SH7709

Hardware Manual

HITACHI

ADE-602-123A
Rev. 3.0
4/13/99
Hitachi, Ltd.

Notice

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi’s permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user’s unit according to this document.
4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi’s semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. MEDICAL APPLICATIONS: Hitachi’s products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi’s sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi’s products are requested to notify the relevant Hitachi sales offices when planning to use the products in MEDICAL APPLICATIONS.

Contents

Preface 1

Section 1 Overview 3
1.1 SH7709 Features 3
1.2 Pin Description 7
1.2.1 Pin Arrangement 7
1.2.2 Pin Functions 9

Section 2 CPU 13
2.1 Register Configuration 13
2.1.1 Privileged Mode and Banks 13
2.1.2 General Registers 16
2.1.3 System Registers 17
2.1.4 Control Registers 18
2.2 Data Formats 20
2.2.1 Data Format in Registers 20
2.2.2 Data Format in Memory 20
2.3 Instruction Features 21
2.3.1 Execution Environment 21
2.3.2 Addressing Modes 22
2.3.3 Instruction Formats 26
2.4 Instruction Set 29
2.4.1 Instruction Set Classified by Function 29
2.4.2 Instruction Code Map 43
2.5 Processor States and Processor Modes 46
2.5.1 Processor States 46
2.5.2 Processor Modes 47

Section 3 Memory Management Unit (MMU) 49
3.1 Overview 49
3.1.1 Features 49
3.1.2 Role of MMU 49
3.1.3 Virtual Address Space 51
3.1.4 Register Configuration 54
3.2 Register Description 55
3.3 TLB Functions 57
3.3.1 Configuration of the TLB 57
3.3.2 TLB Indexing 59
3.3.3 TLB Address Comparison 60
3.3.4 Page Management Information 61

3.4 MMU Functions........ 62
3.4.1 MMU Hardware Management 62
3.4.2 MMU Software Management........ 63
3.4.3 MMU Instruction (LDLTB) 64
3.4.4 Avoiding Synonym Problems........ 65
3.5 MMU Exceptions........ 68
3.5.1 TLB Miss Exception........ 68
3.5.2 TLB Protection Violation Exception........ 69
3.5.3 TLB Invalid Exception 70
3.5.4 Initial Page Write Exception........ 71
3.5.5 Processing Flow in Event of MMU Exception 73
3.6 Memory-Mapped TLB........ 75
3.6.1 Address Array........ 75
3.6.2 Data Array........ 76
3.6.3 Usage Examples 78
3.7 Usage Note 78

Section 4 Exception Handling........ 79
4.1 Overview 79
4.1.1 Features........ 79
4.1.2 Register Configuration........ 79
4.2 Exception Handling Function........ 80
4.2.1 Exception Handling Flow 80
4.2.2 Exception Handling Vector Addresses........ 81
4.2.3 Acceptance of Exceptions........ 83
4.2.4 Exception Codes........ 85
4.2.5 Exception Request Masks 86
4.2.6 Returning from Exception Handling 86
4.3 Register Description........ 87
4.4 Exception Handler Operation........ 88
4.4.1 Reset........ 88
4.4.2 Interrupts 88
4.4.3 General Exceptions........ 89
4.5 Individual Exception Operations........ 89
4.5.1 Resets 89
4.5.2 General Exceptions........ 90
4.5.3 Interrupts 93
4.6 Cautions........ 94

Section 5 Cache 97
5.1 Overview 97
5.1.1 Features........ 97
5.1.2 Cache Structure 97
5.1.3 Register Configuration........ 99

5.2 Register Description.. 100
5.2.1 Cache Control Register (CCR).. 100
5.3 Cache Operation 101
5.3.1 Searching the Cache... 101
5.3.2 Read Access.. 103
5.3.3 Write Access... 103
5.3.4 Write-Back Buffer 104
5.3.5 Coherency of Cache and External Memory 104
5.3.6 RAM Mode... 104
5.4 Memory-Mapped Cache.. 105
5.4.1 Address Array.. 105
5.4.2 Data Array 106
5.5 Usage Examples.. 108
5.5.1 Invalidating Specific Entries 108
5.5.2 Reading the Data of a Specific Entry 108

Section 6 Interrupt Controller (INTC) 109

6.1 Overview 109
6.1.1 Features 109
6.1.2 Block Diagram.. 110
6.1.3 Pin Configuration.. 111
6.1.4 Register Configuration... 111
6.2 Interrupt Sources.. 112
6.2.1 NMI Interrupts.. 112
6.2.2 IRQ Interrupt 112
6.2.3 IRL Interrupts 113
6.2.4 PINT Interrupt 115
6.2.5 On-Chip Peripheral Module Interrupts 115
6.2.6 Interrupt Exception Processing and Priority 116
6.3 INTC Registers 123
6.3.1 Interrupt Priority Registers A to E (IPRA–IPRE) 123
6.3.2 Interrupt Control Register 0 (ICR0)... 124
6.3.3 Interrupt Control Register 1 (ICR1)... 125
6.3.4 Interrupt Control Register 2 (ICR2)... 127
6.3.5 PINT Interrupt Enable Register (PINTER).. 128
6.3.6 Interrupt Request Register 0 (IRR0).. 129
6.3.7 Interrupt Request Register 1 (IRR1).. 131
6.3.8 Interrupt Request Register 2 (IRR2).. 133
6.4 INTC Operation 134
6.4.1 Interrupt Sequence 134
6.4.2 Multiple Interrupts 137
6.5 Interrupt Response Time 138

Section 7 User Break Controller (UBC) 141
7.1 Overview 141
7.1.1 Features 141
7.1.2 Block Diagram 142
7.1.3 Register Configuration... 143
7.1.4 Break Conditions and Register Settings 143
7.2 UBC Register Functions... 144
7.2.1 Break Address Register A (BARA)... 144
7.2.2 Break Address Register B (BARB) 144
7.2.3 Break ASID Register A (BASRA) 145
7.2.4 Break ASID Register B (BASRB).. 145
7.2.5 Break Address Mask Register A (BAMRA)..................................... 145
7.2.6 Break Address Mask Register B (BAMRB) 146
7.2.7 Break Bus Cycle Register A (BBRA)... 146
7.2.8 Break Bus Cycle Register B (BBRB)... 147
7.2.9 Break Data Register B (BDRB) 148
7.2.10 Break Data Mask Register B (BDMRB)... 149
7.2.11 Break Control Register (BRCR)... 150
7.3 UBC Operation 152
7.3.1 User Break Operation Flow... 152
7.3.2 Instruction Fetch Cycle Break.. 153
7.3.3 Data Access Cycle Break... 154
7.3.4 Saved Program Counter (PC) Value... 155
7.3.5 Examples of Use.. 156
7.3.6 Cautions 158

Section 8 Power-Down Modes...159
8.1 Overview 159
8.1.1 Power-Down Modes 159
8.1.2 Pin Configuration 160
8.1.3 Register Configuration... 160
8.2 Register Description... 161
8.2.1 Standby Control Register (STBCR) 161
8.2.2 Standby Control Register 2 (STBCR2).. 162
8.3 Sleep Mode... 164
8.3.1 Transition to Sleep Mode... 164
8.3.2 Canceling Sleep Mode... 164
8.4 Standby Mode... 164
8.4.1 Transition to Standby Mode... 164
8.4.2 Canceling Standby Mode... 165
8.4.3 Clock Pause Function 166
8.5 Module Standby Function 167
8.5.1 Transition to Module Standby Function 167
8.5.2 Clearing the Module Standby Function... 168

