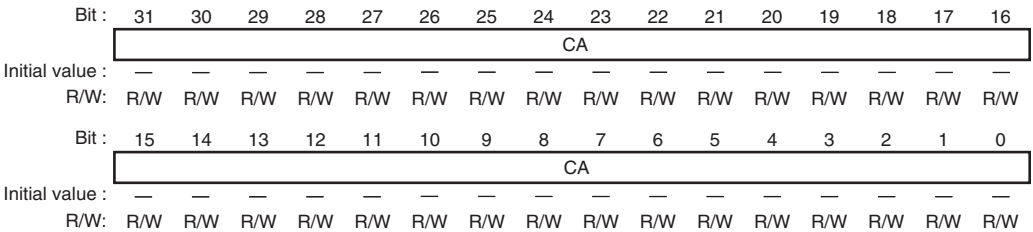
Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
13	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
12 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction execution as the break timing for the instruction fetch cycle. This bit is invalid for breaks other than ones for the instruction fetch cycle. 0: Sets the PC break before instruction execution. 1: Sets the PC break after instruction execution.

Bit	Bit Name	Initial Value	R/W	Description
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break when the match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

41.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)

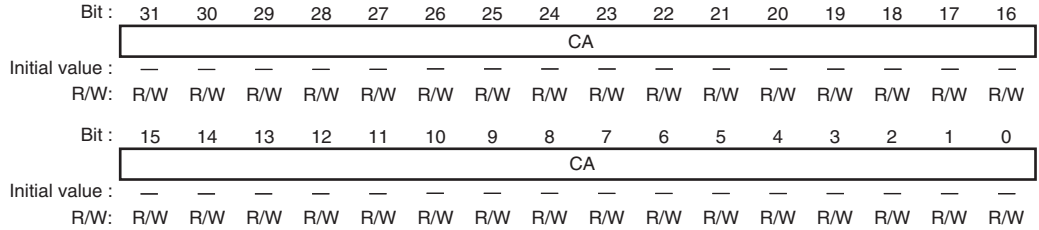
CAR0 and CAR1 are readable/writable 32-bit registers specifying the virtual address to be included in the break conditions for channels 0 and 1, respectively.

- CAR0



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR0 register, specify the SAB address in CA[31:0].

- CAR1



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SAB address in CA[31:0].

41.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)

CMAR0 and CMAR1 are readable/writable 32-bit registers which specify the bits to be masked among the address bits specified by using the match address setting register of the corresponding channel. (Set the bits to be masked to 1.)

- CAMR0

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR0 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

- CAMR1

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAM															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the address bits which are specified using the CAR1 register. (Set the bits to be masked to 1.) 0: Address bits CA[n] are included in the break condition. 1: Address bits CA[n] are masked and not included in the break condition. [n] = any values from 31 to 0

41.2.5 Match Data Setting Register 1 (CDR1)

CDR1 is a readable/writable 32-bit register which specifies the data value to be included in the break conditions for channel 1.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CD															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD															
Initial value :	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CD	Undefined	R/W	Compare Data Value Specifies the data value to be included in the break conditions. When the operand bus has been specified using the CBR1 register, specify the SDB data value in CD[31:0].

Table 41.3 Settings for Match Data Setting Register

Bus and Size Selected Using CBR1	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
Operand bus (byte)	Don't care	Don't care	Don't care	SDB7 to SDB0
Operand bus (word)	Don't care	Don't care	SDB15 to SDB8	SDB7 to SDB0
Operand bus (longword)	SDB31 to SDB24	SDB23 to SDB16	SDB15 to SDB8	SDB7 to SDB0

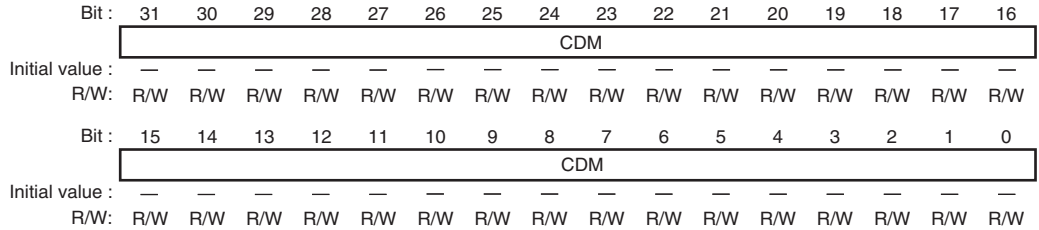
Notes: 1. If the data value is included in the match conditions, be sure to specify the operand size.

2. The OCBI instruction is handled as longword write access without the data value, and the PREF, OCBP, and OCBWB instructions are handled as longword read access without the data value. Therefore, do not include the data value in the match conditions for these instructions.

3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with the contents of both the match data setting register and match data mask setting register.

41.2.6 Match Data Mask Setting Register 1 (CDMR1)

CDMR1 is a readable/writable 32-bit register which specifies the bits to be masked among the data value bits specified using the match data setting register. (Set the bits to be masked to 1.)



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDM	Undefined	R/W	Compare Data Value Mask Specifies the bits to be masked among the data value bits specified using the CDR1 register. (Set the bits to be masked to 1.) 0: Data value bits CD[n] are included in the break condition. 1: Data value bits CD[n] are masked and not included in the break condition. [n] = any values from 31 to 0

41.2.7 Execution Count Break Register 1 (CETR1)

CETR1 is a readable/writable 32-bit register which specifies the number of the channel hits before a break occurs. A maximum value of $2^{12} - 1$ can be specified. When the execution count value is included in the match conditions by using the match condition setting register, the value of this register is decremented by one every time the channel is hit. When the channel is hit after the register value reaches H'001, a break occurs.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CET											
Initial value :	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in the break conditions.

41.2.8 Channel Match Flag Register (CCMFR)

CCMFR is a readable/writable 32-bit register which indicates whether or not the match conditions have been satisfied for each channel. When a channel match condition has been satisfied, the corresponding flag bit is set to 1. To clear the flags, write the data containing value 0 for the bits to be cleared and value 1 for the other bits to this register. (The logical AND between the value which has been written and the current register value is actually written to the register.) Sequential operation using multiple channels is available by using these match flags.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 1 match condition has not been satisfied. 1: Channel 1 match condition has been satisfied.
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 match condition has been satisfied. To clear the flag, write 0 to this bit. 0: Channel 0 match condition has not been satisfied. 1: Channel 0 match condition has been satisfied.

41.2.9 Break Control Register (CBCR)

CBCR is a readable/writable 32-bit register which specifies whether or not to use the user break debugging support function. For details on the user break debugging support function, refer to section 41.4, User Break Debugging Support Function.

Bit :	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UBDE
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved For read/write in this bit, refer to General Precautions on Handling of Product.
0	UBDE	0	R/W	User Break Debugging Support Function Enable Specifies whether or not to use the user break debugging support function. 0: Does not use the user break debugging support function. 1: Uses the user break debugging support function.

41.3 Operation Description

41.3.1 Definition of Words Related to Accesses

"Instruction fetch" refers to an access in which an instruction is fetched. For example, fetching the instruction located at the branch destination after executing a branch instruction is an instruction access. "Operand access" refers to any memory access accompanying execution of an instruction. For example, accessing an address (PC + disp × 2 + 4) in the instruction MOV.W@(disp,PC),Rn is an operand access. "Data" is used in contrast to "address".

All types of operand access are classified into read or write access. Special care must be taken in using the following instructions.

- PREF, OCBP, and OCBWB: Instructions for a read access
- MOVCA.L and OCBI: Instructions for a write access
- TAS.B: Instruction for a single read access or a single write access

The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions is access without the data value; therefore, do not include the data value in the match conditions for these instructions.

The operand size should be defined for all types of operand access. Available operand sizes are byte, word, longword, and quadword. For operand access accompanying the PREF, OCBP, OCBWB, MOVCA.L, and OCBI instructions, the operand size is defined as longword.

41.3.2 User Break Operation Sequence

The following describes the sequence from when the break condition is set until the user break exception handling is initiated.

1. Specify the operand size, bus, instruction fetch/operand access, and read/write as the match conditions using the match condition setting register (CBR0 or CBR1). Specify the break address using the match address setting register (CAR0 or CAR1), and specify the address mask condition using the match address mask setting register (CAMR0 or CAMR1). To include the ASID in the match conditions, set the AIE bit in the match condition setting register and specify the ASID value by the AIV bit in the same register. To include the data value in the match conditions, set the DBE bit in the match condition setting register; specify the break data using the match data setting register (CDR1); and specify the data mask condition using the match data mask setting register (CDMR1). To include the execution count in the match conditions, set the ETBE bit of the match condition setting register; and specify the execution count using the execution count break register (CETR1). To use the sequential break, set the MFE bit of the match condition setting register; and specify the number of the first channel using the MFI bit.
2. Specify whether or not to request a break when the match condition is satisfied and the break timing when the match condition is satisfied as a result of fetching the instruction using the match operation setting register (CRR0 or CRR1). After having set all the bits in the match condition setting register except the CE bit and the other necessary registers, set the CE bit and read the match condition setting register again. This ensures that the set values in the control registers are valid for the subsequent instructions immediately after reading the register. Setting the CE bit of the match condition setting register in the initial state after reset via the control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag (MF1 or MF0) in the channel match flag register (CCMFR) is set. A break is also requested to the CPU according to the set values in the match operation setting register (CRR0 or CRR1). The CPU operates differently according to the BL bit value of the SR register: when the BL bit is 0, the CPU accepts the break request and executes the specified exception handling; and when the BL bit is 1, the CPU does not execute the exception handling.
4. The match flags (MF1 and MF0) can be used to confirm whether or not the corresponding match condition has been satisfied. Although the flag is set when the condition is satisfied, it is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory store instruction to the channel match flag register (CCMFR) in order to use the flag again.
5. Breaks may occur virtually at the same time for channels 0 and 1. In this case, only one break request is sent to the CPU; however, the two condition match flags corresponding to these breaks may be set.

6. While the BL bit in the SR register is 1, no break requests are accepted. However, whether or not the condition has been satisfied is determined. When the condition is determined to be satisfied, the corresponding condition match flag is set.
7. If the sequential break conditions are set, the condition match flag is set every time the match conditions are satisfied for each channel. When the conditions have been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.

41.3.3 Instruction Fetch Cycle Break

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CBR1), the instruction fetch cycle is handled as a match condition. To request a break upon satisfying the match condition, set the BIE bit in the match operation setting register (CRR0 or CRR1) of the corresponding channel. Either before or after executing the instruction can be selected as the break timing according to the PCB bit value. If the instruction fetch cycle is specified as a match condition, be sure to clear the LSB to 0 in the match address setting register (CAR0 or CAR1); otherwise, no break occurs.
2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break is requested when the instruction is fetched and determined to be executed. Therefore, this function cannot be used for the instructions which are fetched through overrun (i.e., the instructions fetched during branching or making transition to the interrupt routine but not executed). For priorities of pre-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If pre-instruction-execution break is specified for the delayed slot of the delayed branch instruction, the break is requested before the delayed branch instruction is executed. However, do not specify pre-instruction-execution break for the delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the break is requested after the instruction which satisfied the match condition has been executed and before the next instruction is executed. Similar to pre-instruction-execution break, this function cannot be used for the instructions which are fetched through overrun. For priorities of post-instruction-execution break and the other exceptions, refer to section 5, Exception Handling. If post-instruction-execution break is specified for the delayed branch instruction and its delayed slot, the break does not occur until the first instruction at the branch destination.
4. If the instruction fetch cycle is specified as the channel 1 match condition, the DBE bit of match condition setting register CBR1 becomes invalid, the settings of match data setting register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, the data value cannot be specified for the instruction fetch cycle break.

41.3.4 Operand Access Cycle Break

1. Table 41.4 shows the relation between the operand sizes specified using the match condition setting register (CBR0 or CBR1) and the address bits to be compared for the operand access cycle break.

Table 41.4 Relation between Operand Sizes and Address Bits to be Compared

Selected Operand Size	Address Bits to be Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Operand size is not included in the match conditions	Address bits A31 to A3 for quadword access Address bits A31 to A2 for longword access Address bits A31 to A1 for word access Address bits A31 to A0 for byte access

The above table means that if address H'00001003 is set in the match address setting register (CAR0 or CAR1), for example, the match condition is satisfied for the following access cycles (assuming that all the other conditions are satisfied):

— Longword access to address H'00001000

— Word access to address H'00001002

— Byte access to address H'00001003

2. When the data value is included in the channel 1 match conditions:

If the data value is included in the match conditions, be sure to select the quadword, longword, word, or byte as the operand size using the operand size select bit (SZ) of the match condition setting register (CBR1), and also set the match data setting register (CDR1) and the match data mask setting register (CDMR1). With these settings, the match condition is satisfied when both of the address and data conditions are satisfied. The data value and mask control for byte access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 to 0 in the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divided into the upper and lower 32-bit data units, and each unit is independently compared with the specified condition. When either the upper or lower 32-bit data unit satisfies the match condition, the match condition for the 64-bit data is determined to be satisfied.

3. The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions are access without the data value; therefore, if the data value is included in the match conditions for these instructions, the match conditions will never be satisfied.
4. If the operand bus is selected, a break occurs after executing the instruction which has satisfied the conditions and immediately before executing the next instruction. However, if the data value is included in the match conditions, a break may occur after executing several instructions after the instruction which has satisfied the conditions; therefore, it is impossible to identify the instruction causing the break. If such a break has occurred for the delayed branch instruction or its delayed slot, the break does not occur until the first instruction at the branch destination.

   However, do not specify the operand break for the delayed slot of the RTE instruction. And if the data value is included in the match conditions, it is not allowed to set the break for the preceding the RTE instruction by one to six instructions.

41.3.5 Sequential Break

1. Sequential break conditions can be specified by setting the MFE and MFI bits in the match condition setting registers (CBR0 and CBR1). (Sequential break involves two cases such that channel 0 break condition is satisfied then channel 1 break condition is satisfied, and vice versa.) To use the sequential break function, clear the MFE bit of the match condition setting register and the BIE bit of the match operation setting register of the first channel in the sequence, and set the MFE bit and specify the number of the second channel in the sequence using the MFI bit in the match condition setting register of the second channel in the sequence. If the sequential break condition is set, the condition match flag is set every time the match condition is satisfied for each channel. When the condition has been satisfied for the first channel in the sequence but not for the second channel in the sequence, clear the condition match flag for the first channel in the sequence in order to release the first channel in the sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequential break conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied within a significantly short time, sequential operation may not be guaranteed in some cases, as shown below.

- When the Match Condition is Satisfied at the Instruction Fetch Cycle for Both the First and Second Channels in the Sequence:

Instruction B is 0 instruction after instruction A	Equivalent to setting the same addresses; do not use this setting.
Instruction B is one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the instruction fetch cycle for the first channel in the sequence whereas the match condition is satisfied at the operand access cycle for the second channel in the sequence:

Instruction B is 0 or one instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for the first channel in the sequence whereas the match condition is satisfied at the instruction fetch cycle for the second channel in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for both the first and second channels in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

41.3.6 Program Counter Value to be Saved

When a break has occurred, the address of the instruction to be executed when the program restarts is saved in the SPC then the exception handling state is initiated. A unique instruction causing a break can be identified unless the data value is included in the match conditions.

1. When the instruction fetch cycle (before instruction execution) is specified as the match condition:

   The address of the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is not executed, but a break occurs instead. However, if the match conditions are satisfied for the delayed slot instruction, the address of the delayed branch instruction is saved in the SPC.
2. When the instruction fetch cycle (after instruction execution) is specified as the match condition:

   The address of the instruction immediately after the instruction which has satisfied the match conditions is saved in the SPC. The instruction which has satisfied the match conditions is executed, then a break occurs before the next instruction. If the match conditions are satisfied for the delayed branch instruction or its delayed slot, these instructions are executed and the address of the branch destination is saved in the SPC.

3. When the operand access (address only) is specified as the match condition:

   The address of the instruction immediately after the instruction which has satisfied the break conditions is saved in the SPC. The instruction which has satisfied the match conditions are executed, then a break occurs before the next instruction. However, if the conditions are satisfied for the delayed slot, the address of the branch destination is saved in the SPC.

4. When the operand access (address and data) is specified as the match condition:

   If the data value is added to the match conditions, the instruction which has satisfied the match conditions is executed. A user break occurs before executing an instruction that is one through six instructions after the instruction which has satisfied the match conditions. The address of the instruction is saved in the SPC; thus, it is impossible to identify exactly where a break will occur. If the conditions are satisfied for the delayed slot instruction, the address of the branch destination is saved in the SPC. If a branch instruction follows the instruction which has satisfied the match conditions, a break may occur after the delayed instruction and delayed slot are executed. In this case, the address of the branch destination is also saved in the SPC.

41.4 User Break Debugging Support Function

By using the user break debugging support function, the branch destination address can be modified when the CPU accepts the user break request. Specifically, setting the UBDE bit of break control register CBCR to 1 allows branching to the address indicated by DBR instead of branching to the address indicated by the [VBR + offset]. Figure 41.2 shows the flowchart of the user break debugging support function.

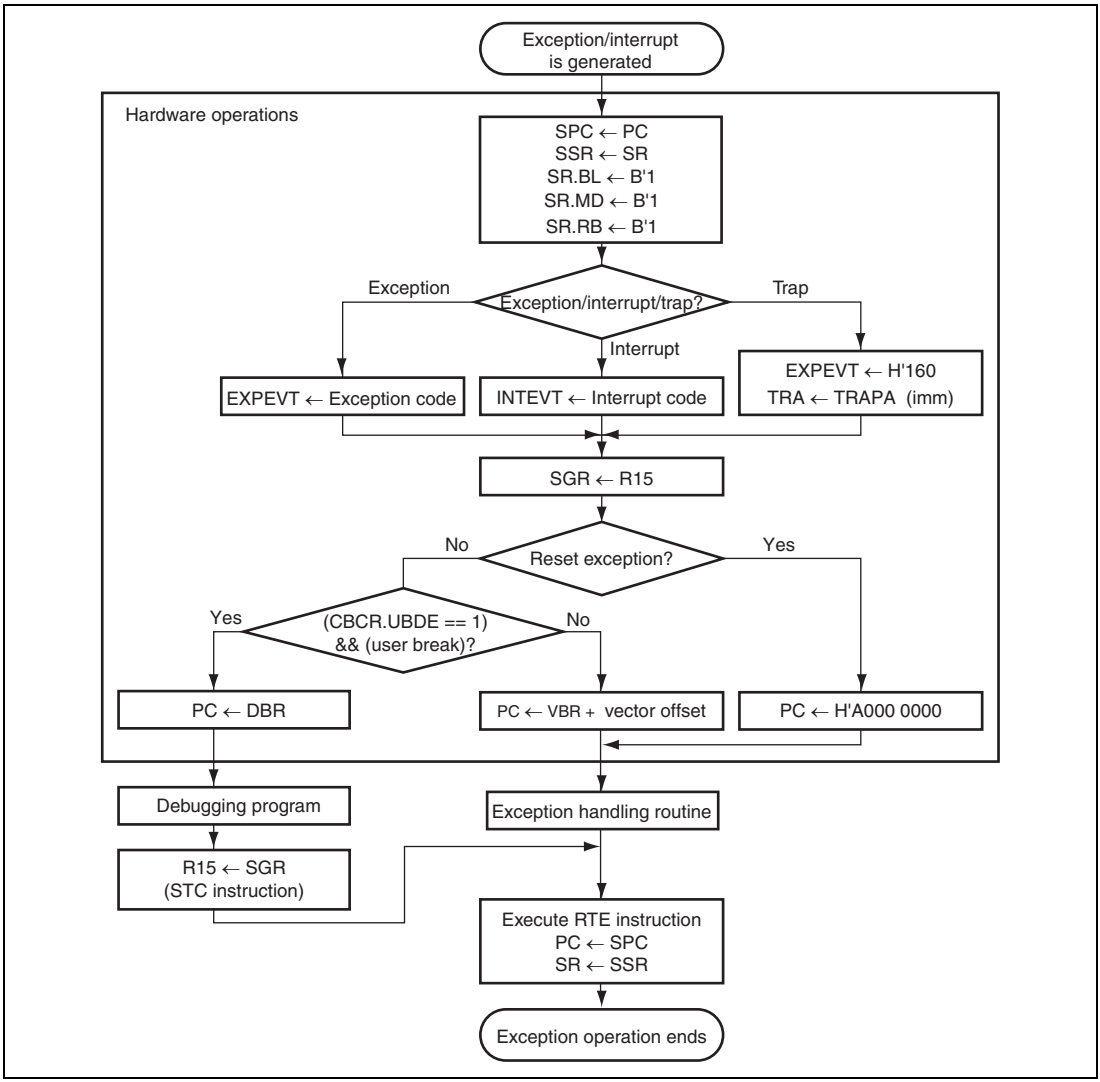


Figure 41.2 Flowchart of User Break Debugging Support Function

41.5 User Break Examples

(1) Match Conditions are Specified for an Instruction Fetch Cycle

- Example 1-1

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002003 / CAR0 = H'00000404 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00008010 / CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00000404 / Address mask: H'00000000

  Bus cycle: Instruction fetch (after executing the instruction)

  ASID is not included in the conditions.

  — Channel 1:

  Address: H'00008010 / Address mask: H'00000006

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing instruction)

  ASID, data values, and execution count are not included in the conditions.

  With the above settings, the user break occurs after executing the instruction at address H'00000404 or before executing the instruction at address H'00008010 to H'00008016.

- Example 1-2

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Channel 0 → Channel1 sequential mode

  — Channel 0

  Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

  Bus cycle: Instruction fetch (before executing the instruction)

  — Channel 1

  Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

  Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-3

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00027128 / CAMR0 = H'00000000 / CBR1 = H'00000013 / CRR1 = H'00002001 / CAR1 = H'00031415 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00027128 / Address mask: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    ASID is not included in the conditions.

  — Channel 1

    Address: H'00031415 / Address mask: H'00000000

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    ASID, data values, and execution count are not included in the conditions.

  With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00027128. No user break occurs for channel 1 since the instruction fetch is executed only at even addresses.

- Example 1-4

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226 / CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0003722E / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H′00000000 / CBCR = H'00000000

  Specified conditions: Channel 0 → Channel 1 sequential mode

  — Channel 0

    Address: H'00037226 / Address mask: H'00000000 / ASID: H'80

    Bus cycle: Instruction fetch (before executing the instruction)

  — Channel 1

    Address: H'0003722E / Address mask: H'00000000 / ASID: H'70

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00037226 where ASID is H'80 and before executing the instruction at address H'0003722E where ASID is H'70.

- Example 1-5

  Register settings: CBR0 = H'00000013 / CRR0 = H'00002001 / CAR0 = H'00000500 / CAMR0 = H'00000000 / CBR1 = H'00000813 / CRR1 = H'00002001 / CAR1 = H'00001000 / CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000005 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00000500 / Address mask: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

    ASID is not included in the conditions.

  — Channel 1

    Address: H'00001000 / Address mask: H'00000000

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000005

    Bus cycle: Instruction fetch (before executing the instruction)

    Execution count: 5

    ASID and data values are not included in the conditions.

  With the above settings, the user break occurs for channel 0 before executing the instruction at address H'00000500. The user break occurs for channel 1 after executing the instruction at address H'00001000 four times; before executing the instruction five times.

- Example 1-6

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002003 / CAR0 = H'00008404 / CAMR0 = H'00000FFF / CBR1 = H'40700013 / CRR1 = H'00002001 / CAR1 = H'00008010 / CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

    Address: H'00008404 / Address mask: H'00000FFF / ASID: H'80

    Bus cycle: Instruction fetch (after executing the instruction)

  — Channel 1

    Address: H'00008010 / Address mask: H'00000006 / ASID: H'70

    Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

    Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at address H'00008000 to H'00008FFE where ASID is H'80 or before executing the instruction at address H'00008010 to H'00008016 where ASID is H'70.

(2) Match Conditions are Specified for an Operand Access Cycle

- Example 2-1

  Register settings: CBR0 = H'40800023 / CRR0 = H'00002001 / CAR0 = H'00123456 / CAMR0 = H'00000000 / CBR1 = H'4070A025 / CRR1 = H'00002001 / CAR1 = H'000ABCDE / CAMR1 = H'000000FF / CDR1 = H'0000A512 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

  Address: H'00123456 / Address mask: H'00000000 / ASID: H'80

  Bus cycle: Operand bus, operand access, and read (operand size is not included in the conditions.)

  — Channel 1

  Address: H'000ABCDE / Address mask: H'000000FF / ASID: H'70

  Data: H'0000A512 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Operand bus, operand access, write, and word size

  Execution count is not included in the conditions.

  With these settings, the user break occurs for channel 0 for the following accesses: longword read access to address H'000123454, word read access to address H'000123456, byte read access to address H'000123456 where ASID is H'80. The user break occurs for channel 1 when word H'A512 is written to address H'000ABC00 to H'000ABCFE where ASID is H'70.

41.6 Usage Notes

1. A desired break may not occur between the time when the instruction for rewriting the UBC register is executed and the time when the written value is actually reflected on the register.

   After the UBC register is updated, execute one of the following three methods.

   A. Read the updated UBC register, and execute a branch using the RTE instruction.
   (It is not necessary that a branch using the RTE instruction is next to a reading UBC register.)

   B. Execute the ICBI instruction for any address (including non-cacheable area).
   (It is not necessary that the ICBI instruction is next to a reading UBC register.)

   C. Set 0(initial value) to IRMCR.R1 before updating the UBC register and update with following sequence.

      a. Write the UBC register.

      b. Read the UBC register which is updated at 1.

      c. Write the value which is read at 2 to the UBC register.

Note: When two or more UBC registers are updated, executing these methods at each updating the UBC registers is not necessary. At only last updating the UBC register, execute one of these methods.

2. The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch is specified as the match condition.
3. If the sequential break conditions are set, the sequential break conditions are satisfied when the conditions for the first and second channels in the sequence are satisfied in this order. Therefore, if the conditions are set so that the conditions for channels 0 and 1 should be satisfied simultaneously for the same bus cycle, the sequential break conditions will not be satisfied, causing no break.
4. For the SLEEP instruction, do not allow the post-instruction-execution break where the instruction fetch cycle is the match condition. For the instructions preceding the SLEEP instruction by one to five instructions, do not allow the break where the operand access is the match condition.
5. If the user break and other exceptions occur for the same instruction, they are determined according to the specified priority. For the priority, refer to section 5, Exception Handling. If the exception having the higher priority occurs, the user break does not occur.

   — The pre-instruction-execution break is accepted prior to any other exception.

— If the post-instruction-execution break and data access break have occurred simultaneously with the re-execution type exception (including the pre-instruction-execution break) having a higher priority, only the re-execution type exception is accepted, and no condition match flags are set. When the exception handling has finished thus clearing the exception source, and when the same instruction has been executed again, the break occurs setting the corresponding flag.

— If the post-instruction-execution break or operand access break has occurred simultaneously with the completion-type exception (TRAPA) having a higher priority, then no user break occurs; however, the condition match flag is set.

6. When conditions have been satisfied simultaneously and independently for channels 0 and 1, resulting in identical SPC values for both of the breaks, the user break occurs only once. However, the condition match flags are set for both channels. For example,

   Instruction at address 110 (post-instruction-execution break for instruction fetch for channel 0) → SPC = 112, CCMFR.MF0 = 1

   Instruction at address 112 (pre-instruction-execution break for instruction fetch for channel 1) → SPC = 112, CCMFR.MF1 = 1

7. It is not allowed to set the pre-instruction-execution break or the operand break in the delayed slot instruction of the RTE instruction. And if the data value is included in the match conditions of the operand break, do not set the break for the preceding the RTE instruction by one to six instructions.

8. If the re-execution type exception and the post-instruction-execution break are in conflict for the instruction requiring two or more execution states, then the re-execution type exception occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 when the break conditions have been satisfied.

Section 42 User Debugging Interface (H-UDI)

The H-UDI is a serial interface which conforms to the JTAG (IEEE 1149.4: IEEE Standard Test Access Port and Boundary-Scan Architecture) standard. The H-UDI is also used for emulator connection.

42.1 Features

The H-UDI is a serial interface which conforms to the JTAG standard. The H-UDI is also used for emulator connection. When using an emulator, H-UDI functions should not be used. Refer to the appropriate emulator users manual for the method of connecting the emulator.

The H-UDI has six pins: TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK. The pin functions except $\overline{\text{ASEBRK}}$/BRKACK and serial communications protocol conform to the JTAG standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDCK, and AUDATA3 to AUDATA0). These six pins for emulator are multiplexed with on-chip modules. And the H-UDI has one chip-mode setting pin: (MPMD).

The H-UDI has two TAP controller blocks; one is for the boundary-scan test and another is H-UDI function except the boundary-scan test. The H-UDI initial state is for the boundary scan after power-on or $\overline{\text{TRST}}$ asserted. It is necessary to set H-UDI switchover command to use the H-UDI function. And the CPU cannot access the boundary scan TAP controller.

Figure 42.1 shows a block diagram of the H-UDI.

The H-UDI has the TAP (Test Access Port) controller and four registers (SDBPR, SDBSR, SDIR, and SDINT). SDBPR supports the JTAG bypass mode, SDBSR supports the JTAG boundary scan mode, SDIR is used for commands, and SDINT is used for H-UDI interrupts. SDIR is directly accessed from the TDI and TDO pins.

The TAP controller, control registers and boundary scan TAP controller are initialized by driving the $\overline{\text{TRST}}$ pin low or by applying the TCK signal for five or more clock cycles with the TMS pin set to 1. This initialization sequence is independent of the reset pin for this LSI. Other circuits are initialized by a normal reset.

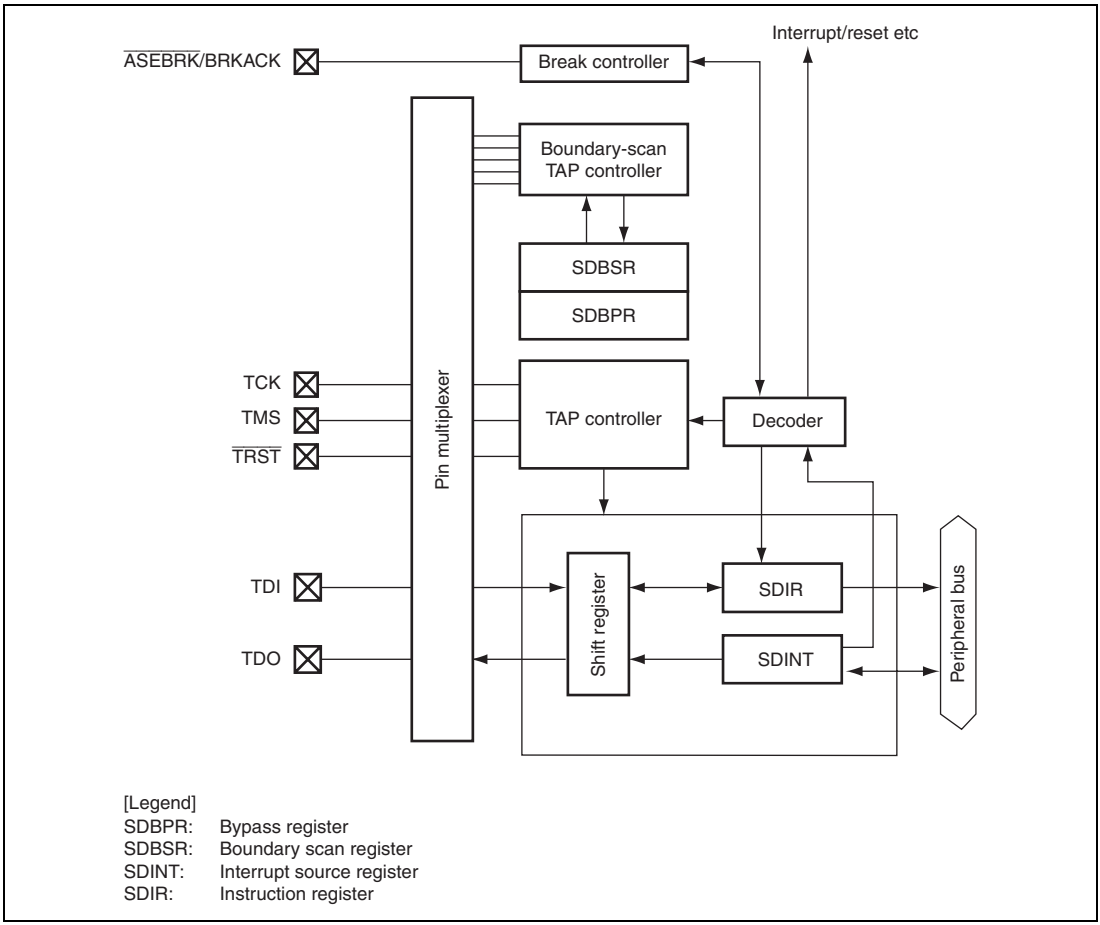


Figure 42.1 H-UDI Block Diagram

42.2 Input/Output Pins

Table 42.1 shows the pin configuration for the H-UDI.

Table 42.1 Pin Configuration

Pin Name	Function	I/O	Description	When Not in Use
TCK	Clock	Input	Functions as the serial clock input pin stipulated in the JTAG standard. Data input to the H-UDI via the TDI pin or data Output via the TDO pin is performed in synchronization with this signal.	Open*[1]
TMS	Mode	Input	Mode Select Input Changing this signal in synchronization with the TCK signal determines the significance of data input via the TDI pin. Its protocol conforms to the JTAG standard (IEEE standard 1149.1).	Open*[1]
$\overline{\text{TRST}}$*[2]	Reset	Input	H-UDI Reset Input This signal is received asynchronously with a TCK signal. Asserting this signal resets the JTAG interface circuit. When a power is supplied, the $\overline{\text{TRST}}$ pin should be asserted for a given period regardless of whether or not the JTAG function is used, which differs from the JTAG standard.	Fixed to ground or connected to the $\overline{\text{PRESET}}$ pin*[3]
TDI	Data input	Input	Data Input Data is sent to the H-UDI by changing this signal in synchronization with the TCK signal.	Open*[1]
TDO	Data output	Output	Data Output Data is read from the H-UDI in synchronization with the TCK signal.	Open
$\overline{\text{ASEBRK}}$/ BRKACK	Emulator	I/O	Pins for an emulator	Open*[1]
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for an emulator	Open
MPMD	Chip-mode	Input	Selects the operation mode of this LSI, whether emulation support mode (Low level) or LSI operation mode (High level).	Open

Notes: 1. This pin is pulled up in this LSI. When using interrupts or resets via the H-UDI or emulator, the use of external pull-up resistors will not cause any problem.

2. When using interrupts or resets via the H-UDI or emulator, the $\overline{\text{TRST}}$ pin should be designed so that it can be controlled independently and can be controlled to retain low level while the $\overline{\text{PRESET}}$ pin is asserted at a power-on reset.

3. This pin should be connected to ground, the $\overline{\text{PRESET}}$, or another pin which operates in the same manner as the $\overline{\text{PRESET}}$ pin. However, when connected to a ground pin, the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground pin. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

The TCK clock or the CPG of this LSI should be set to ensure that the frequency of the TCK clock is less than the peripheral-clock frequency of this LSI.

42.3 Boundary Scan TAP Controllers (IDCODE, EXTEST, SAMPLE/PRELOAD, and BYPASS)

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan function and another for controlling the H-UDI reset and interrupt functions. Assertion of $\overline{\text{TRST}}$, for example at power-on reset, activates the boundary-scan TAP controller and enables the boundary-scan function prescribed in the JTAG standards. Executing a switchover command to the H-UDI allows usage of the H-UDI reset and H-UDI interrupts. This LSI, however, has the following limitations:

- Clock-related pins (EXTAL, XTAL, EXTAL2, and XTAL2) are out of the scope of the boundary-scan test.
- Reset-related pins ($\overline{\text{PRESET}}$, $\overline{\text{MRESET}}$) are out of the scope of the boundary-scan test.
- H-UDI-related pins (TCK, TDI, TDO, TMS, TRST and MPMD) are out of the scope of the boundary-scan test.
- DDRIF-related pins are out of the scope of the boundary-scan test
- $\overline{\text{XRTCTBI}}$, USBP, USBM, DA0, DA1, and AN0 to AN3 pins are out of the scope of the boundary-scan test
- During the boundary scan (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, and H-UDI switchover command), the maximum TCK signal frequency is 2 MHz.
- The external controller has 8-bit access to the boundary-scan TAP controller via the H-UDI.

Note: During the boundary scan, the $\overline{\text{PRESET}}$ pin should be fixed high-level.

Table 42.2 shows the commands supported by the boundary-scan TAP controller.

Table 42.2 Commands Supported by Boundary-Scan TAP Controller

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
0	1	0	1	0	1	0	1	IDCODE
1	1	1	1	1	1	1	1	BYPASS
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	0	0	0	1	0	0	0	H-UDI (switchover command)
Other than above								Setting prohibited

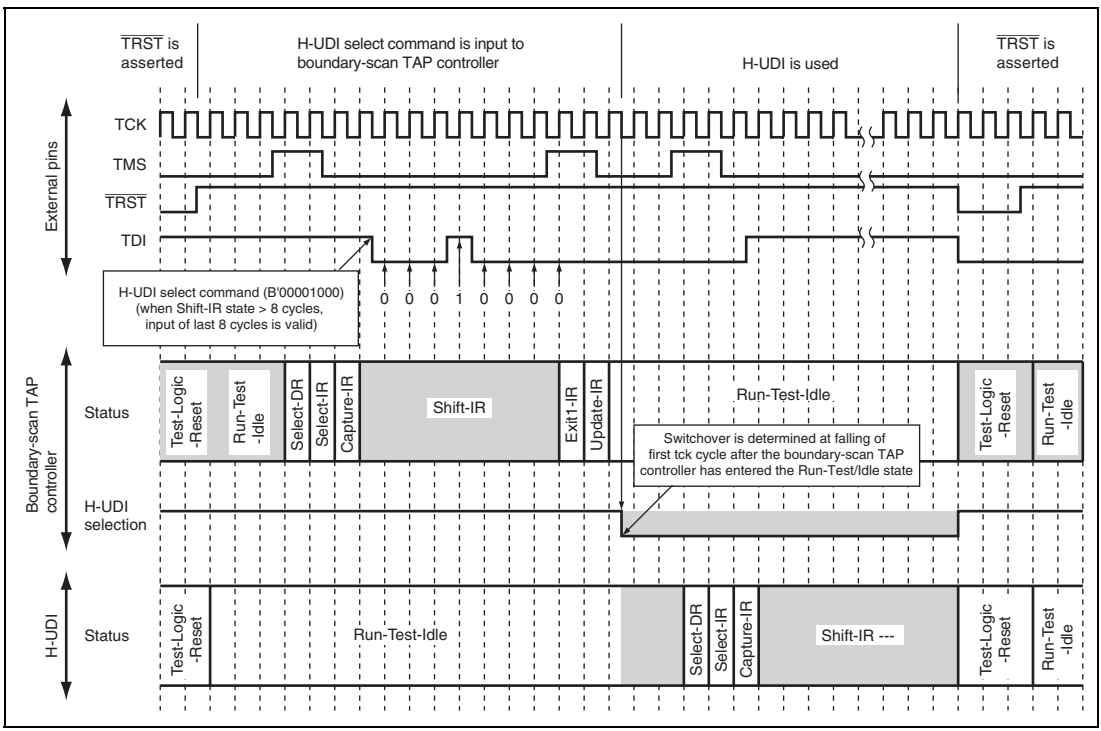


Figure 42.2 Sequence for switching from Boundary-Scan TAP Controller to H-UDI

42.4 Register Descriptions

The H-UDI has the following registers.

Table 42.3 Register Configuration (1)

		CPU Side				
Register Name	**Abbrev.**	**R/W**	**Area P4 Address*[1]**	**Area 7 Address*[1]**	**Size**	**Initial Value*[2]**
Instruction register	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
Interrupt source register	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
Boundary scan register	SDBSR	—	—	—	—	—
Bypass register	SDBPR	—	—	—	—	—

Notes: 1. The area P4 address is an address when accessing through area P4 in a virtual address space. The area 7 address is an address when accessing through area 7 in a physical space using the TLB.
2. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP controller initializes to these values.

Table 42.4 Register Configuration (2)

		H-UDI Side		
Register Name	**Abbrev.**	**R/W**	**Size**	**Initial Value*[1]**
Instruction register	SDIR	R/W	32	H'FFFF FFFD (fixed value*[2])
Interrupt source register	SDINT	W*[3]	32	H'0000 0000
Boundary scan register	SDBSR	—	—	—
Bypass register	SDBPR	R/W	1	Undefined

Note: 1. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP controller initializes to these values.
2. When reading via the H-UDI, the value is always H'FFFF FFFD.
3. Only 1 can be written to the LSB by the H-UDI interrupt command.

Table 42.5 Register Status in Each Processing State

Register Name	**Abbrev.**	**Power-On Reset**	**Manual Reset**	**Sleep**	**Standby**
Instruction register	SDIR	H'0EFF	Retained	Retained	Retained
Interrupt source register	SDINT	H'0000	Retained	Retained	Retained

42.4.1 Instruction Register (SDIR)

SDIR is a 16-bit read-only register that can be read from the CPU. Commands are set via the serial input (TDI). SDIR is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

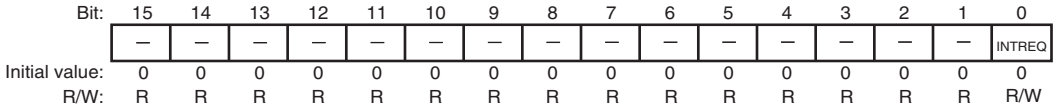
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI	0000 1110	R	Test Instruction Bits 7 to 0 0110 xxxx: H-UDI reset negate 0111 xxxx: H-UDI reset assert 101x xxxx: H-UDI interrupt 0000 1110: Initial state Other than above: Setting prohibited Note: Though H-UDI reset asserted, CPG, watchdog/reset and part of RTC registers are not initialized.
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

42.4.2 Interrupt Source Register (SDINT)

SDINT is a 16-bit register that can be read from or written to by the CPU. Specifying an H-UDI interrupt command in SDIR via H-UDI pin (Update-IR) sets the INTREQ bit to 1. While an H-UDI interrupt command is set in SDIR, SDINT which is connected between the TDI and TDO pins can be read as a 32-bit register. In this case, the upper 16 bits will be 0 and the lower 16 bits represent the SDINT value.

Only 0 can be written to the INTREQ bit by the CPU. While this bit is set to 1, an interrupt request will continue to be generated. This bit, therefore, should be cleared by the interrupt handling routine. It is initialized by $\overline{\text{TRST}}$ or in the Test-Logic-Reset state.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTREQ
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved For reading from or writing to this bit, see General Precautions on Handling of Product.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. When writing 1 to this bit, the previous value is maintained.

42.4.3 Bypass Register (SDBPR)

SDBPR is a one-bit register that supports the J-TAG bypass mode. When the BYPASS command is set to the boundary scan TAP controller, the TDI and TDO are connected by way of SDBPR. This register cannot be accessed from the CPU regardless of the LSI mode. Though this register is not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted, initialized to 0 in the Capture-DR state.

42.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/Output pins, which supports the boundary scan mode of the JTAG standard.

Using the EXTEST and SAMPLE/PRELOAD commands, a boundary-scan test complying with the JTAG standards (IEEE1149.1) can be carried out.

This register cannot be accessed from the CPU regardless of the LSI mode.

This register is not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted.