8.6 Timing of STATUS Pin Changes 168
8.6.1 Timing for Resets 168
8.6.2 Timing for Canceling Standbys 170
8.6.3 Timing for Canceling Sleep Mode 171
8.7 Hardware Standby Function 173
8.7.1 Transition to Hardware Standby Mode 173
8.7.2 Canceling Hardware Standby Mode 173
8.7.3 Hardware Standby Mode Timing 174

Section 9 On-Chip Oscillation Circuits 175
9.1 Overview 175
9.1.1 Features 175
9.2 Overview of the CPG 176
9.2.1 CPG Block Diagram 176
9.2.2 CPG Pin Configuration 178
9.2.3 CPG Register Configuration 178
9.3 Clock Operating Modes 179
9.4 Register Descriptions 184
9.4.1 Frequency Control Register (FRQCR) 184
9.5 Changing the Frequency 187
9.5.1 Changing the Multiplication Rate 187
9.5.2 Changing the Division Ratio 187
9.6 PLL Standby Function 188
9.6.1 Overview of the PLL Standby Function 188
9.6.2 Usage 188
9.7 Controlling Clock Output 189
9.7.1 Clock Modes 0–2 189
9.7.2 Clock Modes 3, 4 189
9.8 Overview of the WDT 190
9.8.1 Block Diagram of the WDT 190
9.8.2 Register Configurations 190
9.9 WDT Registers 191
9.9.1 Watchdog Timer Counter (WTCNT) 191
9.9.2 Watchdog Timer Control/Status Register (WTCSR) 191
9.9.3 Notes on Register Access 193
9.10 Using the WDT 194
9.10.1 Canceling Standbys 194
9.10.2 Changing the Frequency 194
9.10.3 Using Watchdog Timer Mode 195
9.10.4 Using Interval Timer Mode 195
9.11 Notes on Board Design 196

Section 10 Bus State Controller (BSC)..199
10.1 Overview 199
10.1.1 Features 199
10.1.2 Block Diagram.. 202
10.1.3 Pin Configuration.. 204
10.1.4 Register Configuration.. 206
10.1.5 Area Overview.. 207
10.1.6 PCMCIA Support... 210
10.2 BSC, BSCP Registers... 214
10.2.1 Bus Control Register 1 (BCR1)... 214
10.2.2 Bus Control Register 2 (BCR2)... 217
10.2.3 Bus Control Register 3 (BCR3)... 218
10.2.4 Wait State Control Register 1 (WCR1) 221
10.2.5 Wait State Control Register 2 (WCR2) 222
10.2.6 Individual Memory Control Register (MCR).................................. 226
10.2.7 DRAM Control Register (DCR).. 230
10.2.8 PCMCIA Control Register (PCR) 232
10.2.9 SDRAM Mode Register (SDMR).. 234
10.2.10 Refresh Timer Control/Status Register (RTCSR) 235
10.2.11 Refresh Timer Counter (RTCNT) 237
10.2.12 Refresh Time Constant Register (RTCOR) 238
10.2.13 Refresh Count Register (RFCR)... 239
10.2.14 Cautions on Accessing Refresh Control Related Registers 240
10.3 BSC Operation.. 241
10.3.1 Endian/Access Size and Data Alignment 241
10.3.2 Description of Areas 246
10.3.3 Basic Interface.. 249
10.3.4 DRAM Interface 255
10.3.5 Synchronous DRAM Interface 271
10.3.6 Burst ROM Interface.. 288
10.3.7 PCMCIA Interface... 291
10.3.8 Waits between Access Cycles 303
10.3.9 Bus Arbitration 304

Section 11 Direct Memory Access Controller (DMAC) 305
11.1 Overview 305
11.1.1 Features 305
11.1. 2 Block Diagram.. 307
11.1.3 Pin Configuration.. 308
11.1.4 Register Configuration.. 309
11.2 Register Descriptions 310
11.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3) 310
11.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)................................ 311
11.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)........................ 312

11.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3) 313
11.2.5 DMA Operation Register (DMAOR) 318
11.3 Operation 320
11.3.1 DMA Transfer Flow 320
11.3.2 DMA Transfer Requests 322
11.3.3 Channel Priority 324
11.3.4 DMA Transfer Types 327
11.3.5 Number of Bus Cycle States and DREQ Pin Sampling Timing 336
11.3.6 Source Address Reload Function 341
11.3.7 DMA Transfer Ending Conditions 343
11.4 Compare Match Timer (CMT) 344
11.4.1 Overview 344
Features 344
Block Diagram 345
Register Configuration 345
11.4.2 Register Descriptions 346
Compare Match Timer Start Register (CMSTR) 346
Compare Match Timer Control/Status Register (CMCSR) 347
Compare Match Counter (CMCNT) 348
Compare Match Constant Register (CMCOR) 348
11.4.3 Operation 349
11.4.4 Compare Match 350
11.5 Examples of Use 351
11.5.1 Example of DMA Transfer between
On-Chip IRDA and External Memory 351
11.5.2 Example of DMA Transfer between AD0 and External Memory 352
11.5.3 Example of DMA Transfer between
External Memory and SCIF Transmitter 354
11.6 Cautions 355

Section 12 Timer (TMU) 357
12.1 Overview 357
12.1.1 Features 357
12.1.2 Block Diagram 358
12.1.3 Pin Configuration 359
12.1.4 Register Configuration 359
12.2 TMU Registers 360
12.2.1 Timer Output Control Register (TOCR) 360
12.2.2 Timer Start Register (TSTR) 361
12.2.3 Timer Control Register (TCR) 362
12.2.4 Timer Constant Register (TCOR) 365
12.2.5 Timer Counters (TCNT) 366
12.2.6 Input Capture Register (TCPR2) 367

12.3 TMU Operation 368
12.3.1 Overview 368
12.3.2 Basic Functions 368
12.4 Interrupts 373
12.4.1 Status Flag Set Timing 373
12.4.2 Status Flag Clear Timing 373
12.4.3 Interrupt Sources and Priorities 374
12.5 Usage Notes 374
12.5.1 Writing to Registers 374
12.5.2 Reading Registers 374

Section 13 Realtime Clock (RTC) 375

13.1 Overview 375
13.1.1 Features 375
13.1.2 Block Diagram 375
13.1.3 Pin Configuration 377
13.1.4 RTC Register Configuration 378
13.2 RTC Registers 379
13.2.1 64-Hz Counter (R64CNT) 379
13.2.2 Second Counter (RSECCNT) 379
13.2.3 Minute Counter (RMINCNT) 380
13.2.4 Hour Counter (RHRCNT) 380
13.2.5 Day of the Week Counter (RWKCNT) 381
13.2.6 Date Counter (RDAYCNT) 382
13.2.7 Month Counter (RMONCNT) 382
13.2.8 Year Counter (RYRCNT) 383
13.2.9 Second Alarm Register (RSECAR) 383
13.2.10 Minute Alarm Register (RMINAR) 384
13.2.11 Hour Alarm Register (RHRAR) 385
13.2.12 Day of the Week Alarm Register (RWKAR) 386
13.2.13 Date Alarm Register (RDAYAR) 387
13.2.14 Month Alarm Register (RMONAR) 387
13.2.15 RTC Control Register 1 (RCR1) 388
13.2.16 RTC Control Register 2 (RCR2) 389
13.3 RTC Operation 391
13.3.1 Initial Settings of Registers after Power-On 391
13.3.2 Setting the Time 391
13.3.3 Reading the Time 393
13.3.4 Alarm Function 394
13.3.5 Crystal Oscillator Circuit 395