Table 42.6 SDBSR Configuration

Number	Pin Name	I/O*
	From TDI	
517	$\overline{\text{BACK}}$	OUTPUT
516	$\overline{\text{BACK}}$	CONTROL
515	$\overline{\text{BACK}}$	INPUT
514	$\overline{\text{BREQ}}$	OUTPUT
513	$\overline{\text{BREQ}}$	CONTROL
512	$\overline{\text{BREQ}}$	INPUT
511	$\overline{\text{IOIS16}}$/TMU_TCLK	OUTPUT
510	$\overline{\text{IOIS16}}$/TMU_TCLK	CONTROL
509	$\overline{\text{IOIS16}}$/TMU_TCLK	INPUT
508	$\overline{\text{CE2A}}$	OUTPUT
507	$\overline{\text{CE2A}}$	CONTROL
506	$\overline{\text{CE2A}}$	INPUT
505	$\overline{\text{CE2B}}$	OUTPUT
504	$\overline{\text{CE2B}}$	CONTROL
503	$\overline{\text{CE2B}}$	INPUT
502	A25/EX_SIZE2	OUTPUT
501	A25/EX_SIZE2	CONTROL
500	A25/EX_SIZE2	INPUT
499	A24/EX_SIZE1	OUTPUT
498	A24/EX_SIZE1	CONTROL
497	A24/EX_SIZE1	INPUT
496	A23/EX_SIZE0	OUTPUT
495	A23/EX_SIZE0	CONTROL
494	A23/EX_SIZE0	INPUT
493	A22	OUTPUT
492	A22	CONTROL
491	A22	INPUT
490	A21	OUTPUT
489	A21	CONTROL

Number	Pin Name	I/O*
488	A21	INPUT
487	A20	OUTPUT
486	A20	CONTROL
485	A20	INPUT
484	A15	OUTPUT
483	A15	CONTROL
482	A15	INPUT
481	A14	OUTPUT
480	A14	CONTROL
479	A14	INPUT
478	A19	OUTPUT
477	A19	CONTROL
476	A19	INPUT
475	A18	OUTPUT
474	A18	CONTROL
473	A18	INPUT
472	A13	OUTPUT
471	A13	CONTROL
470	A13	INPUT
469	A12	OUTPUT
468	A12	CONTROL
467	A12	INPUT
466	A17	OUTPUT
465	A17	CONTROL
464	A17	INPUT
463	A16	OUTPUT
462	A16	CONTROL
461	A16	INPUT
460	A7	OUTPUT
459	A7	CONTROL
458	A7	INPUT
457	A6	OUTPUT

Number	Pin Name	I/O*
456	A6	CONTROL
455	A6	INPUT
454	A11	OUTPUT
453	A11	CONTROL
452	A11	INPUT
451	A10	OUTPUT
450	A10	CONTROL
449	A10	INPUT
448	A5	OUTPUT
447	A5	CONTROL
446	A5	INPUT
445	A4	OUTPUT
444	A4	CONTROL
443	A4	INPUT
442	A9	OUTPUT
441	A9	CONTROL
440	A9	INPUT
439	A8	OUTPUT
438	A8	CONTROL
437	A8	INPUT
436	A3	OUTPUT
435	A3	CONTROL
434	A3	INPUT
433	A2	OUTPUT
432	A2	CONTROL
431	A2	INPUT
430	A1	OUTPUT
429	A1	CONTROL
428	A1	INPUT
427	A0	OUTPUT
426	A0	CONTROL
425	A0	INPUT

Number	Pin Name	I/O*
424	$\overline{\text{RD}}$/FRAME/EX_FRAME	OUTPUT
423	$\overline{\text{RD}}$/FRAME/EX_FRAME	CONTROL
422	$\overline{\text{RD}}$/FRAME/EX_FRAME	INPUT
421	$\overline{\text{WE0}}$/PCC_REG	OUTPUT
420	$\overline{\text{WE0}}$/PCC_REG	CONTROL
419	$\overline{\text{WE0}}$/PCC_REG	INPUT
418	D1/EX_AD1	OUTPUT
417	D1/EX_AD1	CONTROL
416	D1/EX_AD1	INPUT
415	D0/EX_AD0	OUTPUT
414	D0/EX_AD0	CONTROL
413	D0/EX_AD0	INPUT
412	$\overline{\text{WE1}}$/$\overline{\text{WE}}$	OUTPUT
411	$\overline{\text{WE1}}$/$\overline{\text{WE}}$	CONTROL
410	$\overline{\text{WE1}}$/$\overline{\text{WE}}$	INPUT
409	CLKOUT	CONTROL
408	CLKOUT	OUTPUT
407	D3/EX_AD3	OUTPUT
406	D3/EX_AD3	CONTROL
405	D3/EX_AD3	INPUT
404	D2/EX_AD2	OUTPUT
403	D2/EX_AD2	CONTROL
402	D2/EX_AD2	INPUT
401	D9/EX_AD9	OUTPUT
400	D9/EX_AD9	CONTROL
399	D9/EX_AD9	INPUT
398	D8/EX_AD8	OUTPUT
397	D8/EX_AD8	CONTROL
396	D8/EX_AD8	INPUT
395	D5/EX_AD5	OUTPUT
394	D5/EX_AD5	CONTROL
393	D5/EX_AD5	INPUT

Number	Pin Name	I/O*
392	D4/EX_AD4	OUTPUT
391	D4/EX_AD4	CONTROL
390	D4/EX_AD4	INPUT
389	D11/EX_AD11	OUTPUT
388	D11/EX_AD11	CONTROL
387	D11/EX_AD11	INPUT
386	D10/EX_AD10	OUTPUT
385	D10/EX_AD10	CONTROL
384	D10/EX_AD10	INPUT
383	D7/EX_AD7	OUTPUT
382	D7/EX_AD7	CONTROL
381	D7/EX_AD7	INPUT
380	D6/EX_AD6	OUTPUT
379	D6/EX_AD6	CONTROL
378	D6/EX_AD6	INPUT
377	D13/EX_AD13	OUTPUT
376	D13/EX_AD13	CONTROL
375	D13/EX_AD13	INPUT
374	D12/EX_AD12	OUTPUT
373	D12/EX_AD12	CONTROL
372	D12/EX_AD12	INPUT
371	PTL0/D16/EX_AD16/IRQ4/$\overline{\text{IRL4}}$/ET0_COL/$\overline{\text{DREQ0}}$/LCD_D8	OUTPUT
370	PTL0/D16/EX_AD16/IRQ4/$\overline{\text{IRL4}}$/ET0_COL/$\overline{\text{DREQ0}}$/LCD_D8	CONTROL
369	PTL0/D16/EX_AD16/IRQ4/$\overline{\text{IRL4}}$/ET0_COL/$\overline{\text{DREQ0}}$/LCD_D8	INPUT
368	PTL1/D17/EX_AD17/IRQ5/$\overline{\text{IRL5}}$/ET0_MDC/$\overline{\text{DACK0}}$/LCD_D9	OUTPUT
367	PTL1/D17/EX_AD17/IRQ5/$\overline{\text{IRL5}}$/ET0_MDC/$\overline{\text{DACK0}}$/LCD_D9	CONTROL
366	PTL1/D17/EX_AD17/IRQ5/$\overline{\text{IRL5}}$/ET0_MDC/$\overline{\text{DACK0}}$/LCD_D9	INPUT
365	D15/EX_AD15	OUTPUT
364	D15/EX_AD15	CONTROL
363	D15/EX_AD15	INPUT
362	D14/EX_AD14	OUTPUT
361	D14/EX_AD14	CONTROL

Number	Pin Name	I/O*
360	D14/EX_AD14	INPUT
359	PTL3/D19/EX_AD19/IRQ7/$\overline{\text{IRL7}}$/ET0_MDIO/$\overline{\text{INTC}}$/LCD_D11	OUTPUT
358	PTL3/D19/EX_AD19/IRQ7/$\overline{\text{IRL7}}$/ET0_MDIO/$\overline{\text{INTC}}$/LCD_D11	CONTROL
357	PTL3/D19/EX_AD19/IRQ7/$\overline{\text{IRL7}}$/ET0_MDIO/$\overline{\text{INTC}}$/LCD_D11	INPUT
356	PTL2/D18/EX_AD18/IRQ6/$\overline{\text{IRL6}}$/ET0_ETXD3/$\overline{\text{TEND0}}$/LCD_D10	OUTPUT
355	PTL2/D18/EX_AD18/IRQ6/$\overline{\text{IRL6}}$/ET0_ETXD3/$\overline{\text{TEND0}}$/LCD_D10	CONTROL
354	PTL2/D18/EX_AD18/IRQ6/$\overline{\text{IRL6}}$/ET0_ETXD3/$\overline{\text{TEND0}}$/LCD_D10	INPUT
353	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	OUTPUT
352	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	CONTROL
351	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	INPUT
350	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	OUTPUT
349	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	CONTROL
348	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	INPUT
347	PTK1/ST1_D1/GET0_ETXD5/SIOF1_TXD/LCD_D3	OUTPUT
346	PTK1/ST1_D1/GET0_ETXD5/SIOF1_TXD/LCD_D3	CONTROL
345	PTK1/ST1_D1/GET0_ETXD5/SIOF1_TXD/LCD_D3	INPUT
344	PTK0/ST1_D0/GET0_ETXD4/SIOF1_SYNC/LCD_D2	OUTPUT
343	PTK0/ST1_D0/GET0_ETXD4/SIOF1_SYNC/LCD_D2	CONTROL
342	PTK0/ST1_D0/GET0_ETXD4/SIOF1_SYNC/LCD_D2	INPUT
341	PTL4/D20/EX_AD20/ST0_REQ/ET0_ETXD0/$\overline{\text{INTD}}$/LCD_D12	OUTPUT
340	PTL4/D20/EX_AD20/ST0_REQ/ET0_ETXD0/$\overline{\text{INTD}}$/LCD_D12	CONTROL
339	PTL4/D20/EX_AD20/ST0_REQ/ET0_ETXD0/$\overline{\text{INTD}}$/LCD_D12	INPUT
338	PTJ0/ST0M_REQO/GET0_GTX-CLK/REF50CK	OUTPUT
337	PTJ0/ST0M_REQO/GET0_GTX-CLK/REF50CK	CONTROL
336	PTJ0/ST0M_REQO/GET0_GTX-CLK/REF50CK	INPUT
335	PTK3/ST1_D3/GET0_ETXD7/SIOF2_SYNC/LCD_D5	OUTPUT
334	PTK3/ST1_D3/GET0_ETXD7/SIOF2_SYNC/LCD_D5	CONTROL
333	PTK3/ST1_D3/GET0_ETXD7/SIOF2_SYNC/LCD_D5	INPUT
332	PTK2/ST1_D2/GET0_ETXD6/SIOF1_SCK/LCD_D4	OUTPUT
331	PTK2/ST1_D2/GET0_ETXD6/SIOF1_SCK/LCD_D4	CONTROL
330	PTK2/ST1_D2/GET0_ETXD6/SIOF1_SCK/LCD_D4	INPUT
329	PTL6/D22/EX_AD22/ST0_START/ET0_ETXD2/$\overline{\text{DACK1}}$/LCD_D14	OUTPUT

Number	Pin Name	I/O*
328	PTL6/D22/EX_AD22/ST0_START/ET0_ETXD2/$\overline{\text{DACK1}}$/LCD_D14	CONTROL
327	PTL6/D22/EX_AD22/ST0_START/ET0_ETXD2/$\overline{\text{DACK1}}$/LCD_D14	INPUT
326	PTL5/D21/EX_AD21/ST0_CLK/ST0_STRB/ET0_ETXD1/$\overline{\text{DREQ1}}$/LCD_D13	OUTPUT
325	PTL5/D21/EX_AD21/ST0_CLK/ST0_STRB/ET0_ETXD1/$\overline{\text{DREQ1}}$/LCD_D13	CONTROL
324	PTL5/D21/EX_AD21/ST0_CLK/ST0_STRB/ET0_ETXD1/$\overline{\text{DREQ1}}$/LCD_D13	INPUT
323	$\overline{\text{RDWR}}$/$\overline{\text{EX_RDWR}}$	OUTPUT
322	$\overline{\text{RDWR}}$/$\overline{\text{EX_RDWR}}$	CONTROL
321	$\overline{\text{RDWR}}$/$\overline{\text{EX_RDWR}}$	INPUT
320	PTM0/D24/EX_AD24/ST0_D0/ET0_TX-ER/PINT0/RMII0M0_MDIO	OUTPUT
319	PTM0/D24/EX_AD24/ST0_D0/ET0_TX-ER/PINT0/RMII0M0_MDIO	CONTROL
318	PTM0/D24/EX_AD24/ST0_D0/ET0_TX-ER/PINT0/RMII0M0_MDIO	INPUT
317	PTL7/D23/EX_AD23/ST0_VALID/ET0_TX-EN/$\overline{\text{TEND1}}$/LCD_D15	OUTPUT
316	PTL7/D23/EX_AD23/ST0_VALID/ET0_TX-EN/$\overline{\text{TEND1}}$/LCD_D15	CONTROL
315	PTL7/D23/EX_AD23/ST0_VALID/ET0_TX-EN/$\overline{\text{TEND1}}$/LCD_D15	INPUT
314	$\overline{\text{BS}}$/$\overline{\text{EX_BS}}$	OUTPUT
313	$\overline{\text{BS}}$/$\overline{\text{EX_BS}}$	CONTROL
312	$\overline{\text{BS}}$/$\overline{\text{EX_BS}}$	INPUT
311	PTM2/D26/EX_AD26/ST0_D2/ET0_WOL/RMII0_CRS_DV/PINT2	OUTPUT
310	PTM2/D26/EX_AD26/ST0_D2/ET0_WOL/RMII0_CRS_DV/PINT2	CONTROL
309	PTM2/D26/EX_AD26/ST0_D2/ET0_WOL/RMII0_CRS_DV/PINT2	INPUT
308	PTM1/D25/EX_AD25/ST0_D1/ET0_TX-CLK/RMII0_RX_ER/PINT1	OUTPUT
307	PTM1/D25/EX_AD25/ST0_D1/ET0_TX-CLK/RMII0_RX_ER/PINT1	CONTROL
306	PTM1/D25/EX_AD25/ST0_D1/ET0_TX-CLK/RMII0_RX_ER/PINT1	INPUT
305	REF125CK/SSI_CLK/HAC_BITCLK	OUTPUT
304	REF125CK/SSI_CLK/HAC_BITCLK	CONTROL
303	REF125CK/SSI_CLK/HAC_BITCLK	INPUT
302	PTM3/D27/EX_AD27/ST0_D3/ET0_LINKSTA/RMII0_RXD1/PINT3	OUTPUT
301	PTM3/D27/EX_AD27/ST0_D3/ET0_LINKSTA/RMII0_RXD1/PINT3	CONTROL
300	PTM3/D27/EX_AD27/ST0_D3/ET0_LINKSTA/RMII0_RXD1/PINT3	INPUT

Number	Pin Name	I/O*
299	$\overline{CS0}$	OUTPUT
298	$\overline{CS0}$	CONTROL
297	$\overline{CS0}$	INPUT
296	PTM5/D29/EX_AD29/ST0_D5/ET0_RX-ER/RMII0_TXD_EN/PINT5	OUTPUT
295	PTM5/D29/EX_AD29/ST0_D5/ET0_RX-ER/RMII0_TXD_EN/PINT5	CONTROL
294	PTM5/D29/EX_AD29/ST0_D5/ET0_RX-ER/RMII0_TXD_EN/PINT5	INPUT
293	PTM4/D28/EX_AD28/ST0_D4/ET0_PHY-INT/RMII0_RXD0/PINT4	OUTPUT
292	PTM4/D28/EX_AD28/ST0_D4/ET0_PHY-INT/RMII0_RXD0/PINT4	CONTROL
291	PTM4/D28/EX_AD28/ST0_D4/ET0_PHY-INT/RMII0_RXD0/PINT4	INPUT
290	PTM7/D31/EX_AD31/ST0_D7/ET0_RX-DV/RMII0_TXD0/PINT7	OUTPUT
289	PTM7/D31/EX_AD31/ST0_D7/ET0_RX-DV/RMII0_TXD0/PINT7	CONTROL
288	PTM7/D31/EX_AD31/ST0_D7/ET0_RX-DV/RMII0_TXD0/PINT7	INPUT
287	PTM6/D30/EX_AD30/ST0_D6/ET0_RX-CLK/RMII0_TXD1/PINT6	OUTPUT
286	PTM6/D30/EX_AD30/ST0_D6/ET0_RX-CLK/RMII0_TXD1/PINT6	CONTROL
285	PTM6/D30/EX_AD30/ST0_D6/ET0_RX-CLK/RMII0_TXD1/PINT6	INPUT
284	$\overline{CS1}$/EX_CS0	OUTPUT
283	$\overline{CS1}$/EX_CS0	CONTROL
282	$\overline{CS1}$/EX_CS0	INPUT
281	$\overline{CS2}$/EX_CS1	OUTPUT
280	$\overline{CS2}$/EX_CS1	CONTROL
279	$\overline{CS2}$/EX_CS1	INPUT
278	$\overline{CS5}$/$\overline{CE1A}$	OUTPUT
277	$\overline{CS5}$/$\overline{CE1A}$	CONTROL
276	$\overline{CS5}$/$\overline{CE1A}$	INPUT
275	$\overline{CS4}$	OUTPUT
274	$\overline{CS4}$	CONTROL
273	$\overline{CS4}$	INPUT
272	$\overline{CS6}$/$\overline{CE1B}$	OUTPUT
271	$\overline{CS6}$/$\overline{CE1B}$	CONTROL
270	$\overline{CS6}$/$\overline{CE1B}$	INPUT
269	$\overline{RDY}$/EX_RDY/$\overline{PCC_WAIT}$	OUTPUT
268	$\overline{RDY}$/EX_RDY/$\overline{PCC_WAIT}$	CONTROL

Number	Pin Name	I/O*
267	$\overline{\text{RDY}}$/EX_RDY/PCC_$\overline{\text{WAIT}}$	INPUT
266	PTJ1/ST0M_CLKIO/ST0M_STRBI/RMII1_RX_ER/LCD_CLK	OUTPUT
265	PTJ1/ST0M_CLKIO/ST0M_STRBI/RMII1_RX_ER/LCD_CLK	CONTROL
264	PTJ1/ST0M_CLKIO/ST0M_STRBI/RMII1_RX_ER/LCD_CLK	INPUT
263	PTJ2/ST0M_D0I/ET0_ERXD0/RMII1_TXD1/LCD_M_DISP	OUTPUT
262	PTJ2/ST0M_D0I/ET0_ERXD0/RMII1_TXD1/LCD_M_DISP	CONTROL
261	PTJ2/ST0M_D0I/ET0_ERXD0/RMII1_TXD1/LCD_M_DISP	INPUT
260	PTJ3/ST0M_D1I/ET0_ERXD1/RMII1_CRS_DV/LCD_CL1	OUTPUT
259	PTJ3/ST0M_D1I/ET0_ERXD1/RMII1_CRS_DV/LCD_CL1	CONTROL
258	PTJ3/ST0M_D1I/ET0_ERXD1/RMII1_CRS_DV/LCD_CL1	INPUT
257	PTJ4/ST0M_D2I/ET0_ERXD2/RMII1_RXD1/LCD_CL2	OUTPUT
256	PTJ4/ST0M_D2I/ET0_ERXD2/RMII1_RXD1/LCD_CL2	CONTROL
255	PTJ4/ST0M_D2I/ET0_ERXD2/RMII1_RXD1/LCD_CL2	INPUT
254	PTJ5/ST0M_D3I/ET0_ERXD3/RMII1_RXD0/LCD_DON	OUTPUT
253	PTJ5/ST0M_D3I/ET0_ERXD3/RMII1_RXD0/LCD_DON	CONTROL
252	PTJ5/ST0M_D3I/ET0_ERXD3/RMII1_RXD0/LCD_DON	INPUT
251	PTJ6/ST0M_D4I/ET0_CRS/RMII1_TXD_EN/LCD_FLM	OUTPUT
250	PTJ6/ST0M_D4I/ET0_CRS/RMII1_TXD_EN/LCD_FLM	CONTROL
249	PTJ6/ST0M_D4I/ET0_CRS/RMII1_TXD_EN/LCD_FLM	INPUT
248	PTJ7/$\overline{\text{INTB}}$/ST0M_D5I/$\overline{\text{IRQOUT}}$/RMII1_TXD0/LCD_D0	OUTPUT
247	PTJ7/$\overline{\text{INTB}}$/ST0M_D5I/$\overline{\text{IRQOUT}}$/RMII1_TXD0/LCD_D0	CONTROL
246	PTJ7/$\overline{\text{INTB}}$/ST0M_D5I/$\overline{\text{IRQOUT}}$/RMII1_TXD0/LCD_D0	INPUT
245	PTI6/IRQ2/$\overline{\text{IRL2}}$/ST0M_D6I/IIC1_SCL	OUTPUT
244	PTI6/IRQ2/$\overline{\text{IRL2}}$/ST0M_D6I/IIC1_SCL	INPUT
243	PTI7/IRQ3/$\overline{\text{IRL3}}$/ST0M_D7I/IIC1_SDA	OUTPUT
242	PTI7/IRQ3/IRL3/ST0M_D7I/IIC1_SDA	INPUT
241	PTI4/MD8/ST1_START/ET1_PHY-INT/RMII0M0_MDC/USB_PWREN/USBF_UPLUP	OUTPUT
240	PTI4/MD8/ST1_START/ET1_PHY-INT/RMII0M0_MDC/USB_PWREN/USBF_UPLUP	CONTROL
239	PTI4/MD8/ST1_START/ET1_PHY-INT/RMII0M0_MDC/USB_PWREN/USBF_UPLUP	INPUT
238	PTI5/MD10/ST1_VALID/LCD_D1	OUTPUT

Number	Pin Name	I/O*
237	PTI5/MD10/ST1_VALID/LCD_D1	CONTROL
236	PTI5/MD10/ST1_VALID/LCD_D1	INPUT
235	PTK4/ST1_D4/GET0_ERXD4/SIOF2_TXD/LCD_D6	OUTPUT
234	PTK4/ST1_D4/GET0_ERXD4/SIOF2_TXD/LCD_D6	CONTROL
233	PTK4/ST1_D4/GET0_ERXD4/SIOF2_TXD/LCD_D6	INPUT
232	PTK5/ST1_D5/GET0_ERXD5/SIOF2_RXD/LCD_D7	OUTPUT
231	PTK5/ST1_D5/GET0_ERXD5/SIOF2_RXD/LCD_D7	CONTROL
230	PTK5/ST1_D5/GET0_ERXD5/SIOF2_RXD/LCD_D7	INPUT
229	PTK6/ST1_D6/GET0_ERXD6/SIOF2_SCK/LCD_VEPWC	OUTPUT
228	PTK6/ST1_D6/GET0_ERXD6/SIOF2_SCK/LCD_VEPWC	CONTROL
227	PTK6/ST1_D6/GET0_ERXD6/SIOF2_SCK/LCD_VEPWC	INPUT
226	PTK7/ST1_D7/GET0_ERXD7/SIOF2_MCLK/LCD_VCPWC	OUTPUT
225	PTK7/ST1_D7/GET0_ERXD7/SIOF2_MCLK/LCD_VCPWC	CONTROL
224	PTK7/ST1_D7/GET0_ERXD7/SIOF2_MCLK/LCD_VCPWC	INPUT
223	PTI0/STATUS0/ST1_CLK/ST1_STRB/RMII0_MDC	OUTPUT
222	PTI0/STATUS0/ST1_CLK/ST1_STRB/RMII0_MDC	CONTROL
221	PTI0/STATUS0/ST1_CLK/ST1_STRB/RMII0_MDC	INPUT
220	PTI1/STATUS1/ST1_REQ/RMII0_MDIO	OUTPUT
219	PTI1/STATUS1/ST1_REQ/RMII0_MDIO	CONTROL
218	PTI1/STATUS1/ST1_REQ/RMII0_MDIO	INPUT
217	PTI2/ST0M_STARTI/IIC0_SCL/SIOF1_RXD/$\overline{\text{USB_OVRCRT}}$ /USBF_VBUS	OUTPUT
216	PTI2/ST0M_STARTI/IIC0_SCL/SIOF1_RXD/$\overline{\text{USB_OVRCRT}}$ /USBF_VBUS	INPUT
215	PTI3/ST0M_VALIDI/IIC0_SDA/SIOF1_MCLK/USB_CLK	OUTPUT
214	PTI3/ST0M_VALIDI/IIC0_SDA/SIOF1_MCLK/USB_CLK	INPUT
213	PTF0/$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$/SIM_D/ET1_ETXD3/$\overline{\text{DREQ3}}$	OUTPUT
212	PTF0/$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$/SIM_D/ET1_ETXD3/$\overline{\text{DREQ3}}$	CONTROL
211	PTF0/$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$/SIM_D/ET1_ETXD3/$\overline{\text{DREQ3}}$	INPUT
210	PTG3/$\overline{\text{REQ3}}$/ET1_ETXD2	OUTPUT
209	PTG3/$\overline{\text{REQ3}}$/ET1_ETXD2	CONTROL
208	PTG3/$\overline{\text{REQ3}}$/ET1_ETXD2	INPUT

Number	Pin Name	I/O*
207	PTG2/$\overline{\text{REQ1}}$/ET1_ETXD1	OUTPUT
206	PTG2/$\overline{\text{REQ1}}$/ET1_ETXD1	CONTROL
205	PTG2/$\overline{\text{REQ1}}$/ET1_ETXD1	INPUT
204	PTG1/$\overline{\text{GNT2}}$/ET1_ETXD0	OUTPUT
203	PTG1/$\overline{\text{GNT2}}$/ET1_ETXD0	CONTROL
202	PTG1/$\overline{\text{GNT2}}$/ET1_ETXD0	INPUT
201	PTD7/$\overline{\text{PCIRESET}}$/PCC_RESET/GET1_ETXD7/LCDM_VEPWC	OUTPUT
200	PTE0/$\overline{\text{INTA}}$/$\overline{\text{PCC_DRV}}$/GET1_ETXD6/$\overline{\text{DREQ2}}$	OUTPUT
199	PTE0/$\overline{\text{INTA}}$/$\overline{\text{PCC_DRV}}$/GET1_ETXD6/$\overline{\text{DREQ2}}$	CONTROL
198	PTE0/$\overline{\text{INTA}}$/$\overline{\text{PCC_DRV}}$/GET1_ETXD6/$\overline{\text{DREQ2}}$	INPUT
197	PTD6/$\overline{\text{REQ2}}$/PCC_BVD1/GET1_ETXD5/SSI1_SCK/ LCDM_VCPWC	OUTPUT
196	PTD6/$\overline{\text{REQ2}}$/PCC_BVD1/GET1_ETXD5/SSI1_SCK/ LCDM_VCPWC	CONTROL
195	PTD6/$\overline{\text{REQ2}}$/PCC_BVD1/GET1_ETXD5/SSI1_SCK/ LCDM_VCPWC	INPUT
194	PTE1/PCICLK/GET1_ETXD4/$\overline{\text{DACK2}}$	OUTPUT
193	PTE1/PCICLK/GET1_ETXD4/$\overline{\text{DACK2}}$	CONTROL
192	PTE1/PCICLK/GET1_ETXD4/$\overline{\text{DACK2}}$	INPUT
191	PTG4/AD30/ET1_LINKSTA	OUTPUT
190	PTG4/AD30/ET1_LINKSTA	CONTROL
189	PTG4/AD30/ET1_LINKSTA	INPUT
188	PTG0/$\overline{\text{GNT1}}$/ET1_WOL	OUTPUT
187	PTG0/$\overline{\text{GNT1}}$/ET1_WOL	CONTROL
186	PTG0/$\overline{\text{GNT1}}$/ET1_WOL	INPUT
185	PTF2/AD31/SIM_RST/ET1_MDIO/$\overline{\text{TEND3}}$	OUTPUT
184	PTF2/AD31/SIM_RST/ET1_MDIO/$\overline{\text{TEND3}}$	CONTROL
183	PTF2/AD31/SIM_RST/ET1_MDIO/$\overline{\text{TEND3}}$	INPUT
182	PTF1/$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$/SIM_CLK/ET1_MDC/$\overline{\text{DACK3}}$	OUTPUT
181	PTF1/$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$/SIM_CLK/ET1_MDC/$\overline{\text{DACK3}}$	CONTROL
180	PTF1/$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$/SIM_CLK/ET1_MDC/$\overline{\text{DACK3}}$	INPUT
179	PTG6/AD26/ET1_TX-ER	OUTPUT

Number	Pin Name	I/O*
178	PTG6/AD26/ET1_TX-ER	CONTROL
177	PTG6/AD26/ET1_TX-ER	INPUT
176	PTG7/AD28/ET1_TX-EN	OUTPUT
175	PTG7/AD28/ET1_TX-EN	CONTROL
174	PTG7/AD28/ET1_TX-EN	INPUT
173	PTE5/AD29/SCIF2_TXD/GET1_GTX-CLK/SSI0_SCK	OUTPUT
172	PTE5/AD29/SCIF2_TXD/GET1_GTX-CLK/SSI0_SCK	CONTROL
171	PTE5/AD29/SCIF2_TXD/GET1_GTX-CLK/SSI0_SCK	INPUT
170	PTH0/AD25/TPU_TI3A/ET1_COL/RMII1M_RX_ER	OUTPUT
169	PTH0/AD25/TPU_TI3A/ET1_COL/RMII1M_RX_ER	CONTROL
168	PTH0/AD25/TPU_TI3A/ET1_COL/RMII1M_RX_ER	INPUT
167	PTH6/AD27/TPU_TO2/ET1_CRS/RMII1M_TXD_EN	OUTPUT
166	PTH6/AD27/TPU_TO2/ET1_CRS/RMII1M_TXD_EN	CONTROL
165	PTH6/AD27/TPU_TO2/ET1_CRS/RMII1M_TXD_EN	INPUT
164	PTF3/CBE3/ET1_TX-CLK	OUTPUT
163	PTF3/CBE3/ET1_TX-CLK	CONTROL
162	PTF3/CBE3/ET1_TX-CLK	INPUT
161	PTG5/$\overline{\text{GNT3}}$/ET1_RX-CLK	OUTPUT
160	PTG5/$\overline{\text{GNT3}}$/ET1_RX-CLK	CONTROL
159	PTG5/$\overline{\text{GNT3}}$/ET1_RX-CLK	INPUT
158	PTH2/AD24/TPU_TI2A/ET1_ERXD0/RMII1M_TXD1	OUTPUT
157	PTH2/AD24/TPU_TI2A/ET1_ERXD0/RMII1M_TXD1	CONTROL
156	PTH2/AD24/TPU_TI2A/ET1_ERXD0/RMII1M_TXD1	INPUT
155	PTH5/AD23/TPU_TO1/ET1_ERXD1/RMII1M_TXD0	OUTPUT
154	PTH5/AD23/TPU_TO1/ET1_ERXD1/RMII1M_TXD0	CONTROL
153	PTH5/AD23/TPU_TO1/ET1_ERXD1/RMII1M_TXD0	INPUT
152	PTH1/IDSEL/TPU_TI3B/ET1_RX-ER/RMII1M_CRS_DV	OUTPUT
151	PTH1/IDSEL/TPU_TI3B/ET1_RX-ER/RMII1M_CRS_DV	CONTROL
150	PTH1/IDSEL/TPU_TI3B/ET1_RX-ER/RMII1M_CRS_DV	INPUT
149	PTH3/AD21/TPU_TI2B/ET1_ERXD2/RMII1M_RXD1	OUTPUT
148	PTH3/AD21/TPU_TI2B/ET1_ERXD2/RMII1M_RXD1	CONTROL

Number	Pin Name	I/O*
147	PTH3/AD21/TPU_TI2B/ET1_ERXD2/RMII1M_RXD1	INPUT
146	PTH4/AD19/TPU_TO0/ET1_ERXD3/RMII1M_RXD0	OUTPUT
145	PTH4/AD19/TPU_TO0/ET1_ERXD3/RMII1M_RXD0	CONTROL
144	PTH4/AD19/TPU_TO0/ET1_ERXD3/RMII1M_RXD0	INPUT
143	PTE4/AD22/SCIF2_RXD/GET1_ERXD4/SSI0_SDATA	OUTPUT
142	PTE4/AD22/SCIF2_RXD/GET1_ERXD4/SSI0_SDATA	CONTROL
141	PTE4/AD22/SCIF2_RXD/GET1_ERXD4/SSI0_SDATA	INPUT
140	PTE3/AD20/SCIF2_SCK/GET1_ERXD5/SSI0_WS	OUTPUT
139	PTE3/AD20/SCIF2_SCK/GET1_ERXD5/SSI0_WS	CONTROL
138	PTE3/AD20/SCIF2_SCK/GET1_ERXD5/SSI0_WS	INPUT
137	PTH7/AD17/TPU_TO3/ET1_RX-DV	OUTPUT
136	PTH7/AD17/TPU_TO3/ET1_RX-DV	CONTROL
135	PTH7/AD17/TPU_TO3/ET1_RX-DV	INPUT
134	PTD1/CBE2/$\overline{\text{PCC_VS2}}$/SIOF0_TXD/HAC_SD_OUT/LCDM_D15	OUTPUT
133	PTD1/CBE2/$\overline{\text{PCC_VS2}}$/SIOF0_TXD/HAC_SD_OUT/LCDM_D15	CONTROL
132	PTD1/CBE2/$\overline{\text{PCC_VS2}}$/SIOF0_TXD/HAC_SD_OUT/LCDM_D15	INPUT
131	PTD5/AD18/$\overline{\text{PCC_CD2}}$/GET1_ERXD6/SSI1_SDATA/LCDM_D14	OUTPUT
130	PTD5/AD18/$\overline{\text{PCC_CD2}}$/GET1_ERXD6/SSI1_SDATA/LCDM_D14	CONTROL
129	PTD5/AD18/$\overline{\text{PCC_CD2}}$/GET1_ERXD6/SSI1_SDATA/LCDM_D14	INPUT
128	PTE2/AD16/$\overline{\text{PCC_IOIS16}}$/GET1_ERXD7/$\overline{\text{TEND2}}$	OUTPUT
127	PTE2/AD16/$\overline{\text{PCC_IOIS16}}$/GET1_ERXD7/$\overline{\text{TEND2}}$	CONTROL
126	PTE2/AD16/$\overline{\text{PCC_IOIS16}}$/GET1_ERXD7/$\overline{\text{TEND2}}$	INPUT
125	PTD0/$\overline{\text{IRDY}}$/$\overline{\text{PCC_VS1}}$/SIOF0_SYNC/HAC_SD_IN/LCDM_D13	OUTPUT
124	PTD0/$\overline{\text{IRDY}}$/$\overline{\text{PCC_VS1}}$/SIOF0_SYNC/HAC_SD_IN/LCDM_D13	CONTROL
123	PTD0/$\overline{\text{IRDY}}$/$\overline{\text{PCC_VS1}}$/SIOF0_SYNC/HAC_SD_IN/LCDM_D13	INPUT
122	PTA1/$\overline{\text{DEVSEL}}$/SCIF1_RXD	OUTPUT
121	PTA1/$\overline{\text{DEVSEL}}$/SCIF1_RXD	CONTROL
120	PTA1/$\overline{\text{DEVSEL}}$/SCIF1_RXD	INPUT
119	PTD3/PCIFRAME/PCC-BVD2/SIOFO_SCK/HAC_RES/LCDM_D12	OUTPUT
118	PTD3/PCIFRAME/PCC-BVD2/SIOFO_SCK/HAC_RES/LCDM_D12	CONTROL
117	PTD3/PCIFRAME/PCC-BVD2/SIOFO_SCK/HAC_RES/LCDM_D12	INPUT

Number	Pin Name	I/O*
116	PTD2/$\overline{\text{TRDY}}$/PCC_RDY/SIOF0_RXD/HAC_SYNC/LCDM_D11	OUTPUT
115	PTD2/$\overline{\text{TRDY}}$/PCC_RDY/SIOF0_RXD/HAC_SYNC/LCDM_D11	CONTROL
114	PTD2/$\overline{\text{TRDY}}$/PCC_RDY/SIOF0_RXD/HAC_SYNC/LCDM_D11	INPUT
113	PTA2/$\overline{\text{LOCK}}$/SCIF1_TXD*	OUTPUT
112	PTA2/$\overline{\text{LOCK}}$/SCIF1_TXD	CONTROL
111	PTA2/$\overline{\text{LOCK}}$/SCIF1_TXD	INPUT
110	PTB0/$\overline{\text{PERR}}$/PINT8/LCDM_D10	OUTPUT
109	PTB0/$\overline{\text{PERR}}$/PINT8/LCDM_D10	CONTROL
108	PTB0/$\overline{\text{PERR}}$/PINT8/LCDM_D10	INPUT
107	PTD4/$\overline{\text{STOP}}$/$\overline{\text{PCC_CD1}}$/SIOF0_MCLK/SSI1_WS/LCDM_DON	OUTPUT
106	PTD4/$\overline{\text{STOP}}$/$\overline{\text{PCC_CD1}}$/SIOF0_MCLK/SSI1_WS/LCDM_DON	CONTROL
105	PTD4/$\overline{\text{STOP}}$/$\overline{\text{PCC_CD1}}$/SIOF0_MCLK/SSI1_WS/LCDM_DON	INPUT
104	PTA0/PAR/SCIF1_SCK	OUTPUT
103	PTA0/PAR/SCIF1_SCK	CONTROL
102	PTA0/PAR/SCIF1_SCK	INPUT
101	PTB1/$\overline{\text{SERR}}$/PINT9/LCDM_D9	OUTPUT
100	PTB1/$\overline{\text{SERR}}$/PINT9/LCDM_D9	CONTROL
99	PTB1/$\overline{\text{SERR}}$/PINT9/LCDM_D9	INPUT
98	PTB4/CBE1/PINT12/LCDM_D8	OUTPUT
97	PTB4/CBE1/PINT12/LCDM_D8	CONTROL
96	PTB4/CBE1/PINT12/LCDM_D8	INPUT
95	PTA3/AD15/$\overline{\text{SCIF1_CTS}}$	OUTPUT
94	PTA3/AD15/$\overline{\text{SCIF1_CTS}}$	CONTROL
93	PTA3/AD15/$\overline{\text{SCIF1_CTS}}$	INPUT
92	PTA4/AD13/$\overline{\text{SCIF1_RTS}}$	OUTPUT
91	PTA4/AD13/$\overline{\text{SCIF1_RTS}}$	CONTROL
90	PTA4/AD13/$\overline{\text{SCIF1_RTS}}$	INPUT
89	PTB5/AD14/PINT13/LCDM_M_DISP	OUTPUT
88	PTB5/AD14/PINT13/LCDM_M_DISP	CONTROL
87	PTB5/AD14/PINT13/LCDM_M_DISP	INPUT
86	PTA5/AD12	OUTPUT
85	PTA5/AD12	CONTROL

Number	Pin Name	I/O*
84	PTA5/AD12	INPUT
83	PTB2/AD11/PINT10/LCDM_D7	OUTPUT
82	PTB2/AD11/PINT10/LCDM_D7	CONTROL
81	PTB2/AD11/PINT10/LCDM_D7	INPUT
80	PTB3/AD9/PINT11/LCDM_D6	OUTPUT
79	PTB3/AD9/PINT11/LCDM_D6	CONTROL
78	PTB3/AD9/PINT11/LCDM_D6	INPUT
77	PTC0/AD10/MMC_DAT/LCDM_D5	OUTPUT
76	PTC0/AD10/MMC_DAT/LCDM_D5	CONTROL
75	PTC0/AD10/MMC_DAT/LCDM_D5	INPUT
74	PTC3/AD8/$\overline{\text{MMC_ODMOD}}$/LCDM_D4	OUTPUT
73	PTC3/AD8/$\overline{\text{MMC_ODMOD}}$/LCDM_D4	CONTROL
72	PTC3/AD8/$\overline{\text{MMC_ODMOD}}$/LCDM_D4	INPUT
71	PTB6/CBE0/PINT14/LCDM_D3	OUTPUT
70	PTB6/CBE0/PINT14/LCDM_D3	CONTROL
69	PTB6/CBE0/PINT14/LCDM_D3	INPUT
68	PTB7/AD6/PINT15/LCDM_D2	OUTPUT
67	PTB7/AD6/PINT15/LCDM_D2	CONTROL
66	PTB7/AD6/PINT15/LCDM_D2	INPUT
65	PTC4/AD7/MMC_CMD/LCDM_CL2	OUTPUT
64	PTC4/AD7/MMC_CMD/LCDM_CL2	CONTROL
63	PTC4/AD7/MMC_CMD/LCDM_CL2	INPUT
62	PTC6/AD5/LCDM_CL1	OUTPUT
61	PTC6/AD5/LCDM_CL1	CONTROL
60	PTC6/AD5/LCDM_CL1	INPUT
59	PTC1/AD4/LCDM_D1	OUTPUT
58	PTC1/AD4/LCDM_D1	CONTROL
57	PTC1/AD4/LCDM_D1	INPUT
56	PTC2/AD2/LCDM_D0	OUTPUT
55	PTC2/AD2/LCDM_D0	CONTROL
54	PTC2/AD2/LCDM_D0	INPUT
53	PTC7/AD3/MMC_CLK	OUTPUT

Number	Pin Name	I/O*
52	PTC7/AD3/MMC_CLK	CONTROL
51	PTC7/AD3/MMC_CLK	INPUT
50	PTA6/AD1/MMC_VDDON	OUTPUT
49	PTA6/AD1/MMC_VDDON	CONTROL
48	PTA6/AD1/MMC_VDDON	INPUT
47	PTC5/AD0/MMC_CD/LCDM_FLM	OUTPUT
46	PTC5/AD0/MMC_CD/LCDM_FLM	CONTROL
45	PTC5/AD0/MMC_CD/LCDM_FLM	INPUT
44	PTN0/SCIF0_SCK/MD0	OUTPUT
43	PTN0/SCIF0_SCK/MD0	CONTROL
42	PTN0/SCIF0_SCK/MD0	INPUT
41	PTN1/SCIF0_RXD/MD3	OUTPUT
40	PTN1/SCIF0_RXD/MD3	CONTROL
39	PTN1/SCIF0_RXD/MD3	INPUT
38	PTN2/SCIF0_TXD/MD1	OUTPUT
37	PTN2/SCIF0_TXD/MD1	CONTROL
36	PTN2/SCIF0_TXD/MD1	INPUT
35	PTN3/$\overline{\text{SCIF0_CTS}}$/MD4	OUTPUT
34	PTN3/$\overline{\text{SCIF0_CTS}}$/MD4	CONTROL
33	PTN3/$\overline{\text{SCIF0_CTS}}$/MD4	INPUT
32	PTN4/$\overline{\text{SCIF0_RTS}}$/MD2	OUTPUT
31	PTN4/$\overline{\text{SCIF0_RTS}}$/MD2	CONTROL
30	PTN4/$\overline{\text{SCIF0_RTS}}$/MD2	INPUT
29	PTN5/NMI	OUTPUT
28	PTN5/NMI	CONTROL
27	PTN5/NMI	INPUT
26	PTO6/IRQ0/$\overline{\text{IRL0}}$/$\overline{\text{DACK1M}}$/MD5	OUTPUT
25	PTO6/IRQ0/$\overline{\text{IRL0}}$/$\overline{\text{DACK1M}}$/MD5	CONTROL
24	PTO6/IRQ0/$\overline{\text{IRL0}}$/$\overline{\text{DACK1M}}$/MD5	INPUT
23	PTO7/IRQ1/$\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/SSI3_SCK/MD6	OUTPUT
22	PTO7/IRQ1/$\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/SSI3_SCK/MD6	CONTROL
21	PTO7/IRQ1/$\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/SSI3_SCK/MD6	INPUT

Number	Pin Name	I/O*
20	PTO0/$\overline{\text{AUDSYNCS}}$/RMII_MDC/SSI2_WS	OUTPUT
19	PTO0/$\overline{\text{AUDSYNCS}}$/RMII_MDC/SSI2_WS	CONTROL
18	PTO0/$\overline{\text{AUDSYNCS}}$/RMII_MDC/SSI2_WS	INPUT
17	PTO1/AUDATA0/RMII_MDIO/SSI2_SDATA	OUTPUT
16	PTO1/AUDATA0/RMII_MDIO/SSI2_SDATA	CONTROL
15	PTO1/AUDATA0/RMII_MDIO/SSI2_SDATA	INPUT
14	PTO2/AUDATA1/RMII0M1_MDC	OUTPUT
13	PTO2/AUDATA1/RMII0M1_MDC	CONTROL
12	PTO2/AUDATA1/RMII0M1_MDC	INPUT
11	PTO3/AUDATA2/RMII0M1_MDIO/SSI2_SCK	OUTPUT
10	PTO3/AUDATA2/RMII0M1_MDIO/SSI2_SCK	CONTROL
9	PTO3/AUDATA2/RMII0M1_MDIO/SSI2_SCK	INPUT
8	PTO4/AUDATA3/$\overline{\text{EX_INT}}$/SSI3_WS	OUTPUT
7	PTO4/AUDATA3/$\overline{\text{EX_INT}}$/SSI3_WS	CONTROL
6	PTO4/AUDATA3/$\overline{\text{EX_INT}}$/SSI3_WS	INPUT
5	PTO5/AUDCK/$\overline{\text{DREQ1M}}$/SSI3_SDATA	OUTPUT
4	PTO5/AUDCK/$\overline{\text{DREQ1M}}$/SSI3_SDATA	CONTROL
3	PTO5/AUDCK/$\overline{\text{DREQ1M}}$/SSI3_SDATA	INPUT
2	$\overline{\text{ASEBRK}}$/BRKACK	OUTPUT
1	$\overline{\text{ASEBRK}}$/BRKACK	CONTROL
0	$\overline{\text{ASEBRK}}$/BRKACK	INPUT
	To TDO	

Note: * Control is an active-high signal. When Control is driven high, the corresponding pin is driven according to the OUT value.

42.5 Operation

42.5.1 TAP Control

Figure 42.3 shows the internal states of the TAP controller. The state transitions basically conform to the JTAG standard.

- State transitions occur according to the TMS value at the rising edge of the TCK signal.
- The TDI value is sampled at the rising edge of the TCK signal and shifted at the falling edge of the TCK signal.
- The TDO value is changed at the falling edge of the TCK signal. The TDO signal is in a Hi-Z state other than in the Shift-DR or Shift-IR state.
- A transition to the Test-Logic-Reset by clearing $\overline{\text{TRST}}$ to 0 is performed asynchronously with the TCK signal.

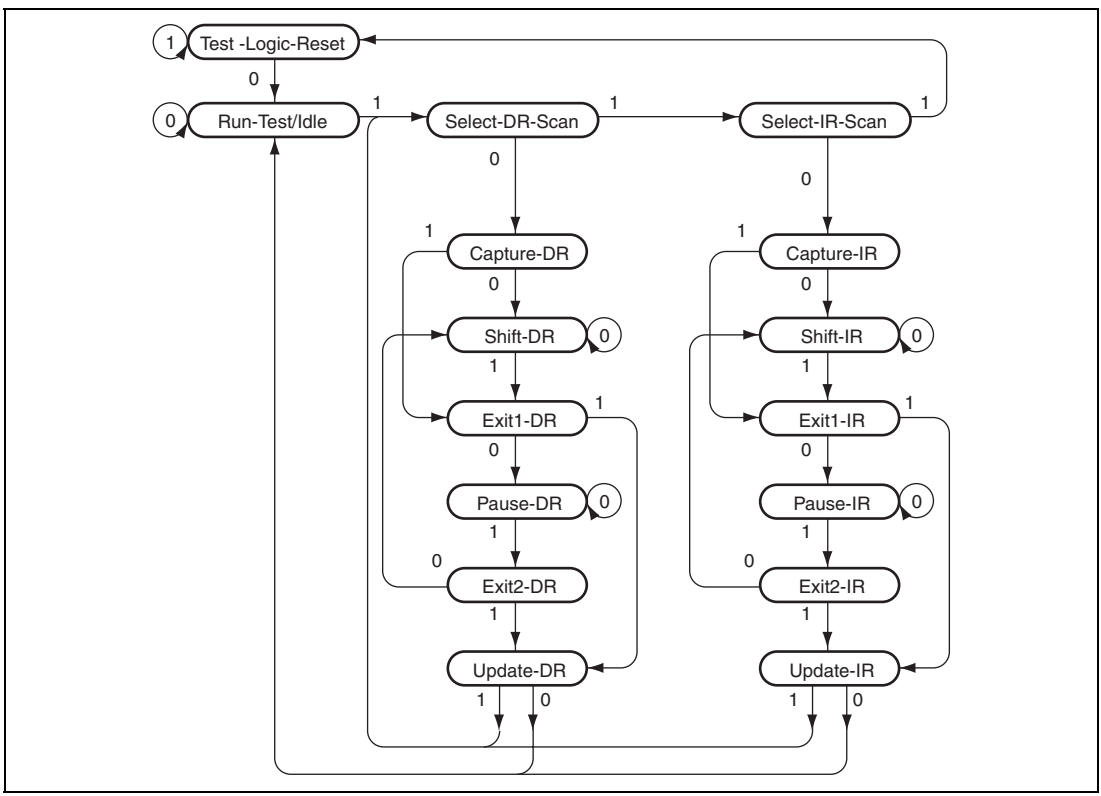


Figure 42.3 TAP Controller State Transitions

42.5.2 H-UDI Reset

A power-on reset is generated by the SDIR command. After the H-UDI reset assert command has been sent from the H-UDI pin, sending the H-UDI reset negate command resets the CPU (see figure 42.4). The required time between the H-UDI reset assert and H-UDI reset negate commands is the same as the time for holding the reset pin low in order to reset this LSI by a power-on reset.

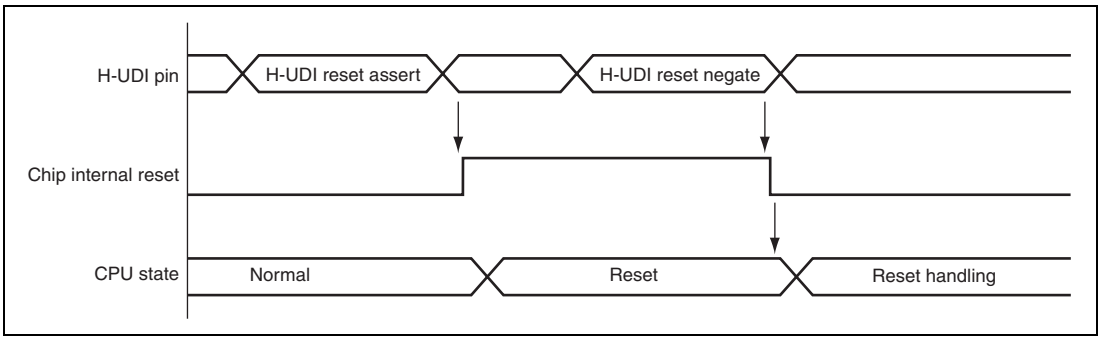


Figure 42.4 H-UDI Reset

42.5.3 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting the appropriate command in SDIR from the H-UDI. An H-UDI interrupt request signal is asserted when the INTREQ bit in SDINT is set to 1 by setting the appropriate command. Since the interrupt request signal is not negated until the INTREQ bit is cleared to 0 by software, it is not possible to lose the interrupt request. While an H-UDI interrupt command is set in SDIR, SDINT is connected between the TDI and TDO pins.

42.6 Usage Notes

Once an SDIR command is set, it will be changed only by an assertion of the $\overline{\text{TRST}}$ signal, making the TAP controller Test-Logic-Reset state, or writing other commands from the H-UDI.

The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

Section 43 Electrical Characteristics

43.1 Absolute Maximum Ratings

Table 43.1 Absolute Maximum Ratings[*1, *2]

Item	Symbol	Value	Unit
I/O power supply voltage	VDDQ	–0.3 to 4.6	V
	VDD-RTC		
	VCCQ-DDR	–0.3 to 2.8	
Internal power supply voltage	VDD	–0.3 to 1.8	V
	$V_{DD\text{-}PLL1/2/3}$		
	$V_{DD\text{-}DLL1/2}$		
Analog power supply voltage	AVcc	–0.3 to 4.6	V
Input voltage	V_{in}	–0.3 to VCCQ + 0.3[*3]	V
		–0.3 to VDD-RTC + 0.3[*3]	
	$V_{in\text{-}DDR}$	–0.3 to VCCQ-DDR + 0.3[*3]	
Analog input voltage	V_{AN}	–0.3 to AVcc + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{stg}	–55 to 125	°C

Notes: 1. The LSI may be permanently damaged if the maximum ratings are exceeded.
2. The LSI may be permanently damaged if any of the V_{SS} pins are not connected to GND.
3. The upper limit of the input voltage must not exceed the power supply voltage.

43.2 Power-On and Power-Off Order

43.2.1 Power-On Order

There are no restrictions on the power-on order. After a single power supply is turned on, all other power supplies should be turned on within 10 ms. It is recommended that this time period is as short as possible. The system should be designed so that a system malfunction is not caused by the undefined states of pins and internal circuits.

The internal circuit states are undefined until a low level is input to the $\overline{\text{RESETP}}$ pin after voltage is applied to all power supplies. While the internal circuit states are undefined, the state of each pin is also undefined. Accordingly, the system should be designed so that a system malfunction is not caused by such undefined state.

43.2.2 Power-Off Order

There are no restrictions on the power-off order. After a single power supply is turned off, all other power supplies should be turned off within 10 ms. It is recommended that this time period is as short as possible.

The system should be designed so that a system malfunction is not caused by the undefined states of pins and internal circuits.