Section 14 Serial Communication Interface (SCI) 397
14.1 Overview 397
14.1.1 Features 397
14.1.2 Block Diagram 398
14.1.3 Pin Configuration 401
14.1.4 Register Configuration 402
14.2 Register Descriptions 403
14.2.1 Receive Shift Register 403
14.2.2 Receive Data Register 403
14.2.3 Transmit Shift Register 403
14.2.4 Transmit Data Register 404
14.2.5 Serial Mode Register 404
14.2.6 Serial Control Register 407
14.2.7 Serial Status Register 410
14.2.8 Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR) 413
14.2.9 Bit Rate Register (SCBRR) 415
14.3 Operation 423
14.3.1 Overview 423
14.3.2 Operation in Asynchronous Mode 425
14.3.3 Multiprocessor Communication 434
14.3.4 Clock Synchronous Operation 442
14.4 SCI Interrupt Sources 450
14.5 Notes on Use 451

Section 15 Smart Card Interface 455
15.1 Overview 455
15.1.1 Features 455
15.1.2 Block Diagram 456
15.1.3 Pin Configuration 457
15.1.4 Register Configuration 457
15.2 Register Descriptions 458
15.2.1 Smart Card Mode Register (SCSCMR) 458
15.2.2 Serial Status Register (SCSSR) 459
15.3 Operation 460
15.3.1 Overview 460
15.3.2 Pin Connections 460
15.3.3 Data Format 461
15.3.4 Register Settings 462
15.3.5 Clock 464
15.3.6 Data Transmission and Reception 466
15.4 Usage Notes 472
15.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode 472
15.4.2 Retransmission (Receive and Transmit Modes) 474

Section 16 Serial Communication Interface with FIFO (SCIF) 477
16.1 Overview 477
16.1.1 Features 477
16.1.2 Block Diagram.. 478
16.1.3 Pin Configuration.. 481
16.1.4 Register Configuration... 482
16.2 Register Descriptions 482
16.2.1 Receive Shift Register 482
16.2.2 Receive FIFO Data Register 483
16.2.3 Transmit Shift Register.. 483
16.2.4 Transmit FIFO Data Register... 483
16.2.5 Serial Mode Register 484
16.2.6 Serial Control Register... 486
16.2.7 Serial Status Register... 488
16.2.8 Bit Rate Register (SCBRR).. 492
16.2.9 FIFO Control Register (SCFCR) 500
16.2.10 Register of the Number of FIFO Data (SCFDR).. 502
16.3 Operation... 503
16.3.1 Overview 503
16.3.2 Serial Operation.. 504
16.4 SCIF Interrupts 513
16.5 Notes on Use.. 514

Section 17 IrDA..517
17.1 Overview 517
17.1.1 Features 517
17.1.2 Block Diagram.. 518
17.1.3 Pin Configuration.. 521
17.1.4 Register Configuration... 522
17.2 Register Description.. 523
17.2.1 Serial Mode Register (SCSMR) 523
17.3 Operation Description 525
17.3.1 Overview 525
17.3.2 Transmit 525
17.3.3 Receive.. 526

Section 18 Pin Function Controller 527
18.1 Overview 527
18.2 Register Configuration 531
18.3 Register Descriptions 532
18.3.1 Port A Control Register (PACR) 532
18.3.2 Port B Control Register (PBCR) 533
18.3.3 Port C Control Register (PCCR) 534
18.3.4 Port D Control Register (PDCR).. 535
18.3.5 Port E Control Register (PECR)... 536

18.3.6 Port F Control Register (PFCR).. 537
18.3.7 Port G Control Register (PGCR) 538
18.3.8 Port H Control Register (PHCR) 539
18.3.9 Port J Control Register (PJCR).. 540
18.3.10 Port K Control Register (PKCR) 541
18.3.11 Port L Control Register (PLCR) 542
18.3.12 Port SC Control Register (SCPCR)... 543

Section 19   I/O Port... 547
19.1 Overview 547
19.2 Port A 547
19.2.1 Register Descriptions 547
19.2.2 Port A Data Register (PADR).. 548
19.3 Port B... 549
19.3.1 Register Descriptions 549
19.3.2 Port B Data Register (PBDR) 550
19.4 Port C... 551
19.4.1 Register Descriptions 551
19.4.2 Port C Data Register (PCDR) 552
19.5 Port D 553
19.5.1 Register Descriptions 553
19.5.2 Port D Data Register (PDDR).. 554
19.6 Port E... 555
19.6.1 Register Descriptions 555
19.6.2 Port E Data Register (PEDR) 556
19.7 Port F 557
19.7.1 Register Descriptions 557
19.7.2 Port F Data Register (PFDR)... 558
19.8 Port G 559
19.8.1 Register Descriptions 559
19.8.2 Port G Data Register (PGDR).. 560
19.9 Port H 561
19.9.1 Register Descriptions 561
19.9.2 Port H Data Register (PHDR).. 562
19.10 Port J... 563
19.10.1 Register Descriptions 563
19.10.2 Port J Data Register (PJDR) 564
19.11 Port K.. 565
19.11.1 Register Descriptions 565
19.11.2 Port K Data Register (PKDR)... 566
19.12 Port L... 567
19.12.1 Register Descriptions 567
19.12.2 Port L Data Register (PLDR) 568

19.13 SC Port 569
19.13.1 Register Descriptions 569
19.13.2 Port SC Data Register (SCPDR).. 570

Section 20 A/D Converter...571
20.1 Overview 571
20.1.1 Features 571
20.1.2 Block Diagram... 572
20.1.3 Input Pins.. 573
20.1.4 Register Configuration... 574
20.2 Register Descriptions 575
20.2.1 A/D Data Registers A to D (ADDRA to ADDRD) 575
20.2.2 A/D Control/Status Register (ADCSR) 576
20.3 Bus Master Interface 579
20.4 Operation... 580
20.4.1 Single Mode (MULTI = 0).. 580
20.4.2 Multi Mode (MULTI = 1) 582
20.4.3 Input Sampling and A/D Conversion Time 584
20.4.4 External Trigger Input Timing 585
20.5 Interrupts 585
20.6 Definitions of A/D Conversion Accuracy 586
20.7 A/D Converter Usage Notes 587
20.7.1 Setting Analog Input Voltage.. 587
20.7.2 Handling of Analog Input Pins.. 587

Section 21 D/A Converter...589
21.1 Overview 589
21.1.1 Features 589
21.1.2 Block Diagram.. 589
21.1.3 Input/Output Pins... 590
21.1.4 Register Configuration.. 590
21.2 Register Descriptions 591
21.2.1 D/A Data Registers 0 and 1 (DADR0/1)... 591
21.2.2 D/A Control Register (DACR) 591
21.3 Operation... 593

Section 22 Electrical Characteristics 595
22.1 Absolute Maximum Ratings... 595
22.2 DC Characteristics 595
22.3 AC Characteristics 599
22.3.1 Clock Timing.. 600
22.3.2 Control Signal Timing 608
22.3.3 Basic Timing 613
22.3.4 Burst ROM Timing... 616
22.3.5 DRAM Timing 619

22.3.6 Synchronous DRAM Timing 631
22.3.7 PCMCIA Timing 642
22.3.8 Peripheral Module Signal Timing 649
22.3.9 AC Characteristics Measurement Conditions 656
22.4 A/D Conversion Characteristics 657
22.5 D/A Conversion Characteristics 657

Appendix A Pin Functions 659
A.1 Pin States 659
A.2 Pin Specifications 662
A.3 Handling of Unused Pins 667
A.4 Pin States in Access to Each Address Space 668

Appendix B Memory-Mapped Control Registers 697
B.1 Register Address Map 697
B.2 Registers bits 703

Appendix C Package Dimensions 715

Preface

SH7709 is a single-chip RISC microprocessor that integrates a Hitachi-original RISC-type SuperH architecture CPU as its core that has an on-chip multiplier, cache memory, and a memory management unit as well as peripheral funPctions required for system configuration such as a timer, a real-time clock, an interrupt controller, and a serial communication interface. This LSI includes data protection and virtual memory functions, and was designed by building a memory management unit onto an SuperH microprocessor (SH-1 or SH-2).

High-speed data transfers on par with a direct memory access controller (DMAC) are implemented. An external memory access support function enables direct connection to each memory. The SH7709 microprocessor also supports, an infrared communication function, an A/D converter, and a D/A converter.

A powerful built-in power management function keeps power consumption low, even during high-apeed operation. This LSI can run at four times the frequency of the system bus operating speed, making it optium for electrical devices such as PDA requring both high speed and low power.

This hardware manuals describes the hardware of the SH7709. Details of instructions can be found in the programming manual.

Related Manuals

SH7709 instructions
SH-3/SH-3E Programming Manual

Please consult your Hitachi sales representative for details of development environment system.

Section 1 Overview

1.1 SH7709 Features

This LSI is a single-chip RISC microprocessor that integrates a Hitachi-original RISC-type SuperH architecture CPU as its core that has an on-chip multiplier, cache memory, and a memory management unit as well as peripheral functions required for system configuration such as a timer, a realtime clock, an interrupt controller, and a serial communication interface. This LSI includes data protection and virtual memory functions, and was designed by building a memory management unit onto an SuperH series microprocessor (SH-1 or SH-2).

High-speed data transfers on par with a direct memory access controller (DMAC) are implemented An external memory access support function enables direct connection to each memory. The SH7709 microprocessor also supports, an infrared communication function, an A/D converter, and a D/A converter.

A powerful built-in power management function keeps power consumption low, even during high-speed operation. This LSI can run at four times the frequency of the system bus operating speed, making it optimum for electrical devices such as PDA requiring both high speed and low power.

The features of this LSI is listed in table 1.1. The specifications are shown in table 1.2.

Table 1.1 SH7709 Features

Item	Features
CPU	• Original Hitachi SuperH architecture • 32-bit internal data bus • General-register files — Sixteen 32-bit general registers (eight 32-bit shadow registers) — Five 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatibility with the SH-2) — Instruction length: 16-bit fixed length for improved code efficiency — Load-store architecture — Delayed branch instructions — Instruction set based on C language • Instruction execution time: one instruction/cycle for basic instructions • Logical address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 logical address space • On-chip multiplier • Five-stage pipeline
Operating modes, clock pulse generator	• Clock mode: selected from an on-chip oscillator module, a frequency-doubling circuit, or a clock output by combining them by PLL synchronization • Processing states: — Power-on reset state — Manual reset state — Exception processing state — Program execution state — Power-down state — Bus-released state • Power-down modes: — Sleep mode — Standby mode — Module standby mode

Table 1.1 SH7709 Features (cont)

Item	Features
Memory management unit	• 4 Gbytes of address space, 256 address spaces (ASID 8 bits) • Page unit sharing • Supports multiple page sizes: 1, 4 kbytes • 128-entry, 4-way set associative TLB • Supports software selection of replacement method and random-replacement algorithms • Contents of TLB are directly accessible by address mapping
Cache memory	• 8-kbyte cache, mixed instruction/data • 128 entries, 4-way set associative (8-kbyte cache), 16-byte block length • Write-back, write through, LRU replacement algorithm • 1-stage write-back buffer • Cache can be divided (4 kB/2-way cache memory + 4-kB memory)
Interrupt controller	• Seven external interrupt pins (NMI, IRQ5—IRQ0, $\overline{\text{IRL3}}$—$\overline{\text{IRL0}}$. Either IRQ or $\overline{\text{IRL}}$ can be selected) • On-chip peripheral interrupts: set priority levels for each module
User break controller	• 2 break channels • Addresses, data values, type of access, and data size can all be set as break conditions • Supports a sequential break function
Bus state controller	• Physical address space divided into six areas, each a maximum 64 Mbytes, with the following features settable for each area: — Bus size (8, 16, or 32 bits) — Number of wait cycles (also supports a hardware wait function) — Setting the type of space enables direct connection to SRAM, DRAM, SDRAM, and burst ROM — Supports PCMCIA interface (2 channels) — Outputs chip select signal (CS0, CS2–CS6) for corresponding area • DRAM/SDRAM refresh function — Programmable refresh interval — Supports CAS-before-RAS refresh and self-refresh modes — Supports power-down DRAM • DRAM/SDRAM burst access function • Usable as either big or little endian machine

Table 1.1 SH7709 Features (cont)

Item	Features
Timer	• 3-channel auto-reload type 32-bit timer • Input capture function • 6 types of counter input clocks can be selected • Maximum resolution: 2 MHz
Realtime clock	• Built-in clock, calendar functions, and alarm functions • On-chip 32-kHz crystal oscillator circuit with a maximum resolution (cycle interrupt) of 1/256 second
Serial communication interface 0	• Select start-stop sync mode or clock sync system • Full-duplex communication • Supports smart card interface
Serial communication interface 1	• 16-byte FIFO for transmit/receive • DMA can be transferred • IrDA: interface based on 1.0
Serial communication interface 2	• 16-byte FIFO for transmit/receive • DMA can be transferred • Hardware flow control
DMA controller	• 4 channels • Burst mode and cycle-steel mode
I/O port	• 16 bits (at 16-bit external bus)
A/D converter	• 10 bits ± 4 LSB, 8 channels • Conversion time: 10 µs • Input range: 0 – Vcc (max. 3.6 V)
D/A converter	• 8 bits ± 4 LSB, 2 channels • Conversion time: 10 µs • Output range: 0 – Vcc (max. 3.6 V)
Package	• 208-pin plastic LQFP (FP-208C)

Table 1.2 Characteristics

Power supply voltage	• 3.3 ± 0.3 V
Operation frequency	• Internal frequency: max. 80 MHz, and external frequency: max. 40 MHz
Process	• 0.35-µm CMOS/3-layer metal

1.2 Pin Description

1.2.1 Pin Arrangement

Figure 1.1 shows the pin arrangement and figure 1.2 shows a block diagram.