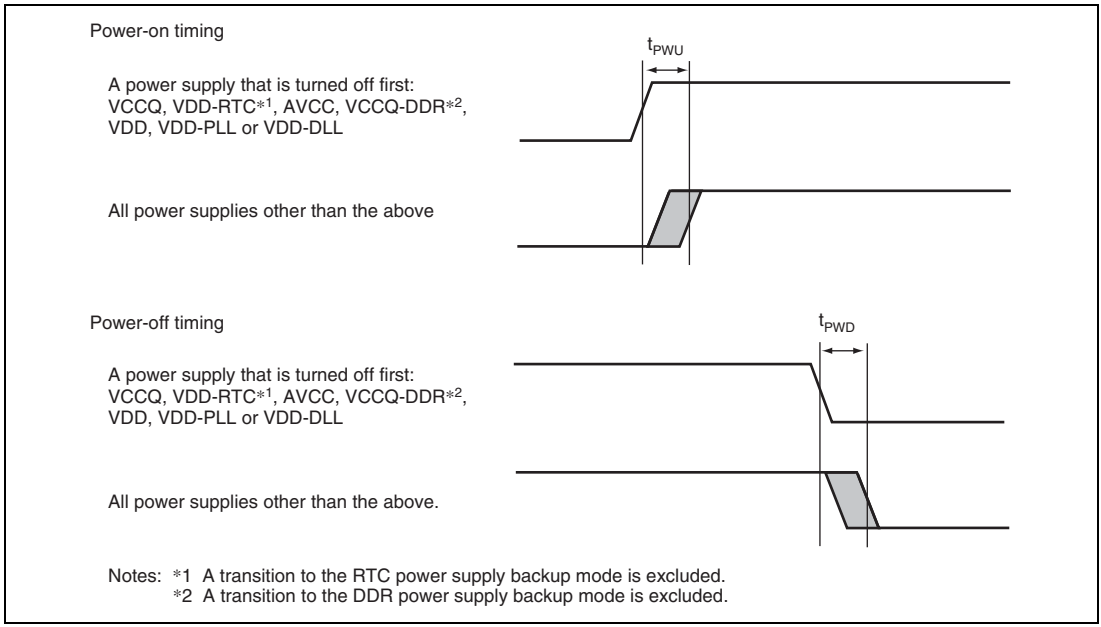


Figure 43.1 Power-On and Power-Off Timing

Table 43.2 Power-On and Power-Off Timing

Item	Symbol	Maximum time	Unit
Time lags among powering on (VCCQ, VDD-RTC, AVCC), (VCCQ-DDR), and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3)	t_{PWU}	10	ms
Time lags among powering off (VCCQ, VDD-RTC, AVCC), (VCCQ-DDR), and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3)	t_{PWD}	10	ms

43.2.3 Power-Off and Power-On Order in RTC Power-Supply Backup Mode (Hardware Standby)

To use RTC power supply backup mode, the RTC clock should be supplied.

First bring the $\overline{\text{XRTCSTBI}}$ pin low, and then make sure that the STATUS0 and STATUS1 pins have been pulled high and low, respectively. After that, turn off the power supplies (VCCQ, AVCC), (VCCQ-DDR), and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3). The power supply VDD-RTC should remain on, and the $\overline{\text{XRTCSTBI}}$ pin should remain low.

Turn on the power supplies (VCCQ, AVCC), (VCCQ-DDR), and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3) while the $\overline{\text{XRTCSTBI}}$ pin is low. After these power supplies become stable, bring the XRTCSTBI pin high and negate the $\overline{\text{PRESET}}$ pin to high level.

43.2.4 Power-Off and Power-On Order in DDR-SDRAM Power-Supply Backup Mode

To use DDR-SDRAM power-supply backup mode, the DDR-SDRAM should be placed in the self-refresh state. After the DDR-SDRAM is placed in the self-refresh state, bring the M-CKE pin low. Make sure that the SELFS bit in the MIM register is set to 1, and then bring the $\overline{\text{M_BKPRST}}$ pin low. After that, turn off the power supplies (VCCQ, AVCC, VDD-RTC) and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3). The power supply VCCQ-DDR should remain on and the $\overline{\text{M_BKPRST}}$ pin should remain low.

Turn on the power supplies (VCCQ, AVCC, VDD-RTC) and (VDD, VDD-PLL1 to 3, VDD-DLL1 to 3) while the $\overline{\text{M_BKPRST}}$ pin is low. After these power supplies become stable, negate the $\overline{\text{M_BKPRST}}$ and $\overline{\text{PRESET}}$ pins to high level.

43.3 DC Characteristics

Table 43.3 DC Characteristics (1) [common]

Condition: T_a = –20 to 75°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		VCCQ	3.0	3.3	3.6	V	
		VDD-RTC	3.0	3.3	3.6	V	
		VCCQ-DDR	2.3	2.5	2.7	V	
		VDD $V_{DD-PLL1/2/3}$ $V_{DD-DLL1/2}$	1.15	1.25	1.35	V	
Analog power supply voltage		AV_{CC}	3.0	3.3	3.6	V	When not in use the same voltage as VCCQ
Reference voltage		DDR-V_{REF}	1.15	1.25	1.35	V	
Current dissipation	Normal operation	I_{DD}	—	950	1200	mA	Ick = 266MHz
		I_{DD-PLL}	—	—	10	mA	
		I_{DD-DLL}	—	—	12	mA	
		I_{CCQ}	—	200	300	mA	
		I_{DD-RTC}	—	—	0.9	mA	VDD-RTC = 3.3V
		$I_{CCQ-DDR}$	—	—	250	mA	DDRck = 133MHz
	Sleep mode	I_{DD}	—	—	800	mA	Ick = 266MHz
		I_{CCQ}	—	—	25	mA	
	RTC backup mode	I_{DD-RTC}	—	—	50	µA	
	DDR backup mode	$I_{CCQ-DDR}$	—	—	155	µA	
Analog power supply current	A/D conversion period	AI_{CC}	—	—	30	mA	
	A/D and D/A conversion period			—	30		
	Idle		—	—	20		

Note: Note that a heat radiation countermeasure, such as a heat sinks, is required when the ambient temperature exceeds 60 degrees.

Table 43.4 DC Characteristics (2-a) [Except of USB Transceiver and I²C Related Pins]

Condition: T_a = –20 to 75°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	Input pin group*[1]	V_{IH}	VCCQ × 0.9	—	VCCQ + 0.3	V	VCCQ = 3.0 to 3.6 V
	DDR pins		DDR-V_{REF} + 0.15	—	VCCQ-DDR + 0.3		DDR-V_{REF} = 1.15 to 1.35V VCCQ-DDR = 2.3 to 2.7V
	PCICLK		VCCQ × 0.6	—	VCCQ + 0.3		VCCQ = 3.0 to 3.6V
	Other PCI pins		VCCQ × 0.5	—	VCCQ + 0.3		
	Other input pins		2	—	VCCQ + 0.3		
	Input pin group*[1]	V_{IL}	–0.3	—	VCCQ × 0.1	V	VCCQ = 3.0 to 3.6 V
	DDR pins		–0.3	—	DDR-V_{REF} – 0.18		DDR-V_{REF} = 1.15 to 1.35V VCCQ-DDR = 2.3 to 2.7V
	M_BKPRST		–0.3	—	VCCQ-DDR × 0.2		
	PCICLK		–0.3	—	VCCQ × 0.2		VCCQ = 3.0 to 3.6V
	Other PCI pins		–0.3	—	VCCQ × 0.3		
	Other input pins		–0.3	—	VCCQ × 0.2		
Input leak current	DDR pins	|L|	—	—	2	µA	V_{IN} = 0.5 to VCCQ-DDR – 0.5V
	All input pins	|Iin|	—	—	1		V_{IN} = 0.5 to VCCQ – 0.5V
Output voltage	PCI pins	V_{OH}	2.4	—	—	V	VCCQ = 3.0 to 3.6V I_{OH} = –4mA
	DDR pins		1.84	—	—		VCCQ-DDR = 2.3V I_{OH} = –7.6mA
	Other output pins		2.4	—	—		VCCQ = 3.0 to 3.6V I_{OH} = –2mA

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output voltage	PCI pins	V_{OL}	—	—	0.55	V	VCCQ = 3.0 to 3.6V I_{OL} = 4mA
	DDR pins		—	—	0.54		VCCQ-DDR = 2.3 to 2.7 V I_{OL} = 7.6mA
	Other output pins		—	—	0.55		VCCQ = 3.0 to 3.6V I_{OL} = 2mA
Pull-up resistance	All pins	R_{pull}	20	60	180	kΩ	
Pin capacitance	DDR pins	C_L	—	—	5	pF	
	Other pins		—	—	10		

Notes: 1. Input pin group: EXTAL, EXTAL2, PRESET, XRTCSTBI, MRESET, TRST, MD0 to MD6, MD8, MD10, MPMD, PTN0 to PTN5, PTO6, PTO7, PTI2 to PTI7, NMI, IRQ0 to IRQ7, IRL0 to IRL7, PINT0 to PINT15, DACK1M, SSI3_SCK, SSI_CLK, SIOF1_MCLK, SIOF1_RXD, SCIF0_SCK, SCIF0_RTS, SCIF0_RXD, SCIF0_CTS, ET1_PHY-INT, ST1_START, ST1_VALID, ST0M_VALIDI, ST0M_D7I, ST0M_D6I, REF125CK, HAC_BITCLK, USB_CLK, USB_OVRCRT/USBF_VBUS

2. The current dissipation values are for V_{IH} min = V_{DDQ} –0.5 V and V_{IL} max = 0.5 V with all output pins unload.

Table 43.5 DC Characteristics (2-b) [I²C Related Pins]

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage	VCCQ	3.0	3.3	3.6	V	
Input high voltage	V_{IH}	VCCQ × 0.7	—	VCCQ + 0.3	V	
Input low voltage	V_{IL}	–0.3	—	VCCQ × 0.3	V	
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 3 mA
Permissible output low current	I_{OL}	—	—	10	mA	

Note: I²C related pins: IIC0_SCL, IIC1_SCL, IIC0_SDA, and IIC1_SDA pins (open-drain pins).

Table 43.6 DC Characteristics (2-c) [USB Transceiver Related Pins]

Condition: Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage	VCCQ	3.0	3.3	3.6	V	
Differential input sensitivity	V_{DI}	0.2	—	—	V	|(DP) – (DM)|
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Single ended receiver threshold voltage	V_{SE}	0.8	—	2.0	V	
Output high voltage	V_{OH}	2.5	—	VCCQ	V	
Output low voltage	V_{OL}	—	—	0.3	V	
Tray state leakage voltage	I_{LO}	–10	—	10	µA	$0V < V_{IN} < 3.3V$

Note: Transceiver related pins: USBP, USBM.

Table 43.7 Permissible Output Currents

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin; DDR1 pins)	I_{OL}	—	—	16	mA
Permissible output low current (per pin; PCI1 pins)		—	—	4	
Permissible output low current (per pin; other than DDR and PCI pins)		—	—	2	
Permissible output low current (total)	ΣI_{OL}	—	—	120	mA
Permissible output high current (per pin; DDR1 pins)	$-I_{OH}$	—	—	16	mA
Permissible output high current (per pin; PCI1 pins)		—	—	4	
Permissible output high current (per pin; other than DDR and PCI pins)		—	—	2	
Permissible output high current (total)	$\Sigma\lvert-I_{OH}\rvert$	—	—	40	mA

Note: To protect chip reliability, do not exceed the output current values in table 43.7.

43.4 AC Characteristics

In principle, this LSI's input should be synchronous. Unless specified otherwise, ensure that the setup time and hold times for each input signal are observed.

Table 43.8 Maximum Operating Frequency

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	CPU, FPU, cache, TLB	f	200	—	267	MHz
	DDR-SDRAM bus		100	—	134	
	External bus		50	—	67	
	PCI bus		32	—	67	
	Peripheral modules 0		50	—	67	
	Peripheral modules 1		25	—	34	
	RTC oscillator		32	—	33	kHz

43.4.1 Clock and Control Signal Timing

Table 43.9 Clock and Control Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency PLL1 PLL2 operation*[1]	f_{EX}	25	33.4	MHz	
EXTAL clock input cycle time	t_{EXcyc}	30	40	ns	43.2
EXTAL clock input low-level pulse width	t_{EXL}	3.5	—	ns	43.2
EXTAL clock input high-level pulse width	t_{EXH}	3.5	—	ns	43.2
EXTAL clock input rise time	t_{EXr}	—	4	ns	43.2
EXTAL clock input fall time	t_{EXf}	—	4	ns	43.2
CLKOUT clock output PLL1/PLL2 operation*[2]	t_{OP}	50	67	MHz	
CLKOUT clock output cycle time	$t_{CLKOUTcyc}$	15	20	ns	43.3
CLKOUT clock output low-level pulse width	$t_{CLKOUTL1}$	3	—	ns	43.3
CLKOUT clock output high-level pulse width	$t_{CLKOUTH1}$	3	—	ns	43.3
CLKOUT clock output rise time	$t_{CLKOUTr}$	—	3	ns	43.3
CLKOUT clock output fall time	$t_{CLKOUTf}$	—	3	ns	43.3
CLKOUT clock output low-level pulse width	$t_{CLKOUTL2}$	3	—	ns	43.4
CLKOUT clock output high-level pulse width	$t_{CLKOUTH2}$	3	—	ns	43.4
Power-on oscillation settling time	t_{OSC1}	30	—	ms	43.5
Power-on oscillation settling time/mode settling time	t_{OSCMD}	30	—	ms	43.5
MDn reset hold time	t_{MDRH}	0	—	ns	43.5
$\overline{\text{TRST}}$ reset hold time	t_{TRSTRH}	0	—	ns	43.5
Reset holding time	t_{RESH}	0	—	ms	43.5
$\overline{\text{PRESET}}$ pulse width	t_{RESPW}	20	—	t_{cyc}*[3]	43.8
Power-on RTC oscillation settling time	$t_{RTC-OSC}$	—	3	s	
PLL synchronization settling time	t_{PLL}	200	—	µs	43.6
Oscillation settling time on return from standby 2	T_{SOC2}	10	—	ms	43.7
$\overline{\text{MRESET}}$ pulse width	t_{RESMW}	20	—	t_{cyc}*[3]	43.8

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{MRESET}}$ setup time	t_{RESMS}	23	—	ns	43.8
$\overline{\text{MRESET}}$ hold time	t_{RESMH}	2	—	ns	43.8

Notes: 1. When a crystal resonator is connected to EXTAL and XTAL, the maximum frequency is 34MHz. when a 3rd overtone crystal resonator is used, an external tank circuit is necessary.
2. The load capacitance connected to the CLKOUT pin should be a maximum of 50 pF.
3. t_{cyc} shows 1 cycle time of a CLKOUT clock.

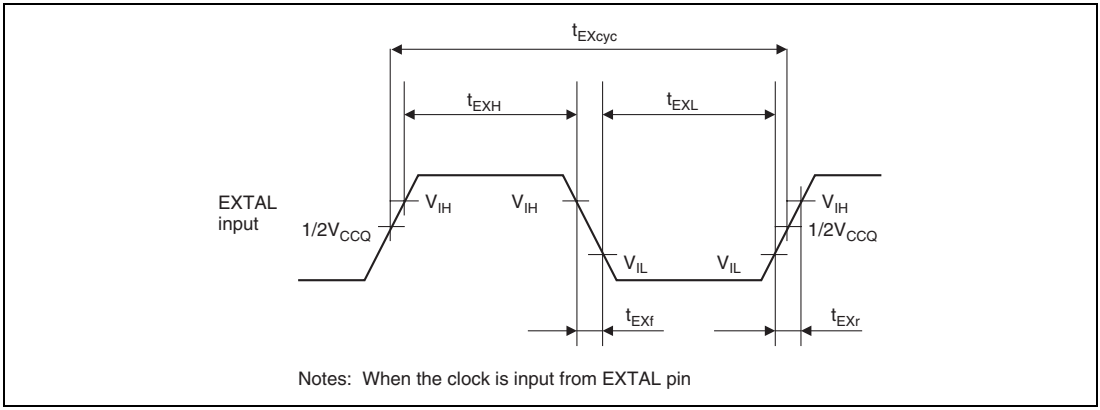


Figure 43.2 EXTAL Clock Input Timing

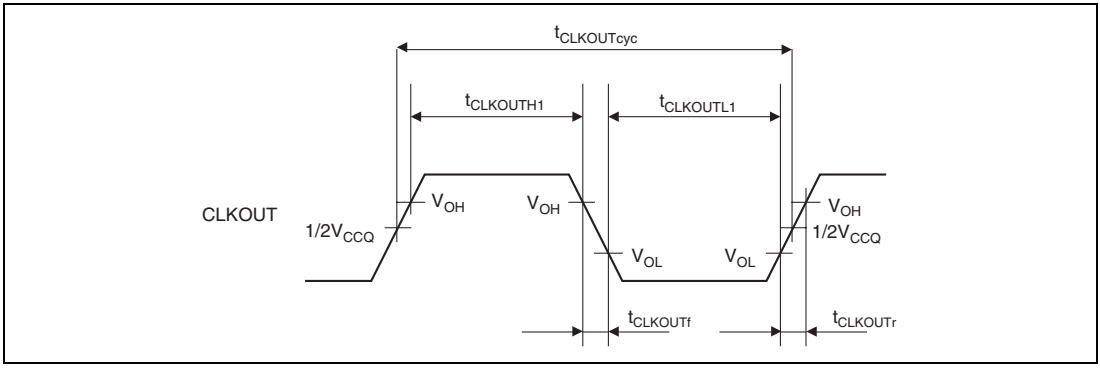


Figure 43.3 CLKOUT Clock Output Timing (1)

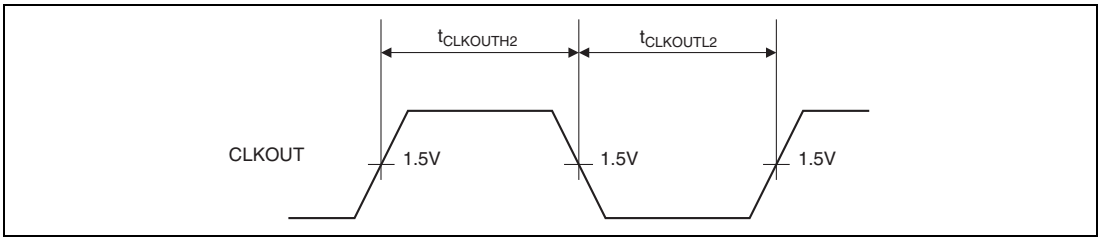
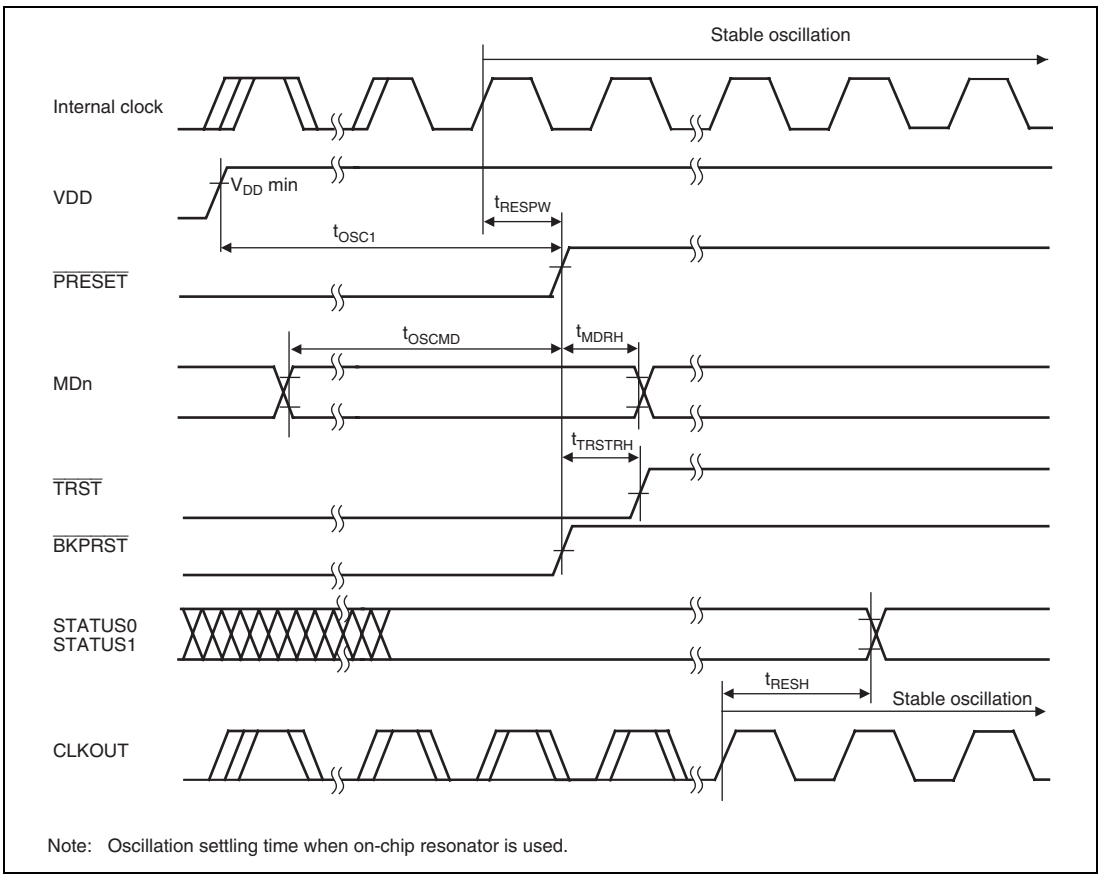


Figure 43.4 CLKOUT Clock Output Timing (2)



Note: Oscillation settling time when on-chip resonator is used.

Figure 43.5 Power-On Oscillation Settling Time

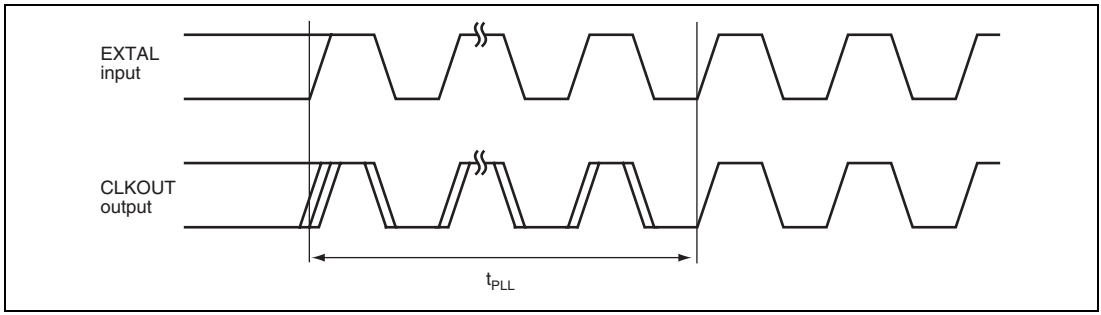


Figure 43.6 PLL Synchronization Settling Time

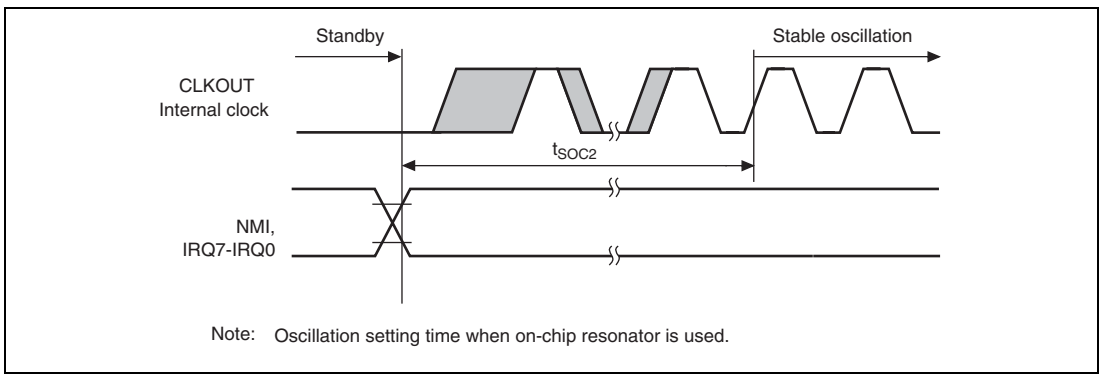


Figure 43.7 Oscillation Settling Time on Return from Standby NMI or IRQ

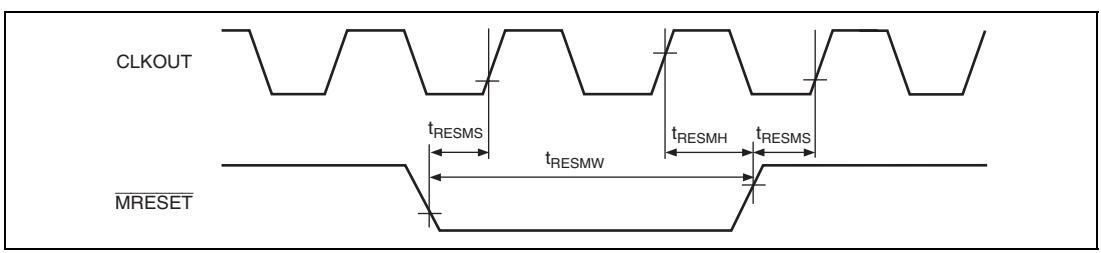


Figure 43.8 Reset Input Timing

43.4.2 Control Signal Timing

Table 43.10 Control Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{BREQ}$ setup time	t_{BREQS}	6	—	ns	43.9
$\overline{BREQ}$ hold time	t_{BREQH}	3	—	ns	43.9
$\overline{BACK}$ delay time	t_{BACKD}	1	13	ns	43.9
Bus three-state delay time	t_{BOFF1}	—	13	ns	43.9
Bus buffer on time	t_{BON1}	—	13	ns	43.9
STATUS0, STATUA1 delay time	t_{STD}	—	20	ns	43.10

Note: t_{cyc} : One CLK cycle time

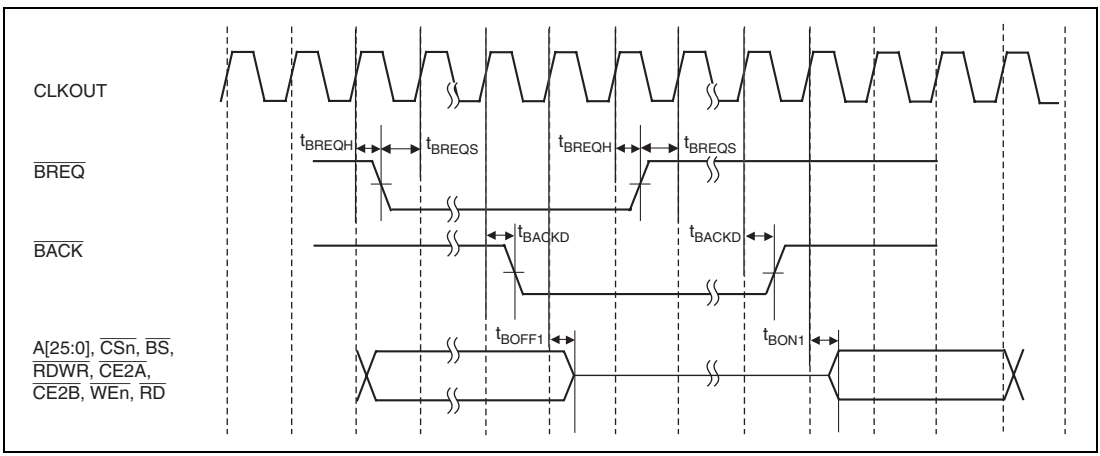


Figure 43.9 Control Signal Timing

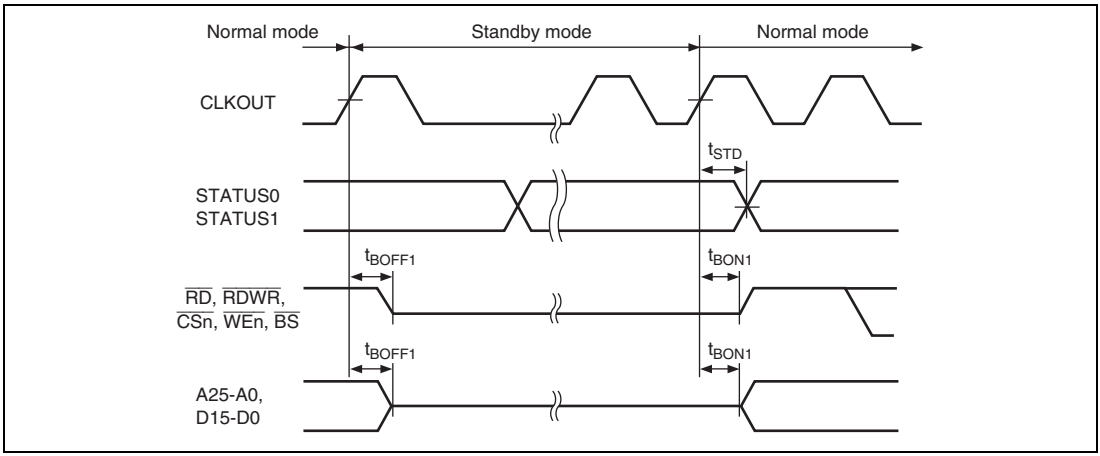


Figure 43.10 Pin Drive Timing in Standby Mode

43.4.3 Bus Timing

Table 43.11 Bus Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Address delay time	t_{AD}	1	13	ns		43.11 to 43.21, 43.26, 43.27
$\overline{BS}$ delay time	t_{BSD}	1	13	ns		43.11 to 43.27
$\overline{CSn}$ delay time	t_{CSD}	1	13	ns		43.11 to 43.27
$\overline{RDWR}$ delay time	t_{RWD}	1	13	ns		43.11 to 43.27
$\overline{RD}$ delay time	t_{RSD}	1	13	ns		43.11 to 43.19, 43.26, 43.27
Read data setup time	t_{RDS}	6	—	ns		43.11 to 43.22, 43.24, 43.26, 43.27
Read data hold time	t_{RDH}	2	—	ns		43.11 to 43.22, 43.24, 43.26, 43.27
$\overline{WEn}$ delay time (falling edge)*	t_{WEDF}	—	13	ns		43.11 to 43.14, 43.19, 43.26, 43.27
$\overline{WEn}$ delay time	t_{WED1}	1	13	ns		43.11 to 43.14, 43.19, 43.22, 43.23, 43.26, 43.27
Write data delay time	t_{WDD}	1	13	ns		43.11 to 43.14, 43.19 to 43.25
$\overline{RDY}$ setup time	t_{RDYS}	6	—	ns		43.12, 43.13, 43.16, 43.18 to 43.27
$\overline{RDY}$ hold time	t_{RDYH}	2.5	—	ns		43.12, 43.13, 43.16, 43.18 to 43.27
$\overline{FRAME}$ delay time	t_{FMD}	1	13	ns	MPX	43.22 to 43.25
$\overline{IOIS16}$ setup time	t_{IO16S}	6	—	ns	PCMCIA	43.20, 43.21
$\overline{IOIS16}$ hold time	t_{IO16H}	2.5	—	ns	PCMCIA	43.20, 43.21
$\overline{IOWR}$ delay time (falling edge)	t_{IOWSDF}	1	13	ns	PCMCIA	43.20, 43.21
$\overline{IORD}$ delay time	t_{IORSD}	1	13	ns	PCMCIA	43.20, 43.21

Note: * Delay time from the rising edge of CLKOUT

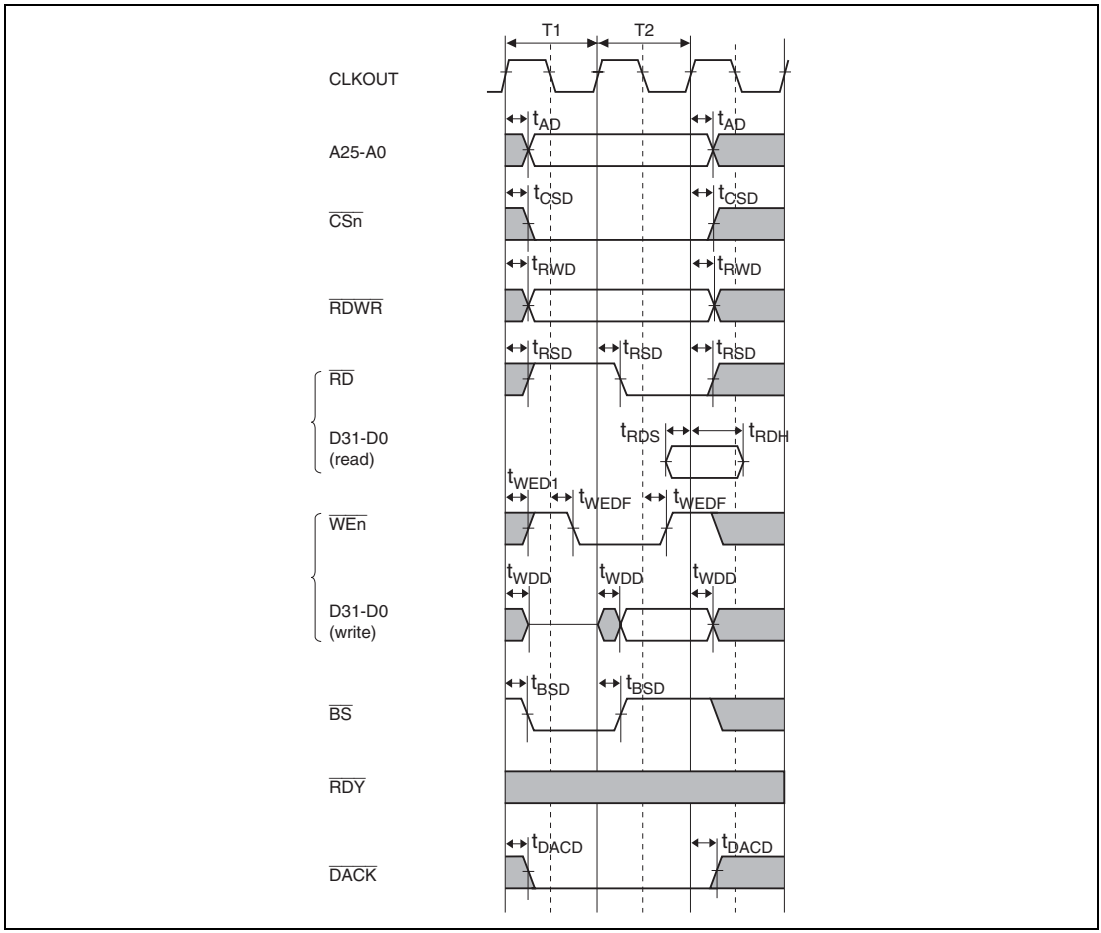


Figure 43.11 SRAM Bus Cycle: Basic Bus Cycle (No Wait)

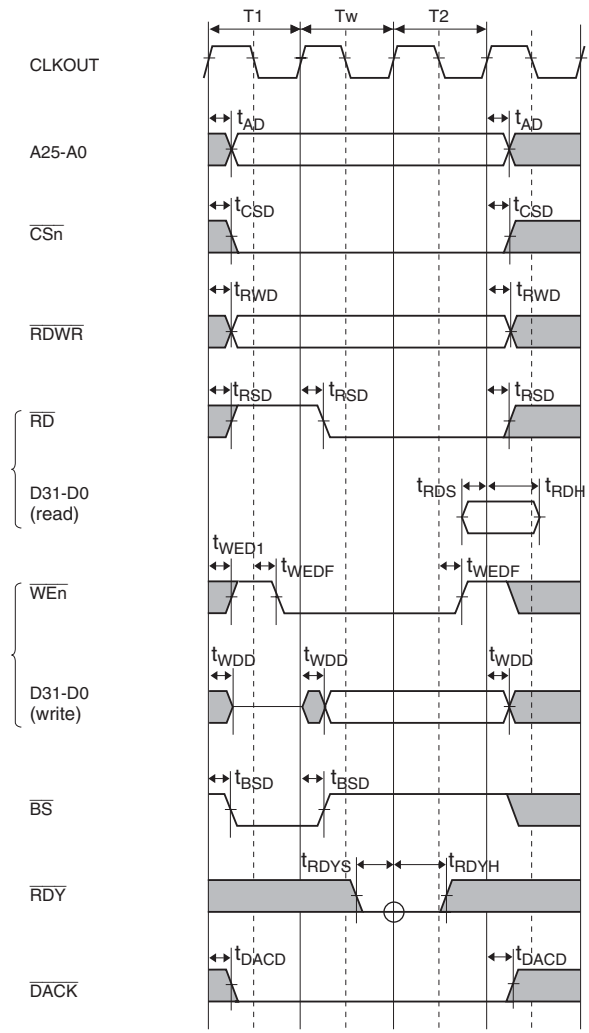


Figure 43.12 SRAM Bus Cycle: Basic Bus Cycle (One Wait only by Software)

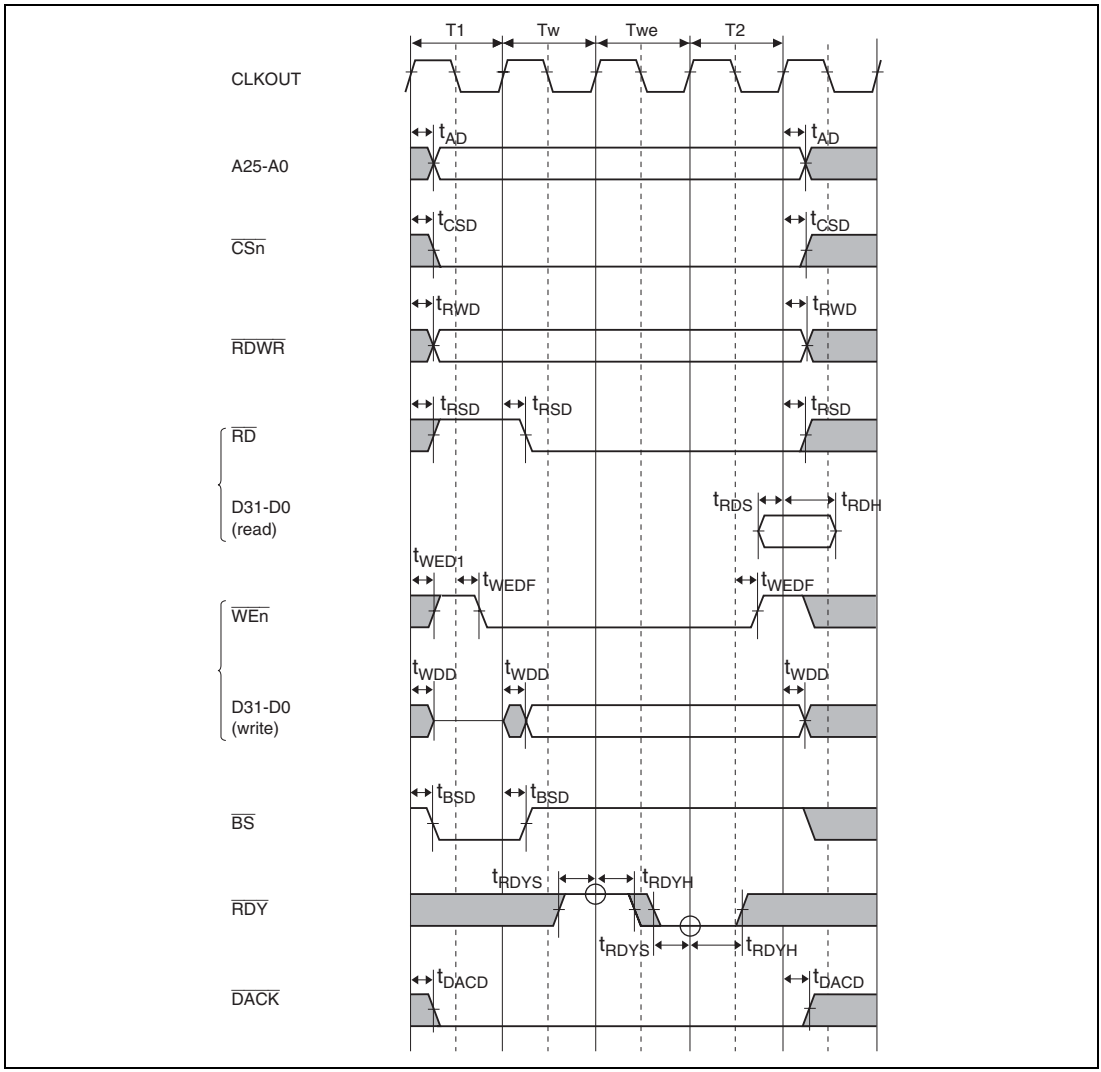


Figure 43.13 SRAM Bus Cycle: Basic Bus Cycle (One Wait by Software + One Wait by $\overline{\text{RDY}}$, $\overline{\text{RDY}}$ Signal is Synchronous Input)

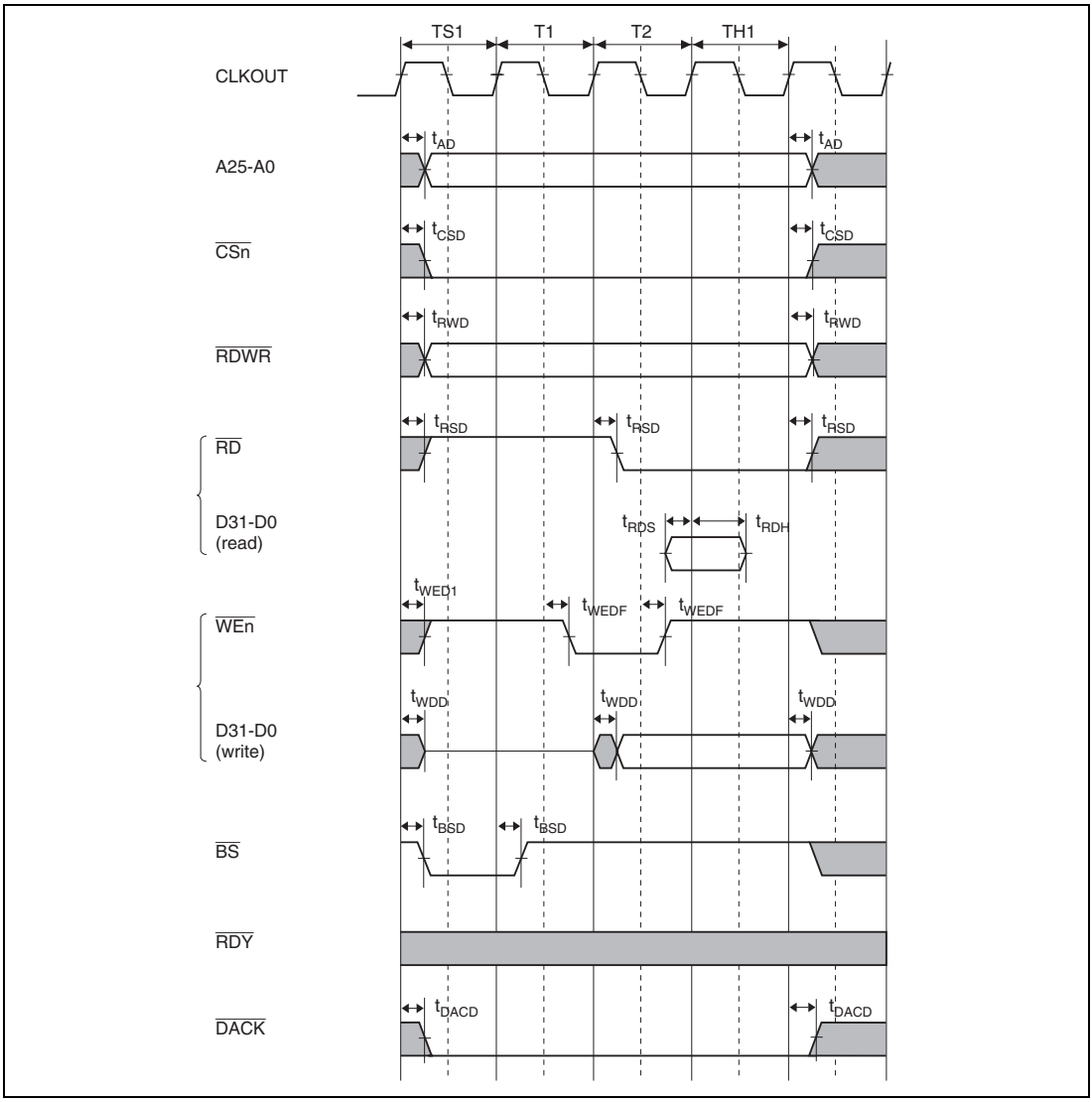


Figure 43.14 SRAM Bus Cycle: Basic Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0, WTS = 1, WTH = 1)

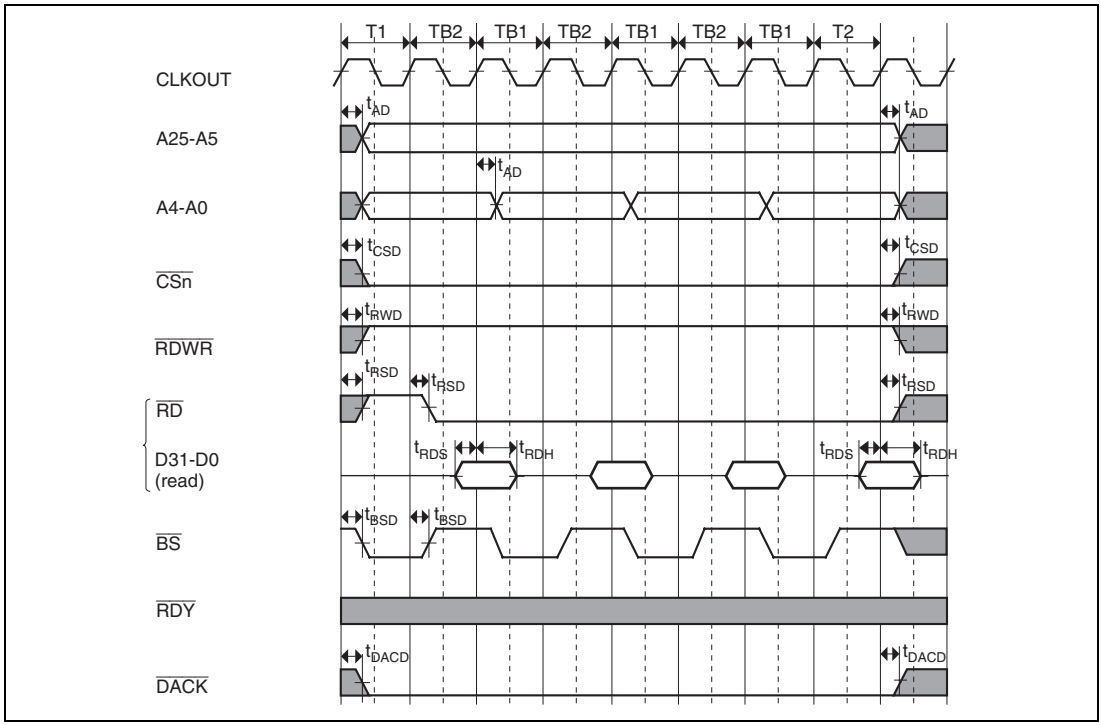


Figure 43.15 Burst ROM Bus Cycle (No Wait)

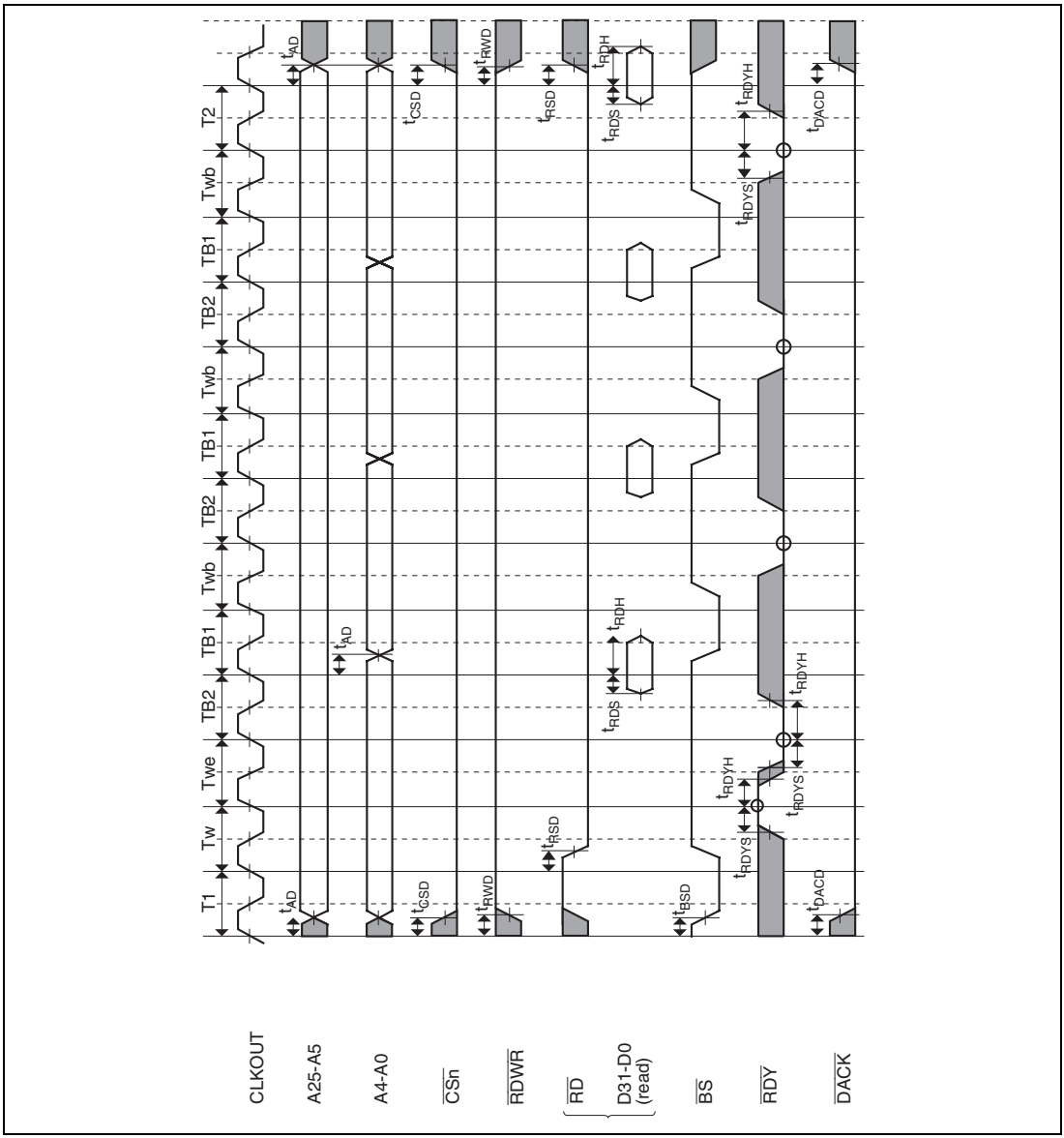


Figure 43.16 Burst ROM Bus Cycle
(1st Data: One Wait by Software + One Wait by $\overline{\text{RDY}}$;
2nd/3rd/4th Data: One Wait only by software)