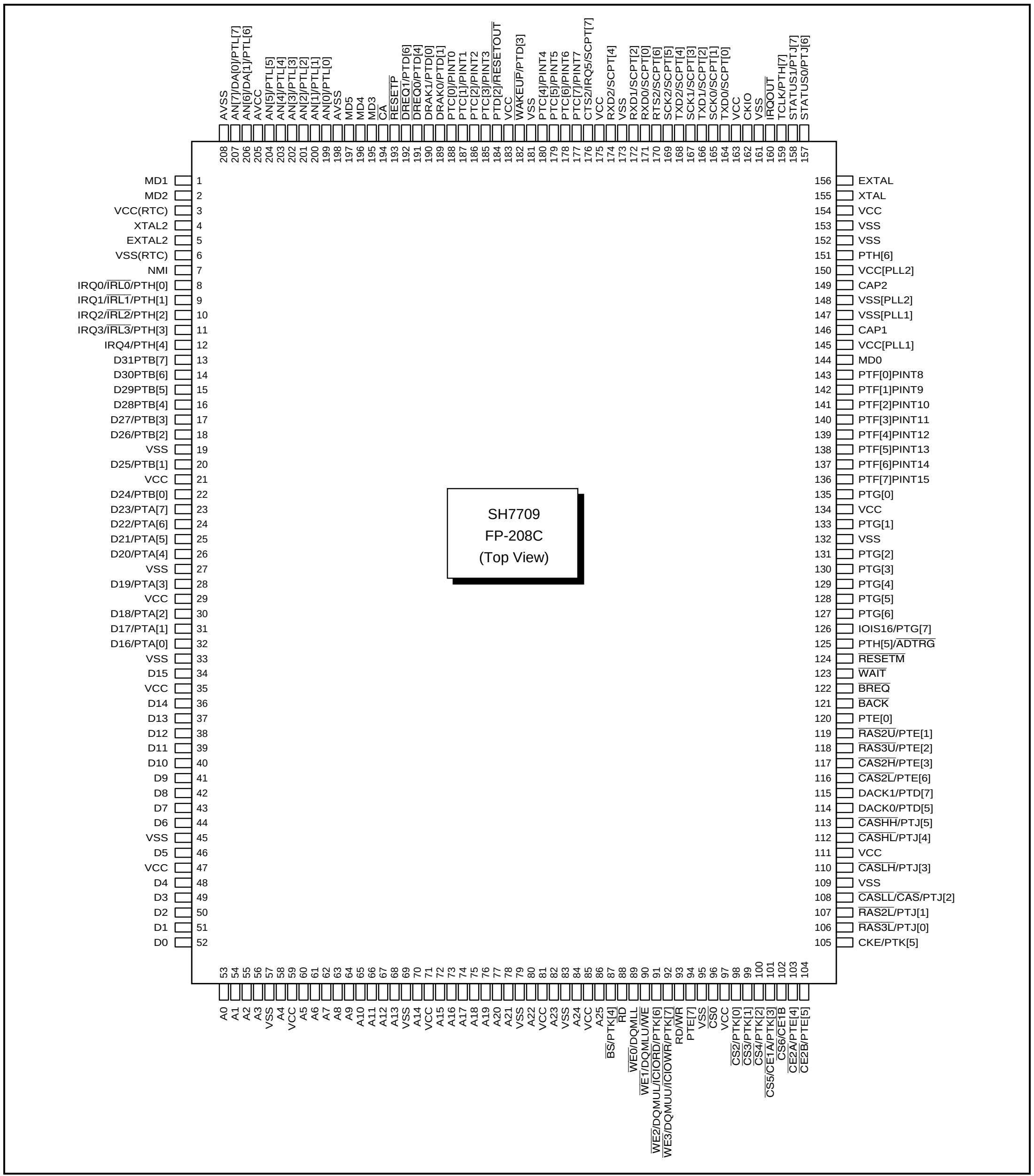
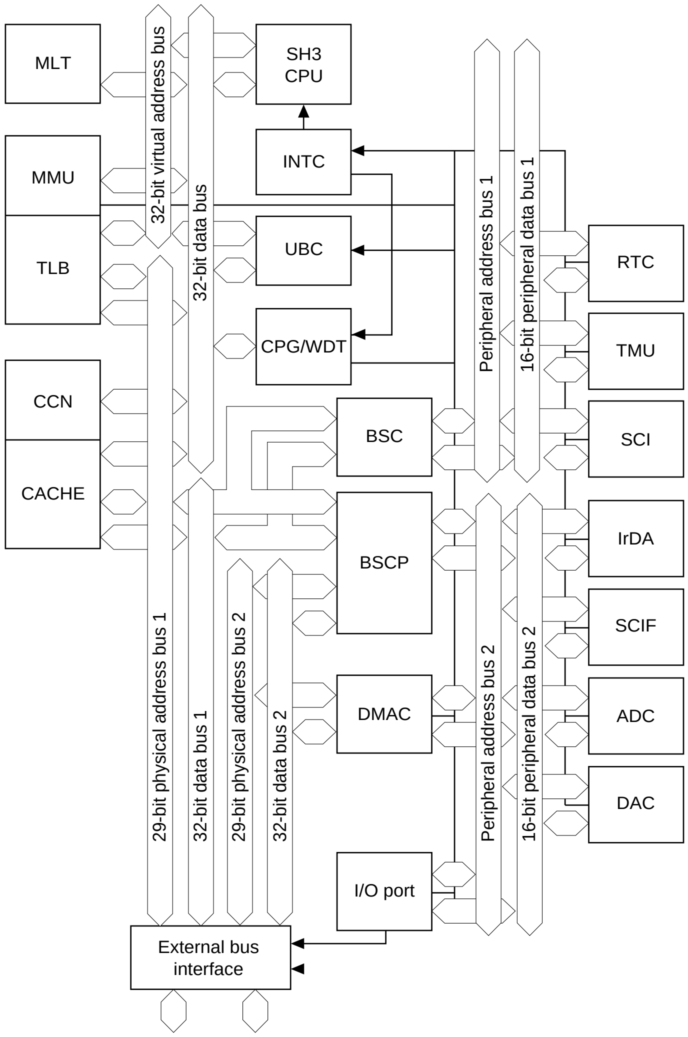


Figure 1.1 Pin Arrangement



Legend:

ADC: Analog-digital converter
BSC: Bus state controller 1
BSCP: Bus state controller 2
CACHE: Cache memory
CCN: Cache memory controller
CPG/WDT: Clock pulse generator/watchdog timer
CPU: Central processing unit
DAC: Digital-analog converter
DMAC: Direct memory access controller
INTC: Interrupt controller

MLT: Multiplier
MMU: Memory management unit
RTC: Real-time clock
SCI: Serial communication interface (with smart card interface)
IrDA: Serial communication interface (with IrDA)
SCIF: Serial communication interface (with FIFO)
TLB: Translation look-aside buffer
TMU: Timer unit
UBC: User break controller

Figure 1.2 SH7709 Block Diagram

1.2.2 Pin Functions

Table 1.3 summarizes the pin functions.

Table 1.3 SH7709 Pin Function

Function	Pin Name	Description	I/O	Number of Pins
Data Bus	D[15:0]	Data bus D[15:0]	IO	16
	D[23:16]/PTA[7:0]	Data bus D[23:16]/I/O port A[7:0]	IO	8
	D[31:24]/PTB[7:0]	Data bus D[31:24]/I/O port B[7:0]	IO	8
Address Bus	A[25:0]	Address bus A[25:0]	O	26
Bus Control	$\overline{\text{CS0}}$	Chip select 0	O	1
	$\overline{\text{CS2}}$/PTK[0]	Chip select 2/I/O port K[0]	O/IO	1
	$\overline{\text{CS3}}$/PTK[1]	Chip select 3/I/O port K[1]	O/IO	1
	$\overline{\text{CS4}}$/PTK[2]	Chip select 4/I/O port K[2]	O/IO	1
	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTK[3]	Chip select 5/CE1 (area 5 PCMCIA) /I/O port K[3]	O/IO	1
	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	Chip select 6/CE1 (area 6 PCMCIA)	O	1
	$\overline{\text{BS}}$/PTK[4]	Bus cycle start signal/I/O port K[4]	O/IO	1
	$\overline{\text{RAS3U}}$/PTE[2]	RAS (area 3 DRAM, area 2, 3 SDRAM upper 32MByte address)/I/O port E[2]	O/IO	1
	$\overline{\text{RAS3L}}$/PTJ[0]	RAS (area 3 DRAM, area 2, 3 SDRAM lower 32Mbyte address)/I/O port J[0]	O/IO	1
	$\overline{\text{RAS2U}}$/PTE[1]	RAS (area 2 DRAM upper 32MByte address) /I/O port E[1]	O/IO	1
	$\overline{\text{RAS2L}}$/PTJ[1]	RAS (area 2 DRAM lower 32Mbyte address) /I/O port J[1]	O/IO	1
	$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]	CAS for D7-D0 (DRAM) /CAS (SDRAM) /I/O port J[2]	O/IO	1
	$\overline{\text{CASLH}}$/PTJ[3]	CAS for D15-D8 (DRAM) /I/O port J[3]	O/IO	1
	$\overline{\text{CASHL}}$/PTJ[4]	CAS for D23-D16 (DRAM) /I/O port J[4]	O/IO	1
	$\overline{\text{CASHH}}$/PTJ[5]	CAS for D31-D24 (DRAM) /I/O port J[5]	O/IO	1
	$\overline{\text{CAS2L}}$/PTE[6]	CAS for D7-D0 (area 2 DRAM)/I/O port E[6]	O/IO	1
	$\overline{\text{CAS2H}}$/PTE[3]	CAS for D15-D8 (area 2 DRAM)/I/O port E[3]	O/IO	1
	$\overline{\text{WE0}}$/DQMLL	D7-D0 select signal/DQM (SDRAM)	O	1
	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	D15-D8 select signal/DQM (SDRAM)/ PCMCIA WE	O	1
	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$/ PTK[6]	D23-D16 select signal/DQM (SDRAM)/ PCMCIA I/O-read/I/O port K[6]	O/IO	1
	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$/ PTK[7]	D31-D24 select signal/DQM (SDRAM)/ PCMCIA I/O-write/I/O port K[7]	O/IO	1

Table 1.3 SH7709 Pin Function (cont)

Function	Pin Name	Description	I/O	Number of Pins
Bus Control	RD$\overline{\text{WR}}$	Read/Write specification	O	1
	$\overline{\text{RD}}$	Read strobe	O	1
	CKE/PTK[5]	CK enable (only for SDRAM)/I/O port K[5]	O/IO	1
	$\overline{\text{WAIT}}$	Hardware wait request	I	1
Interrupt	IRQ[3:0]/$\overline{\text{IRL[3:0]}}$/ PTH[3:0]	External interrupt request/I/O port H[3:0]	I	4
	IRQ4/PTH[4]	External interrupt request / I/O port H[4]	I	1
	NMI	Nonmaskable interrupt request	I	1
	$\overline{\text{IRQOUT}}$	Interrupt request notification	O	1
	$\overline{\text{WAKEUP}}$/PTD[3]	Interrupt request notification in standby mode/I/O port D[3]	O/IO	1
Timer	TCLK/PTH[7]	Clock input for TMU/RTC clock output/ I/O port H[7]	IO	1
DMAC	$\overline{\text{DREQ0}}$/PTD[4]	DMA request 0/Input port D[4]	I	1
	DACK0/PTD[5]	DMA acknowledge 0/I/O port D[5]	O/IO	1
	$\overline{\text{DREQ1}}$/PTD[6]	DMA request 1/Input port D[6]	I	1
	DACK1/PTD[7]	DMA acknowledge 1/I/O port D[7]	O/IO	1
	DRAK0/PTD[1]	DMA request acknowledge 0/I/O port D[1]	O/IO	1
	DRAK1/PTD[0]	DMA request acknowledge 1/I/O port D[0]	O/IO	1
SCI/Smart Card IF without FIFO	RxD0/SCPT[0]	Recieve data 0/Input port for SCI[0]	I	1
	TxD0/SCPT[0]	Transmit data 0/Output port for SCI[0]	O	1
	SCK0/SCPT[1]	Serial clock 0/I/O port for SCI[1]	IO	1
SCIF/IrDA with FIFO	RxD1/SCPT[2]	Recieve data 1/Input port for SCI[2]	I	1
	TxD1/SCPT[2]	Transmit data 1/Output port for SCI[2]	O	1
	SCK1/SCPT[3]	Serial clock 1/I/O port for SCI[3]	IO	1
SCIF with FIFO	RxD2/SCPT[4]	Recieve data 2/Input port for SCI[4]	I	1
	TxD2/SCPT[4]	Transmit data 2/Output port for SCI[4]	O	1
	SCK2/SCPT[5]	Serial clock 2/I/O port for SCI[5]	IO	1
	$\overline{\text{RTS2}}$/SCPT[6]	Request to send 2/I/O port for SCI[6]	O/IO	1

Table 1.3 SH7709 Pin Function (cont)

Function	Pin Name	Description	I/O	Number of Pins
SCIF with FIFO	$\overline{\text{CTS2}}$/IRQ5/SCPT[7]	Clear to send 2/External interrupt request/Input port for SCI[7]	I	1
PCMCIA control	$\overline{\text{CE2B}}$/PTE[5]	Chip ebable 2 for PC card 0/I/O port E[5]	O/IO	1
-PCC0:area 6 -PCC1:area 5	$\overline{\text{IOIS16}}$/PTG[7]	Write protect/I/O is 16 for PC card 0/ Input port G[7]	I	1
	$\overline{\text{CE2A}}$/PTE[4]	Chip ebable 2 for PC card 1/I/O port E[4]	O/IO	1
Clock	CAP[1:2]	External capacitance pin for PLL	–	2
	EXTAL	External clock/crystal oscillator pin	I	1
	XTAL	Crystal oscillator pin	O	1
	CKIO	System clock I/O	IO	1
	EXTAL2	Crystal oscillator pin for onchip RTC	I	1
	XTAL2	Crystal oscillator pin for onchip RTC	O	1
System	$\overline{\text{RESETP}}$	Power on reset request	I	1
Control	$\overline{\text{RESETM}}$	Manual reset request	I	1
	$\overline{\text{BREQ}}$	Bus request	I	1
	$\overline{\text{BACK}}$	Bus acknowledge	O	1
	MD[2:0]	Setting clock mode	I	3
	MD[4:3]	Setting bus width for area 0	I	2
	MD5	Setting endian	I	1
	$\overline{\text{CA}}$	Chip Active	I	1
	STATUS[1:0]/PTJ[7:6]	Processor status/I/O port J[7:6]	O/IO	2
Analog	AN[5:0]/PTL[5:0]	A/D converter input/Input port L[5:0]	I	6
	AN[6:7]/DA[1:0]/ PTL[6:7]	A/D converter input/D/A converter output [1:0]/Input port L[6:7]	I/O	2
Port	PTC[7:0]/PINT[7:0]	I/O port C[7:0]/Port interrupt [7:0]	IO	8
	PTD[2]/ $\overline{\text{RESETOUT}}$	I/O port D[2]/ RESETOUT	IO	1
	PTE[0]/PTE[7]	I/O port E[0], E[7]	IO	2

Table 1.3 SH7709 Pin Function (cont)

Function	Pin Name	Description	I/O	Number of Pins
Port	PTF[7:0]/PINT[15:8]	Input port F[7:0]/Port interrupt [15:8]	I	8
	PTG[6:0]	Input port G[6:0]	I	7
	PTH[5]/$\overline{\text{ADTRG}}$	Input port H[5]/analog trigger	I	1
	PTH[6]	Input port H[6]	I	1
V_{cc}, GND	V_{cc}	Power supply (3.3 V)	–	15
	V_{cc} (PLL)	Power supply (3.3 V)	–	2
	V_{cc} (RTC Oscilator)	Power supply (3.3 V)	–	1
	AV_{cc}	Power supply for analog (3.3 V)	–	1
	V_{ss}	Power supply (0 V)	–	16
	V_{ss} (PLL)	Power supply (0 V)	–	2
	V_{ss} (RTC Oscilator)	Power supply (0 V)	–	1
	AV_{ss}	Power supply for analog (0 V)	–	2
Total				208

Section 2 CPU

2.1 Register Configuration

2.1.1 Privileged Mode and Banks

Processor Modes: There are two processor modes: user mode and privileged mode. The SH7709 normally operates in user mode, and enters privileged mode when an exception occurs or an interrupt is accepted. There are three kinds of registers—general registers, system registers, and control registers—and the registers that can be accessed differ in the two processor modes.

General Registers: There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers which are switched by a processor mode change. 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, BANK1 general registers R0_BANK1–R7_BANK1 and non-banked general registers R8–R15 function as the general register set, with BANK0 general registers R0_BANK0–R7_BANK0 accessed only by the LDC/STC instructions.