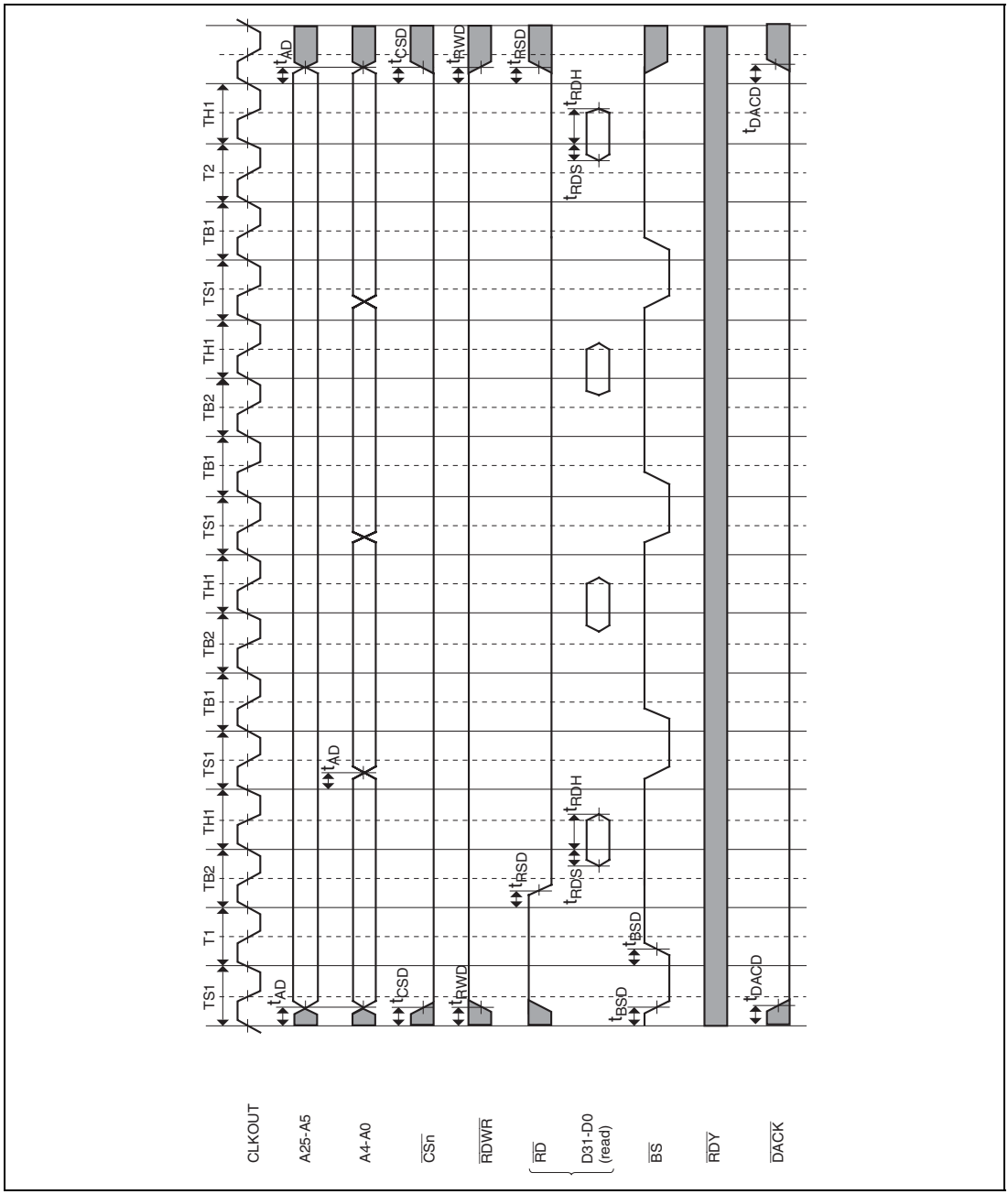


Figure 43.17 Burst ROM Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

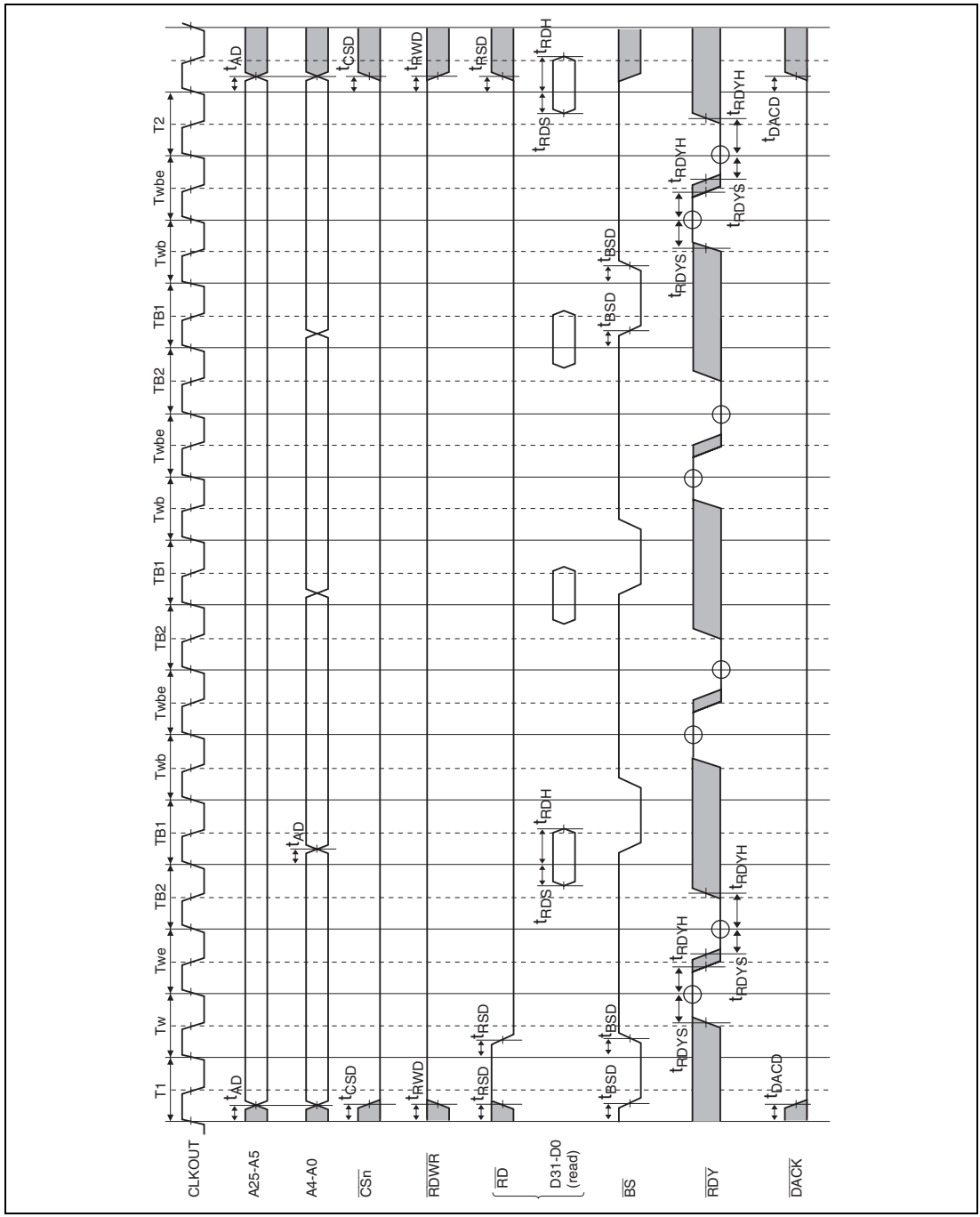


Figure 43.18 Burst ROM Bus Cycle (One Wait by Software + One Wait by $\overline{\text{RDY}}$)

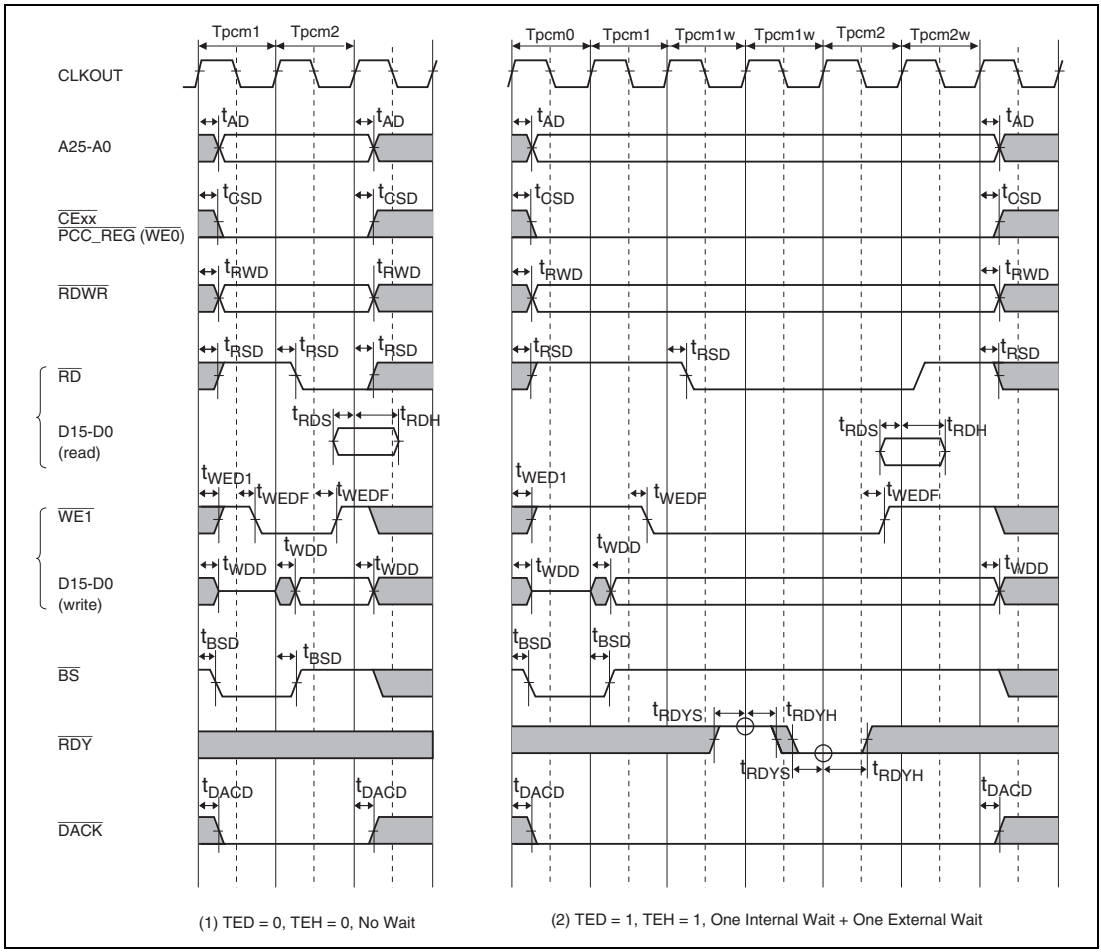


Figure 43.19 PCMCIA Memory Bus Cycle

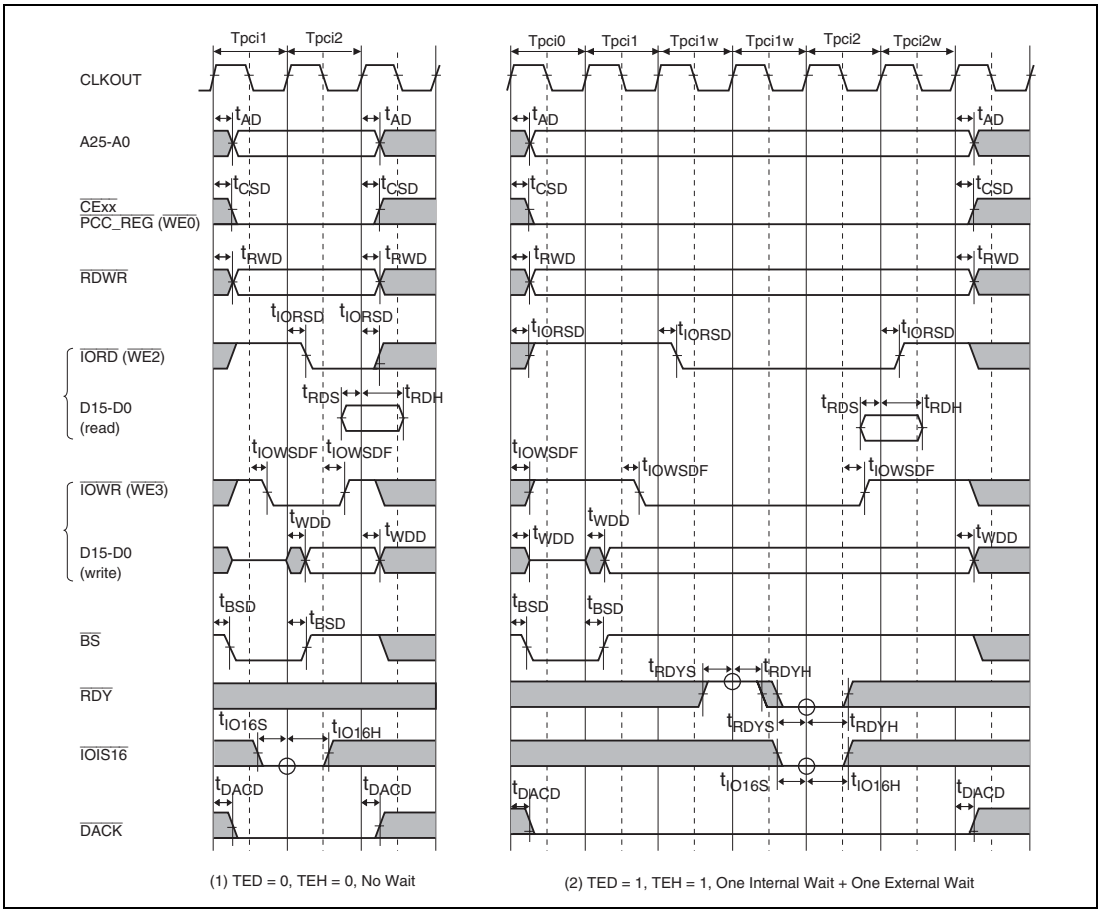


Figure 43.20 PCMCIA I/O Bus Cycle

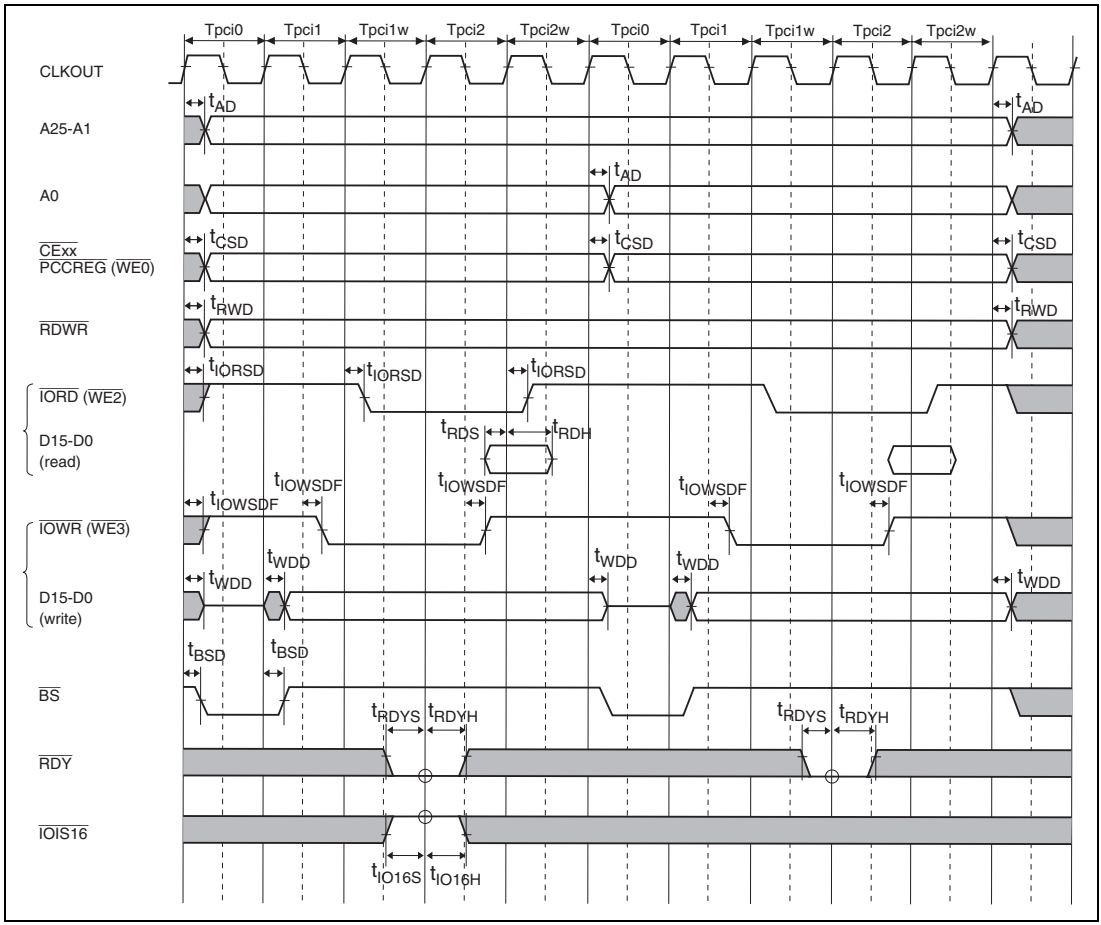


Figure 43.21 PCMCIA I/O Bus Cycle
(TEDA/TEDB = 1, TEHA/TEHB = 1, IW/PCIW = 1, Dynamic Bus Sizing)

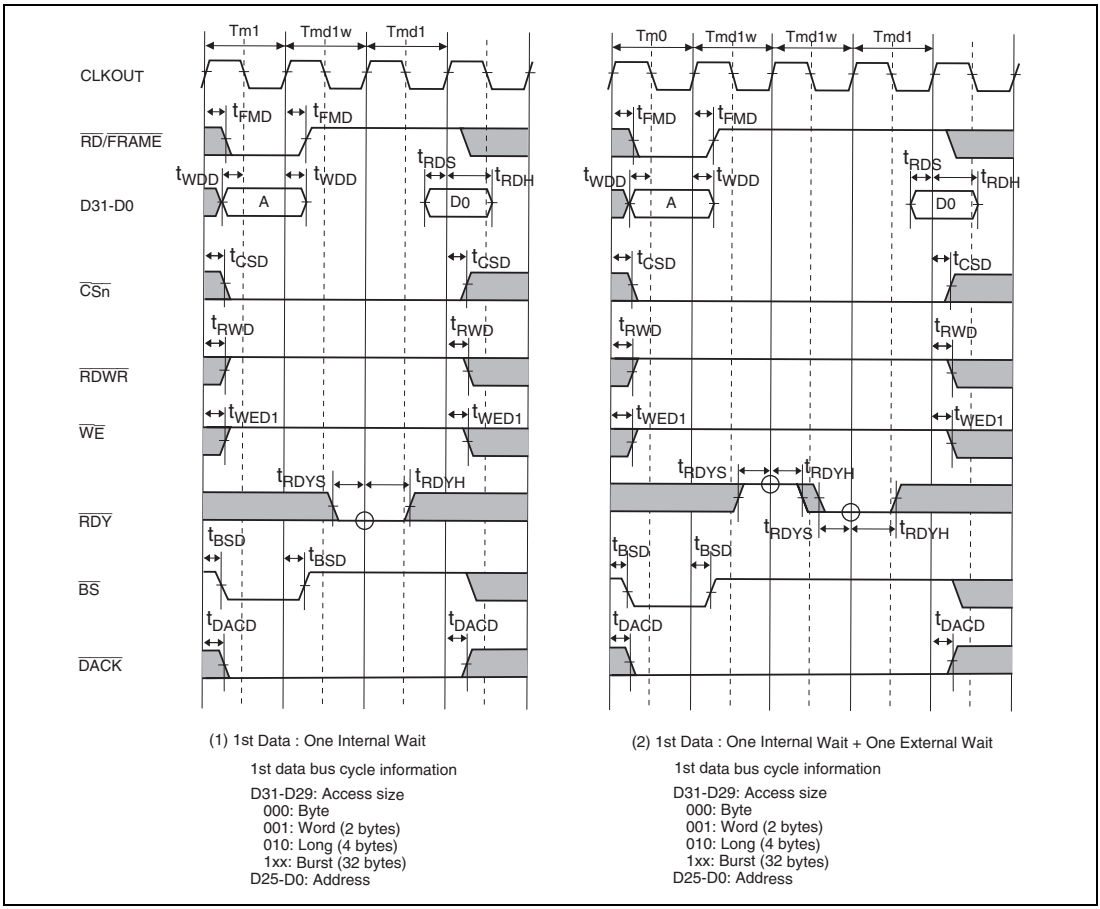


Figure 43.22 MPX Basic Bus Cycle: Read

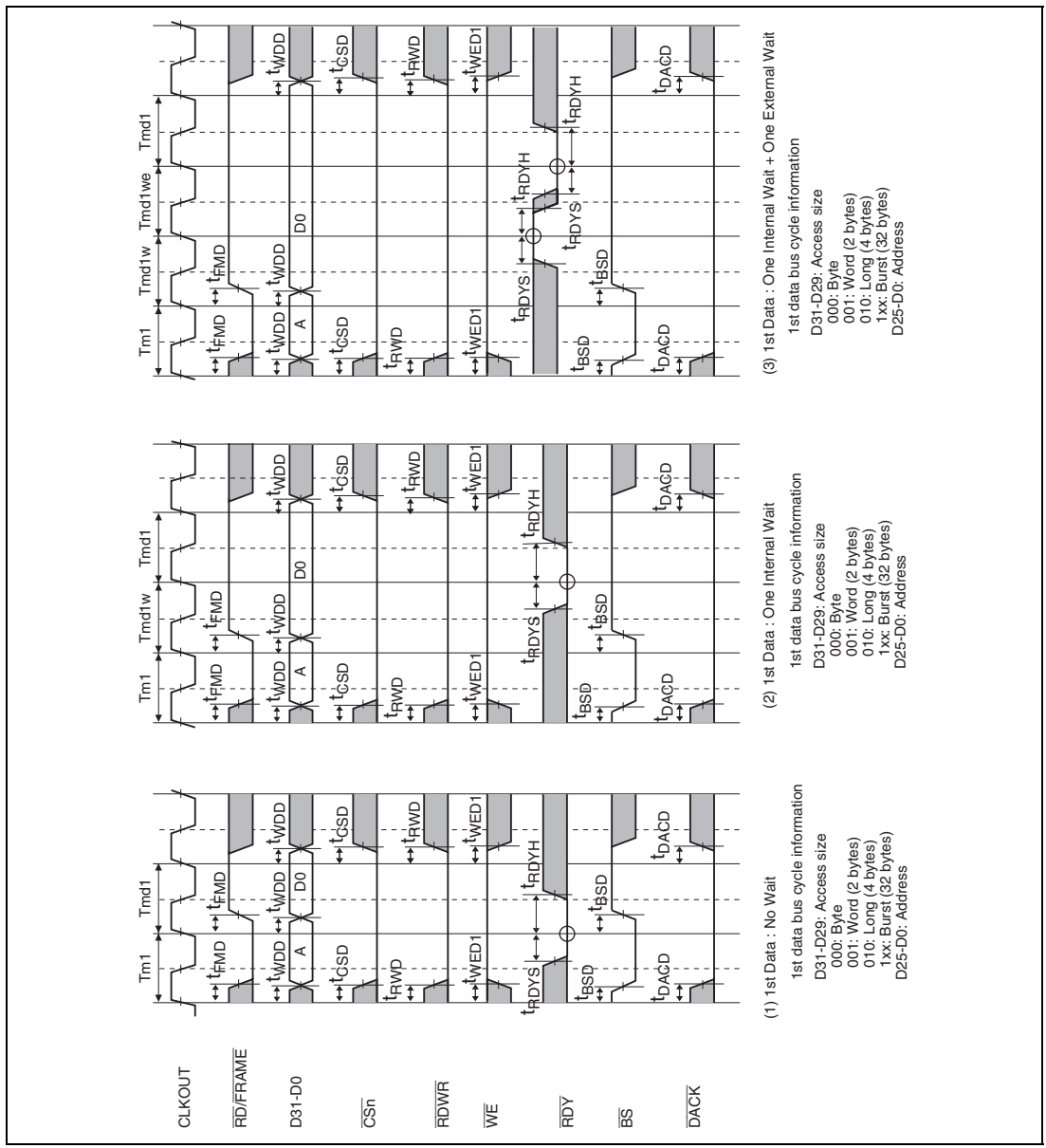


Figure 43.23 MPX Basic Bus Cycle: Write

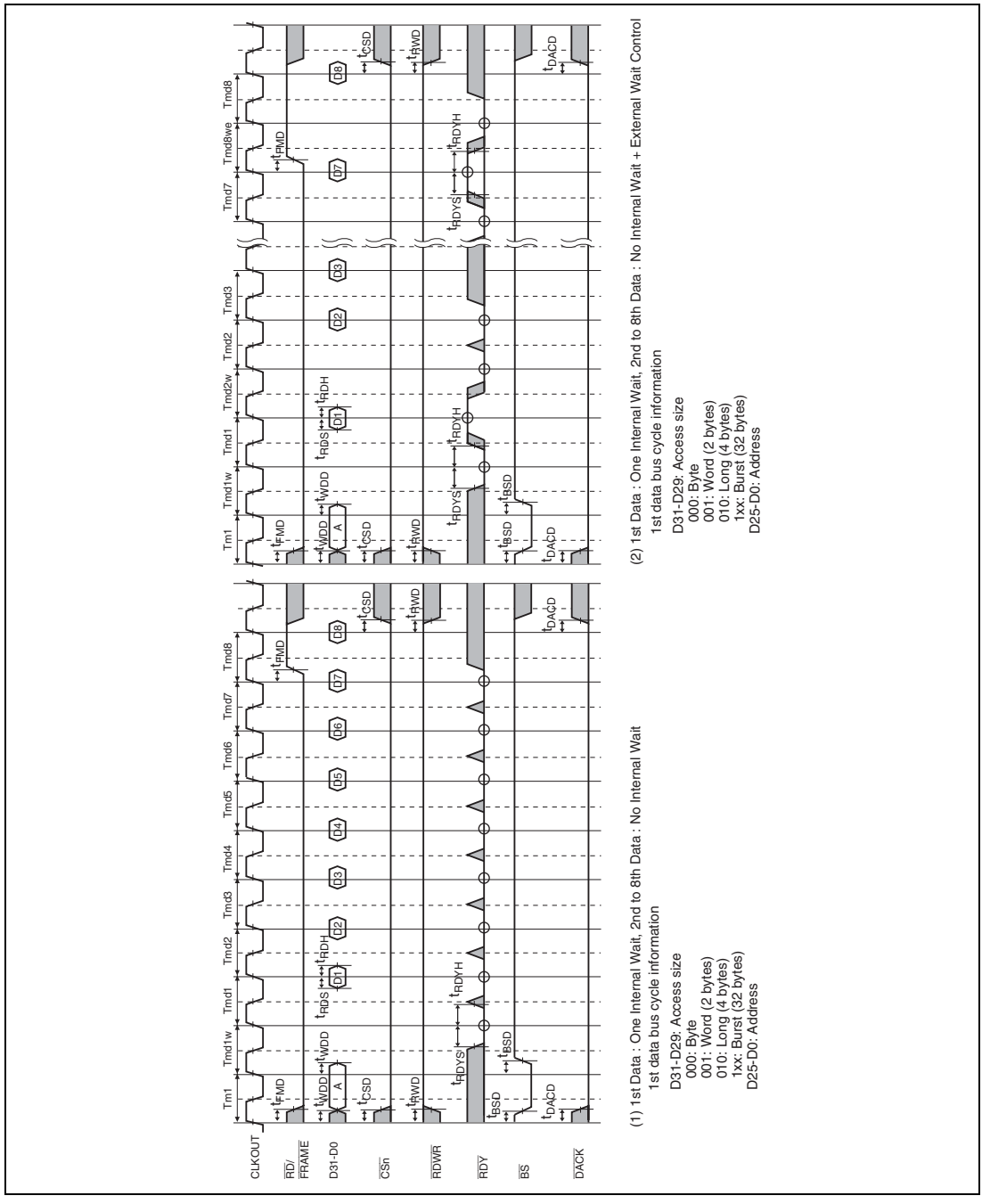


Figure 43.24 MPX Bus Cycle: Burst Read

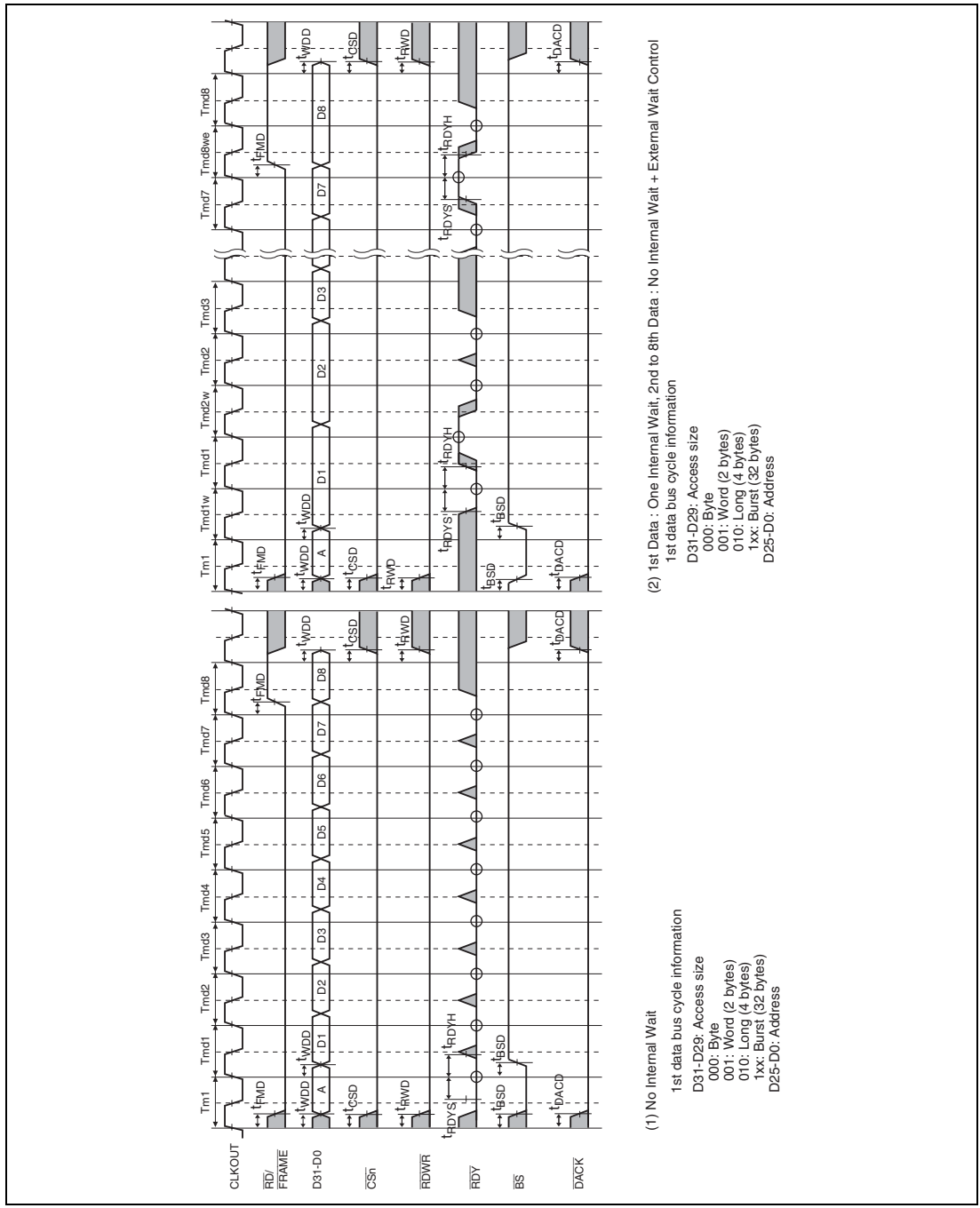


Figure 43.25 MPX Bus Cycle: Burst Write

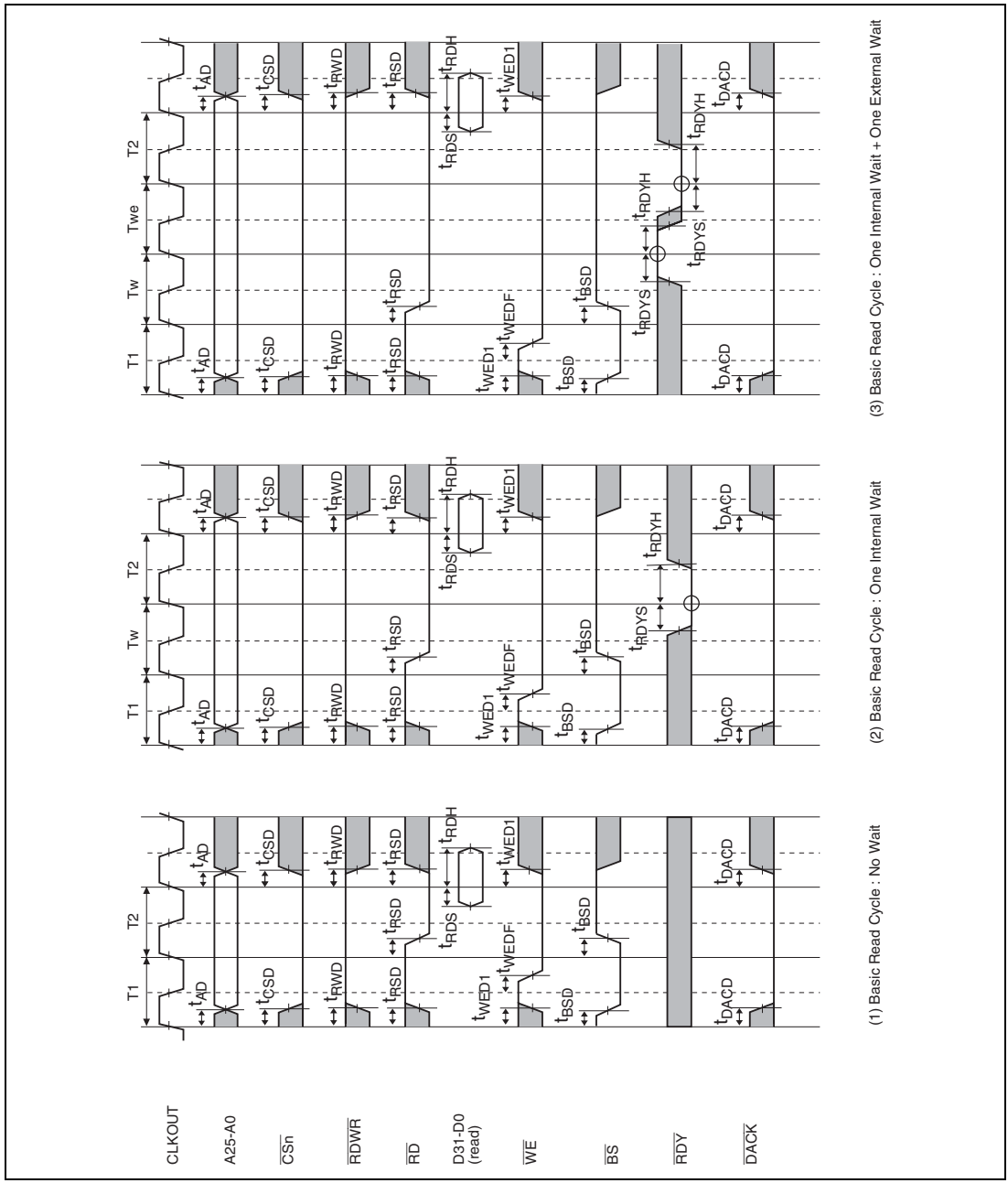


Figure 43.26 Byte Control SRAM Bus Cycle

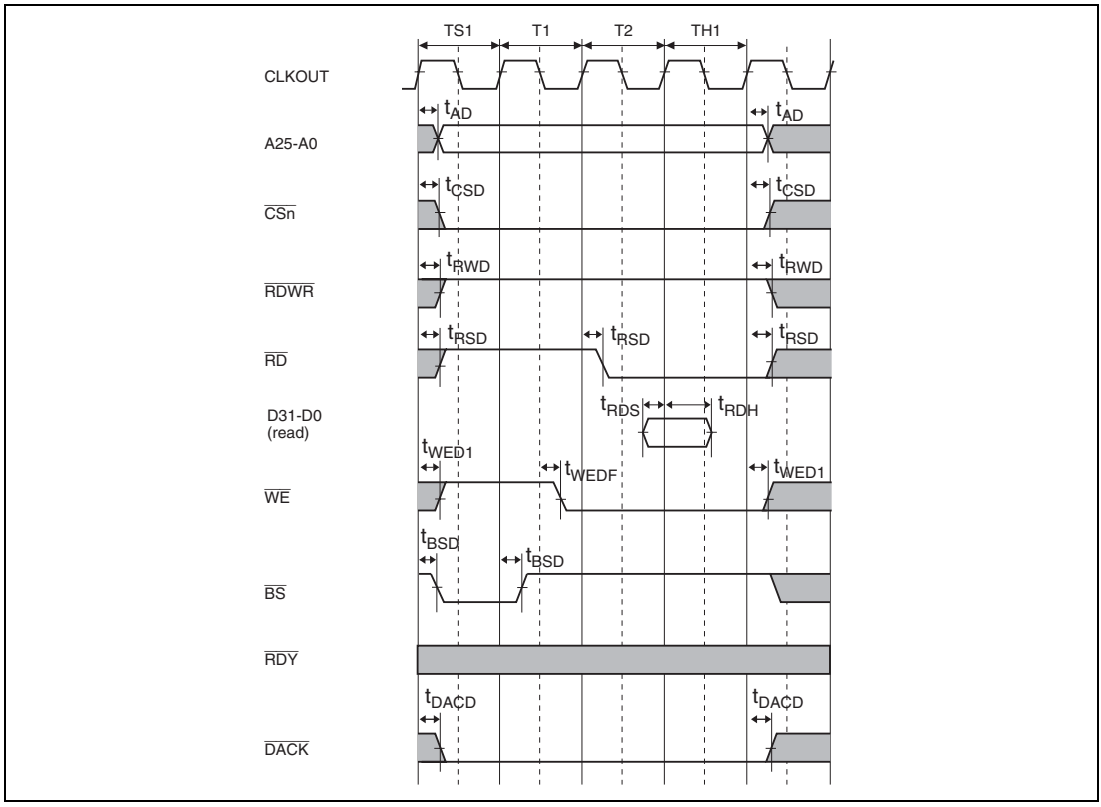


Figure 43.27 Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

43.4.4 DDRIF Signal Timing

Table 43.12 DDRIF Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
M_CLK output cycle	t_{MCLK}	10	12	ns	43.28	DDR200
		7.5	12			DDR266
M_CLK output high-level pulse width	t_{MCLKH}	0.45	0.55	t_{MCLK}	43.28	
M_CLK output low-level pulse width	t_{MCLKL}	0.45	0.55	t_{MCLK}	43.28	
Address and command signal setup time to M_CLK rising edge	t_{ADCTLS}	1.5	—	ns	43.29, 43.30	DDR200
		1.2	—			DDR266
Address and command signal hold time to M_CLK rising edge	t_{ADCTLH}	1.5	—	ns	43.29, 43.30	DDR200
		1.2	—			DDR266
M_CLK-to-M_DQSn skew time (read)	$t_{RMDQS-MCLK}$	–1.0	1.0	ns	43.29	DDR200
		–0.8	0.8			DDR266
M_DQSn-to-M_Dn skew (read)	t_{RMDQSQ}	—	0.7	ns	43.29	DDR200
		—	0.6			DDR266
Write command to first M_DQSn delay time (rising edge)	t_{WMDQSS}	0.75	1.2	t_{MCLK}	43.30	
M_DQSn falling edge setup time to M_CLK rising edge (write)	t_{WDSS}	0.25	—	t_{MCLK}	43.30	
M_DQSn falling edge hold time to M_CLK rising edge (write)	t_{WDSH}	0.25	—	t_{MCLK}	43.30	
M_DQS high-level pulse width (write)	t_{WMDQSH}	0.35	—	t_{MCLK}	43.30	
M_DQS low-level pulse width (write)	t_{WMDQSL}	0.35	—	t_{MCLK}	43.30	
M_Dn and M_DQMn setup time to M_DQSn rising edge (write)	t_{WDS}	1.0	—	ns	43.30	DDR200
		0.75	—			DDR266
M_Dn and M_DQMn hold time to M_DQSn rising edge (write)	t_{WDH}	1.0	—	ns	43.30	DDR200
		0.75	—			DDR266

Note: t_{MCLK} : One M_CLK cycle time

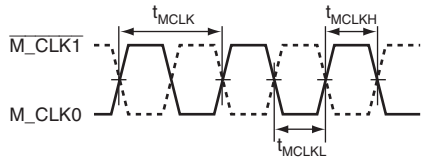


Figure 43.28 DDRIF MCLK Output Timing

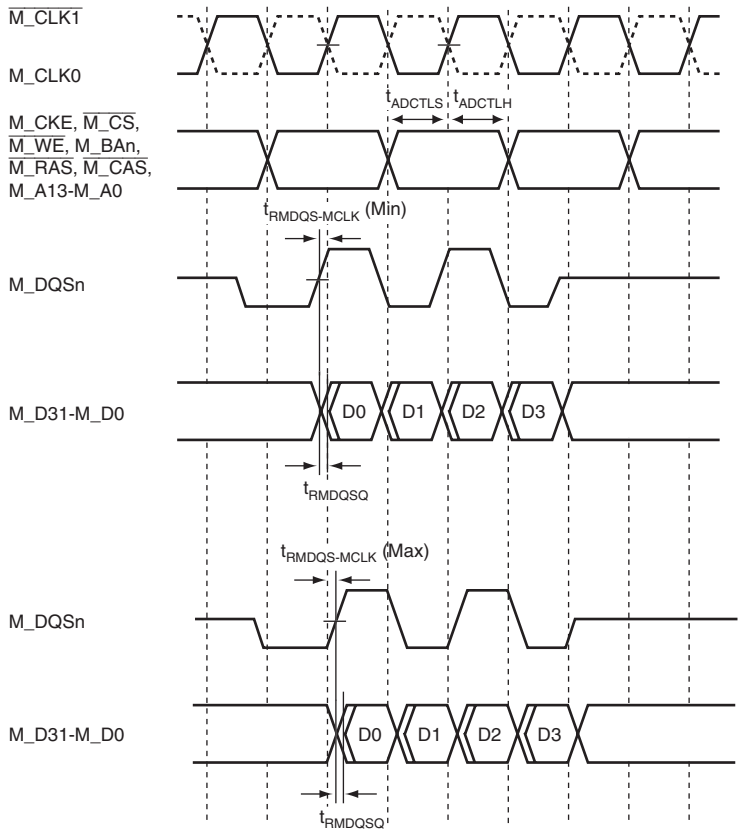


Figure 43.29 Read Timing of DDR-SDRAM (2 Burst Read)

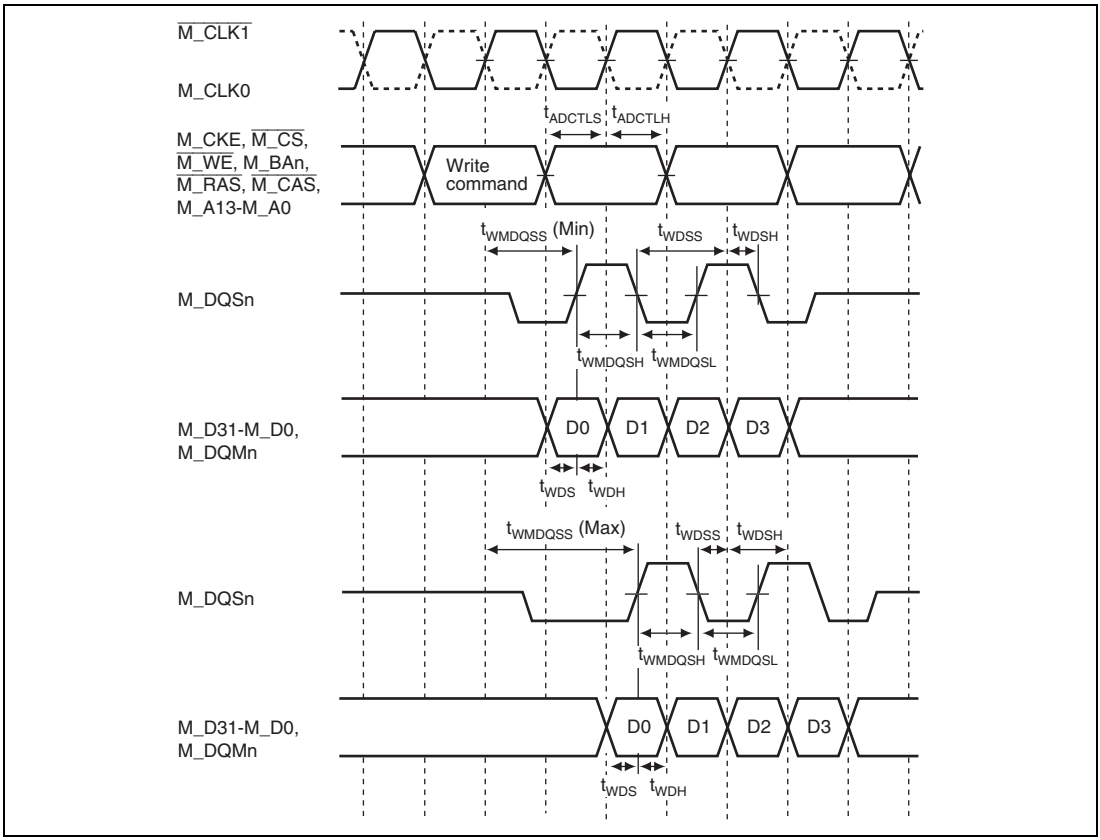


Figure 43.30 Write Timing of DDR-SDRAM (2 Burst Write)

43.4.5 INTC Module Signal Timing

Table 43.13 INTC Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
NMI pulse width (High)	t_{NMIH}	5	—	t_{cyc}	43.31	normal mode sleep mode
NMI pulse width (Low)	t_{NMIL}	5	—	t_{cyc}	43.31	normal mode sleep mode
IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ setup time	t_{IRQS}	8	—	ns	43.32	IRQ input
IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ hold time	t_{IRQH}	3	—	ns	43.32	IRQ input
IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ setup time	t_{IRLS}	8	—	ns	43.32	IRL input
IRQ7/$\overline{\text{IRL7}}$ to IRQ0/$\overline{\text{IRL0}}$ hold time	t_{IRLH}	3	—	ns	43.32	IRL input
PINTn interrupt setup	t_{GPIOS}	15	—	ns	43.32	GPIO interrupt input
PINTn interrupt hold time	t_{GPIOH}	8	—	ns	43.32	GPIO interrupt input
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	—	13	ns	43.32	$\overline{\text{IRQOUT}}$ output