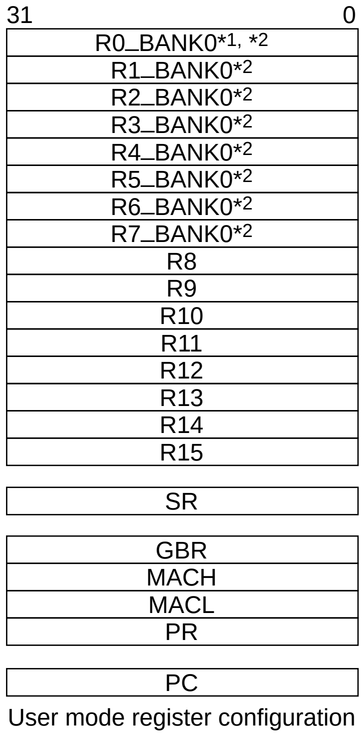
When the RB bit is 0, BANK0 general registers R0_BANK0–R7_BANK0 and nonbanked general registers R8–R15 function as the general register set, with BANK1 general registers R0_BANK1–R7_BANK1 accessed only by the LDC/STC instructions. In user mode, the 16 registers comprising bank 0 general registers R0_BANK0–R7_BANK0 and non-banked registers R8–R15 can be accessed as general registers R0–R15, and bank 1 general registers R0_BANK1–R7_BANK1 cannot be accessed.

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

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

The register configuration in each mode is shown in figures 2.1 and 2.2.

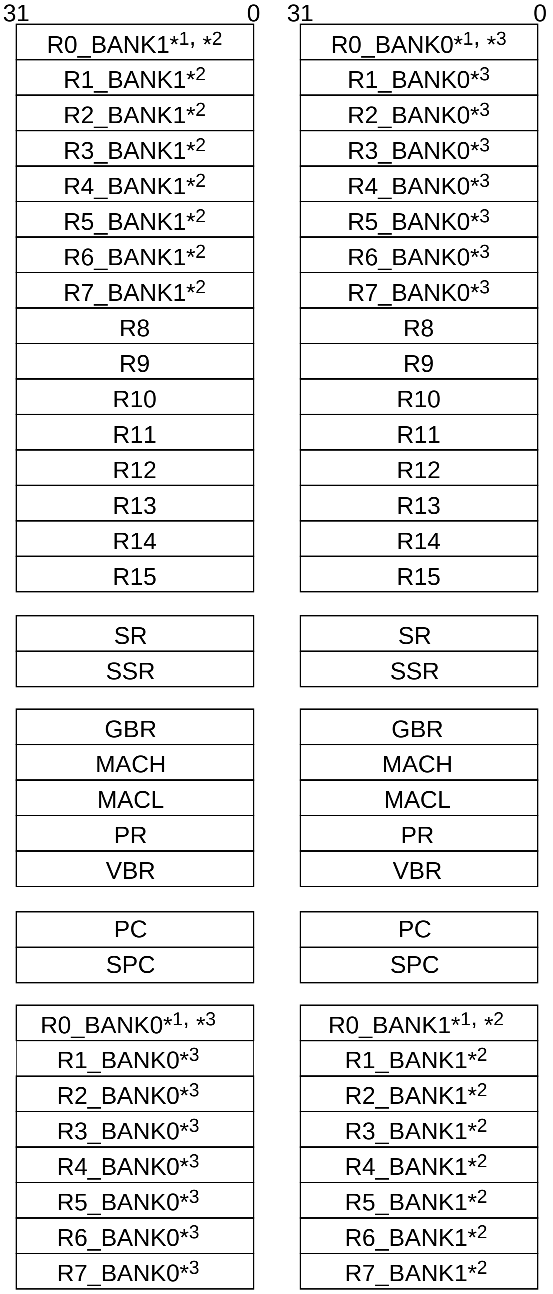
Switching between user mode and privileged mode is controlled by the processor mode bit (MD) in the status register.



Notes: 1. R0 functions as an index register in the indexed register-indirect addressing mode and indexed GBR-indirect addressing mode.

2. Banked register

Figure 2.1 User Mode Register Configuration



a. Privileged mode register configuration (RB = 1)

b. Privileged mode register configuration (RB = 0)

Notes: 1. R0 functions as an index register in the indexed register-indirect addressing mode and indexed GBR-indirect addressing mode.

2. Banked register
When the RB bit of the SR register is 1, the register can be accessed for general use. When the RB bit is 0, it can only be accessed with the LDC/STC instruction.

3. Banked register
When the RB bit of the SR register is 0, the register can be accessed for general use. When the RB bit is 1, it can only be accessed with the LDC/STC instruction.

Figure 2.2 Privileged Mode Register Configuration

Register values after a reset are shown in table 2.1.

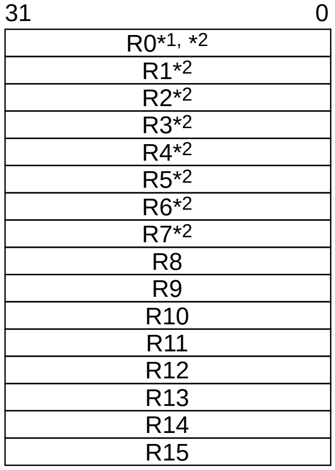
Table 2.1 Initial Register Values

Type	Registers	Initial Value
General registers	R0 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3–I0 = 1111 (H'F), reserved bits = 0, others undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000

2.1.2 General Registers

There are 16 general registers, designated R0 to R15 (figure 2.3). General registers R0 to R7 are banked registers, with a different R0–R7 register bank (R0_BANK0–R7_BANK0 or R0_BANK1–R7_BANK1) being accessed according to the processor mode. For details, see section 2.1.1, Privileged Mode and Banks.

General Registers



Notes:

1. R0 functions as an index register in the indexed register-indirect addressing mode and indexed GBR-indirect addressing mode. In some instructions, only R0 can be used as the source register or destination register.

2. R0–R7 are banked registers.
   In privileged mode, SR.RB specifies which banked registers are accessed as general registers (R0_BANK0–R7_BANK0 or R0_BANK1–R7_BANK1).

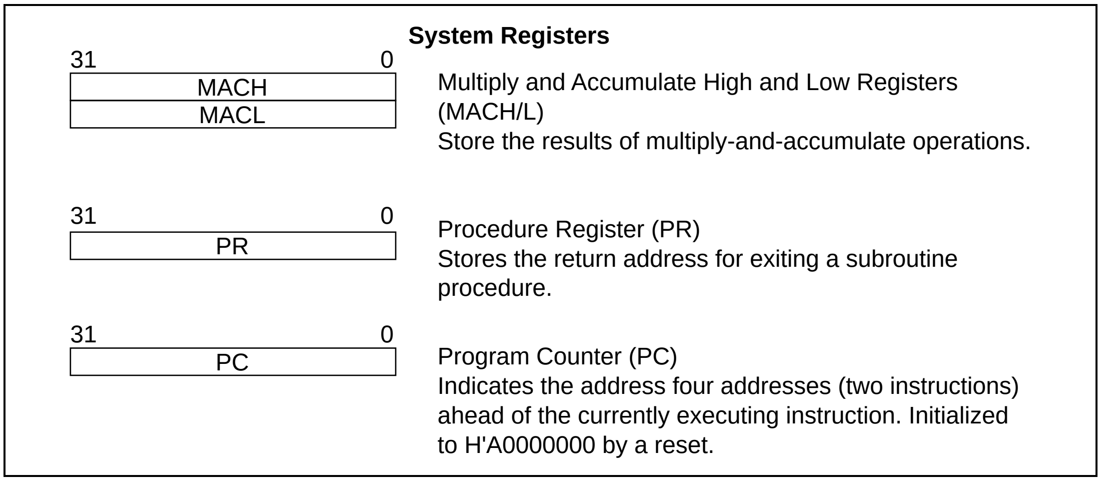
Figure 2.3 General Registers

2.1.3 System Registers

System registers can be accessed by the LDS and STS instructions. When an exception occurs, the contents of the program counter (PC) are saved in the saved program counter (SPC). The SPC contents are restored to the PC by the RTE instruction used at the end of the exception handling. There are four system registers, as follows.

- Multiply and accumulate high register (MACH)
- Multiply and accumulate low register (MACL)
- Procedure register (PR)
- Program counter (PC)

The system register configuration is shown in figure 2.4.



Multiply and Accumulate High and Low Registers (MACH/L)
Store the results of multiply-and-accumulate operations.

Procedure Register (PR)
Stores the return address for exiting a subroutine procedure.

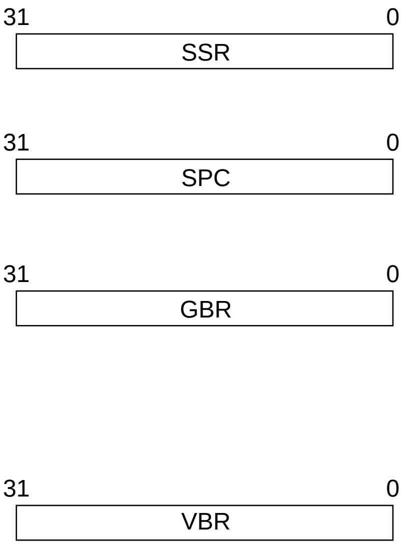
Program Counter (PC)
Indicates the address four addresses (two instructions) ahead of the currently executing instruction. Initialized to H'A0000000 by a reset.

Figure 2.4 System Registers

2.1.4 Control Registers

Control registers can be accessed in privileged mode using the LDC and STC instructions. The GBR register can also be accessed in user mode. There are five control registers, as follows:

- Status register (SR)
- Saved status register (SSR)
- Saved program counter (SPC)
- Global base register (GBR)
- Vector base register (VBR)



Saved Status Register (SSR)
Stores current SR value at time of exception to indicate processor status in return to instruction stream from exception handler.

Saved Program Counter (SPC)
Stores current PC value at time of exception to indicate return address at completion of exception handling.

Global Base Register (GBR)
Stores base address of GBR-indirect addressing mode. The GBR-indirect addressing mode is used for on-chip supporting module register area data transfers and logic operations.
The GBR register can also be accessed in user mode. Its contents are undefined after a reset.

Vector Base Register (VBR)
Stores base address of exception handling vector area. Initialized to H'0000000 by a reset.

Status register (SR)

31	30	29	28	27 … 10	9	8	7 … 4	3	2	1	0
0	MD	RB	BL	0———0	M	Q	I3 I2 I1 I0	0	0	S	T

MD: Processor operation mode bit: Indicates the processor operation mode as follows:
MD =1: Privileged mode; MD = 0: User mode
MD is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a reset.

RB: Register bank bit: Determines the bank of general registers R0–R7 used in processing mode.
RB = 1: R0_BANK1–R7_BANK1 and R8–R15 are general registers, and R0_BANK0–R7_BANK0 can be accessed by LDC/STC instructions.
RB = 0: R0_BANK0–R7_BANK0 and R8–R15 are general registers, and R0_BANK1–R7_BANK1 can be accessed by LDC/STC instructions.
RB is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a reset.

BL: Block bit
BL = 1: Exceptions and interrupts are suppressed. See section 4, Exception Handling, for details.
BL = 0: Exceptions and interrupts are accepted.
BL is set to 1 on generation of an exception or interrupt , and is initialized to 1 by a reset.

M and Q bits: Used by the DIV0S/U and DIV1 instructions.

I3–I0 bits: Interrupt mask bits: 4-bit field indicating the interrupt request mask level.
I3–I0 do not change to the interrupt acceptance level when an interrupt is generated.
Initialized to B'1111 by a reset.

S bit: Used by the MAC instruction.

T bit: Used by the MOVT, CMP/cond, TAS, TST, BT, BF, SETT, CLRT, and DT instructions to indicate true (1) or false (0).
Used by the ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L, and ROTCR/L instructions to indicate a carry, borrow, overflow, or underflow.

0 bits: These bits always read 0, and the write value should always be 0.

Note: The M, Q, S, and T bits can be set or cleared by special instructions in user mode.
Their values are undefined after a reset. All other bits can be read or written in privileged mode.

Figure 2.5 Register Set Overview, Control Registers

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits, figure 2.6). 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.

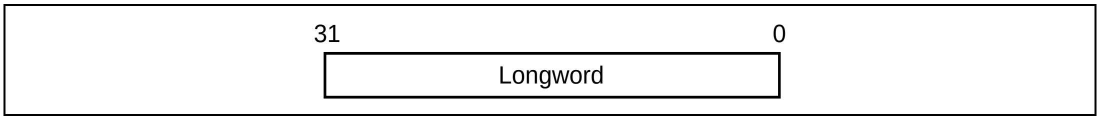


Figure 2.6 Longword

2.2.2 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 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 stored in 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 mode should be set with the MD5 external pin in a power-on reset. Big-endian mode is selected when the MD5 pin is low, and little-endian when high. The endian mode 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. In little-endian mode, data written in byte-size (8-bit) units should be read in byte-size units, and data written in word-size (16-bit) units should be read in word-size units.

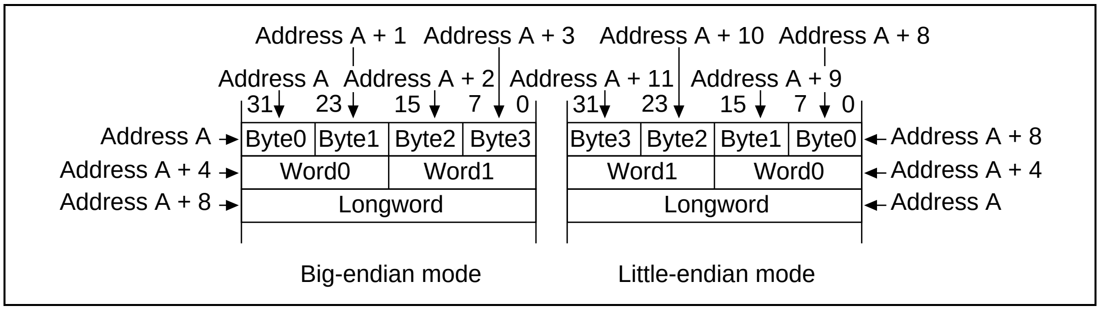


Figure 2.7 Byte, Word, and Longword Alignment

2.3 Instruction Features

2.3.1 Execution Environment

Data Length: The SH7709 instruction set is implemented with fixed-length 16-bit wide instructions executed in a pipelined sequence with single-cycle execution for most instructions. All operations are executed in 32-bit longword units. Memory can be accessed in 8-bit byte, 16-bit word, or 32-bit longword units, with byte or word units sign-extended into 32-bit longwords. Literals are sign-extended in arithmetic operations (MOV, ADD, and CMP/EQ instructions) and zero-extended in logical operations (TST, AND, OR, and XOR instructions).

Load/Store Architecture: The SH7709 features a load-store architecture in which basic operations are executed in registers. Operations requiring memory access are executed in registers following register loading, except for bit-manipulation operations such as logical AND functions, which are executed directly in memory.

Delayed Branching: Unconditional branching is implemented as delayed branch operations. Pipeline disruptions due to branching are minimized by the execution of the instruction following the delayed branch instruction prior to branching. Conditional branch instructions are of two kinds, delayed and normal.

```
BRA     TRGET
ADD     R1, R0    ;ADD is executed prior to branching to TRGET
```

T bit: The T bit in the status register (SR) is used to indicate the result of compare operations, and is read as a TRUE/FALSE condition determining if a conditional branch is taken or not. To improve processing speed, the T bit logic state is modified only by specific operations. An example of how the T bit may be used in a sequence of operations is shown below