Note: t_{cyc}: One CLKOUT cycle time

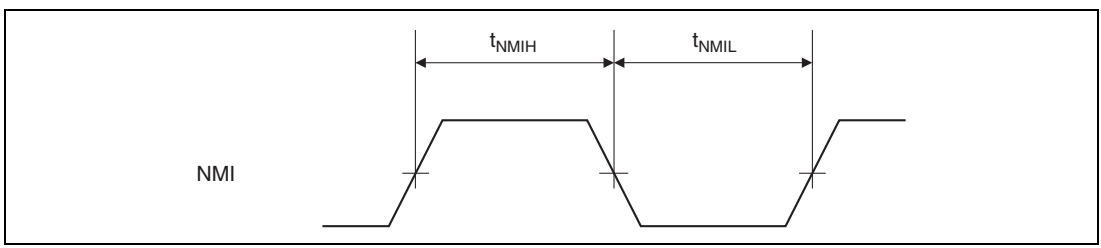


Figure 43.31 NMI Input Timing

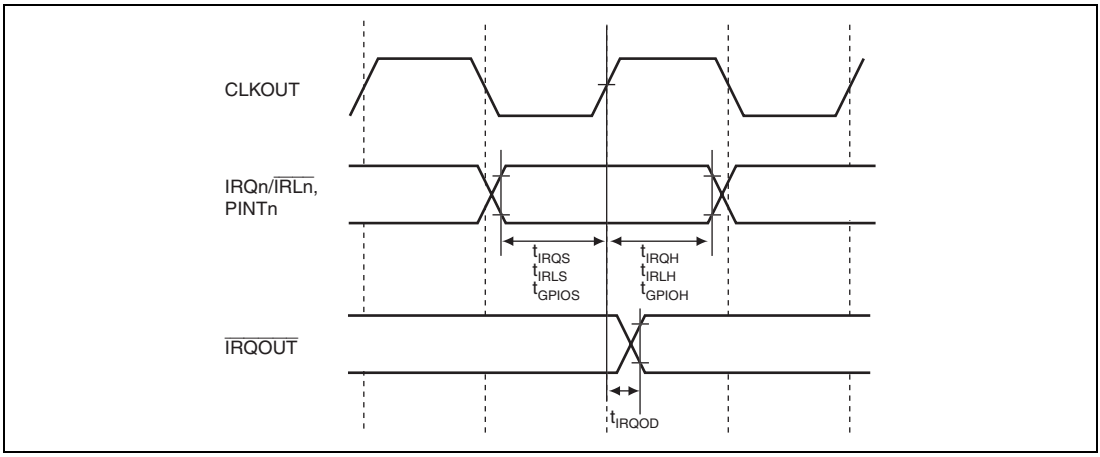


Figure 43.32 IRQ/IRL, PINT Input and IRQOUT Output Timing

43.4.6 External CPU Interface Read/Write Access Timing

Table 43.14 External CPU Interface Access Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
External CPU bus release request ($\overline{BREQ}$) setup time	t_{SEBRQ}	6	—	—	ns	43.33
External CPU bus release request ($\overline{BREQ}$) hold time	t_{TMS}	3	—	—	ns	43.33
External CPU bus request acknowledge ($\overline{BACK}$) delay time	t_{DEBAK}	1	—	13	ns	43.33
Address/write data setup time	t_{SEDA}	6	—	—	ns	43.33
Address/write data hold time	t_{HEDA}	3	—	—	ns	43.33
Read data delay time	t_{DED}	1	—	13	ns	43.33
$\overline{EX_CSn}$ setup time	t_{SECS}	6	—	—	ns	43.33
$\overline{EX_CSn}$ hold time	t_{HECS}	3	—	—	ns	43.33
$\overline{EX_BS}$ setup time	t_{SEBS}	6	—	—	ns	43.33
$\overline{EX_BS}$ hold time	t_{HEBS}	3	—	—	ns	43.33
$\overline{EX_FRAME}$ setup time	t_{SEFR}	6	—	—	ns	43.33
$\overline{EX_FRAME}$ hold time	t_{HEFR}	3	—	—	ns	43.33
$\overline{EX_RDWR}$ setup time	t_{SERW}	6	—	—	ns	43.33
$\overline{EX_RDWR}$ hold time	t_{HERW}	3	—	—	ns	43.33
$\overline{EX_RDY}$ delay time	t_{DERY}	1	—	13	ns	43.33

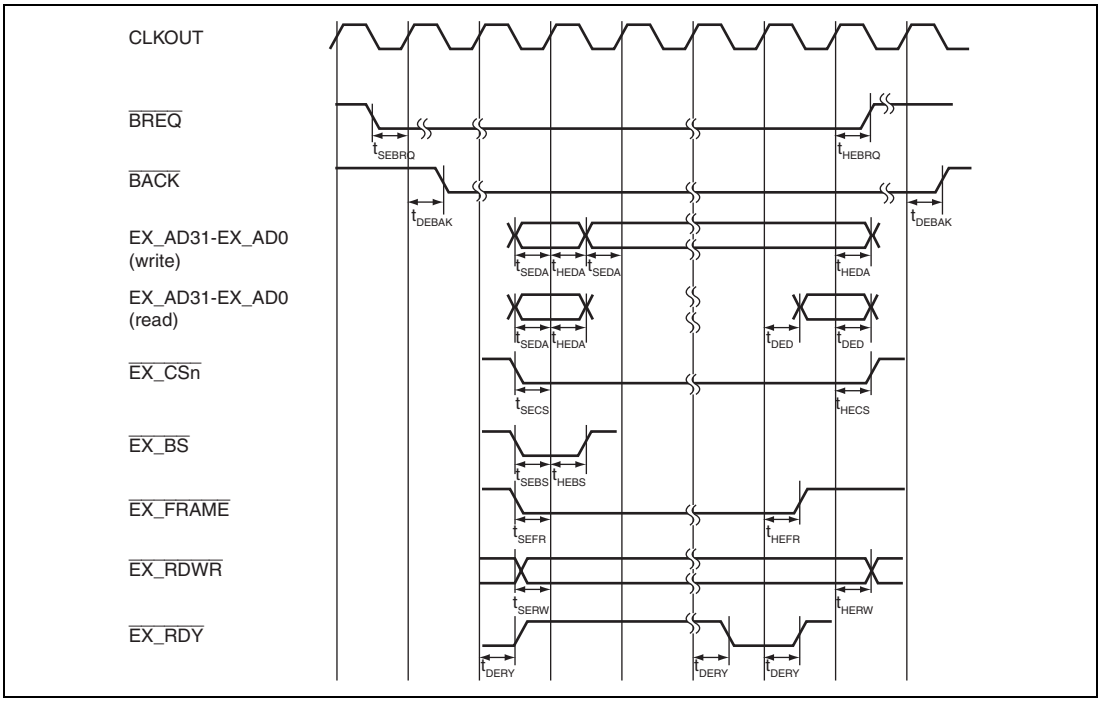


Figure 43.33 External CPU Interface Read/Write Access Timing

43.4.7 PCIC Module Signal Timing

Table 43.15 PCIC Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

			33 MHz		66 MHz			
Pin	**Item**	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Figure**
PCICLK	Clock cycle	t_{PCICYC}	30	—	15	30	ns	43.34
	Clock pulse width (high)	$t_{PCIHIGH}$	11	—	6	—	ns	43.34
	Clock pulse width (low)	t_{PCILOW}	11	—	6	—	ns	43.34
	Clock rise time	t_{PCIr}	—	4	—	1.5	ns	43.34
	Clock fall time	t_{PCIf}	—	4	—	1.5	ns	43.34
$\overline{\text{PCIRESET}}$	Output delay time	t_{PCIRES}	—	14	—	14	ns	—
IDSEL	Input setup time	t_{PCISU}	4	—	4	—	ns	43.36
	Input hold time	t_{PCIH}	0	—	0	—	ns	43.36
AD31 to AD0	Output data delay time	t_{PCIVAL}	—	10	—	10	ns	43.35
CBE3 to CBE0	Tri-state drive delay time	t_{PCION}	—	10	—	10	ns	43.35
PAR	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	—	12	ns	43.35
$\overline{\text{PCIFRAME}}$	Input setup time	t_{PCISU}	4	—	4	—	ns	43.36
$\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$	Input hold time	t_{PCIH}	0	—	0	—	ns	43.36
$\overline{\text{STOP}}$, $\overline{\text{LOCK}}$								
$\overline{\text{DEVSEL}}$								
$\overline{\text{PERR}}$								
$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$	Output data delay time	t_{PCIVAL}	—	10	—	10	ns	43.35
	Tri-state drive delay time	t_{PCION}	—	10	—	10	ns	43.35
$\overline{\text{REQ3}}$ to $\overline{\text{REQ1}}$ $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	—	12	ns	43.35
$\overline{\text{GNT3}}$ to $\overline{\text{GNT1}}$	Input setup time	t_{PCISU}	4	—	4	—	ns	43.36
	Input hold time	t_{PCIH}	0	—	0	—	ns	43.36
$\overline{\text{SERR}}$	Tri-state drive delay time	t_{PCION}	—	10	—	10	ns	43.35
$\overline{\text{INTA}}$ to $\overline{\text{INTD}}$	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	—	12	ns	43.35
	Input setup time	t_{PCISU}	4	—	4	—	ns	43.36
	Input hold time	t_{PCIH}	0	—	0	—	ns	43.36

Note: When the ratio of the clocks (SHwy clock : PCICLK clock) is in the ranges of (2.1 : 1) to (3.3 : 1), the PCIC cannot be used.

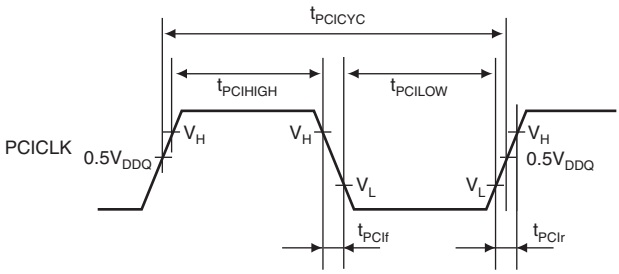


Figure 43.34 PCI Clock Input Timing

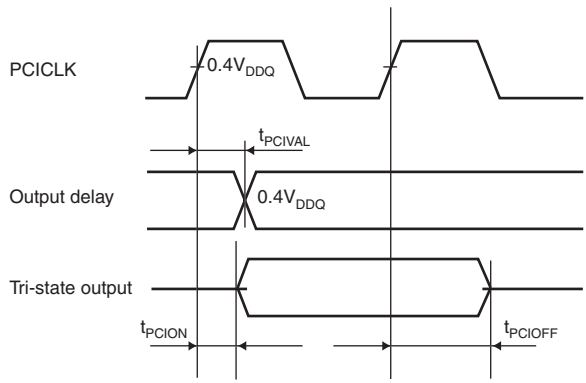


Figure 46.35 Output Signal Timing

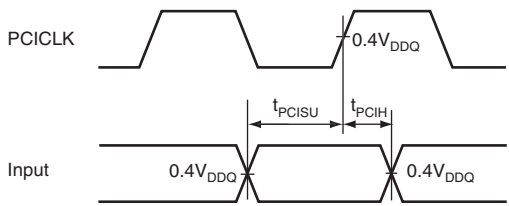


Figure 43.36 Input Signal Timing

43.4.8 DMAC Module Signal Timing

Table 43.16 DMAC Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
$\overline{\text{DREQ}}$n setup time	t_{DRQS}	6	—	ns	43.37	
$\overline{\text{DREQ}}$n hold time	t_{DRQH}	5	—	ns	43.37	
$\overline{\text{TEND}}$n delay time	t_{TENDD}	—	13	ns	43.37	
$\overline{\text{DACK}}$n delay time	t_{DACKD}	—	13	ns	43.37	

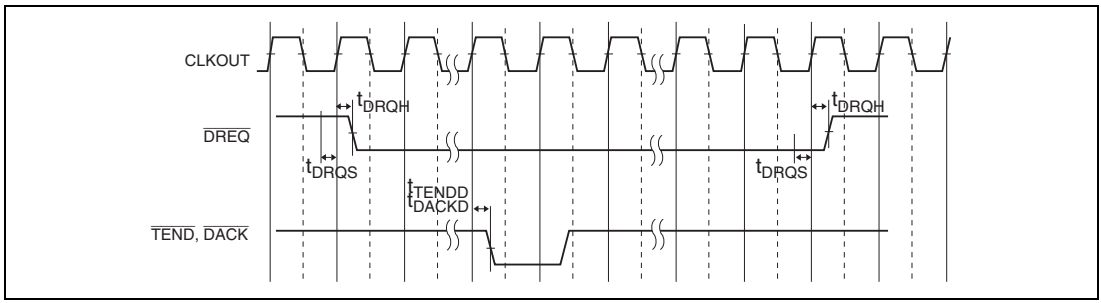


Figure 43.37 $\overline{\text{DREQ}}$, $\overline{\text{TEND}}$, and $\overline{\text{DACK}}$ Timing

43.4.9 TMU Module Signal Timing

Table 43.17 TMU Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure	Remarks
Timer clock pulse width (High)	t_{TCLKWH}	4	—	t_{Pcyc0}	43.38	
Timer clock pulse width (Low)	t_{TCLKWL}	4	—	t_{Pcyc0}	43.38	
Timer clock rise time	t_{TCLKr}	—	0.8	t_{Pcyc0}	43.38	
Timer clock fall time	t_{TCLKf}	—	0.8	t_{Pcyc0}	43.38	

Note: t_{Pcyc0}: One Pck0 cycle time

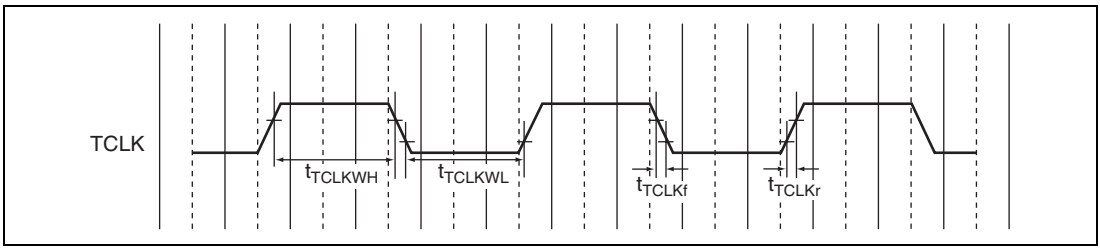


Figure 43.38 TCLK Input Timing

43.4.10 16-bit Timer Pulse Unit (TPU) Timing

Table 43.18 16-bit Timer Pulse Unit

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item		Symbol	Min.	Max.	Unit	Figure	Remarks
Timer output delay		t_{TOD}	—	15	Ns	43.39	
Timer clock input setup time		t_{TCKS}	15	—	ns	43.40	
Timer clock pulse width	Count at rising or falling edge	t_{TCKWH}, t_{TCKWL}	2	—	t_{Pcyc0}*	43.40	
	Count at both edge	t_{TCKWH}, t_{TCKWL}	3	—	t_{Pcyc0}*	43.40	

Note: * t_{Pcyc0} is a cycle time of a peripheral clock 0 (Pck0).

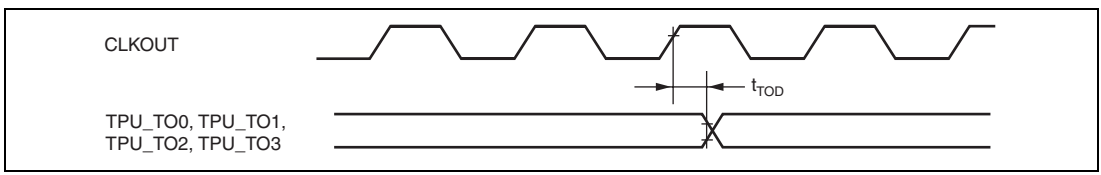


Figure 43.39 TPU Output Timing

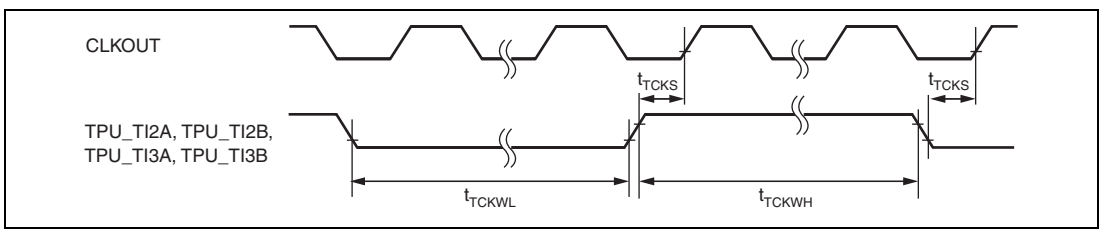


Figure 43.40 TPU Clock Input Timing

43.4.11 GETHER Module Signal Timing

(1) Ethernet Controller Signal Timing (MII)

Table 43.19 Ethernet Controller Signal Timing (MII)

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
ETn_TX-CLK cycle time	t_{Tcyc}	40	—	—	ns	43.41
ETn_TX-EN output delay time	t_{TEND}	3	—	20	ns	43.41
ETn_ETXD[3:0] output delay time	t_{TEDD}	3	—	20	ns	43.41
ETn_RX-CLK cycle time	t_{Rcyc}	40	—	—	ns	43.42
ETn_RX-DV setup time	t_{RDVS}	10	—	—	ns	43.42
ETn_RX-DV hold time	t_{RDVH}	3	—	—	ns	43.42
ETn_ERXD[3:0] setup time	t_{ERDS}	10	—	—	ns	43.42
ETn_ERXD[3:0] hold time	t_{ERDH}	3	—	—	ns	43.42
ETn_RX-ER setup time	t_{ERRS}	10	—	—	ns	43.43
ETn_RX-ER hold time	t_{ERRH}	3	—	—	ns	43.43
ETn_WOL output delay time	t_{WOLD}	1	—	18	ns	43.44

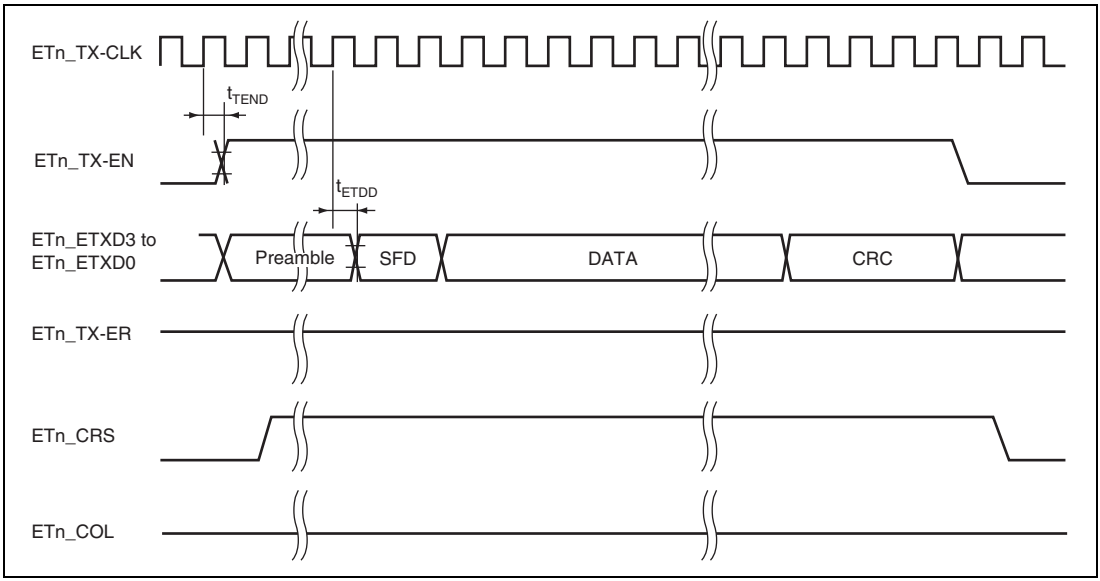


Figure 43.41 MII Transmit Timing (normal operation)

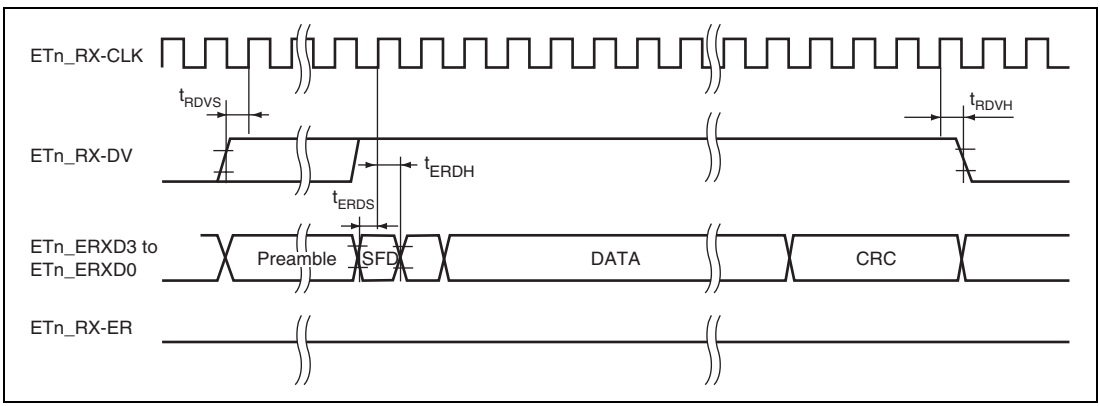


Figure 43.42 MII Receive Timing (normal operation)

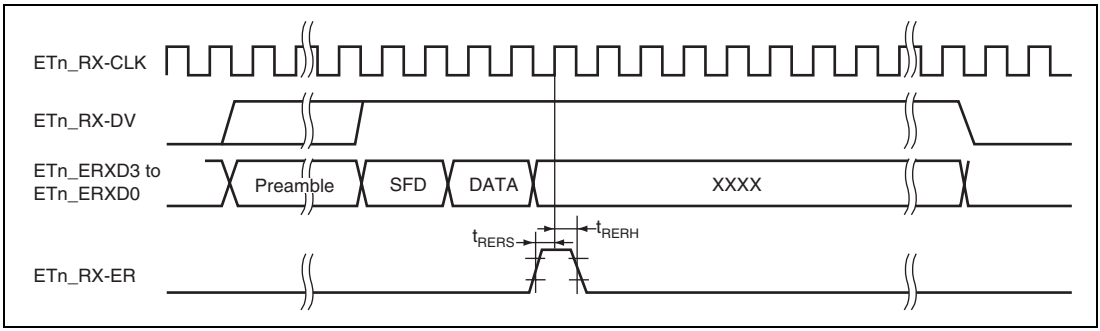


Figure 43.43 MII Receive Timing (When an Error is Detected)

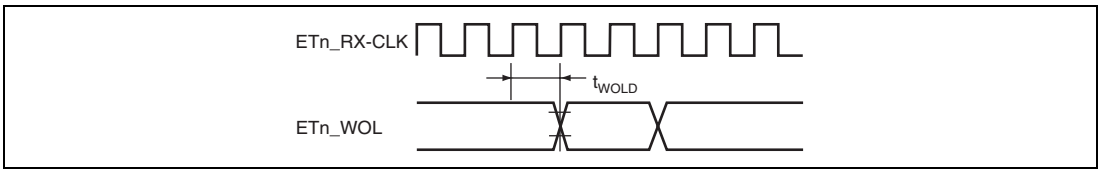


Figure 43.44 WOL Output Timing

(2) Ethernet Controller Signal Timing (GMII)

Table 43.20 Ethernet Controller Signal Timing (GMII)

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
GETn_GTX-CLK cycle time	t_{Gtcyc}	8	—	—	ns	43.45
Etn_TX-EN output delay time	t_{GTEND}	0.5	—	5.5	ns	43.45
GETn_ETXD[7:4], ETn_ETXD[3:0] output delay time	t_{GETDD}	0.5	—	5.5	ns	43.45
ETn_RX-CLK cycle time	t_{GRcyc}	8	—	—	ns	43.46
ETn_RX-DV setup time	t_{GRDVS}	2.5	—	—	ns	43.46
ETn_RX-DV hold time	t_{GRDVH}	0.5	—	—	ns	43.46
GETn_ERXD[7:4], ETn_ERXD[3:0] setup time	t_{GERDS}	2.5	—	—	ns	43.46

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
GETn_ERXD[7:4], ETn_ERXD[3:0] hold time	t_{GERDH}	0.5	—	—	ns	43.46
ETn_RX-ER setup time	t_{GRERS}	2.5	—	—	ns	43.47
ETn_RX-ER hold time	t_{GRERH}	0.5	—	—	ns	43.47
ETn_WOL output delay time	t_{GWOLD}	0	—	18	ns	43.48

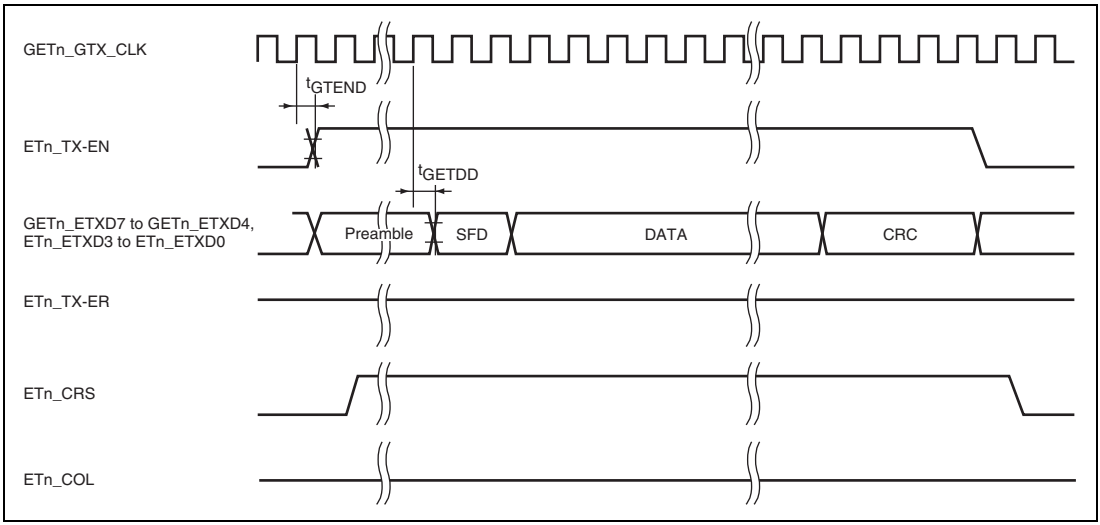


Figure 43.45 GMII Transmit Timing (normal operation)

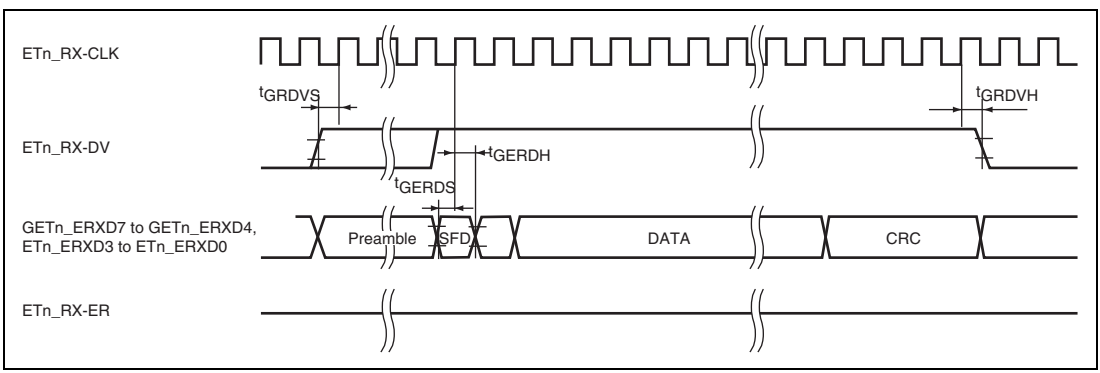


Figure 43.46 GMII Receive Timing (normal operation)

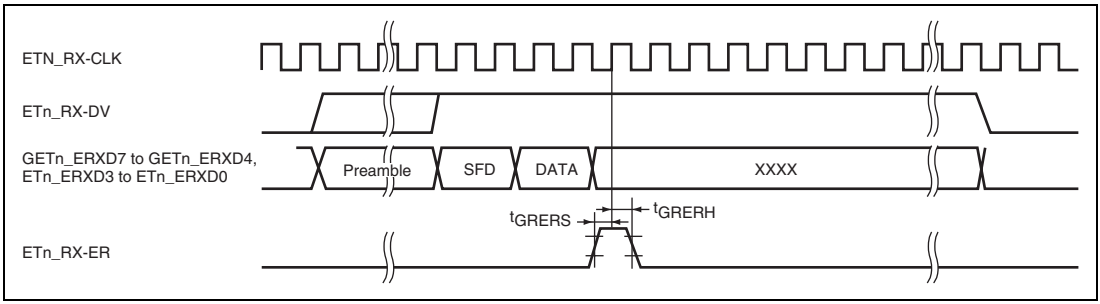


Figure 43.47 GMII Receive Timing (When an Error is Detected)

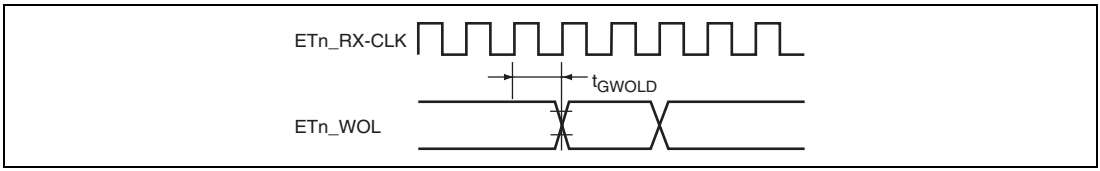


Figure 43.48 WOL Output Timing

(3) Ethernet Controller Signal Timing (RMII)

Table 43.21 Ethernet Controller Signal Timing (RMII)

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, V_{DD} = 1.15 to 1.35 V, Ta = −20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
REF50CK cycle time	t_{Rtcyc}	20	—	—	ns	43.49
RMIIn_TXD-EN, RMII1M_TXD-EN output delay time	t_{RTEND}	2.5	—	10	ns	43.49
RMIIn_TXD1, RMIIn_TXD0, RMII1M_TXD1, RMII1M_TXD0 output delay time	t_{RETDD}	2.5	—	10	ns	43.49
RMIIn_CRS_DV, RMII1M_CRS_DV setup time	t_{RRDVS}	4	—	—	ns	43.50
RMIIn_CRS_DV, RMII1M_CRS_DV hold time	t_{RRDVH}	2.5	—	—	ns	43.50
RMIIn_RXD1, RMIIn_RXD0, RMII1M_RXD1, RMII1M_RXD0 setup time	t_{RERDS}	4	—	—	ns	43.50

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
RMIIn_RXD1, RMIIn_RXD0, RMII1M_RXD1, RMII1M_RXD0 hold time	t_{RERDH}	2.5	—	—	ns	43.50
RMIIn_RX_ER setup time	t_{RRERS}	4	—	—	ns	43.51
RMIIn_RX_ER hold time	t_{RRERH}	2.5	—	—	ns	43.51

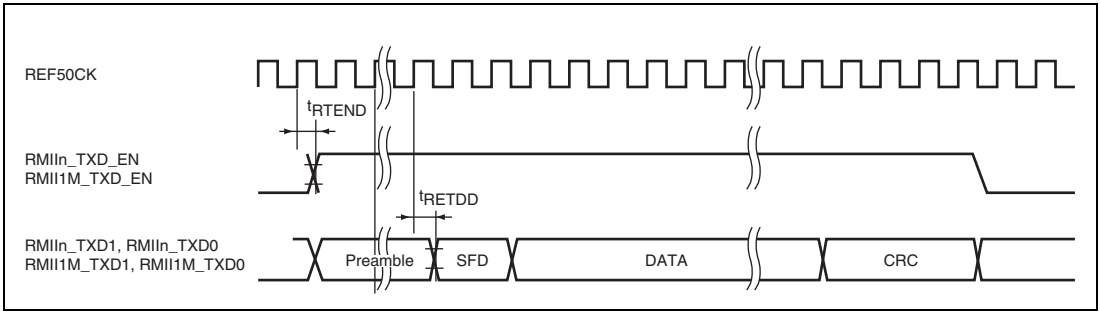


Figure 43.49 RMII Transmit Timing

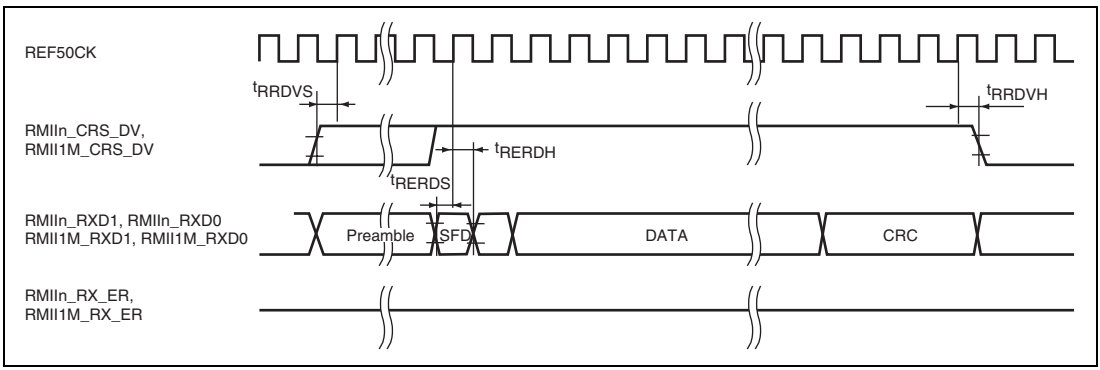


Figure 43.50 RMII Receive Timing (normal operation)

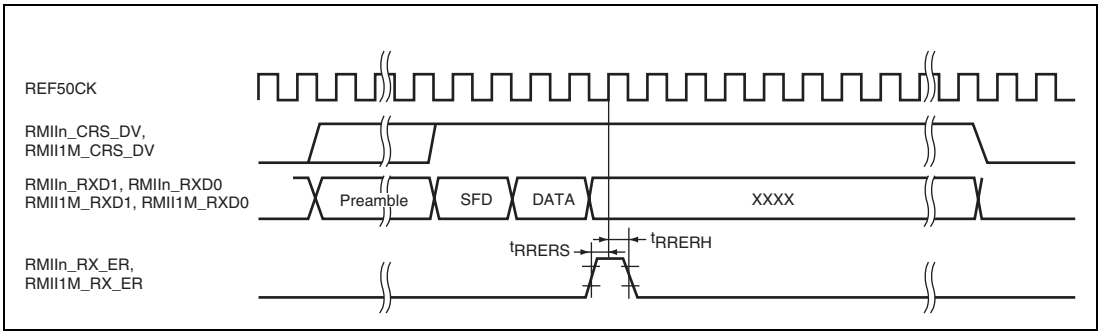


Figure 43.51 RMII Receive Timing (When an Error is Detected)

43.4.12 Stream Interface Module Timing

(1) Clock Valid Reception

Table 43.22 STIF Clock Valid Reception Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
ST_CLK cycle time	t_{STCYC}	30	—	ns	43.52
ST_REQ delay time	t_{STRQD}	4	21	ns	43.52
ST_START setup time	t_{STSTS}	7	—	ns	43.52
ST_START hold time	t_{STSTH}	4	—	ns	43.52
ST_VALID setup time	t_{STVLS}	7	—	ns	43.52
ST_VALID hold time	t_{STVLH}	4	—	ns	43.52
ST_DATA setup time	t_{STDS}	7	—	ns	43.52
ST_DATA hold time	t_{STDH}	4	—	ns	43.52

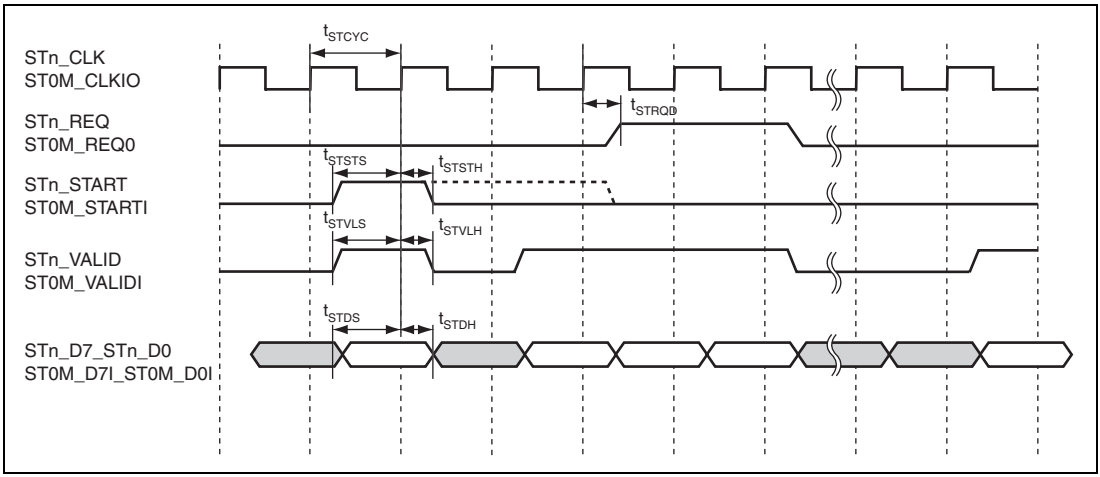


Figure 43.52 STIF Clock Valid Receive Timing

(2) Clock Valid Transmission

Table 43.23 STIF Clock Valid Transmission Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
ST_CLK cycle time	t_{STCYC}	30	—	ns	43.53
ST_REQ setup time	t_{STRQS}	7	—	ns	43.53
ST_REQ hold time	t_{STRQH}	5	—	ns	43.53
ST_START delay time	t_{STSTD}	3	21	ns	43.53
ST_VALID delay time	t_{STVLD}	3	21	ns	43.53
ST_DATA delay time	t_{STDD}	3	21	ns	43.53

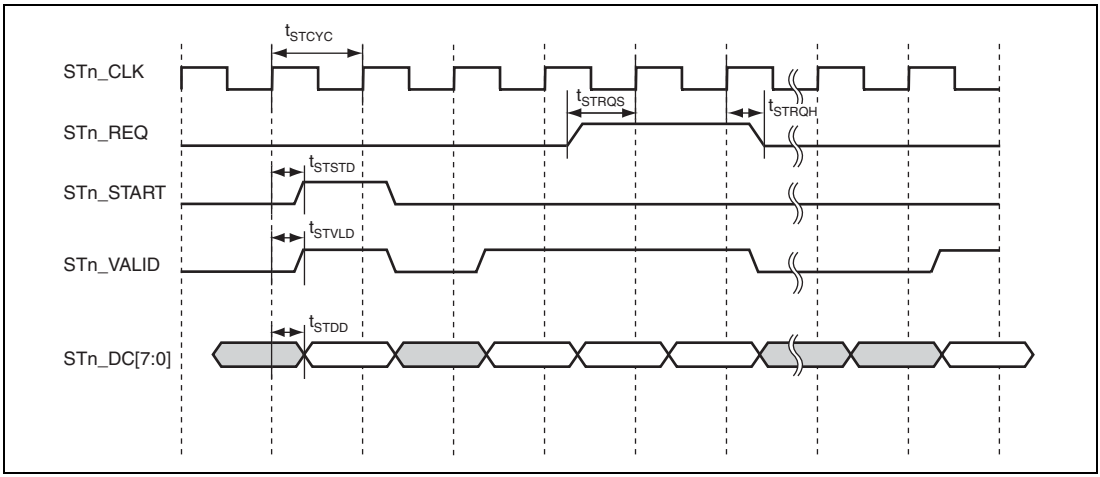


Figure 43.53 STIF Clock Valid Transmit Timing

(3) Strobe Reception

Table 43.24 STIF Strobe Reception Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
ST_STRB low level width	t_{STSLW}	30	—	ns	43.54
ST_STRB high level width	t_{STSHW}	30	—	ns	43.54
ST_REQ output delay time	t_{STRQD}	0	—	ns	43.54
ST_DATA setup time	t_{STDS}	7	—	ns	43.54
ST_DATA hold time	t_{STDH}	4	—	ns	43.54

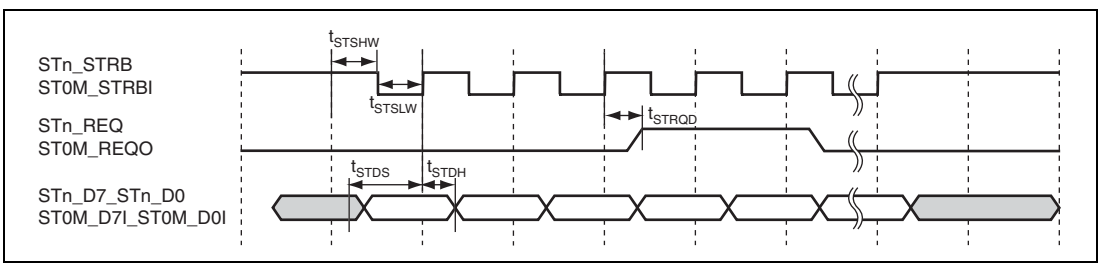


Figure 43.54 STIF Strobe Receive Timing

(4) Strobe Transmission

Table 43.25 STIF Strobe Transmission Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
STn_REQ hold time to STn_VALID	$t_{STRQRDH}$	0	—	ns	43.55
STn_STRB delay time from STn_REQ	$t_{STSRRQD}$	2	3	t_{STYCYC}	43.55
STn_STRB hold time from STn_REQ	$t_{STRQSRH}$	0	—	ns	43.55
ST_START delay time from STn_STRB	$t_{STSTSRD}$	–2	—	ns	43.55
ST_DATA delay time from STn_STRB	t_{STDSRD}	–1	—	ns	43.55

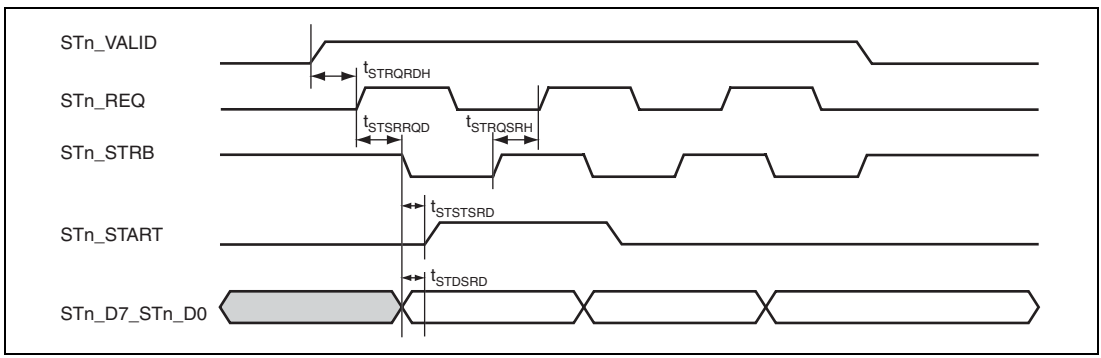


Figure 43.55 STIF Strobe Transmit Timing

43.4.13 I^2C Bus Interface Timing

Table 43.26 I^2C Bus Interface Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
IICn_SCL frequency	t_{ICYC}	0	—	400	kHz	43.56, 43.57
IICn_SCL, IICn_SDA fall time	t_{ICF}	—	—	300	ns	RP•CB = 257×10^{-9} to 275×10^{-9} [Ω•pF]
IICn_SDA input bus free time	t_{ICBF}	1.3	—	—	ns	VPU = 3.3V
IICn_SCL start condition input hold time	t_{ICH}	0.6	—	—	ns	
IICn_SCL retransmission start condition input setup time	t_{ICS}	0.6	—	—	ns	
IICn_SDA stop condition input setup time	t_{ICST}	0.6	—	—	ns	
IICn_SDA setup time	t_{DAS}	100	—	—	ns	
IICn_SDA hold time	t_{ICDH}	0	—	0.9	ns	

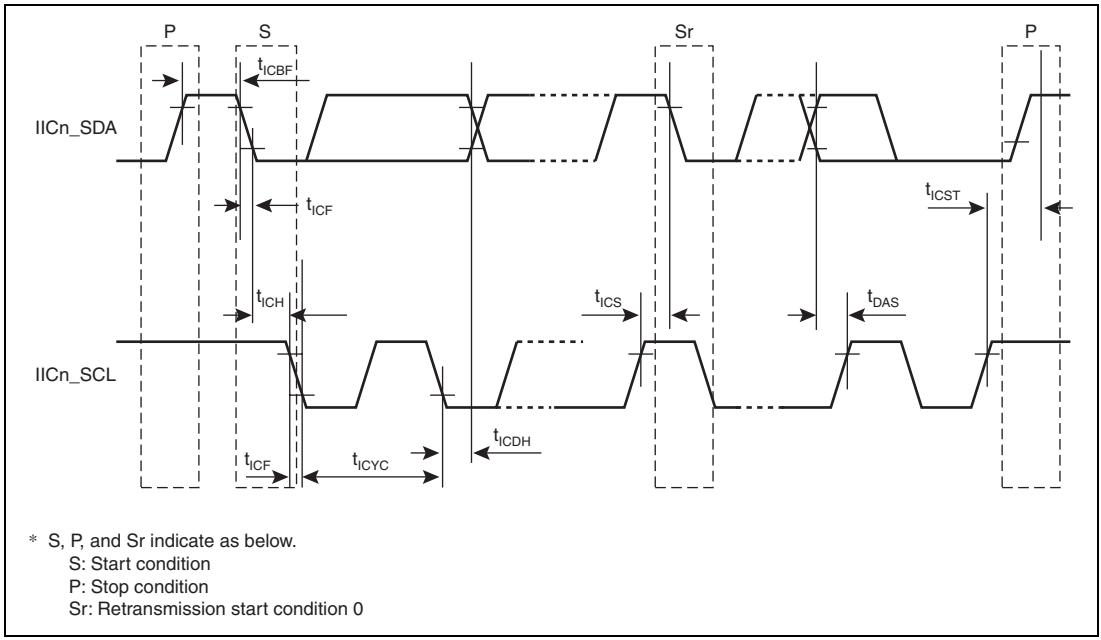


Figure 43.56 I²C Bus Interface Input/Output Timing

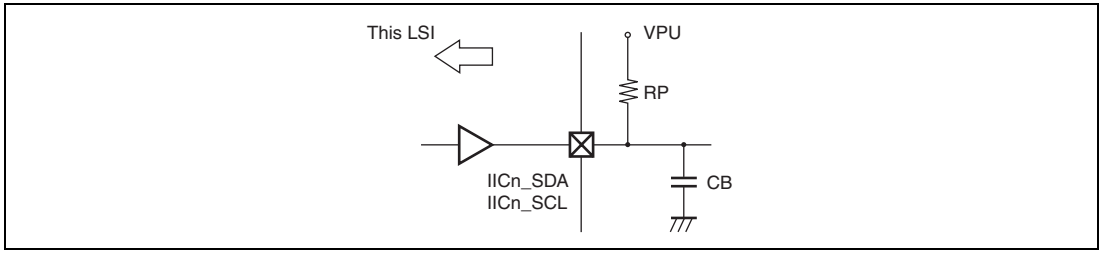


Figure 43.57 AC Characteristic Load Condition

43.4.14 SCIF Module Signal Timing

Table 43.27 SCIF Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
Input clock cycle (asynchronous)	t_{Scyc}	8	—	t_{Pcyc0}	43.58
Input clock cycle (synchronous)		24	—	t_{Pcyc0}	43.58
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Pcyc0}	43.58
Input clock rise time	t_{SCKr}	—	0.8	t_{Pcyc0}	43.58
Input clock fall time	t_{SCKf}	—	0.8	t_{Pcyc0}	43.58
Transfer data delay time	t_{TXD}	—	$6 \times t_{Pcyc0} + 50$	ns	43.59
Receive data setup time (synchronous)	t_{RXS}	$4 \times t_{Pcyc0}$	—	ns	43.59
Receive data hold time (synchronous)	t_{RXH}	$4 \times t_{Pcyc0}$	—	ns	43.59

Note: t_{Pcyc0}: One Pck0 cycle time

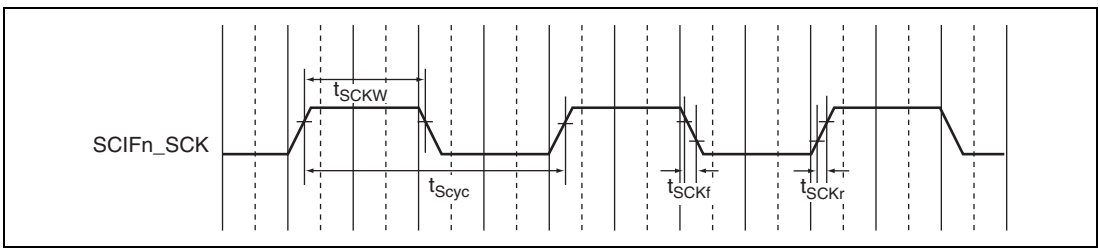


Figure 43.58 SCIFn_SCK Input Clock Timing

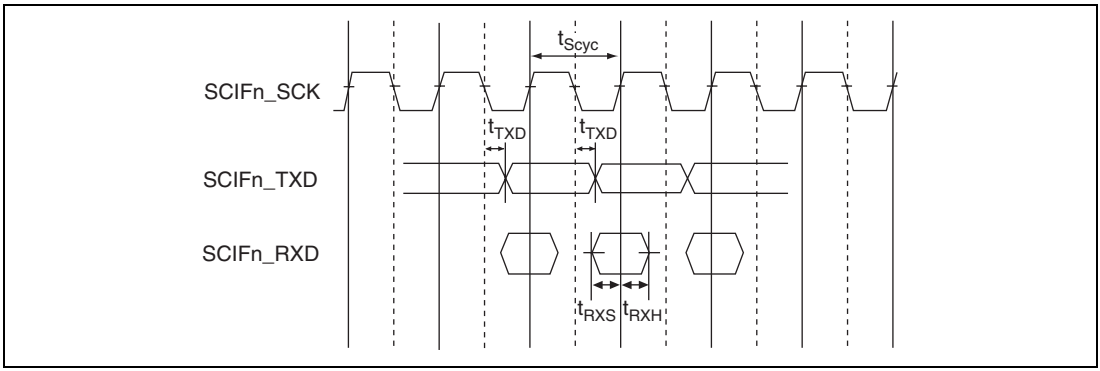


Figure 43.59 SCIFn I/O Synchronous Mode Clock Timing

43.4.15 SIOF Module Signal Timing

Table 43.28 SIOF Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = −20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
SIOFn_MCK clock input cycle time	t_{MCYC}	$2 \times t_{Pcyc0}$*	—	ns	43.60
SIOFn_MCK input high level width	t_{MWH}	$0.4 \times t_{MCYC}$	—	ns	43.60
SIOFn_MCK input low level width	t_{MWL}	$0.4 \times t_{MCYC}$	—	ns	43.60
SIOFn_SCK clock cycle time	t_{SICYC}	$2 \times t_{Pcyc0}$*	—	ns	43.61 to 43.65
SIOFn_SCK output high level width	t_{SWHO}	$0.4 \times t_{MCYC}$	—	ns	43.61 to 43.64
SIOFn_SCK output low level width	t_{SWLO}	$0.4 \times t_{MCYC}$	—	ns	43.61 to 43.64
SIOFn_SYNC output delay time	t_{FSD}	—	20	ns	43.61 to 43.64
SIOFn_SCK input high level width	t_{SWHI}	$0.4 \times t_{SICYC}$	—	ns	43.65
SIOFn_SCK input low level width	t_{SWLI}	$0.4 \times t_{SICYC}$	—	ns	43.65
SIOFn_SYNC input setup time	t_{FSS}	20	—	ns	43.65
SIOFn_SYNC input hold time	t_{FSH}	20	—	ns	43.65
SIOFn_TXD output delay time	t_{STDD}	—	20	ns	43.61 to 43.65
SIOFn_RXD input setup time	t_{SRDS}	20	—	ns	43.61 to 43.65
SIOFn_RXD input hold time	t_{SRDH}	20	—	ns	43.61 to 43.65

Note: * t_{Pcyc0} is a cycle time of a peripheral clock 0(Pck0).

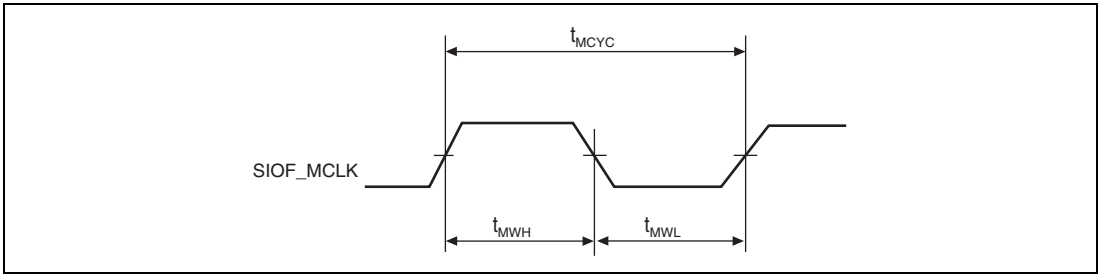


Figure 43.60 SIOF_MCLK Input Timing

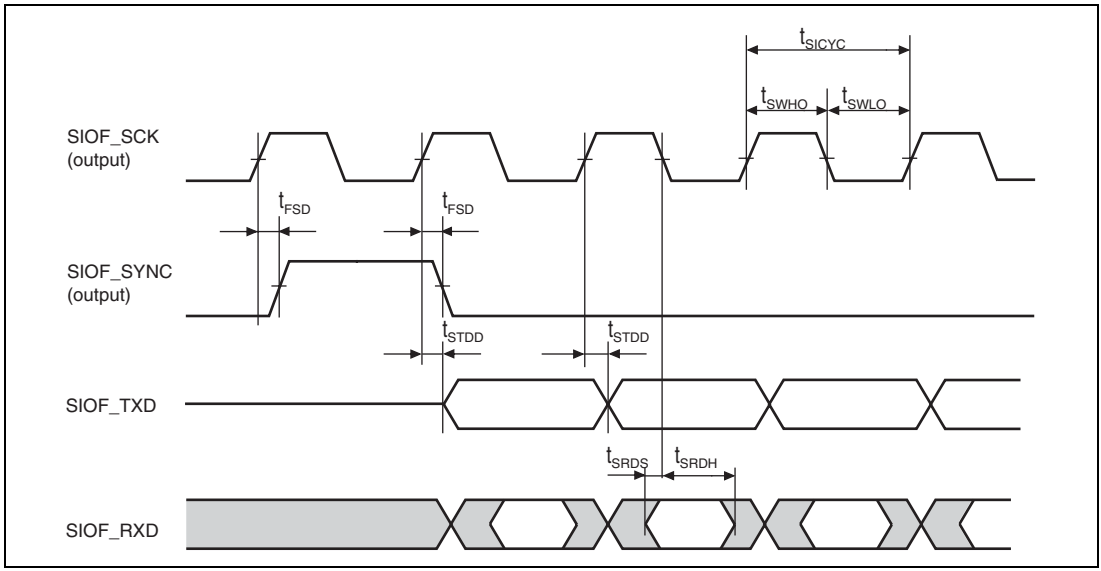


Figure 43.61 SIOF Transmission/Reception Timing (Master Mode 1, Sampling at the Falling Edge)

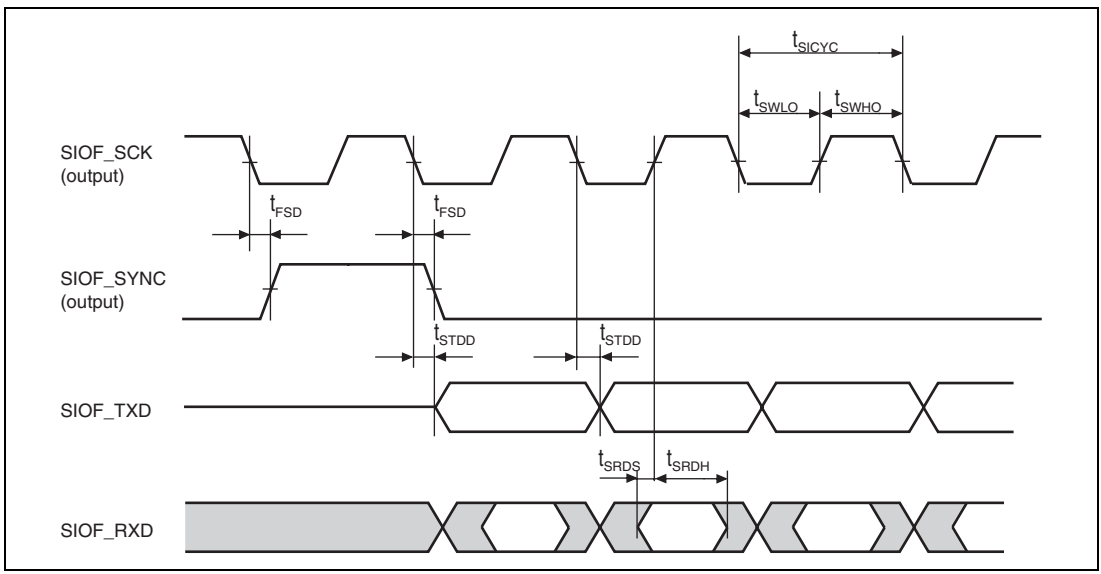


Figure 43.62 SIOF Transmission/Reception Timing (Master Mode 1, Sampling at the Rising Edge)

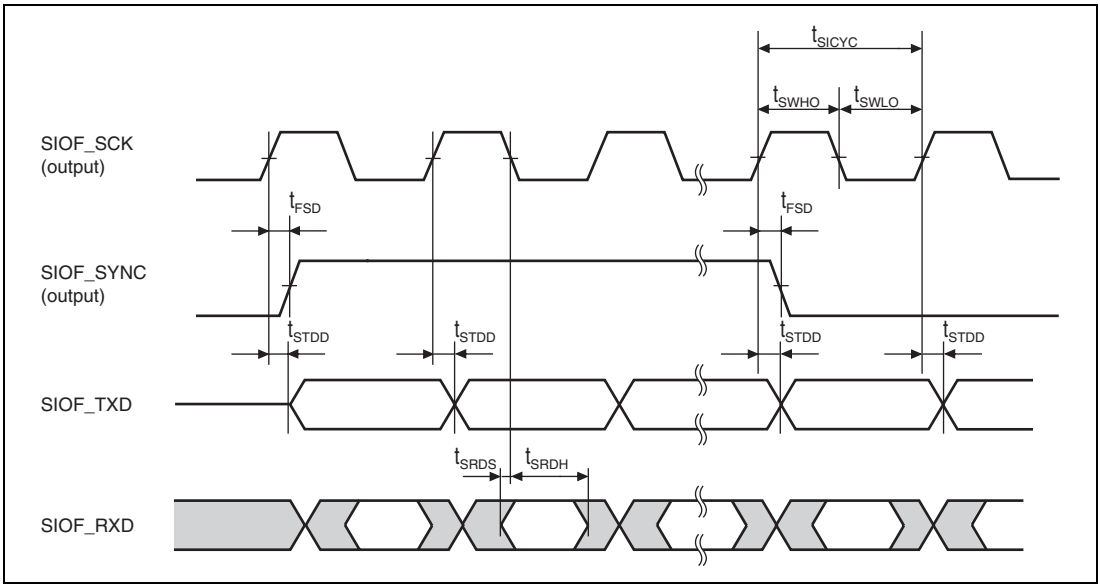


Figure 43.63 SIOF Transmission/Reception Timing (Master Mode 2, Sampling at the Falling Edge)

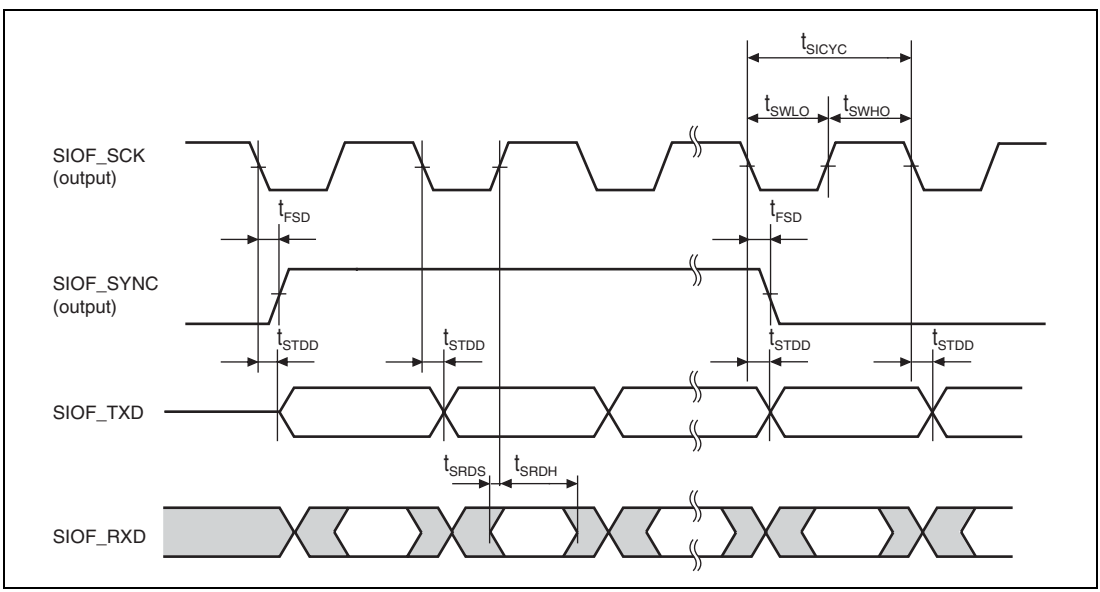


Figure 43.64 SIOF Transmission/Reception Timing (Master Mode 2, Sampling at the Rising Edge)

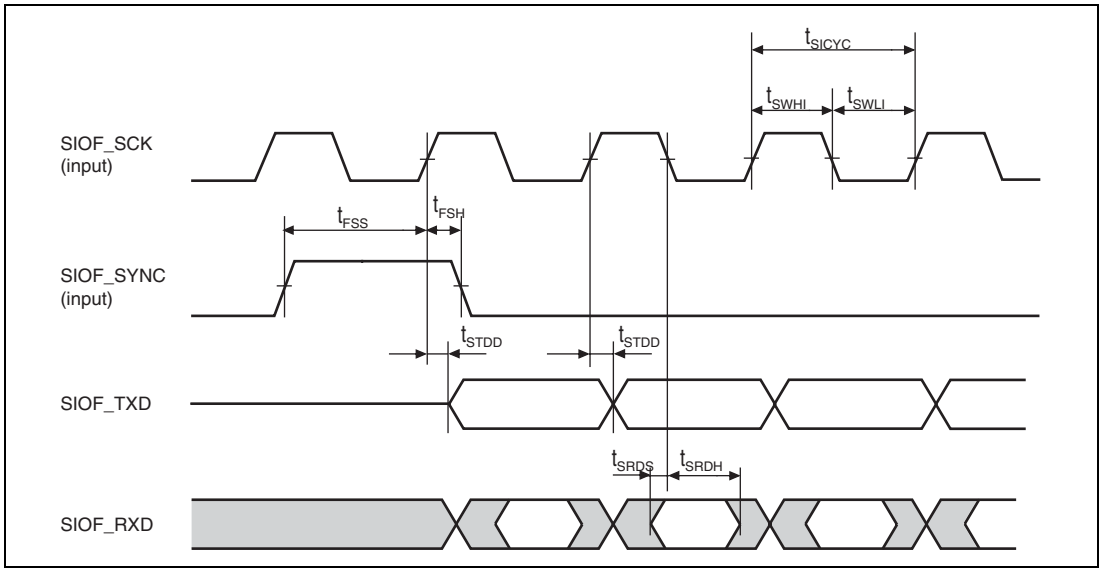


Figure 43.65 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2)

43.4.16 SIM Module Signal Timing

Table 43.29 SIM Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
SIM_CLK clock cycle	t_{SMCYC}	$2/t_{Pcyc0}$	$16/t_{Pcyc0}$	ns	43.66
SIM_CLK clock high level width	t_{SMCWH}	$0.4 \times t_{SMCYC}$	—	ns	
SIM_CLK clock low level width	t_{SMCWL}	$0.4 \times t_{SMCYC}$	—	ns	
SIM_RST reset output delay	t_{SMRD}	—	20	ns	

Note: t_{Pcyc0} is a cycle time of a peripheral clock (Pch0).

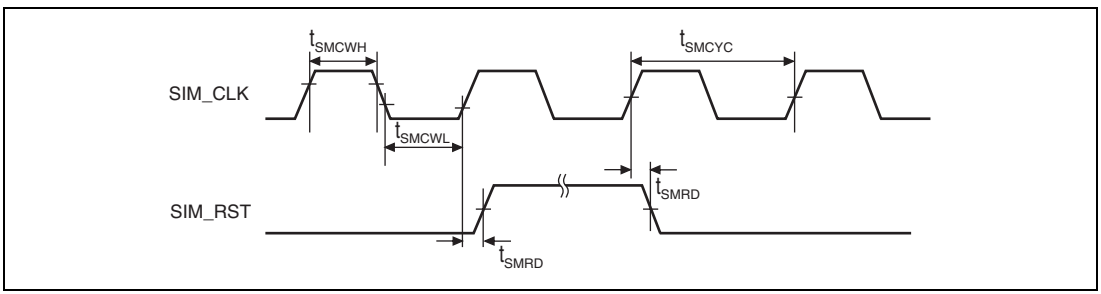


Figure 43.66 SIM Module Signal Timing

43.4.17 MMCIF Module Signal Timing

Table 43.30 MMCIF Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
MMC_CLK clock cycle time	t_{MMCYC}	60	—	ns	43.67
MMC_CLK clock high level width	t_{MMWH}	0.4 × t_{MMCYC}	—	ns	43.67
MMC_CLK clock low level width	t_{MMWL}	0.4 × t_{MMcyc}	—	ns	43.67
MMC_CMD output data delay time	t_{MMCD}	—	10	ns	43.67
MMC_CMD input data setup time	t_{MMRCS}	10	—	ns	43.68
MMC_CMD input data hold time	t_{MMRCH}	10	—	ns	43.68
MMC_DAT output data delay time	t_{MMTDD}	—	10	ns	43.67
MMC_DAT input data setup time	t_{MMRDS}	10	—	ns	43.68
MMC_DAT input data hold time	t_{MMRDH}	10	—	ns	43.68

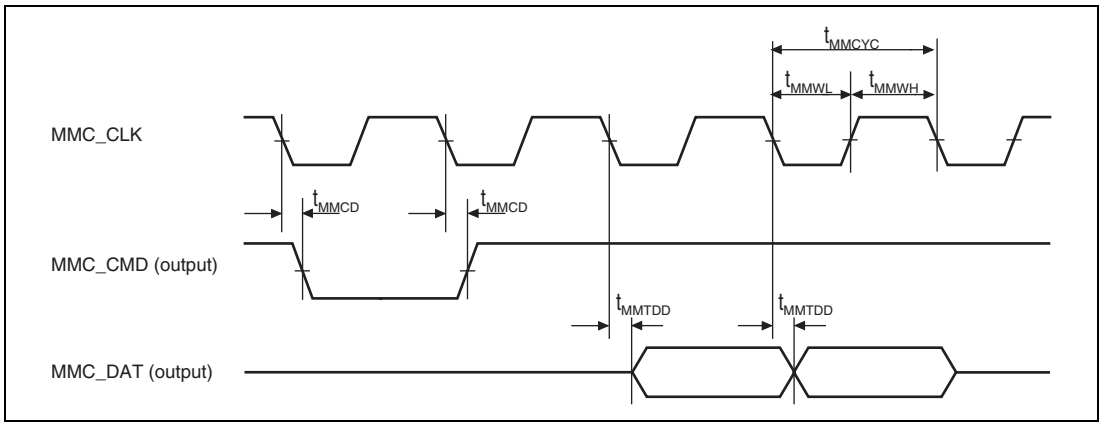


Figure 43.67 MMCIF Transmit Timing

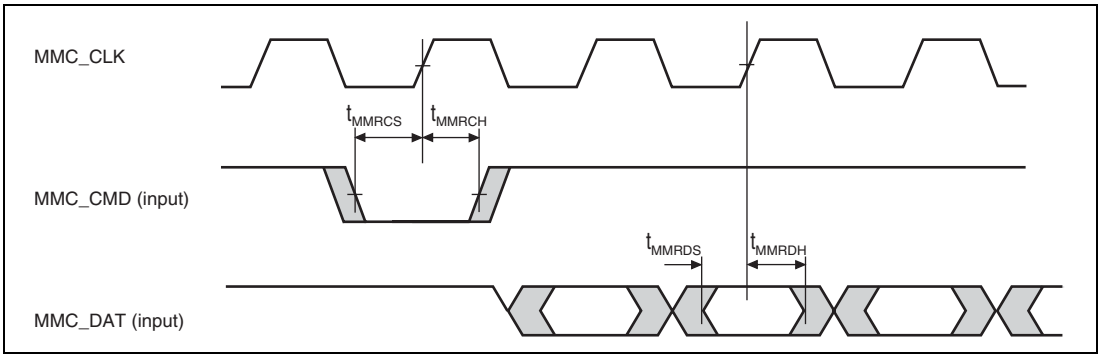


Figure 43.68 MMCIF Receive Timing (Sampling at the Rising Edge)

43.4.18 HAC Interface Module Signal Timing

Table 43.31 HAC Interface Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
HAC_RES active low pulse width	t_{RST_LOW}	1000	—	ns	43.69
HAC_SYNC active high pulse width	t_{SYN_HIGH}	1000	—	ns	43.70
HAC_SYNC delay time 1	t_{SYNCD1}	0	15	ns	43.72
HAC_SYNC delay time 2	t_{SYNCD2}	0	15	ns	43.72
HAC_SD_OUT delay time	t_{SDOUTD}	0	15	ns	43.72
HAC_SD_IN setup time	t_{SDINS}	10	—	ns	43.72
HAC_SD_IN hold time	t_{SDINH}	10	—	ns	43.72
HAC_BITCLK input high level width	t_{ICL_HIGH}	t_{Pcyc0}	—	ns	43.71
HAC_BITCLK input low level width	t_{ICL_LOW}	t_{Pcyc0}	—	ns	43.71

Note: t_{Pcyc0}: One Pck0 cycle time

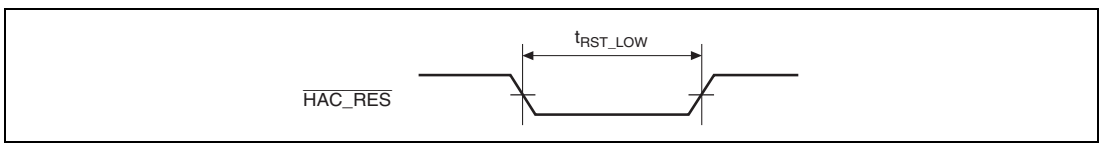


Figure 43.69 HAC Cold Reset Timing

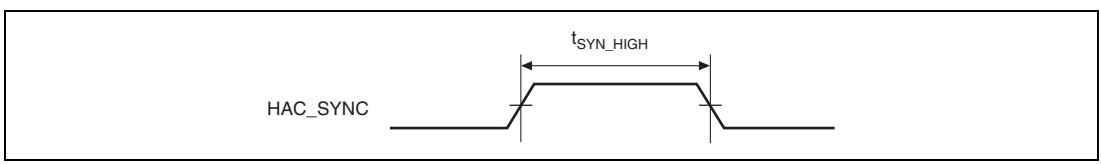


Figure 43.70 HAC SYNC Output Timing

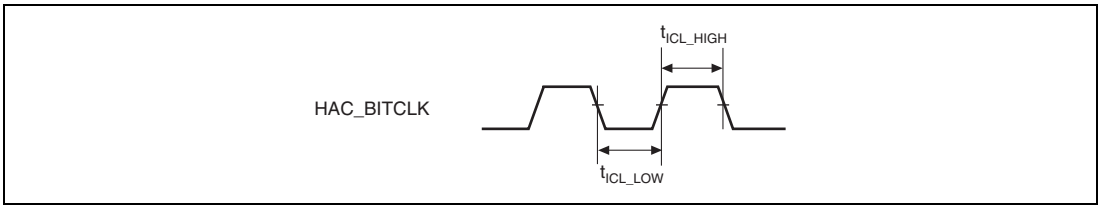


Figure 43.71 HAC Clock Input Timing

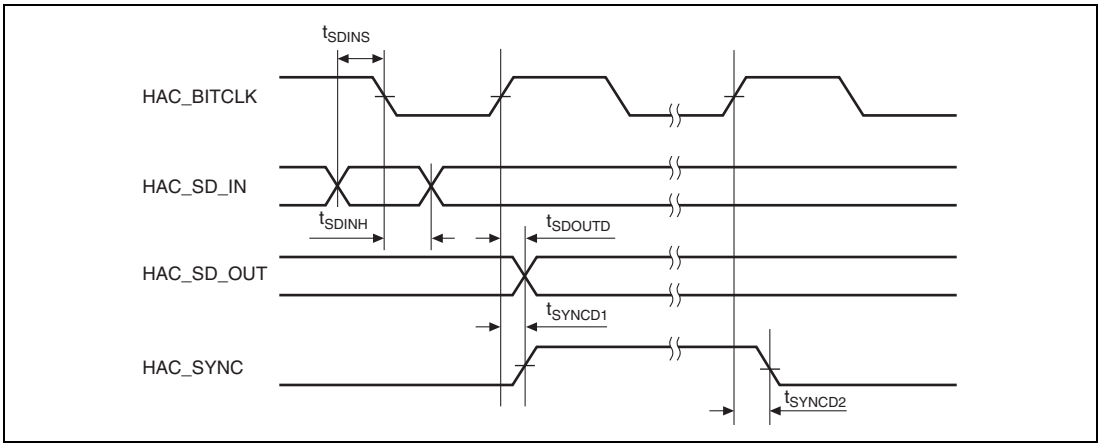


Figure 43.72 HAC Interface Module Signal Timing

43.4.19 SSI Interface Module Signal Timing

Table 43.32 SSI Interface Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = −20 to 75°C

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output cycle time	t_{OSCK}	60	960	ns	output	43.73
Input cycle time	t_{ISCK}	60	3300	ns	input	43.73
Input high level width/Output high level width	t_{IHC}/t_{OHC}	15	—	ns	input, output	43.73
Input low level width/Output low level width	t_{ILC}/t_{OLC}	15	—	ns	input, output	43.73
SSI_SCK output rise time	t_{RC}	—	10	ns	output	43.73
SSI_SDATA/WS output delay time	t_{DTR}	—	25	ns	transmit	43.74, 43.75
SSI_SDATA/WS input setup time	t_{SR}	10	—	ns	receive	43.76, 43.77
SSI_SDATA/WS input hold time	t_{HTR}	10	—	ns	receive	43.76, 43.77

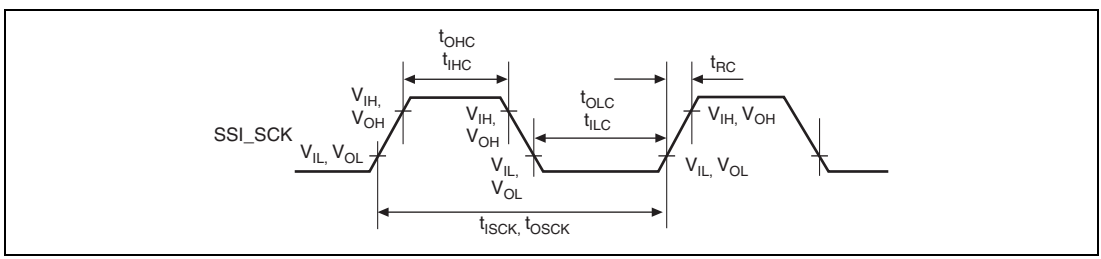


Figure 43.73 SSI Clock Input/Output Timing

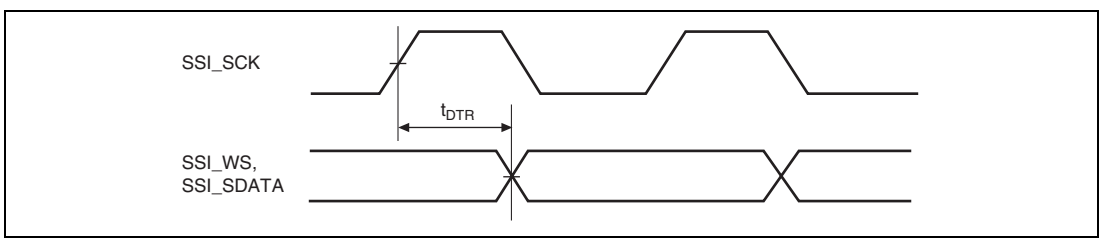


Figure 43.74 SSI Transmit Timing (1)

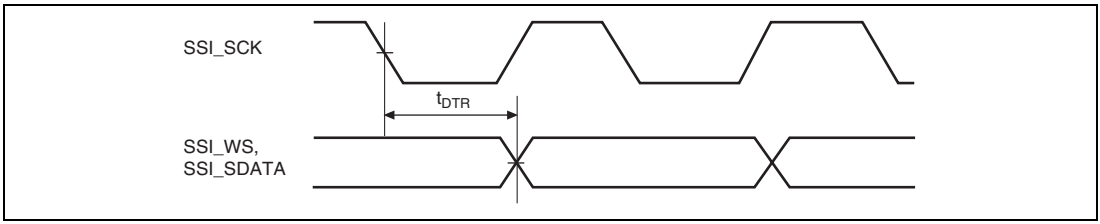


Figure 43.75 SSI Transmit Timing (2)

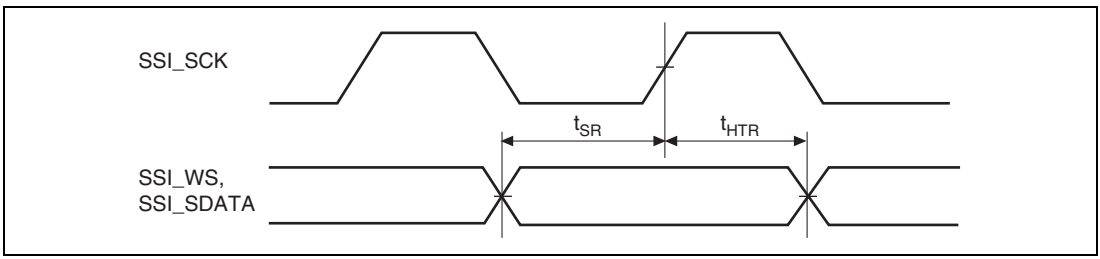


Figure 43.76 SSI Receive Timing (1)

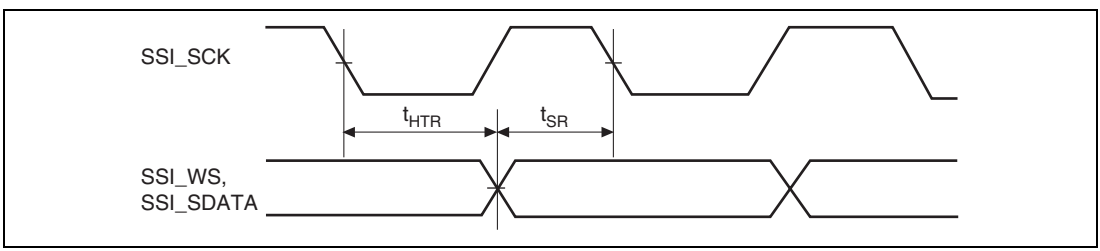


Figure 43.77 SSI Receive Timing (2)

43.4.20 USB Module Signal Timing

Table 43.33 USB Module Clock Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
USB_CLK external input clock frequency (48 MHz)	t_{FREQ}	47.9	48.1	MHz	43.78
Clock rise time	t_{R48}	—	4	ns	
Clock fall time	t_{F48}	—	4	ns	
Duty (t_{HIGH}/t_{LOW})	t_{DUTY}	90	110	%	

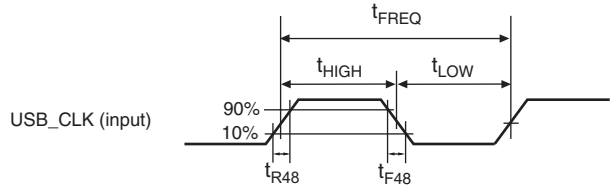


Figure 43.78 USB Clock Timing

Table 43.34 USB Electrical Characteristics (Full-Speed)

Item	Symbol	Min.	Max.	Unit	Condition*[1]
Transition time (rise)*[2]	t_R	4	20	ns	CL = 50 pF
Transition time (fall)*[2]	t_F	4	20	ns	CL = 50 pF
Rise/fall time matching	t_{RFM}	90	111	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.0	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 22 Ω.

1. Value when CL = 50 pF unless specified.
2. Value within 10% to 90% of the signal power supply voltage.

Table 43.35 USB Electrical Characteristics (Low-Speed)

Item	Symbol	Min.	Max.	Unit	Condition
Transition time (rise)*	t_R	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Transition time (fall)*	t_F	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Rise/fall time matching	t_{RFM}	80	125	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.0	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 22 Ω.

* Value within 10% to 90% of the signal power supply voltage.

43.4.21 LCDC Module Signal Timing

Table 43.36 LCDC Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
LCD_CLK input clock frequency	t_{FREQ}	—	66	MHz	
LCD_CLK input clock rise time	t_r	—	3	ns	
LCD_CLK input clock fall time	t_f	—	3	ns	
LCD_CLK input clock duty	t_{DUTY}	90	110	%	
Clock (LCD_CL2) cycle time	t_{CC}	25	—	ns	43.79
Clock (LCD_CL2) high level pulse width	t_{CHW}	7	—	ns	
Clock (LCD_CL2) low level pulse width	t_{CLW}	7	—	ns	
Clock (LCD_CL2) transition time (rise/fall)	t_{CT}	—	3	ns	
Data (LCD_DATA) delay time	t_{DDdo}	–3.5	3	ns	
Display enable (LCD_M_DISP) delay time	t_{IDdo}	–3.5	3	ns	
Horizontal synchronous signal (LCD_CL1) delay time	t_{HDdo}	–3.5	3	ns	
Vertical synchronous signal (LCD_FLM) delay time	t_{VDdo}	–3.5	3	ns	

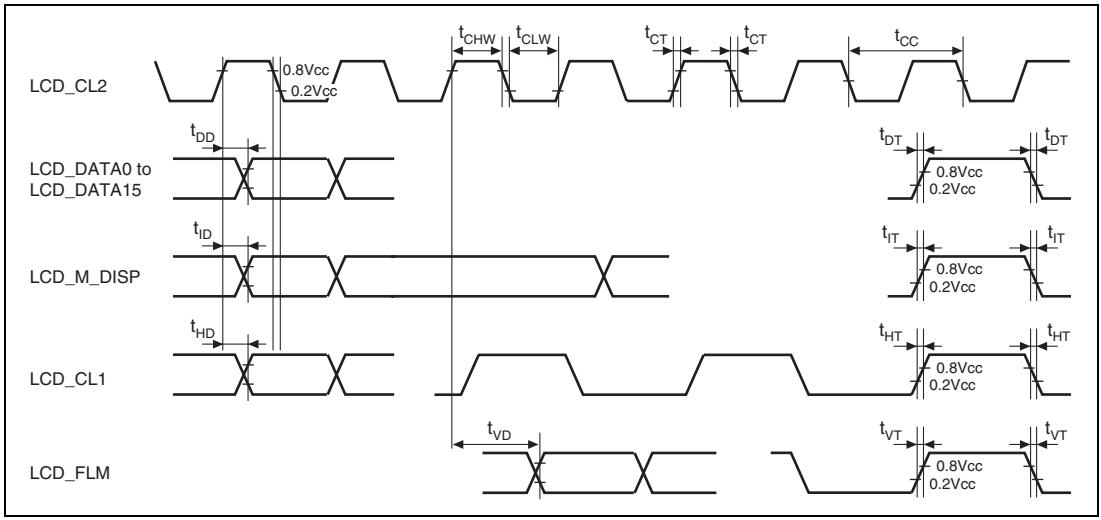


Figure 43.79 LCDC Module Signal Timing

43.4.22 GPIO Signal Timing

Table 43.37 GPIO Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
GPIO output delay time	t_{IOPD}	0	15	ns	43.80
GPIO input setup time	t_{IOPS}	15	—	ns	43.80
GPIO input hold time	t_{IOPH}	5	—	ns	43.80

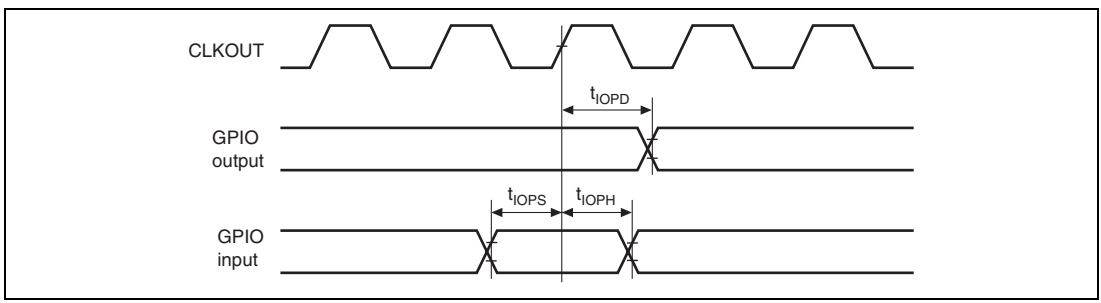


Figure 43.80 GPIO Timing

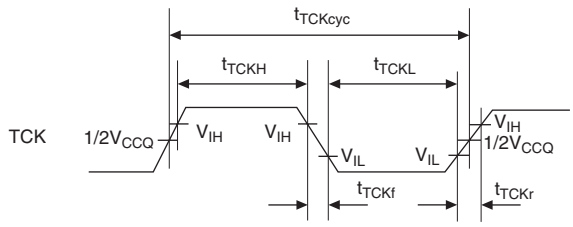
43.4.23 H-UDI Module Signal Timing

Table 43.38 H-UDI Module Signal Timing

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ-DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Symbol	Min.	Max.	Unit	Figure
Input clock cycle	t_{TCKcyc}	50	—	ns	43.81, 43.83
Input clock pulse width (High)	t_{TCKH}	15	—	ns	43.81
Input clock pulse width (Low)	t_{TCKL}	15	—	ns	43.81
Input clock rise time	t_{TCKr}	—	10	ns	43.81
Input clock fall time	t_{TCKf}	—	10	ns	43.81
$\overline{\text{ASEBRK}}$ setup time	$t_{ASEBRKS}$	10	—	t_{cyc}	43.82
$\overline{\text{ASEBRK}}$ hold time	$t_{ASEBRKH}$	10	—	t_{cyc}	43.82
TDI/TMS setup time	t_{TDIS}	15	—	ns	43.83
TDI/TMS hold time	t_{TDIH}	15	—	ns	43.83
TDO data delay time	t_{TDO}	0	10	ns	43.83
$\overline{\text{ASEBRK}}$ pin break pulse width	t_{PINBRK}	2	—	t_{Pcyc0}	43.84

Notes: 1. t_{cyc}: One $\overline{\text{CLKOUT}}$ cycle time
2. t_{Pcyc0}: One Pck0 cycle time



Note: When clock is input from TCK pin.

Figure 43.81 TCK Input Timing

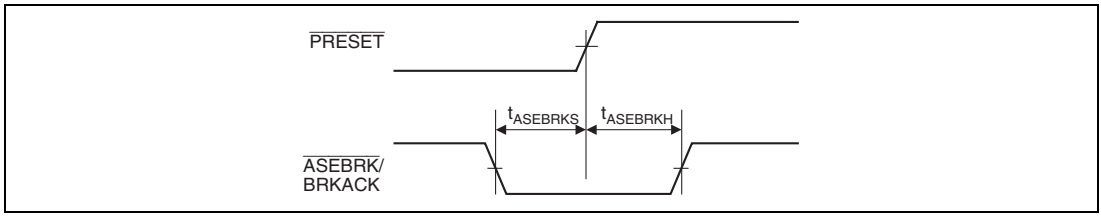


Figure 43.82 $\overline{\text{PRESET}}$ Hold Timing

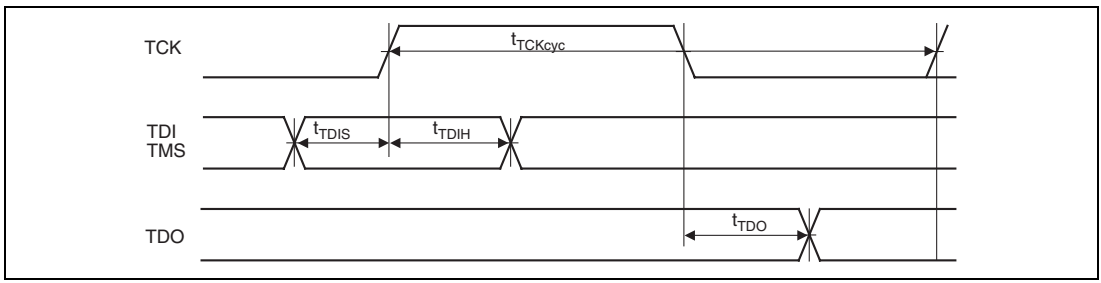


Figure 43.83 H-UDI Data Transfer Timing

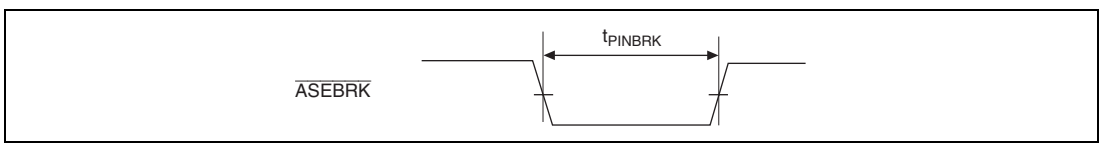


Figure 43.84 $\overline{\text{ASEBRK}}$ Pin Break Timing

43.5 A/D, D/A Converter Characteristics

43.5.1 A/D Converter Characteristics

Table 43.39 A/D Converter Characteristics

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	8.5	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal source (single source) impedance	—	—	3.5	kΩ
Absolute accuracy	—	—	±4.0	LSB

43.5.2 D/A Converter Characteristics

Table 43.40 D/A Converter Characteristics

Conditions: V_{CCQ} = VDD_RTC = AV_{CC} = 3.0 to 3.6 V, $V_{CCQ\text{-}DDR}$ = 2.3 to 2.7 V, VDD = 1.15 to 1.35 V, Ta = –20 to 75°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20 pF capacitive load
Absolute accuracy	—	—	±4.0	LSB	2 MΩ resistance load

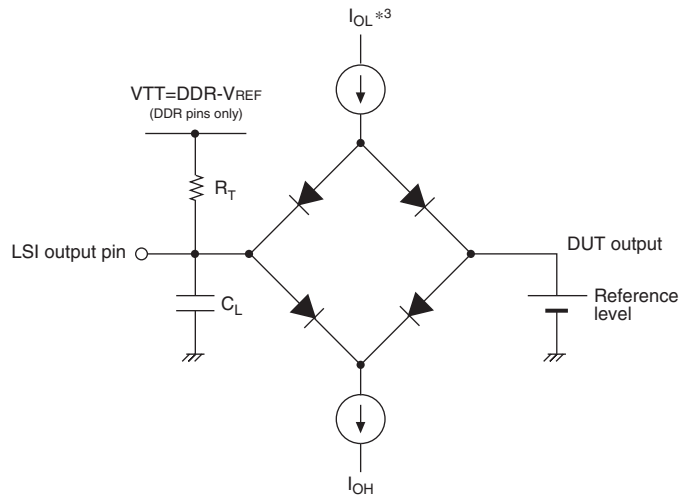
43.6 AC Characteristic Test Conditions

The AC characteristic test conditions are as follows:

- Input/output signal reference level: V*/2
- Input pulse level: V_{SSQ} to V*
- Input rise/fall time: 1 ns

Note: V*: VCCQ, $V_{CCQ\text{-}DDR}$ (V_{DDQ} = 3.0 to 3.6V, VCCQ_DDR = 2.3 to 2.7V)

The output load circuit is shown in figure 43.85



Notes: 1. C_L = 30pF (All pins). C_L is the total value that includes the capacitance of measurement instruments.
The capacitance of each pin is set to 30 pF.
2. R_T = 50Ω (DDR pins only)
3. I_{OL} = 7.6 mA (DDR pins),
4 mA (PCI pins),
2 mA (Other output pins)
I_{OH} = –7.6 mA (DDR pins),
–4 mA (PCI pins),
–2 mA (Other output pins)

Figure 43.85 Output Load Circuit

43.7 Change in Delay Time Based on Load Capacitance

Figure 43.86 is a chart showing the changes in the delay time (reference data) when a load capacitance equal to or larger than the stipulated value (30 pF) is connected to the LSI pins. When connecting an external device with a load capacitance exceeding the regulation, use the chart in figure 43.86 as reference for system design.

Note that if the load capacitance to be connected exceeds the range shown in figure 43.86 the graph will not be a straight line.

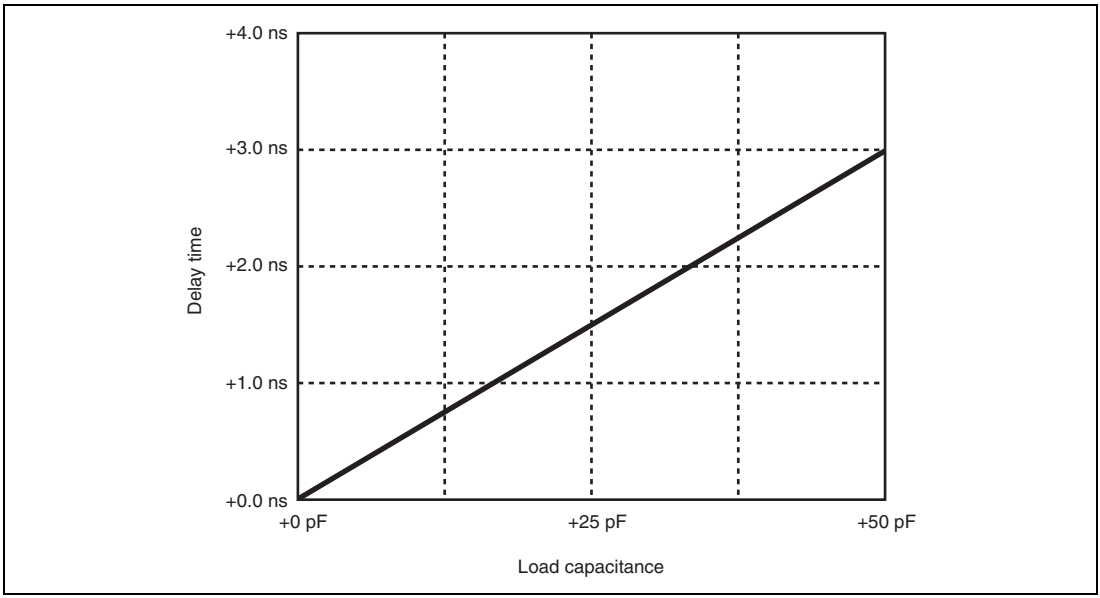


Figure 43.86 Load Capacitance - Delay Time

Appendix

A. CPU Operation Mode Register (CPUOPM)

The CPUOPM is used to control the CPU operation mode. This register can be read from or written to the address H'FF2F0000 in P4 area or H'1F2F0000in area 7 as 32-bit size. The write value to the reserved bits should be the initial value. The operation is not guaranteed if the write value is not the initial value.

The CPUOPM register should be updated by the CPU store instruction not the access from SuperHyway bus master except CPU.

After the CPUOPM is updated, read CPUOPM once, and execute one of the following two methods.

1. Execute a branch using the RTE instruction.
2. Execute the ICBI instruction for any address (including non-cacheable area).

After one of these methods are executed, it is guaranteed that the CPU runs under the updated CPUOPM value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RABD	—	INTMU	—	—	—
Initial value:	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	H'000000F	R	Reserved The write value must be the initial value.
5	RABD	1	R/W	Speculative execution bit for subroutine return 0: Instruction fetch for subroutine return is issued speculatively. When this bit is set to 0, refer to appendix C, Speculative Execution for Subroutine Return. 1: Instruction fetch for subroutine return is not issued speculatively.
4	—	0	R	Reserved The write value must be the initial value.
3	INTMU	0	R/W	Interrupt mode switch bit 0: SR.IMASK is not changed when an interrupt is accepted. 1: SR.IMASK is changed to the accepted interrupt level.
2 to 0	—	All 0	R	Reserved The write value must be the initial value.

B. Instruction Prefetching and Its Side Effects

This LSI is provided with an internal buffer for holding pre-read instructions, and always performs pre-reading. Therefore, program code must not be located in the last 64-byte area of any memory space. If program code is located in these areas, a bus access for instruction prefetch may occur exceeding the memory areas boundary. A case in which this is a problem is shown below.

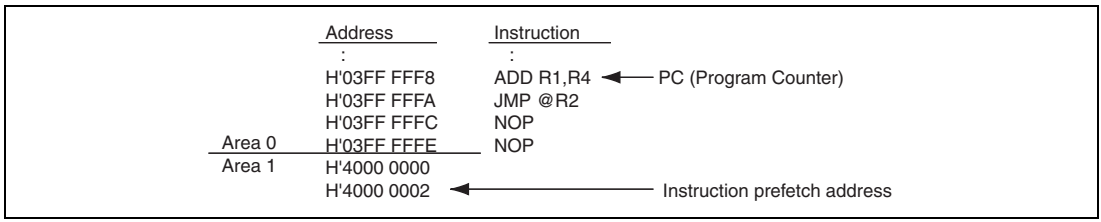


Figure B.1 Instruction Prefetch

Figure B.1 presupposes a case in which the instruction (ADD) indicated by the program counter (PC) and the address H'04000002 instruction prefetch are executed simultaneously. It is also assumed that the program branches to an area other than area 1 after executing the following JMP instruction and delay slot instruction.

In this case, a bus access (instruction prefetch) to area 1 may unintentionally occur from the programming flow.

Instruction Prefetch Side Effects:

1. It is possible that an external bus access caused by an instruction prefetch may result in misoperation of an external device, such as a FIFO, connected to the area concerned.
2. If there is no device to reply to an external bus request caused by an instruction prefetch, hang-up will occur.

Remedies:

1. These illegal instruction fetches can be avoided by using the MMU.
2. The problem can be avoided by not locating program code in the last 64 bytes of any area.

C. Speculative Execution for Subroutine Return

The SH-4A has the mechanism to issue an instruction fetch speculatively when returning from subroutine. By issueing an instruction fetch speculatively, the execution cycles to return from subroutine may be shortened.

This function is enabled by setting 0 to the bit 5 (RABD) of CPU Operation Mode register (CPUOPM). But this speculative instruction fetch may issue the access to the address that should not be accessed from the program. Therefore a bus access to an unexpected area or an internal instruction address error may cause a problem. As for the effect of this bus access to unexpected memory area, refer to appendix B, Instruction Prefetching and Its Side Effects.

Usage Condition: When the speculative execution for subroutine return is enabled, the RTS instruction should be used to return to the address set in PR by the JSR, BSR or BSRF instructions. It can prevent the access to unexpected address and avoid the problem.

D. List of Mode Control Pins and Schematic Diagram of External Cicuits

Table D.1 shows the list of mode control pins of this LSI.

Table D.1 Mode Control Pins

Pin Name	Function	I/O	Description
MD0, MD1, and MD2	Mode control pins 0,1 and 2	Input	Selects the clock-operating mode at a power-on reset.
MD3 and MD4	Mode control pins 3 and 4	Input	Selects the bus width and MPX interface of area 0 at a power-on reset
MD5	Mode control pin 5	Input	Selects the endian at a power-on reset
MD6	Mode control pin 6	Input	Selects the PCI operating mode at a power-on reset.
MD8	Mode control pin 8	Input	Selects whether to use the crystal resonator at a power-on reset.
MD10	Mode control pin 10	Input	Selects the external CPU at a power-on reset
MPMD	Chip mode control pin	Input	Selects either emulation support mode or LSI operation mode at a power-on reset.

The state of the mode control pins of this LSI is sampled white a power-on reset is applied. Therefore, these pins can serve as other functions in the other periods.

The method for switching between the control functions and other functions should be examined referring to figure D.1. See section 40, General Purpose I/O (GPIO), or the relevant sections of the applicable modules for whether the pins of the other functions are the input, output, or input/output pins.

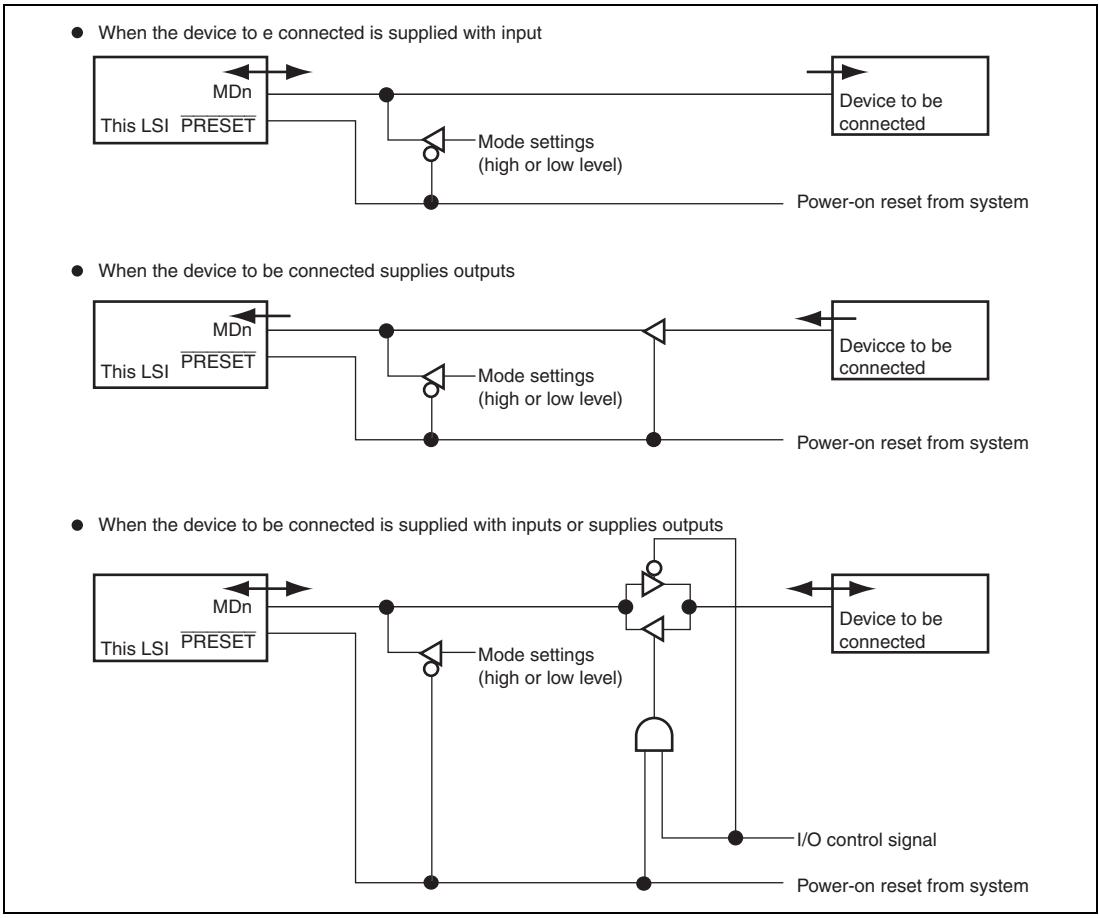


Figure D.1 Schematic Diagram of External Circuits

E. Notes on Board Design

A multi-layer ceramic capacitor should be used as a bypass capacitor for each pair of a Vss or a Vdd power supply pin and a Vcc pin.
The following table shows the pairs of a Vss or Vdd power supply pin and a Vcc pin in terms of the specific pin numbers.

Power Supply Pin Pair for	Vss/Vdd Pin Number	Vcc Pin Number
Internal circuits: 1.25V (Vdd – V CC)	V1	K10
	V2	L10
	V3	M10
	V4	N10
	V5	P10
	W5	Y5
	AA9	R10
	AA10	T10
	AA11	T11
	W21	V21
	U21	T21
	N21	M21
	K21	J21
	G21	F21
	D20	E20
	E16	E17
	E8	E9
DDR-SDRAM I/O: 2.5V (VCCQ-DDR - VSSQ-DDR	B1	B2
	C2	C3
	E4	F4
	E5	F5
	G4	H4
	G5	H5
	K4	J4
	K5	J5
	N4	P4

Power Supply Pin Pair for	Vss/Vdd Pin Number	Vcc Pin Number
DDR-SDRAM I/O: 2.5V (VCCQ-DDR - VSSQ-DDR	N5	P5
	R4	T3
	R5	U3
	T4	U4
	T5	U5
	D11	D10
	E11	E10
	D8	D9
	D7	D6
	E7	E6
	C4	D5
	D3	D4
	A2	A1
I/O: 3.3V (VCCQ – VSSQ	AE2	AE1
	AD3	AD2
	AC4	AC3
	AB5	AB4
	AA6	AA5
	AA8	AA7
	AA12	AA14
	AA13	AB14
	AA15	AA16
	AB15	AA17
	AA18	AA19
	AA20	AA21
	Y21	AB22
	AB23	AA22
	R21	P21
	L21	B23
	H21	A24
	E21	A25

Power Supply Pin Pair for	Vss/Vdd Pin Number	Vcc Pin Number
I/O: 3.3V (VCCQ - VSSQ	C21	C22
	D22	D21
	E18	E19
	D16	E15
	E14	D14
	AD23	AE25
	C13	D13
Analog circuits: 3.3V (AVcc-AVss	AD25/AA25/AA24*	AC23/T16*
DLL: 1.25V (Vcc-DLL - Vss-DLL)	M4	L4
	M5	L5
PLL: 1.25V (Vcc-PLL - Vss-PLL)	AE24	AD24
	AE21	AD21
	AE20	AD20
RTC: 3.3V (VDD-RTC - VSS-RTC)	D12	E12

Note * See figure E.1, Connection Exmple of Bypass Capacitors for Analog Power Supply.

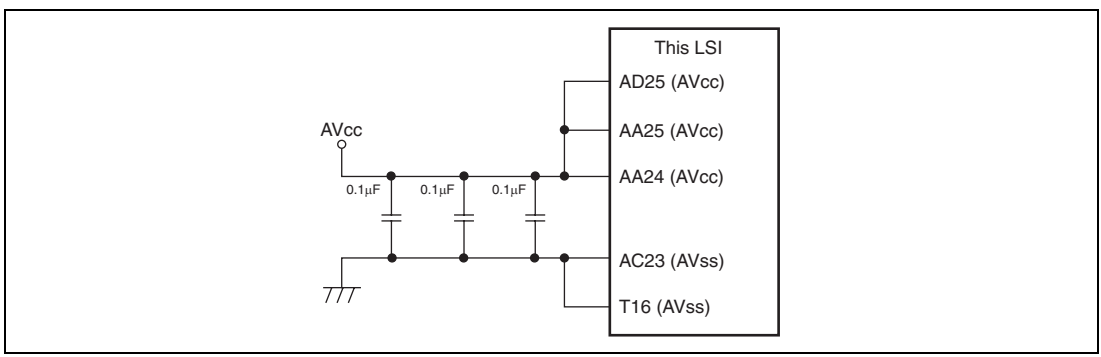


Figure E.1 Connection Example of Bypass Capacitors for Analog Power Supply

F. Package Dimensions

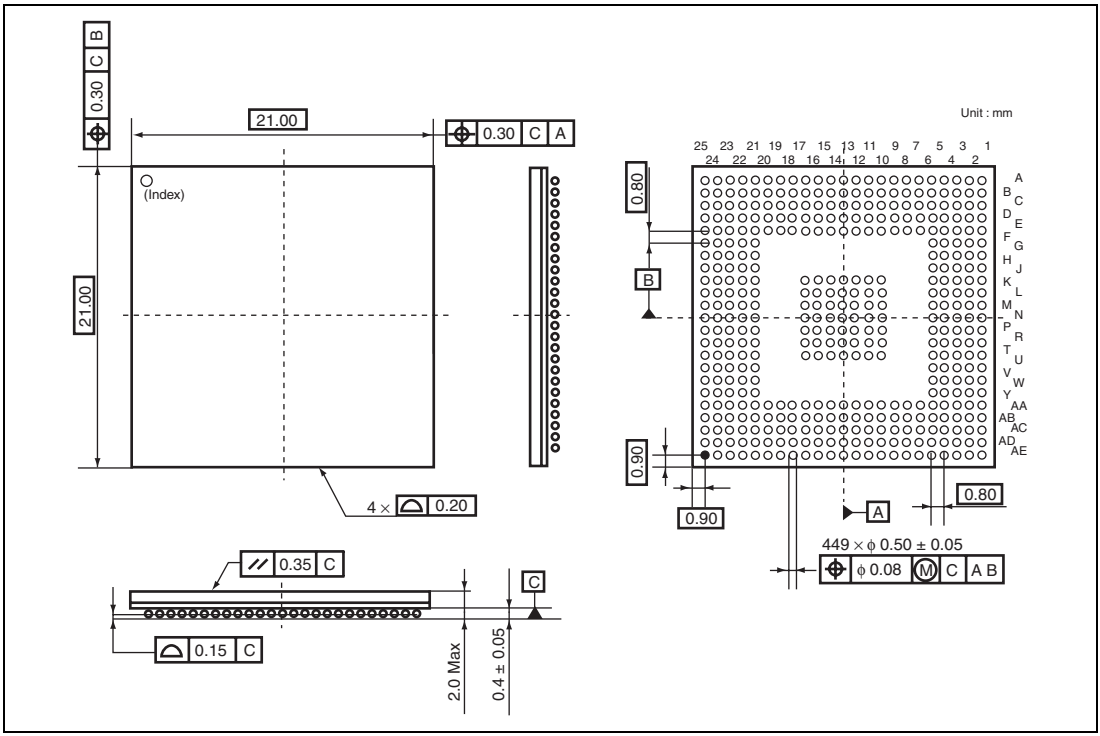


Figure F.1 Package Dimensions (449-Pin)

G. Pin States

Table G.1 Pin States

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
A4	M_CLK0	O	O	O	O	O	O
A5	$\overline{\text{M_CLK1}}$	O	O	O	O	O	O
A6	$\overline{\text{M_WE}}$	O	H	H	H	O	O
A7	$\overline{\text{M_RAS}}$	O	H	H	H	O	O
A8	M_BA0	O	L	L	L	O	O
A9	M_A10	O	L	L	L	O	O
A10	M_A1	O	L	L	L	O	O
A11	M_A3	O	L	L	L	O	O
A12	XTAL2	O	O	O	O	O	O
A13	USBM	IO	I	I	I	I	I
A14	PTI2/ ST0M_STARTI/ IIC0_SCL/ SIOF1_RXD/ $\overline{\text{USB_OVRCRT}}$/ USBF_VBUS	I/I/IO/I/I/I	I	I	I/I/IO/I/I/I	I	I/I/IO/I/I/I
A15	PTI0/STATUS0/ ST1_CLK/ RMII0_MDC	IO/O/IO/O	H	H	P/O/O/O	Z/O/Z/O	P/O/IO/O
A16	PTK4/ST1_D4/ GET0_ERXD4/ SIOF2_TXD/ LCD_D6	IO/IO/I/O/O	M	M	P/IV/I/O/O	Z/Z/Z/Z/O	P/IO/I/O/O
A17	PTI6/IRQ2/ $\overline{\text{IRL2}}$/ ST0M_D6I/ IIC1_SCL	I/I/I/I/IO	I	I	I/I/I/I/IO	I	I/I/I/I/IO
A18	PTJ5/ ST0M_D3I/ ET0_ERXD3/ RMII1_RXD0/ LCD_DON	IO/I/I/I/O	M	M	P/IV/I/I/O	Z/Z/Z/Z/O	P/I/I/I/O

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
A19	PTJ1/ ST0M_CLKIO/ RMII1_RX_ER/ LCK_CLK	IO/IO/I/I	M	M	P/O/I/I	Z/Z/Z/I	P/IO/I/I
A20	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	O/O	H	H	H	Z	ZV
A21	PTM6/D30/ EX_AD30/ ST0_D6/ ET0_RX-CLK/ RMII0_TXD1/ PINT6	IO/IO/IO/ IO/IO/I	Z	Z	P/Z/Z/IV/I/ O/IV	Z	P/Z/Z/IO/I/ O/I
A22	PTM4/D28/ EX_AD28/ ST0_D4/ ET0_PHY-INT/ RMII0_RXD0/ PINT4	IO/IO/IO/ IO/I/I/I	Z	Z	P/Z/Z/IV/I/I/ IV	Z	P/Z/Z/IO/I/ I/I
A23	$\overline{\text{CS0}}$	O	H	H	H	Z	ZV
B3	$\overline{\text{M_BKPRST}}$	I	M	M	M	M	I
B4	M_CKE	O	O	O	O	O	O
B5	M_A13	O	L	L	L	O	O
B6	$\overline{\text{M_CAS}}$	O	H	H	H	O	O
B7	$\overline{\text{M_CS}}$	O	H	H	H	O	O
B8	M_BA1	O	L	L	L	O	O
B9	M_A0	O	L	L	L	O	O
B10	M_A2	O	L	L	L	O	O
B11	M_A4	O	L	L	L	O	O
B12	EXTAL2	I	I	I	I	I	I
B13	USBP	IO	I	I	I	I	I
B14	PTI3/ ST0M_VALIDI/ IIC0_SDA/ SIOF1_MCLK/ USB_CLK	I/I/IO/I/I	I	I	I/I/IO/I/I	I	I/I/IO/I/I

Pin No.	Pin Name	I/O	Power-On Reset MD6 = 0	Power-On Reset MD6 = 1	Manual Reset	Standby	Bus Release
B15	PTK7/ST1_D7/ GET0_ERXD7/ SIOF2_MXLK/ LCD_VCPWC	IO/IO/I/I/O	M	M	P/IV/I/IV/O	Z/Z/Z/Z/O	P/IO/I/I/O
B16	PTI5/MD10/ ST1_VALID/ LCD_D1	IO/I/IO/O	I	I	P/-/I/O	Z/-/Z/O	P/-/IO/O
B17	PTI7/IRQ3/ $\overline{\text{IRL3}}$/ ST0M_D7I/ IIC1_SDA	I/I/I/I/IO	I	I	I/I/I/I/IO	I	I/I/I/I/IO
B18	PTJ4/ ST0M_D2I/ ET0_ERXD2/ RMII1_RXD1/ LCD_CL2	IO/I/I/I/O	M	M	P/IV/I/I/O	Z/Z/Z/Z/O	P/I/I/I/O
B19	$\overline{\text{RDY}}$/$\overline{\text{EX_RDY}}$/ $\overline{\text{PCC_WAIT}}$	I/O/I	M	M	IV/O/IV	IV/O/IV	IV/O/IV
B20	$\overline{\text{CS2}}$/$\overline{\text{EX_CS1}}$	O/I	H	H	H/I	Z	ZV
B21	PTM7/D31/ EX_AD31/ ST0_D7/ ET0_RX-DV/ RMII0_TXD0/ PINT7	IO/IO/IO/ IO/I/O/I	Z	Z	P/Z/Z/IV/I/ O/IV	Z	P/Z/Z/IO/I/ O/I
B22	PTM5/D29/ EX_AD29/ ST0_D5/ ET0/RX-ER/ RMII0_TXD_EN /PINT5	IO/IO/IO/ IO/I/O/I	Z	Z	P/Z/Z/IV/I/ O/IV	Z	P/Z/Z/IO/I/ O/I
B24	PTM3/D27/ EX_AD27/ ST0_D3/ ET0_LINKSTA/ RMII0_RXD1/ PINT3	IO/IO/IO/ IO/I/I/I	Z	Z	P/Z/Z/IV/I/I/ IV	Z	P/Z/Z/IO/I/ I/I

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
B25	REF125CK/ SSI_CLK/ HAC_BITCLK	I/I/I	I	I	I/I/I	I	I/I/I
C1	M_D0	IO	Z	Z	Z	Z	IO
C5	M_A12	O	L	L	L	O	O
C6	M_A11	O	L	L	L	O	O
C7	M_A9	O	L	L	L	O	O
C8	M_A8	O	L	L	L	O	O
C9	M_A7	O	L	L	L	O	O
C10	M_A6	O	L	L	L	O	O
C11	M_A5	O	L	L	L	O	O
C12	$\overline{\text{XRTCSTBI}}$	I	I	I	I	I	I
C14	PTI1/STATUS1/ ST1_REQ/ RMII0_MDIO	IO/O/IO/IO	H	H	P/O/O/I	Z/O/Z/Z	P/O/IO/IO
C15	PTK6/ST1_D6/ GET0_ERXD6/ SIOF2_SCK/ LCD_VEPWC	IO/IO/I/IO/ O	M	M	P/IV/I/O/O	Z/Z/Z/Z/O	P/IO/I/IO/O
C16	PTI4/MD8/ ST1_START/ ET1_PHY-INT/ RMII0M0_MDC/ USB_PWREN/ USBF_UPLUP	IO/I/IO/I/O/ O/O	I	I	P/-/I/I/O/ O/O	Z/-/Z/Z/O/ O/O	P/-/IO/I/O/ O/O
C17	PTJ7/$\overline{\text{INTB}}$/ ST0M_D5I/ $\overline{\text{IRQOUT}}$/ IRMII1_TXD0/ LCD_D0	IO/I/I/O/O/ O	M	M	IO/IV/IV/O/ O/O	Z/Z/Z/Z/O/ O	P/I/I/O/O/O
C18	PTJ3/ ST0M_D1I/ ET0_ERXD1/ RMII1_CRS_ DV/LCD_CL1	IO/I/I/I/O	M	M	P/IV/I/I/O	Z/Z/Z/Z/O	P/I/I/I/O
C19	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O/O	H	H	H	Z	ZV

Pin No.	Pin Name	I/O	Power-On Reset MD6 = 0	Power-On Reset MD6 = 1	Manual Reset	Standby	Bus Release
C20	$\overline{\text{CS1}}$/$\overline{\text{EX_CS0}}$	O/I	H	H	H/I	Z	ZV
C23	$\overline{\text{BS}}$/$\overline{\text{EX_BS}}$	O/I	H	H	H/I	Z	ZV
C24	PTM2/D26/ EX_AD26/ ST0_D2/ ET0_WOL/ RMII0_CRS_ DV/PINT2	IO/IO/IO/ IO/O/I/I	Z	Z	P/Z/Z/IV/O/ I/IV	Z	P/Z/Z/IO/ O/I/I
C25	PTM1/D25/ EX_AD25/ ST0_D1/ ET0_TX-CLK/ RMII0_RX_ER/ PINT1	IO/IO/IO/ IO/I/I/I	Z	Z	P/Z/Z/IV/I/I/ IV	Z	P/Z/Z/IO/I/ I/I
D1	M_D1	IO	Z	Z	Z	Z	IO
D2	M_D16	IO	Z	Z	Z	Z	IO
D15	PTK5/ST1_D5/ GET0_ERXD5/ SIOF2_RXD/ LCD_D7	IO/IO/I/I/O	M	M	P/IV/I/IV/O	Z/Z/Z/Z/O	P/IO/I/I/O
D17	PTJ6/ ST0M_D4I/ ET0_CRS/ RMII1_TXD_EN /LCD_FLM	IO/I/I/O/O	M	M	P/IV/I/O/O	Z/Z/Z/O/O	P/I/I/O/O
D18	PTJ2/ ST0M_D0I/ ET0_ERXD0/ RMII1_TXD1/ LCD_M_DISP	IO/I/I/O/O	M	M	P/IV/I/O/O	Z/Z/Z/O/O	P/I/I/O/O
D19	$\overline{\text{CS4}}$	O	H	H	H	Z	ZV
D23	$\overline{\text{PDWR}}$/ $\overline{\text{EX_RDWR}}$	O/I	H	H	H/I	Z	ZV

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
D24	PTM0/D24/ EX_AD24/ ST0_D0/ ET0_TX-ER/ PINT0/ RMII0M0_MDIO	IO/IO/IO/ IO/O/I/IO	Z	Z	P/Z/Z/IV/O/ IV/IV	Z/Z/Z/Z/O/ Z/Z	P/Z/Z/IO/ O/I/IO
D25	PTL7/D23/ EX_AD23/ ST0_VALID/ ET0_TX-EN/ $\overline{\text{TEND1}}$/ LCD_D15	IO/IO/IO/ IO/O/O/O	Z	Z	P/Z/Z/IV/O/ O/O	Z/Z/Z/Z/O/ O/O	P/Z/Z/IO/ O/O/O
E1	M_D2	IO	Z	Z	Z	Z	IO
E2	M_D17	IO	Z	Z	Z	Z	IO
E3	M_D18	IO	Z	Z	Z	Z	IO
E22	PTK3/ST1_D3/ GET0_ETXD7/ SIOF2_SYNC/ LCD_D5	IO/IO/O/IO/ O	M	M	P/IV/O/O/O	Z/Z/O/Z/O	P/IO/O/IO/ O
E23	PTK2/ST1_D2/ GET0_ETXD6/ SIOF1_SCK/ LCD_D4	IO/IO/O/IO/ O	M	M	P/IV/O/O/O	Z/Z/O/Z/O	P/IO/O/IO/ O
E24	PTL6/D22/ EX_AD22/ ST0_START/ ET0_ETXD2/ $\overline{\text{DACK1}}$/ LCD_D14	IO/IO/IO/ IO/O/O/O	Z	Z	P/Z/Z/IV/O/ O/O	Z/Z/Z/Z/O/ O/O	P/Z/Z/IO/ O/O/O
E25	PTL5/D21/ EX_AD21/ ST0_CLK/ ET0_ETXD1/ $\overline{\text{DREQ1}}$/ LCD_D13	IO/IO/IO/ IO/O/I/O	Z	Z	P/Z/Z/O/O/ IV/O	Z/Z/Z/Z/O/ Z/O	P/Z/Z/IO/ O/I/O
F1	M_D3	IO	Z	Z	Z	Z	IO
F2	M_D19	IO	Z	Z	Z	Z	IO

Pin No.	Pin Name	I/O	Power-On Reset MD6 = 0	Power-On Reset MD6 = 1	Manual Reset	Standby	Bus Release
F3	M_D20	IO	Z	Z	Z	Z	IO
F22	PTK1/ST1_D1/ GET0_ETXD5/ SIOF1_TXD/ LCD_D3	IO/IO/O/O/ O	M	M	P/IV/O/O/O	Z/Z/O/Z/O	P/IO/O/O/ O
F23	PTK0/ST1_D0/ GET0_ETXD4/ SIOF1_SYNC/ LCD_D2	IO/IO/O/I/ O/O	M	M	P/IV/O/O/O	Z/Z/O/Z/O	P/IO/O/IO/ O
F24	PTL4/D20/ EX_AD20/ ST0_REQ/ ET0_ETXD0/ $\overline{\text{INTD}}$/LCD_D12	IO/IO/IO/ IO/O/I/O	Z	Z	P/Z/Z/O/O/ IV/O	Z/Z/Z/Z/O/ IV/O	P/Z/Z/IO/ O/I/O
F25	PTJ0/ ST0M_REQO/ GET0_ GTX-CLK/ REF50CK	IO/O/O/I	M	M	P/O/O/I	Z/Z/O/Z	P/O/O/I
G1	M_D4	IO	Z	Z	Z	Z	IO
G2	M_D21	IO	Z	Z	Z	Z	IO
G3	M_D22	IO	Z	Z	Z	Z	IO
G22	PTL3/D19/ EX_AD19/ IRQ7/$\overline{\text{IRL7}}$/ ET0_MDIO/ $\overline{\text{INTC}}$/LCD_D11	IO/IO/IO/I/ I/IO/I/O	Z	Z	P/Z/Z/IV/ IV/I/IV/O	Z/Z/Z/IV/ IV/Z/IV/O	P/Z/Z/I/I/ IO/I/O
G23	PTL2/D18/ EX_AD18/ IRQ6/$\overline{\text{IRL6}}$/ ET0_ETXD3/ $\overline{\text{TEND0}}$/ LCD_D10	IO/IO/IO/I/ I/O/O/O	Z	Z	P/Z/Z/IV/ IV/O/O/O	Z/Z/Z/IV/ IV/O/O/O	P/Z/Z/I/I/O/ O/O
G24	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	O/O	H	H	H	Z	ZV
G25	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	O/O	H	H	H	Z	ZV
H1	M_D5	IO	Z	Z	Z	Z	IO
H2	M_D23	IO	Z	Z	Z	Z	IO

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
H3	M_DQS2	IO	Z	Z	Z	Z	IO
H22	PTL0/D16/ EX_AD16/ IRQ4/$\overline{\text{IRL4}}$/ ET0_COL/ $\overline{\text{DREQ0}}$/ LCD_D8	IO/IO/IO/I/ I/I/I/O	Z	Z	P/Z/Z/IV/ IV/I/IV/O	Z/Z/Z/IV/ IV/Z/Z/O	P/Z/Z/I/I/I/I/ O
H23	PTL1/D17/ EX_AD17/ IRQ5/$\overline{\text{IRL5}}$/ ET0_MDC/ $\overline{\text{DACK0}}$/ LCD_D9	IO/IO/IO/I/ I/O/O/O	Z	Z	P/Z/Z/IV/ IV/O/O/O	Z/Z/Z/IV/ IV/O/O/O	P/Z/Z/I/I/O/ O/O
H24	D15/EX_AD15	IO/IO	Z	Z	Z	ZV	Z
H25	D14/EX_AD14	IO/IO	Z	Z	Z	ZV	Z
J1	M_D7	IO	Z	Z	Z	Z	IO
J2	M_D6	IO	Z	Z	Z	Z	IO
J3	M_DQM2	O	H	H	H	O	O
J22	D7/EX_AD7	IO/IO	Z	Z	Z	ZV	Z
J23	D6/EX_AD6	IO/IO	Z	Z	Z	ZV	Z
J24	D13/EX_AD13	IO/IO	Z	Z	Z	ZV	Z
J25	D12/EX_AD12	IO/IO	Z	Z	Z	ZV	Z
K1	M_DQM0	O	H	H	H	O	O
K2	M_DQS0	IO	Z	Z	Z	Z	IO
K3	M_DQS3	IO	Z	Z	Z	Z	IO
K22	D5/EX_AD5	IO/IO	Z	Z	Z	ZV	Z
K23	D4/EX_AD4	IO/IO	Z	Z	Z	ZV	Z
K24	D11/EX_AD11	IO/IO	Z	Z	Z	ZV	Z
K25	D10/EX_AD10	IO/IO	Z	Z	Z	ZV	Z
L1	M_DQS1	IO	Z	Z	Z	Z	IO
L2	M_DQM1	O	H	H	H	O	O
L3	M_DQM3	O	H	H	H	O	O
L22	D3/EX_AD3	IO/IO	Z	Z	Z	ZV	Z

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
L23	D2/EX_AD2	IO/IO	Z	Z	Z	ZV	Z
L24	D9/EX_AD9	IO/IO	Z	Z	Z	ZV	Z
L25	D8/EX_AD8	IO/IO	Z	Z	Z	ZV	Z
M1	M_D8	IO	Z	Z	Z	Z	IO
M2	M_D24	IO	Z	Z	Z	Z	IO
M3	M_D25	IO	Z	Z	Z	Z	IO
M22	D1/EX_AD1	IO/IO	Z	Z	Z	ZV	Z
M23	D0/EX_AD0	IO/IO	Z	Z	Z	ZV	Z
M24	$\overline{WE1}/\overline{WE}$	O/O	H	H	H	Z	ZV
M25	CLKOUT	O	O	O	O	O	O
N1	M_D9	IO	Z	Z	Z	Z	IO
N2	M_D26	IO	Z	Z	Z	Z	IO
N3	M_D27	IO	Z	Z	Z	Z	IO
N22	$\overline{RD}/\overline{FRAME}/$ $\overline{EX_FRAME}$	O/O/I	H	H	H/H/I	Z	ZV
N23	$\overline{WE0}/$ $\overline{PCC_REG}$	O/O	H	H	H/O	Z	ZV
N24	A1	O	O(V*[1])	O(V*[1])	O	Z	ZV
N25	A0	O	O(V*[1])	O(V*[1])	O	Z	ZV
P1	M_D10	IO	Z	Z	Z	Z	IO
P2	M_D28	IO	Z	Z	Z	Z	IO
P3	M_D29	IO	Z	Z	Z	Z	IO
P22	A9	O	O(V*[1])	O(V*[1])	O	Z	ZV
P23	A8	O	O(V*[1])	O(V*[1])	O	Z	ZV
P24	A3	O	O(V*[1])	O(V*[1])	O	Z	ZV
P25	A2	O	O(V*[1])	O(V*[1])	O	Z	ZV
R1	M_D11	IO	Z	Z	Z	Z	IO
R2	M_D30	IO	Z	Z	Z	Z	IO
R3	M_D31	IO	Z	Z	Z	Z	IO
R22	A11	O	O(V*[1])	O(V*[1])	O	Z	ZV
R23	A10	O	O(V*[1])	O(V*[1])	O	Z	ZV

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
R24	A5	O	O(V*[1])	O(V*[1])	O	Z	ZV
R25	A4	O	O(V*[1])	O(V*[1])	O	Z	ZV
T1	M_D13	IO	Z	Z	Z	Z	IO
T2	M_D12	IO	Z	Z	Z	Z	IO
T22	A17	O	O(V*[1])	O(V*[1])	O	Z	ZV
T23	A16	O	O(V*[1])	O(V*[1])	O	Z	ZV
T24	A7	O	O(V*[1])	O(V*[1])	O	Z	ZV
T25	A6	O	O(V*[1])	O(V*[1])	O	Z	ZV
U1	M_D15	IO	Z	Z	Z	Z	IO
U2	M_D14	IO	Z	Z	Z	Z	IO
U22	A19	O	O(V*[1])	O(V*[1])	O	Z	ZV
U23	A18	O	O(V*[1])	O(V*[1])	O	Z	ZV
U24	A13	O	O(V*[1])	O(V*[1])	O	Z	ZV
U25	A12	O	O(V*[1])	O(V*[1])	O	Z	ZV
V22	A21	O	O(V*[1])	O(V*[1])	O	Z	ZV
V23	A20	O	O(V*[1])	O(V*[1])	O	Z	ZV
V24	A15	O	O(V*[1])	O(V*[1])	O	Z	ZV
V25	A14	O	O(V*[1])	O(V*[1])	O	Z	ZV
W1	PTG1/$\overline{\text{GNT2}}$/ ET1_ETXD0	IO/O/O	Z	Z	IO/Z/O	Z/Z/O	IO/O/O
W2	PTG2/$\overline{\text{REQ1}}$/ ET1_ETXD1	IO/I/O	I	I	IO/I/O	Z/Z/O	IO/I/O
W3	PTG3/$\overline{\text{REQ3}}$/ ET1_ETXD2	IO/I/O	I	I	IO/I/O	Z/Z/O	IO/I/O
W4	PTF0/$\overline{\text{GNT0}}$/ $\overline{\text{GNTIN}}$/SIM_D/ ET1_ETXD3/ $\overline{\text{DREQ3}}$	IO/IO/I/IO/ O/I	I	I	IO/I(IO*[2])/I/ I/O/I	Z/Z/Z/Z/O/ Z	IO/IO/I/IO/ O/I
W22	A25/EX_SIZE2	O/I	O(M*[1])	O(M*[1])	O/I	Z	ZV
W23	A24/EX_SIZE1	O/I	O(M*[1])	O(M*[1])	O/I	Z	ZV
W24	A23/EX_SIZE0	O/I	O(M*[1])	O(M*[1])	O/I	Z	ZV
W25	A22	O	O(V*[1])	O(V*[1])	O	Z	ZV

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
Y1	PTE1/PCICLK/ GET1_ETXD4/ $\overline{\text{DACK2}}$	IO/I/O/O	I	I	IO/I/O/O	I/I/O/O	IO/I/O/O
Y2	PTD6/$\overline{\text{REQ2}}$/ PCC_BVD1/ GET1_ETXD5/ SSI1_SCK/ LCDM_VCPWC	IO/I/I/O/IO/ O	I	I	IO/I/I/O/I/O	Z/Z/Z/O/Z/ O	IO/I/I/O/IO/ O
Y3	PTE0/$\overline{\text{INTA}}$/ $\overline{\text{PCC_DRV}}$/ GET1_ETXD6/ $\overline{\text{DREQ2}}$	IO/IO/O/O/I	I	I	IO/I/O/O/I	Z/Z/O/O/Z	IO/IO/O/O/I
Y4	PTD7/ $\overline{\text{PCIRESET}}$/ PCC_RESET/ GET1_ETXD7/ LCDM_VEPWC	O/O/O/O/O	O	O	O/O/O/O/O	O	O/O/O/O/O
Y22	$\overline{\text{CE2A}}$	O	V	V	O	ZV	ZV
Y23	$\overline{\text{CE2B}}$	O	V	V	O	ZV	ZV
Y24	DA1	O	O	O	O	O	O
Y25	DA0	O	O	O	O	O	O
AA1	PTF1/$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$/ SIM_CLK/ ET1_MDC/ $\overline{\text{DACK3}}$	IO/IO/O/O/ O/O	I	I	IO/IO(I*[1])/ O/O/O/O	Z/Z/Z/Z/O/ O	IO/IO/O/O/ O/O
AA2	PTF2/AD31/ SIM_RST/ ET1_MDIO/ $\overline{\text{TEND3}}$	IO/IO/O/IO/ O	I	O	IO/I(IO*[2])/ O/I/O	Z/Z/Z/Z/O	IO/IO/O/IO/ O
AA3	PTG0/$\overline{\text{GNT1}}$/ ET1_WOL	IO/O/O	Z	Z	IO/Z/O	Z/Z/O	IO/O/O
AA4	PTG4/AD30/ ET1_LINKSTA	IO/IO/I	I	O	IO/I(IO*[2])/I	Z	IO/IO/I
AA23	$\overline{\text{IOIS16}}$/ TMU_TCLK	I/I	M(V*[1])	M(V*[1])	IV	ZV	IV/I

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AB1	PTE5/AD29/ SCIF2_TXD/ GET1_ GTX-CLK/ SSI0_SCK	IO/IO/O/O/ IO	I	O	IO/I(IO*2)/ O/O/I	Z/Z/Z/O/Z	IO/IO/O/O/ IO
AB2	PTG7/AD28/ ET1_TX-EN	IO/IO/O	I	O	IO/I(IO*2)/ O	Z/Z/O	IO/IO/O
AB3	PTG6/AD26/ ET1_TX-ER	IO/IO/O	I	O	IO/I(IO*2)/ O	Z/Z/O	IO/IO/O
AB6	PTE4/AD22/ SCIF2_RXD/ GET1_ERXD4/ SSI0_SDATA	IO/IO/I/I/IO	I	O	IO/I(IO*2)I/I /I	Z	IO/IO/I/I/IO
AB7	PTD5/AD18/ $\overline{\text{PCC_CD2}}$/ GET1_ERXD6/ SSI1_SDATA/ LCDM_D14	IO/IO/I/I/ IO/O	I	O	IO/I(IO*2)/I/ I/I/O	Z/Z/Z/Z/Z/ O	IO/IO/I/I/ IO/O
AB8	PTD3/ $\overline{\text{PCIFRAME}}$/ PCC_BVD2/ SIOF0_SCK/ $\overline{\text{HAC_RES}}$/ LCDM_D12	IO/IO/I/IO/ O/O	I	I	IO/I/I/O/O/ O	Z/Z/Z/Z/O/ O	IO/IO/I/IO/ O/O
AB9	PTD4/$\overline{\text{STOP}}$/ $\overline{\text{PCC_CD1}}$/ SIOF0_MCLK/ SSI1_WS/ LCDM_DON	IO/IO/I/I/ IO/O	I	I	IO/I/I/I/I/O	Z/Z/Z/Z/Z/ O	IO/IO/I/I/ IO/O
AB10	PTA3/AD15/ $\overline{\text{SCIF1_CTS}}$	IO/IO/IO	I	O	IO/I(IO*2)/I	Z	IO/IO/IO
AB11	PTB2/AD11/ PINT10/ LCDM_D7	IO/IO/I/O	I	O	IO/I(IO*2)/I/ O	Z/Z/Z/O	IO/IO/I/O
AB12	PTB6/CBE0/ PINT14/ LCDM_D3	IO/IO/I/O	I	O	IO/I(IO*2)/I/ O	Z/Z/Z/O	IO/IO/I/O
AB13	PTC1/AD4/ LCDM_D1	IO/IO/O	I	O	IO/I(IO*2)/ O	Z/Z/O	IO/IO/O

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AB16	MPMD	I	M	M	M	M	M
AB17	PTO6/IRQ0/ $\overline{\text{IRL0}}$/$\overline{\text{DACK1M}}$/ MD5	IO/I/I/O/I	I	I	P/I/I/O/-	Z/I/I/O/-	P/I/I/O/-
AB18	PTO2/ AUDATA1/ RMII0M1_MDC	IO/O/O	M	M	P/O/O	Z/O/O	P/O/O
AB20	TDO	O	Z	Z	O	Z	O
AB24	AN3	I	Z	Z	I	I	I
AB25	AN2	I	Z	Z	I	I	I
AC1	PTH6/AD27/ TPU_TO2/ ET1_CRS/ RMII1M_TXD_ EN	IO/IO/O/I/O	I	O	IO/I(IO*[2])/ O/I/O	Z/Z/O/Z/O	IO/IO/O/I/O
AC2	PTH0/AD25/ TPU_TI3A/ ET1_COL/ RMII1M_RX_ ER	IO/IO/I/I/I	I	O	IO/I(IO*[2])/I/ I/I	Z	IO/IO/I/I/I
AC5	PTH1/IDSEL/ TPU_TI3B/ ET1_RX-ER/ RMII1M_CRS_ DV	IO/I/I/I/I	I	I	IO/I/I/I/I	Z	IO/I/I/I/I
AC6	PTE3/AD20/ SCIF2_SCK/ GET1_ERXD5/ SSI0_WS	IO/IO/IO/I/ IO	I	O	IO/I(IO*[2])/I/ I/I	Z	IO/IO/IO/I/ IO
AC7	PTE2/AD16/ $\overline{\text{PCC_IOIS16}}$/ GET1_ERXD7/ $\overline{\text{TEND2}}$	IO/IO/I/I/O	I	O	IO/I(IO*[2])/I/ I/O	Z/Z/Z/Z/O	IO/IO/I/I/O
AC8	PTD2/$\overline{\text{TRDY}}$/ PCC_RDY/ SIOF0_RXD/ HAC_SYNC/ LCDM_D11	IO/IO/I/I/O/ O	I	I	IO/I/I/I/O/O	Z/Z/Z/Z/O/ O	IO/IO/I/I/O/ O

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AC9	PTA0/PAR/ SCIF1_SCK	IO/IO/IO	I	O	IO/I(IO*[2])/I	Z	IO/IO/IO
AC10	PTA4/AD13/ $\overline{\text{SCIF1_RTS}}$	IO/IO/IO	I	O	IO/I(IO*[2])/ O	Z	IO/IO/IO
AC11	PTB3/AD9/ PINT11/ LCDM_D6	IO/IO/I/O	I	O	IO/I(IO*[2])/I/ O	Z/Z/Z/O	IO/IO/I/O
AC12	PTB7/AD6/ PINT15/ LCDM_D2	IO/IO/I/O	I	O	IO/I(IO*[2])/I/ O	Z/Z/Z/O	IO/IO/I/O
AC13	PTC2/AD2/ LCDM_D0	IO/IO/O	I	O	IO/I(IO*[2])/ O	Z/Z/O	IO/IO/O
AC14	PTC5/AD0/ MMC_CD/ LCDM_FLM	IO/IO/I/O	I	O	IO/I(IO*[2])/I/ O	Z/Z/Z/O	IO/IO/I/O
AC15	PTN2/ SCIF0_TXD/ MD1	IO/O/I	I	I	IO/O/-	Z/Z/-	IO/O/-
AC16	$\overline{\text{MRESET}}$	I	I	I	I	I	I
AC17	PTO7/IRQ1/ $\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/ SSI3_SCK/MD6	IO/I/I/O/IO/ I	I	I	P/I/I/O/I/-	Z/I/I/O/Z/-	P/I/I/O/IO/-
AC18	PTO3/ AUDATA2/ RMII0M1_MDIO /SSI2_SCK	IO/O/IO/IO	M	M	P/O/I/IV	Z/O/Z/Z	P/O/IO/IO
AC19	$\overline{\text{TRST}}$	I	M	M	M	M	M
AC20	TDI	I	M	M	M	M	M
AC21	TMS	I	M	M	M	M	M
AC22	$\overline{\text{BACK}}$	O	O	O	O	O	O
AC24	AN1	I	Z	Z	I	I	I
AC25	AN0	I	Z	Z	I	I	I
AD1	PTF3/CBE3/ ET1_TX-CLK	IO/IO/I	I	O	IO/I(IO*[2])/I	Z	IO/IO/I

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AD4	PTH2/AD24/ TPU_TI2A/ ET1_ERXD0/ RMII1M_TXD1	IO/IO/I/I/O	I	O	IO/I(IO*2)/I/ I/O	Z/Z/Z/Z/O	IO/IO/I/I/O
AD5	PTH3/AD21/ TPU_TI2B/ ET1_ERXD2/ RMII1M_RXD1	IO/IO/I/I/I	I	O	IO/I(IO*2)/I/ I/I	Z	IO/IO/I/I/I
AD6	PTH7/AD17/ TPU_TO3/ ET1_RX-DV	IO/IO/O/I	I	O	IO/I(IO*2)/ O/I	Z/Z/O/Z	IO/IO/O/I
AD7	PTD0/$\overline{\text{IRDY}}$/ $\overline{\text{PCC_VS1}}$/ SIOF0_SYNC/ HAC_SD_IN/ LCDM_D13	IO/IO/I/IO/ I/O	I	I	IO/I/I/O/I/O	Z/Z/Z/Z/Z/ O	IO/IO/I/IO/ I/O
AD8	PTA2/$\overline{\text{LOCK}}$/ SCIF1_TXD	IO/IO/O	I	I	IO/I/O	Z	IO/IO/O
AD9	PTB1/$\overline{\text{SERR}}$/ PINT9/ LCDM_D9	IO/IO/I/O	I	I	IO/I/I/O	Z/Z/Z/O	IO/IO/I/O
AD10	PTB5/AD14/ PINT13/ LCDM_M_DISP	IO/IO/I/O	I	O	IO/I(IO*2)/I/ O	Z/Z/Z/O	IO/IO/I/O
AD11	PTC0/AD10/ MMC_DAT/ LCDM_D5	IO/IO/IO/O	I	O	IO/I(IO*2)/ IO/O	Z/Z/Z/O	IO/IO/IO/O
AD12	PTC4/AD7/ MMC_CMD/ LCDM_CL2	IO/IO/IO/O	I	O	IO/I(IO*2)/ IO/O	Z/Z/Z/O	IO/IO/IO/O
AD13	PTC7/AD3/ MMC_CLK	IO/IO/O	I	O	IO/I(IO*2)/ O	Z	IO/IO/O
AD14	PTN0/ SCIF0_SCK/ MD0	IO/IO/I	I	I	IO/I/-	Z/Z/-	IO/IO/-
AD15	PTN3/ $\overline{\text{SCIF0_CTS}}$/ MD4	IO/IO/I	I	I	IO/I/-	Z/Z/-	IO/IO/-

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AD16	PTN5/NMI	IO/I	M	M	P/IV	Z/IV	P/I
AD17	PTO0/ AUDSYNC/ RMII1_MDC/ SSI2_WS	IO/O/O/IO	M	M	P/O/O/IV	Z/O/O/Z	P/O/O/IO
AD18	PTO4/ AUDATA3/ $\overline{\text{EX_INT}}$/ SSI3_WS	IO/O/O/IO	M	M	P/O/O/IV	Z/O/O/Z	P/O/O/IO
AD19	$\overline{\text{ASEBRK}}$/ BRKACK	IO	M	M	M/O	M	M/O
AD22	$\overline{\text{BREQ}}$	I	M	M	IV	IV	IV
AE3	PTG5/$\overline{\text{GNT3}}$/ ET1_RX-CLK	IO/O/I	Z	Z	IO/Z/I	Z	IO/O/I
AE4	PTH5/AD23/ TPU_TO1/ ET1_ERXD1/ RMII1M_TXD0	IO/IO/O/I/O	I	O	IO/I(IO*[2])/ O/I/O	Z/Z/O/Z/O	IO/IO/O/I/O
AE5	PTH4/AD19/ TPU_TO0/ ET1_ERXD3/ RMII1M_RXD0	IO/IO/O/I/I	I	O	IO/I(IO*[2])/ O/I/I	Z/Z/O/Z/Z	IO/IO/O/I/I
AE6	PTD1/CBE2/ $\overline{\text{PCC_VS2}}$/ SIOF0_TXD/ HAC_SD_OUT/ LCDM_D15	IO/IO/I/O/ O/O	I	O	IO/I(IO*[2])/I/ O/O/O	Z/Z/Z/Z/O/ O	IO/IO/I/O/ O/O
AE7	PTA1/$\overline{\text{DEVSEL}}$/ SCIF1_RXD	IO/IO/I	I	I	IO/I/I	Z	IO/IO/I
AE8	PTB0/$\overline{\text{PERR}}$/ PINT8/ LCDM_D10	IO/IO/I/O	I	I	IO/I/I/O	Z/Z/Z/O	IO/IO/I/O
AE9	PTB4/CBE1/ PINT12/ LCDM_D8	IO/IO/I/O	I	O	IO/I(IO*[2])/I/ O	Z/Z/Z/O	IO/IO/I/O
AE10	PTA5/AD12	IO/IO	I	O	IO/I(IO*[2])	Z	IO/IO

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AE11	PTC3/AD8/ MMC_ODMOD/ LCDM_D4	IO/IO/O/O	I	O	IO/I(IO*2)/ O/O	Z/Z/Z/O	IO/IO/O/O
AE12	PTC6/AD5/ LCDM_CL1	IO/IO/O	I	O	IO/I(IO*2)/ O	Z/Z/Z/O	IO/IO/O
AE13	PTA6/AD1/ MMC_VDDON	IO/IO/O	I	O	IO/I(IO*2)/ O	Z	IO/IO/O
AE14	PTN1/ SCIF0_RXD/ MD3	IO/I/I	I	I	IO/Z/-	Z/Z/-	IO/I/-
AE15	PTN4/ SCIF0_RTS/ MD2	IO/IO/I	I	I	IO/O/-	Z/Z/-	IO/IO/-
AE16	PRESET	I	I	I	I	I	I
AE17	PTO1/ AUDATA0/ RMII1_MDIO/ SSI2_SDATA	IO/O/IO/IO	M	M	P/O/I/IV	Z/O/Z/Z	P/O/IO/IO
AE18	PTO5/AUDCK/ DREQ1M/ SSI3_SDATA	IO/O/I/IO	M	M	P/O/IV/IV	Z/O/Z/Z	P/O/I/IO
AE19	TCK	I	M	M	M	M	M
AE22	EXTAL	I	I	I	I	I	I

Pin No.	Pin Name	I/O	Power-On Reset		Manual Reset	Standby	Bus Release
			MD6 = 0	MD6 = 1			
AE23	XTAL	O	O	O	O	O	O

[Legend]

I: Input

O: Output

IO: Input or output

H: High level output (input buffer: off, output buffer: on)

V: Input buffer: off, output buffer: off, pulled-up

M: Input buffer: on, output buffer: off, pulled-up

IV: Input buffer: on, output buffer: off, pulled-up or not according to the GPIO register settings

Z: High impedance state (input buffer: off, output buffer: off)

ZV: High impedance or pulled up according to the LBSC, GPIO register settings

P: Input buffer on or off, output buffer on or off, pulled-up or not according to register settings

-: Invalid

Notes: 1. Pin states until the internal status is determined.

2. When MD6 = 1.

H. Handling of Unused Pins

Table H.1 Handling of Unused Pins

Pin No.	Pin Name	I/O	Handling of Unused Pins
A4	M_CLK0	O	Open
A5	$\overline{\text{M_CLK1}}$	O	Open
A6	$\overline{\text{M_WE}}$	O	Open
A7	$\overline{\text{M_RAS}}$	O	Open
A8	M_BA0	O	Open
A9	M_A10	O	Open
A10	M_A1	O	Open
A11	M_A3	O	Open
A12	XTAL2	O	Always used
A13	USBM	IO	Pulled-down
A14	PTI2/ST0M_$\overline{\text{STARTI}}$/IIC0_SCL/SIOF1_RXD/ $\overline{\text{USB_OVRCRT}}$/USBF_VBUS	I/I/IO/I/I/I	Pulled-up
A15	PTI0_STATUS0/ST1_CLK/RMII0_MDC	IO/O/IO/O	Open
A16	PTK4/ST1_D4/SIOF2_TXD/LCD_D6	IO/IO/O/O	Open
	GET0_ERXD4	I	Pulled-up
A17	PTI6/IRQ2/$\overline{\text{IRL2}}$/ST0M_D6I/IIC1_SCL	I/I/I/I/IO	Pulled-up
A18	PTJ5/LCD_DON	IO/O	Open
	ST0M_D3I/ET0_ERXD3/RMII1_RXD0	I/I/I	Pulled-up
A19	PTJ1/ST0M_CLKIO	IO/IO	Open
	RMII1_RX_ER/LCK_CLK	I/I	Pulled-up
A20	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	O/O	Open
A21	PTM6/D30/EX_AD30/ST0_D6/RMII0_TXD1	IO/IO/IO/IO /O	Open
	ET0_RX-CLK/PINT6	I/I	Pulled-up
A22	PTM4/D28/EX_AD28/ST0_D4	IO/IO/IO/IO	Open
	ET0_PHY-INT/RMII0_RXD0/PINT4	I/I/I	Pulled-up
A23	$\overline{\text{CS0}}$	O	Open
B3	$\overline{\text{M_BKPRST}}$	I	Pulled-up
B4	M_CKE	O	Open

Pin No.	Pin Name	I/O	Handling of Unused Pins
B5	M_A13	O	Open
B6	$\overline{\text{M_CAS}}$	O	Open
B7	$\overline{\text{M_CS}}$	O	Open
B8	M_BA1	O	Open
B9	M_A0	O	Open
B10	M_A2	O	Open
B11	M_A4	O	Open
B12	EXTAL2	I	Always used
B13	USBP	IO	Pulled-up
B14	PTI3/ST0M_VALIDI/IIC0_SDA/SIOF1_MCLK /USB_CLK	I/I/IO/I/I	Pulled-up
B15	PTK7/ST1_D7/LCD_VCPWC	IO/IO/O	Open
	GET0_ERXD7/SIOF2_MCLK	I/I	Pulled-up
B16	PTI5/MD10/ST1_VALID/LCD_D1	IO/I/IO/O	Always used
B17	PTI7/IRQ3/$\overline{\text{IRL3}}$/ST0M_D7I/IIC1_SDA	I/I/I/I/IO	Pulled-up
B18	PTJ4/LCD_CL2	IO/O	Open
	ST0M_D2I/ET0_ERXD2/RMII1_RXD1	I/I/I	Pulled-up
B19	$\overline{\text{RDY}}$	I	Pulled-down
	$\overline{\text{EX_RDY}}$	O	Open
	$\overline{\text{PCC_WAIT}}$	I	Pulled-up
B20	$\overline{\text{CS2}}$	O	Open
	$\overline{\text{EX_CS1}}$	I	Pulled-up
B21	PTM7/D31/EX_AD31/ST0_D7/RMII0_TXD0	IO/IO/IO/IO /O	Open
	ET0_RX-DV/PINT7	I/I	Pulled-up
B22	PTM5/D29/EX_AD29/ST0_D5/ RMII0_TXD_EN	IO/IO/IO/IO /O	Open
	ET0_RX-ER/PINT5	I/I	Pulled-up
B24	PTM3/D27/EX_AD27/ST0_D3	IO/IO/IO/IO	Open
	ET0_LINKSTA/RMII0_RXD1/PINT3	I/I/I	Pulled-up
B25	REF125CK/SSI_CLK/HAC_BITCLK	I/I/I	Pulled-up
C1	M_D0	IO	Open

Pin No.	Pin Name	I/O	Handling of Unused Pins
C5	M_A12	O	Open
C6	M_A11	O	Open
C7	M_A9	O	Open
C8	M_A8	O	Open
C9	M_A7	O	Open
C10	M_A6	O	Open
C11	M_A5	O	Open
C12	$\overline{\text{XRTCSTBI}}$	I	Pulled-up
C14	PTI1/STATUS1/ST1_REQ/RMII0_MDIO	IO/O/IO/IO	Open
C15	PTK6/ST1_D6/SIOF2_SCK/LCD_VEPWC	IO/IO/IO/O	Open
	GET0_ERXD6	I	Pulled-up
C16	PTI4/MD8/ST1_START/ET1_PHY-INT/ RMII0M0_MDC/USB_PWREN/ USBF_UPLUP	IO/I/IO/I/O/ O/O	Always used
C17	PTJ7/$\overline{\text{IRQOUT}}$/IRMII1_TXD0/LCD_D0	IO/O/O/O	Open
	$\overline{\text{INTB}}$/ST0M_D5I	I/I	Pulled-up
C18	PTJ3/LCD_CL1	IO/O	Open
	ST0M_D1I/ET0_ERXD1/RMII1_CRS_DV	I/I/I	Pulled-up
C19	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O/O	Open
C20	$\overline{\text{CS1}}$	O	Open
	$\overline{\text{EX_CS0}}$	I	Pulled-up
C23	$\overline{\text{BS}}$	O	Open
	$\overline{\text{EX_BS}}$	I	Pulled-up
C24	PTM2/D26/EX_AD26/ST0_D2/ET0_WOL	IO/IO/IO/IO /O	Open
	RMII0_CRS_DV/PINT2	I/I	Pulled-up
C25	PTM1/D25/EX_AD25/ST0_D1/	IO/IO/IO/IO	Open
	ET0_TX-CLK/RMII0_RX_ER/PINT1	I/I/I	Pulled-up
D1	M_D1	IO	Open
D2	M_D16	IO	Open
D15	PTK5/ST1_D5/LCD_D7	IO/IO/O	Open
	GET0_ERXD5/SIOF2_RXD	I/I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
D17	PTJ6/RMII1_TXD_EN/LCD_FLM	IO/O/O	Open
	ST0M_D4I/ET0_CRS	I/I	Pulled-up
D18	PTJ2/RMII1_TXD1/LCD_M_DISP	IO/O/O	Open
	ST0M_D0I/ET0_ERXD0	I/I	Pulled-up
D19	$\overline{CS4}$	O	Open
D23	$\overline{PDWR}$	O	Open
	$\overline{EX_RDWR}$	I	Pulled-up
D24	PTM0/D24/EX_AD24/ST0_D0/ET0_TX-ER/ RMII0M0_MDIO	IO/IO/IO/IO /O/IO	Open
	PINT0	I	Pulled-up
D25	PTL7/D23/EX_AD23/ST0_VALID/ ET0_TX-EN/$\overline{TEND1}$/LCD_D15	IO/IO/IO/IO /O/O/O	Open
E1	M_D2	IO	Open
E2	M_D17	IO	Open
E3	M_D18	IO	Open
E22	PTK3/ST1_D3/GET0_ETXD7/SIOF2_SYNC/ LCD_D5	IO/IO/O/IO/ O	Open
E23	PTK2/ST1_D2/GET0_ETXD6/SIOF1_SCK/ LCD_D4	IO/IO/O/IO/ O	Open
E24	PTL6/D22/EX_AD22/ST0_START/ ET0_ETXD2/$\overline{DACK1}$/LCD_D14	IO/IO/IO/IO /O/O/O	Open
E25	PTL5/D21/EX_AD21/ST0_CLK/ET0_ETXD1/ LCD_D13	IO/IO/IO/IO /O/O	Open
	$\overline{DREQ1}$	I	Pulled-up
F1	M_D3	IO	Open
F2	M_D19	IO	Open
F3	M_D20	IO	Open
F22	PTK1/ST1_D1/GET0_ETXD5/SIOF1_TXD/ LCD_D3	IO/IO/O/O/ O	Open
F23	PTK0/ST1_D0/GET0_ETXD4/SIOF1_SYNC/ LCD_D2	IO/IO/O/IO/ O	Open
F24	PTL4/D20/EX_AD20/ST0_REQ/ET0_ETXD0 /LCD_D12	IO/IO/IO/IO /O/O	Open
	$\overline{INTD}$	I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
F25	PTJ0/ST0M_REQO/GET0_GTX-CLK	IO/O/O	Open
	REF50CK	I	Pulled-up
G1	M_D4	IO	Open
G2	M_D21	IO	Open
G3	M_D22	IO	Open
G22	PTL3/D19/EX_AD19/ET0_MDIO/LCD_D11	IO/IO/IO/IO/O	Open
	IRQ7/$\overline{\text{IRL7}}$/$\overline{\text{INTC}}$	I/I/I	Pulled-up
G23	PTL2/D18/EX_AD18/ET0_ETXD3/$\overline{\text{TEND0}}$/LCD_D10	IO/IO/IO/O/O/O	Open
	IRQ6/$\overline{\text{IRL6}}$	I/I	Pulled-up
G24	$\overline{\text{WE3}}$/$\overline{\text{IOWR}}$	O/O	Open
G25	$\overline{\text{WE2}}$/$\overline{\text{IORD}}$	O/O	Open
H1	M_D5	IO	Open
H2	M_D23	IO	Open
H3	M_DQS2	IO	Open
H22	PTL0/D16/EX_AD16/LCD_D8	IO/IO/IO/O	Open
	IRQ4/$\overline{\text{IRL4}}$/ET0_COL/$\overline{\text{DREQ0}}$	I/I/I/I	Pulled-up
H23	PTL1/D17/EX_AD17/ET0_MDC/$\overline{\text{DACK0}}$/LCD_D9	IO/IO/IO/O/O/O	Open
	IRQ5/$\overline{\text{IRL5}}$	I/I	Pulled-up
H24	D15/EX_AD15	IO/IO	Open
H25	D14/EX_AD14	IO/IO	Open
J1	M_D7	IO	Open
J2	M_D6	IO	Open
J3	M_DQM2	O	Open
J22	D7/EX_AD7	IO/IO	Open
J23	D6/EX_AD6	IO/IO	Open
J24	D13/EX_AD13	IO/IO	Open
J25	D12/EX_AD12	IO/IO	Open
K1	M_DQM0	O	Open
K2	M_DQS0	IO	Open

Pin No.	Pin Name	I/O	Handling of Unused Pins
K3	M_DQS3	IO	Open
K22	D5/EX_AD5	IO/IO	Open
K23	D4/EX_AD4	IO/IO	Open
K24	D11/EX_AD11	IO/IO	Open
K25	D10/EX_AD10	IO/IO	Open
L1	M_DQS1	IO	Open
L2	M_DQM1	O	Open
L3	M_DQM3	O	Open
L22	D3/EX_AD3	IO/IO	Open
L23	D2/EX_AD2	IO/IO	Open
L24	D9/EX_AD9	IO/IO	Open
L25	D8/EX_AD8	IO/IO	Open
M1	M_D8	IO	Open
M2	M_D24	IO	Open
M3	M_D25	IO	Open
M22	D1/EX_AD1	IO/IO	Open
M23	D0/EX_AD0	IO/IO	Open
M24	$\overline{\text{WE1}}/\overline{\text{WE}}$	O/O	Open
M25	CLKOUT	O	Open
N1	M_D9	IO	Open
N2	M_D26	IO	Open
N3	M_D27	IO	Open
N22	$\overline{\text{RD}}/\overline{\text{FRAME}}$	O/O	Open
	$\overline{\text{EX_FRAME}}$	I	Pulled-up
N23	$\overline{\text{WE0}}/\overline{\text{PCC_REG}}$	O/O	Open
N24	A1	O	Open
N25	A0	O	Open
P1	M_D10	IO	Open
P2	M_D28	IO	Open
P3	M_D29	IO	Open
P22	A9	O	Open

Pin No.	Pin Name	I/O	Handling of Unused Pins
P23	A8	O	Open
P24	A3	O	Open
P25	A2	O	Open
R1	M_D11	IO	Open
R2	M_D30	IO	Open
R3	M_D31	IO	Open
R22	A11	O	Open
R23	A10	O	Open
R24	A5	O	Open
R25	A4	O	Open
T1	M_D13	IO	Open
T2	M_D12	IO	Open
T22	A17	O	Open
T23	A16	O	Open
T24	A7	O	Open
T25	A6	O	Open
U1	M_D15	IO	Open
U2	M_D14	IO	Open
U22	A19	O	Open
U23	A18	O	Open
U24	A13	O	Open
U25	A12	O	Open
V22	A21	O	Open
V23	A20	O	Open
V24	A15	O	Open
V25	A14	O	Open
W1	PTG1/$\overline{\text{GNT2}}$/ET1_ETXD0	IO/O/O	Open
W2	PTG2/ET1_ETXD1	IO/O	Open
	$\overline{\text{REQ1}}$	I	Pulled-up
W3	PTG3/ET1_ETXD2	IO/O	Open
	$\overline{\text{REQ3}}$	I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
W4	PTF0/SIM_D/ET1_ETXD3	IO/IO/O	Open
	$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$/$\overline{\text{DREQ3}}$	IO/I/I	Pulled-up
W22	A25	O	Open
	EX_SIZE2	I	Pulled-up
W23	A24	O	Open
	EX_SIZE1	I	Pulled-up
W24	A23	O	Open
	EX_SIZE0	I	Pulled-up
W25	A22	O	Open
Y1	PTE1/GET1_ETXD4/$\overline{\text{DACK2}}$	IO/O/O	Open
	PCICLK	I	Pulled-up
Y2	PTD6/GET1_ETXD5/SSI1_SCK/ LCDM_VCPWC	IO/O/IO/O	Open
	$\overline{\text{REQ2}}$/PCC_BVD1	I/I	Pulled-up
Y3	PTE0/$\overline{\text{PCC_DRV}}$/GET1_ETXD6	IO/O/O	Open
	$\overline{\text{INTA}}$/$\overline{\text{DREQ2}}$	IO/I	Pulled-up
Y4	PTD7/$\overline{\text{PCIRESET}}$/PCC_RESET/ GET1_ETXD7/LCDM_VEPWC	O/O/O/O/O	Open
Y22	$\overline{\text{CE2A}}$	O	Open
Y23	$\overline{\text{CE2B}}$	O	Open
Y24	DA1	O	Open
Y25	DA0	O	Open
AA1	PTF1/$\overline{\text{REQOUT}}$/SIM_CLK/ET1_MDC/$\overline{\text{DACK3}}$	IO/O/O/O/ O	Open
	$\overline{\text{REQ0}}$	IO	Pulled-up
AA2	PTF2/AD31/SIM_RST/ET1_MDIO/$\overline{\text{TEND3}}$	IO/IO/O/IO/ O	Open
AA3	PTG0/$\overline{\text{GNT1}}$/ET1_WOL	IO/O/O	Open
AA4	PTG4/AD30	IO/IO	Open
	ET1_LINKSTA	I	Pulled-up
AA23	$\overline{\text{IOIS16}}$/TMU_TCLK	I/I	Pulled-up
AB1	PTE5/AD29/SCIF2_TXD/GET1_GTX-CLK/ SSI0_SCK	IO/IO/O/O/ IO	Open

Pin No.	Pin Name	I/O	Handling of Unused Pins
AB2	PTG7/AD28/ET1_TX-EN	IO/IO/O	Open
AB3	PTG6/AD26/ET1_TX-ER	IO/IO/O	Open
AB6	PTE4/AD22/SSI0_SDATA	IO/IO/IO	Open
	SCIF2_RXD/GET1_ERXD4	I/I	Pulled-up
AB7	PTD5/AD18/GET1_ERXD6/SSI1_SDATA/ LCDM_D14	IO/IO/I/IO/ O	Open
	$\overline{\text{PCC_CD2}}$	I	Pulled-up
AB8	PTD3/SIOF0_SCK/$\overline{\text{HAC_RES}}$/LCDM_D12	IO/IO/O/O	Open
	$\overline{\text{PCIFRAME}}$/PCC_BVD2	IO/I	Pulled-up
AB9	PTD4/SSI1_WS/LCDM_DON	IO/IO/O	Open
	$\overline{\text{STOP}}$/$\overline{\text{PCC_CD1}}$/SIOF0_MCLK	IO/I/I	Pulled-up
AB10	PTA3/AD15/$\overline{\text{SCIF1_CTS}}$	IO/IO/IO	Open
AB11	PTB2/AD11/LCDM_D7	IO/IO/O	Open
	PINT10	I	Pulled-up
AB12	PTB6/CBE0/LCDM_D3	IO/IO/O	Open
	PINT14	I	Pulled-up
AB13	PTC1/AD4/LCDM_D1	IO/IO/O	Open
AB16	MPMD	I	Always used
AB17	PTO6/IRQ0/$\overline{\text{IRL0}}$/$\overline{\text{DACK1M}}$/MD5	IO/I/I/O/I	Always used
AB18	PTO2/AUDATA1/RMII0M1_MDC	IO/O/O	Open
AB20	TDO	O	Open
AB24	AN3	I	Open
AB25	AN2	I	Open
AC1	PTH6/AD27/TPU_TO2/RMII1M_TXD_EN	IO/IO/O/O	Open
	ET1_CRS	I	Pulled-up
AC2	PTH0/AD25	IO/IO	Open
	TPU_TI3A/ET1_COL/RMII1M_RX_ER	I/I/I	Pulled-up
AC5	PTH1	IO	Open
	IDSEL	I	Pulled-down
	TPU_TI3B/ET1_RX-ER/RMII1M_CRS_DV	I/I/I	Pulled-up
AC6	PTE3/AD20/SCIF2_SCK/SSI0_WS	IO/IO/IO/IO	Open
	GET1_ERXD5	I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
AC7	PTE2/AD16/$\overline{\text{TEND2}}$	IO/IO/O	Open
	$\overline{\text{PCC-IOIS16}}$/GET1_ERXD7	I/I	Pulled-up
AC8	PTD2/HAC_SYNC/LCDM_D11	IO/O/O	Open
	$\overline{\text{TRDY}}$/PCC_RDY/SIOF0_RXD	IO/I/I	Pulled-up
AC9	PTA0/PAR/SCIF1_SCK	IO/IO/IO	Open
AC10	PTA4/AD13/$\overline{\text{SCIF1_RTS}}$	IO/IO/IO	Open
AC11	PTB3/AD9/LCDM_D6	IO/IO/O	Open
	PINT11	I	Pulled-up
AC12	PTB7/AD6/LCDM_D2	IO/IO/O	Open
	PINT15	I	Pulled-up
AC13	PTC2/AD2/LCDM_D0	IO/IO/O	Open
AC14	PTC5/AD0/LCDM_FLM	IO/IO/O	Open
	MMC_CD	I	Pulled-up
AC15	PTN2/SCIF0_TXD/MD1	IO/O/I	Always used
AC16	$\overline{\text{MRESET}}$	I	Always used
AC17	PTO7/IRQ1/$\overline{\text{IRL1}}$/$\overline{\text{TEND1M}}$/SSI3_SCK/MD6	IO/I/I/O/IO/I	Always used
AC18	PTO3/AUDATA2/RMII0M1_MDIO/SSI2_SCK	IO/O/IO/IO	Open
AC19	$\overline{\text{TRST}}$	I	Always used
AC20	TDI	I	Open
AC21	TMS	I	Open
AC22	$\overline{\text{BACK}}$	O	Open
AC24	AN1	I	Open
AC25	AN0	I	Open
AD1	PTF3/CBE3	IO/IO	Open
	ET1_TX-CLK	I	Pulled-up
AD4	PTH2/AD24/RMII1M_TXD1	IO/IO/O	Open
	TPU_TI2A/ET1_ERXD0	I/I	Pulled-up
AD5	PTH3/AD21	IO/IO	Open
	TPU_TI2B/ET1_ERXD2/RMII1M_RXD1	I/I/I	Pulled-up
AD6	PTH7/AD17/TPU_TO3	IO/IO/O	Open
	ET1_RX-DV	I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
AD7	PTD0/SIOF0_SYNC/LCDM_D13	IO/IO/O	Open
	$\overline{\text{IRDY}}$/$\overline{\text{PCC_VS1}}$/HAC_SD_IN	IO/I/I	Pulled-up
AD8	PTA2/SCIF1_TXD	IO/O	Open
	$\overline{\text{LOCK}}$	IO	Pulled-up
AD9	PTB1/LCDM_D9	IO/O	Open
	$\overline{\text{SERR}}$/PINT9	IO/I	Pulled-up
AD10	PTB5/AD14/LCDM_M_DISP	IO/IO/O	Open
	PINT13	I	Pulled-up
AD11	PTC0/AD10/MMC_DAT/LCDM_D5	IO/IO/IO/O	Open
AD12	PTC4/AD7/MMC_CMD/LCDM_CL2	IO/IO/IO/O	Open
AD13	PTC7/AD3/MMC_CLK	IO/IO/O	Open
AD14	PTN0/SCIF0_SCK/MD0	IO/IO/I	Always used
AD15	PTN3/$\overline{\text{SCIF0_CTS}}$/MD4	IO/IO/I	Always used
AD16	PTN5	IO	Open
	NMI	I	Pulled-up
AD17	PTO0/AUDSYNC/RMII1_MDC/SSI2_WS	IO/O/O/IO	Open
AD18	PTO4/AUDATA3/$\overline{\text{EX_INT}}$/SSI3_WS	IO/O/O/IO	Open
AD19	$\overline{\text{ASEBRK}}$/BRKACK	IO	Open
AD22	$\overline{\text{BREQ}}$	I	Pulled-up
AE3	PTG5/$\overline{\text{GNT3}}$	IO/O	Open
	ET1_RX-CLK	I	Pulled-up
AE4	PTH5/AD23/TPU_TO1/RMII1M_TXD0	IO/IO/O/O	Open
	ET1_ERXD1	I	Pulled-up
AE5	PTH4/AD19/TPU_TO0	IO/IO/O	Open
	ET1_ERXD3/RMII1M_RXD0	I/I	Pulled-up
AE6	PTD1/CBE2/SIOF0_TXD/HAC_SD_OUT/ LCDM_D15	IO/IO/O/O/ O	Open
	$\overline{\text{PCC_VS2}}$	I	Pulled-up
AE7	PTA1	IO	Open
	$\overline{\text{DEVSEL}}$/SCIF1_RXD	IO/I	Pulled-up
AE8	PTB0/LCDM_D10	IO/O	Open
	$\overline{\text{PERR}}$/PINT8	IO/I	Pulled-up

Pin No.	Pin Name	I/O	Handling of Unused Pins
AE9	PTB4/CBE1/LCDM_D8	IO/IO/O	Open
	PINT12	I	Pulled-up
AE10	PTA5/AD12	IO/IO	Pulled-up
AE11	PTC3/AD8/$\overline{\text{MMC_ODMOD}}$/LCDM_D4	IO/IO/O/O	Pulled-up
AE12	PTC6/AD5/LCDM_CL1	IO/IO/O	Open
AE13	PTA6/AD1/MMC_VDDON	IO/IO/O	Open
AE14	PTN1/SCIF0_RXD/MD3	IO/I/I	Always used
AE15	PTN4/$\overline{\text{SCIF0_RTS}}$/MD2	IO/IO/I	Always used
AE16	$\overline{\text{PRESET}}$	I	Always used
AE17	PTO1/AUDATA0/RMII1_MDIO/SSI2_SDATA	IO/O/IO/IO	Open
AE18	PTO5/AUDCK/SSI3_SDATA	IO/O/IO	Open
	$\overline{\text{DREQ1M}}$	I	Pulled-up
AE19	TCK	I	Open
AE22	EXTAL	I	Always used
AE23	XTAL	O	Always use

I. Version Registers

The registers related to the version registers are shown below.

Table I.1 Register Configuration

Name	Abbrev.	R/W	Initial Value	P4 Address	Area 7 Address	Size
Processor version register	PVR	R	H'1020 0Axx	H'FF00 0030	H'1F00 0030	32
Product register	PRR	R	H'0000 092x	H'FF00 0044	H'1F00 0044	32

[Legend]

x: Undefined

(1) Processor Version Register (PVR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Version information															
Initial value:	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Version information								—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	0	1	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	—	—	—	—	—	—	—	—

(2) Product Register (PRR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Version information															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Version information												—	—	—	—
Initial value:	0	0	0	0	1	0	0	1	0	0	1	0	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	—	—	—	—

J. Heat Radiation

This LSI is assumed to be used under the condition of Tj (junction temperature) ≤ 125°C. Accordingly, if Tj can possibly exceed 125°C, some heat radiation measures should be taken. For reference, the following sections show the heat resistance simulation results of this LSI; determine the measures to be taken so that a condition of Tj ≤ 125°C should be satisfied. Here, note that the following sections only show the simulation results, not the guaranteed values or measures.

J.1 Heat Resistance Simulation Conditions

(1) Circuit Board Model on which this LSI is Mounted: JEDEC Standard Board (PCB)

- Structure: Four-layer board of 101.5 mm × 114.5 mm × t1.6 mm (2S2P)
- Routing rate: 20% - 100% - 100% - 20%

(2) Heat Resistance Simulation Environment

- Heat resistance simulation environment: JEDEC standard environment (300-mm cubic casing); Ta (ambient temperature) = 75°C or 60°C; windless.
- Casing: Acrylic (300 mm × 300 mm × 300 mm in size)

Figure J.1 shows the overall view of the simulation model.

(3) Power Consumption

- Power consumption: 2.4 W (maximum value when this LSI is operating in normal operating mode)

(4) Heat Sink Model

- Heat sink: Aluminum
- Thermal conductivity: 215.9 W/mK

Figure J.2 shows the heat sink model.

J.2 Analysis Results of Heat Resistance Simulation

Table J.1 shows the analysis results of heat resistance simulation. As shown, Tj exceeds 125°C when TA = 60°C and a heat sink is not provided. Therefore, if Ta possibly exceeds 60°C, heat radiation measures are indispensable.

Table J.1 Heat Resistance Simulation Results

Power Consumption (W)	Heat Sink	Ta (°C)	Tj (°C)	θja (°C/W)
2.4	Not provided	75	136.9	25.8
2.4	Provided	75	116.9	17.4
2.4	Not provided	60	122.9	26.2

Note: Tj and θja vary depending on the PCB, casing, heat source, and other environmental factors.

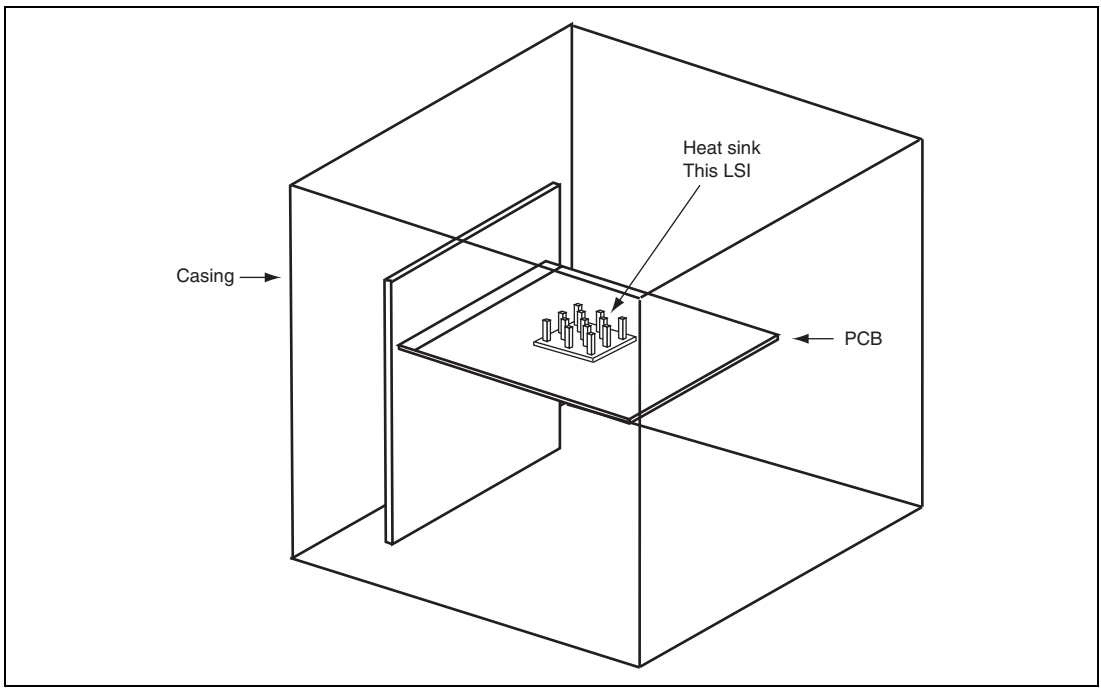
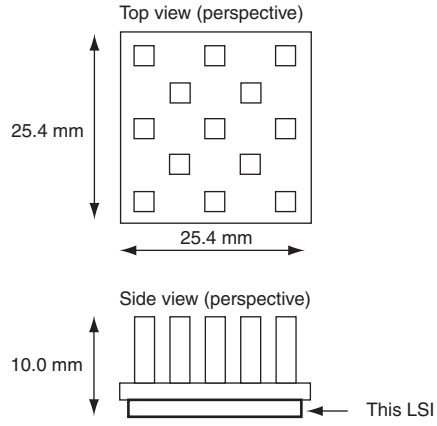


Figure J.1 Overall View of Simulation Model (with heat sink)



Note: A heat sink is attached to this LSI using adhesive (0.125 mm thick and 0.6 W/mK).

Figure J.2 Heat Sink Model

Index

Numerics

0-Time PAUSE frame control 981
10-Bit address format 1048
16-Bit timer pulse unit (TPU) 707
32-Bit address extended mode 177
7-Bit address format 1047

A

A/D converter 1655
AC 97 Frame Slot Structure 1409
Address space identifier (ASID) 145
Address translation 145
Addressing modes 59
Alarm function 780
Arithmetic operation instructions 68
ASID 156
Asynchronous Mode 1151
Asynchronous serial communication mode 1059
ATI 781
Audio Codec Interface (HAC) 1391
Automatic PAUSE frame transmission 980
Auto-Reload Count Operation 701
Auto-Request mode 591

B

Baud rate generator 1217
Baud Rate Generator for External Clock (BRG) 1181
Big endian 52
Block diagram 13
Branch instructions 72
Break 1115, 1176
BRI 1114, 1175
Buffer operation 734
Burst mode 602
Bus Arbitration 405

C

Cacheability bit 158
Caches 187
CAM function 969
Clock pulse generator (CPG) 633
Clock valid reception 1017
Clock valid transmission 1021
Clocked Synchronous Mode 1152
Clocked synchronous serial communication mode 1059
Compare match timer (CMT) 747
Control registers 39
Crystal Oscillator Circuit 781
CUI 781
Cycle-steal mode 600

D

D/A converter (DAC) 1673
Data address error 123
Data alignment 412
Data TLB miss exception 118, 169
Data TLB multiple hit exception 169
Data TLB multiple-hit exception 117
Data TLB protection violation exception 121, 170
DDR-SDRAM commands 432
DDR-SDRAM interface (DDRIF) 409
Delay slot 57
Delayed branches 57
Descriptor and Transmit/Receive Buffer 952

Descriptor Pointer 954
Direct memory access controller (DMAC) 565
Dirty bit 158
Double-precision floating-point extended registers 43
Double-precision floating-point registers 43
Dual address mode 598

E

effective address 59
ERI 1114, 1175
Ethernet reception 962
Ethernet relay 968
Ethernet transmission 956
Exception flow 112
Exception handling 105
Exception/interrupt codes 110
Execution cycles 94
External CPU interface (EXCPU) 621
External request mode 591

F

Fixed mode 595
Fixed-point transfer instructions 66
Floating-point control instructions 76
Floating-point double-precision instructions 76
Floating-point graphics acceleration instructions 77
Floating-point registers 39, 43
Floating-point single-precision instructions 75
FPU exception 131
Free-running operation 755

G

GEINT0 974
GEINT1 974
GEINT2 974
General FPU disable exception 128
General illegal instruction exception 126
General interrupt request 133
General Purpose I/O (GPIO) 1679
General registers 38
Gigabit Ethernet controller (GETHER) 783
GMII/MII frame 986

H

Hardware ITLB miss handling 164
H-UDI reset 116

I

I^2C bus data format 1046
I^2C bus interface (IIC) 1025
Infrared Data Communication Interface 1179
Initial page write exception 120, 171
Input Capture Function 703
Instruction address error 124
Instruction execution state 53
Instruction fetch cycle break 1784
Instruction set 57
Instruction TLB miss exception 119, 167
Instruction TLB multiple hit exception 166
Instruction TLB multiple-hit exception 117
Instruction TLB protection violation exception 122, 168
Intermittent mode 601
Interrupt controller (INTC) 233
Interrupt response time 308

IP Security Accelerator (SECURITY).... 995
IRL interrupts 293
IRQ interrupts 292
Issue rates 94
ITLB 159
ITLB address array 173
ITLB data array 174

L

L memory 217
LCD controller (LCDC) 1585
LCD module power-supply states..... 1637
List processor..... 1491
Little endian..... 52
Load-store architecture 57
Logic operation instructions 70

M

Magic packet detection..... 981
Manual PAUSE frame transmission..... 980
Manual reset 116
Master Mode..... 407
Master Mode 1..... 1226
Memory management unit..... 137, 324
Memory-mapped registers..... 51
MII frame..... 984
MII registers 989, 990
MII-RMII conversion circuit..... 991
MMC Mode..... 1326
Module standby mode..... 680
Multi-Buffer frame 971
Multiple interrupts..... 307
Multiple virtual memory mode..... 145

N

NMI (nonmaskable interrupt)..... 132
NMI interrupt..... 292
Notes on display-off mode
(LCDC stopped)..... 1638

O

On-chip module interrupts 295
On-Chip peripheral module request
mode 592
One-shot operation..... 755
Operand access cycle break 1785

P

P0, P3, and U0 areas 142
P1 area..... 142
P2 area..... 142
P4 area..... 142
Padding insertion 973
Page size bits..... 157
PAUSE frame reception..... 981
PC card controller (PCC) 1359
PECR 1700
Phase counting mode 740
PHY-LSI 984
Pipelining..... 79
Power-Down mode 434, 667
Power-down state..... 53
Power-on reset 116
Power-supply control sequences 1633
PPN 157
Pre-execution user break/
post-execution user break..... 130
Prefetch instruction 229
PRI 781
Privileged mode 38
Processing modes..... 38
Programming model..... 37
Protection key data..... 157
PWM modes 737

Q

Qtag 982

R

Realtime clock (RTC) 759
Receive Descriptor 945
Reception in Master Mode 1230
Reception in Slave Mode 1232
Registers
ADCSR 1660
ADDR 1658
APR 834
ARSTR 807
BCULR 840
BRGCKS2 1149
BRGDL2 1148
CAMR0 1774
CAMR1 1774
CAR0 1772
CAR1 1772
CBCR 1780
CBR0 1763
CBR1 1763
CCMFR 1779
CCR 191
CDCR 824
CDMR1 1777
CDR1 1776
CEECR 832
CEFCR 826
CERCR 831
CETR1 1778
CHATR 1324
CHCR 576
CLKON 1316
CMCNT 754
CMCOR 754
CMCSR 752
CMDR 1298
CMDSTRT 1302
CMDTYR 1292
CMSTR 751
CPUOPM 1903
CRR0 1770
CRR1 1770
CSTR 1308
CSWR 1322
CTLR0 1546
CTLR1 1548
CTOCR 1306
CVR 1543
DACR 1676
DADR 1675
DAR 573
DARB 574
DASTS 1537
DBK 429
DBR 47
DMA 1540
DMACR 1319
DMAOR 584
DMARS 587
DR 1318
DTOUTR 1307
ECMR 808
ECSIPR 816
ECSR 814
EDMR 904
EDRRR 907
EDSR 903
EDTRR 906
EESIPR 916
EESR 910
EPDR0i 1525
EPDR0o 1526
EPDR0s 1527
EPDR1 1528
EPDR2 1529
EPDR3 1530

EPDR4 1531
EPDR5 1532
EPIR 1549
EPSTL0 1541
EPSTL1 1542
EPSZ0o 1533
EPSZ1 1534
EPSZ4 1535
EXCCTRL 624
EXCINOR 626
EXCMSETR 625
EXPEVT 107
FCFTR 933
FCLR0 1538
FCLR1 1539
FDR 925
FIFOCLR 1318
FPSCR 48
FRECR 827
FRQCR 639
FWALCR0 896
FWALCR1 902
FWNLCR0 895
FWNLCR1 901
GBR 47
GECMR 839
HACACR 1407
HACCR 1394
HACCSAR 1396
HACCSDR 1398
HACPCML 1399
HACPCMR 1401
HACRIER 1405
HACRSR 1406
HACTIER 1402
HACTSR 1403
ICCCR 1041
ICMAR 1041
ICMCR 1036
ICMIER 1040
ICMSR 1038
ICR0 246
ICR1 248
ICRXD 1043
ICSAR 1035
ICSCR 1029
ICSIER 1034
ICSSR 1031
ICTXD 1043
IER0 1520
IFR0 1504
INT2A0 268
INT2A01 269
INT2A1 272
INT2A11 274
INT2B 283
INT2GPIC 289
INT2MSKCR 279
INT2MSKCR1 281
INT2MSKR 276
INT2MSKR1 277
INT2PRI 266
INTCR 1310
INTCR2 1320
INTEVT 108
INTMSK0 251
INTMSK1 253
INTMSK2 254
INTMSKCLR0 257
INTMSKCLR1 259
INTMSKCLR2 260
INTPRI 249
INTREQ 250
INTSTR 1312
INTSTR2 1321
IRMCR 153
ISR0 1515
LCCR 825
LDA0 225
LDA1 227

LDACLNR 1610
LDCNTR 1618
LDDFR 1596
LDHCNR 1605
LDHSYNR 1606
LDICKR 1591
LDINTR 1611
LDLAOR 1602
LDLIRNR 1622
LDMTR 1593
LDPALCR 1603
LDPMMR 1614
LDPR 1604
LDPSPR 1616
LDSARL 1601
LDSARU 1599
LDSMR 1598
LDUINTLNR 1621
LDUINTR 1619
LDVDLNR 1607
LDVSYNR 1609
LDVTLNR 1608
LSA0 221
LSA1 223
MACH 47
MACL 47
MAFCR 833
MAHR 818
MALR 819
MIM 417
MMUCR 149
MPR 835
MSTPCR0 672
MSTPCR1 673
NMIFCR 263
OPCR 1304
PACR 1693
PADR 1721
PASCR 152
PBCR 1695
PBDR 1722
PC 47
PCC0CSCIER 1376
PCC0CSCR 1372
PCC0GCR 1369
PCC0ISR 1366
PCCR 1696
PCDR 1723
PDCR 1698
PDDR 1724
PEDR 1725
PFCR 1701
PFDR 1726
PFRCR 838
PFTCR 837
PGCR 1703
PGDR 1727
PHCR 1705
PHDR 1728
PICR 1707
PIDR 1729
PIPR 822
PIPUPR 1736
PIR 817
PJCR 1709
PJDR 1730
PJPUPR 1737
PKCR 1711
PKDR 1731
PKPUPR 1738
PLCR 1713, 1719
PLDR 1732
PLLCR 641
PLPUPR 1739
PMDR 1733
PMPUPR 1740
PNCR 1717
PNDR 1734
PNPUPR 1742
PODR 1735

POPUPR 1743
PPUPR 1744
PR 47
PRPRICR 316
PSEL0 1745
PSEL1 1746
PSEL2 1749
PSEL3 1752
PSEL4 1755
PSR 821
PTEH 147
PTEL 148
QACR0 193
QACR1 194
R64CNT 764
RAMCR 195, 220
RCR1 772
RCR2 774
RCR3 777
RDAYAR 771
RDAYCNT 767
RDFAR 927
RDFFR 929
RDFXR 928
RDLAR 909
RFCR 830
RFLR 820
RHRAR 770
RHRCNT 765
RMCR 926
RMFCR 923
RMINAR 769
RMINCNT 765
RMONAR 772
RMONCNT 768
RPADIR 935
RSECAR 769
RSECCNT 764
RSPR 1300
RSPTYR 1293
RWKAR 770
RWKCNT 766
RXALCR0 894
RXALCR1 900
RXNLCR0 893
RXNLCR1 899
RYRAR 777
RYRCNT 768
SAR 572
SARB 573
SBRIVCLV 315
SCBRR 1083, 1141, 1247
SCFCR 1084, 1142
SCFDR 1144
SCFRDR 1068, 1126
SCFSR 1077, 1135
SCFTDR 1069, 1127
SCGRD 1262
SCLSR 1090, 1147
SCR 421
SCRDR 1258
SCRER 1091
SCRFDR 1086
SCRSR 1068, 1126, 1258
SCSC2R 1261
SCSCMR 1259
SCSCR 1073, 1131, 1248
SCSMPL 1264
SCSMR 1070, 1128, 1246
SCSMRIR 1150
SCSPTR 1087, 1145
SCSSR 1252
SCTDR 1251
SCTFDR 1086
SCTSR 1069, 1127, 1250
SCWAIT 1263
SDBSR 1805
SDINT 1804
SDIR 1803
SDMR 427

SDR 426
SGR 47
SICDAR 1215
SICTR 1196
SIFCTR 1211
SIIER 1209
SIMDR 1192
SIRCR 1202
SIRDAR 1214
SIRDR 1200
SISCR 1194
SISTR 1203
SITCR 1201
SITDAR 1213
SITDR 1199
SPC 47
SR 45
SSICR 1424
SSIRDR 1436
SSISR 1431
SSITDR 1436
SSR 47
STBCR 671
STICR 1006
STIFIFO 1014
STIIER 1009
STIISR 1007
STIMDR 1002
STIPCR 1013
STIPNR 1012
STITSC 1011
STR 423
SWSR 1323
TBCR 1297
TBNCR 1298
TCNT 695
TCOR 695
TCPR2 698
TCR 574, 696
TCRB 575
TDFAR 930
TDFFR 932
TDFXR 931
TDLAR 908
TFTR 924
TGR 727
TIER 723
TIOR 721
TLFRCR 829
TMDR 719
TOCR 692
TPAUSER 836
TRA 106
TRG 1536
TROCR 823
TRSCER 920
TSFRCR 828
TSR 725
TSTR 693
TSU_ADQT0 867
TSU_ADQT1 868
TSU_ADRH 889
TSU_ADRL 890
TSU_ADSBSY 871
TSU_BSYSL0 845
TSU_BSYSL1 847
TSU_CTRST 841
TSU_FCM 844
TSU_FWEN0 842
TSU_FWEN1 843
TSU_FWINMK 864
TSU_FWSL0 853
TSU_FWSL1 855
TSU_FWSLC 857
TSU_FWSR 861
TSU_POST1 877
TSU_POST2 880
TSU_POST3 883
TSU_POST4 886
TSU_PRISL0 849

TSU_PRISL1 851
TSU_QTAG0 859
TSU_QTAG1 860
TSU_TEN 872
TSU_VTAG0 869
TSU_VTAG1 870
TTB 149
TXALCR0 892
TXALCR1 898
TXNLCR0 891
TXNLCR1 897
USBHBCED 1473
USBHBHED 1473
USBHC 1463
USBHCCED 1472
USBHCHED 1472
USBHCS 1465
USBHDHED 1474
USBHFI 1475
USBHFN 1477
USBHFR 1476
USBHHCCA 1471
USBHID 1469
USBHIE 1468
USBHIS 1466
USBHLST 1479
USBHPCED 1471
USBHPS 1478
USBHR 1462
USBHRDA 1480
USBHRDB 1482
USBHRPS2 1486
USBHRS 1484
USBHSC 1489
USERIMASK 264
VBR 47
VDCNT 1317
WDTBST 652
WDTCNT 653
WDTCSR 650
WDTST 649
Relative priorities 110
Reset state 53
RMII frame 988
Root hub 1492
Round-robin mode 595
RTC power supply backup 684
RXI 1114, 1175

S

SCIF Interrupt Sources 1175
Self-Refresh and initialization 681
Self-Refresh mode 433
Sequential break 1787
Serial communication interface with FIFO 1059
Serial communication interface with FIFO/IrDA interface (SCIF/IrDA) 1119
Serial I/O with FIFO (SIOF) 1185
Serial interface engine (SIE) 1491
Serial Sound Interface (SSI) 1419
Setting the display resolution 1633
Share status bit 157
Shift instructions 71
Sign-extended 52
SIM Card Module (SIM) 1241
Single virtual memory mode 145
Single-precision floating-point extended register matrix 44
Single-precision floating-point extended registers 43
single-precision floating-point registers 43
Single-precision floating-point registers 43
Single-precision floating-point vector registers 43
Slave Mode 1 1226
Slave Mode 2 1226

Sleep mode 678
Slot FPU disable exception..... 129
Slot illegal instruction exception 127
Smart card interface..... 1266
Software standby mode 679
Stall operations 1577
Stream interface (STIF)..... 997
Strobe reception..... 1019
Strobe transmission 1023
SuperHyway bus bridge (SBR) 313
System control instructions 72
System registers..... 39
System registers related to FPU..... 39

T

T bit 58
TAP control 1823
TCNT Count Timing 701
TICPI..... 704
Time setting..... 778
Timer Unit 687
Transmission in Master Mode 1229
Transmission in Slave Mode 1231
Transmit Descriptor..... 939
Transmit/Receive Reset..... 1233
TUNI 704
TXI 1114, 1175
Types of exceptions 110

U

Unconditional trap 125
USB function controller (USBF) 1495
User break controller 1759
User break operation..... 1782
User debugging interface 1797
User mode..... 38
UTLB..... 156
UTLB address array..... 175
UTLB data array 176

V

Validity bit..... 157
Vector addresses 110
Virtual address space 140
VPN 156

W

Watchdog timer and reset 645
Write-back instruction 229
Write-through bit 158

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7763

Publication Date: Rev.1.00, Oct. 01, 2007
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

© 2007. 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) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7763
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan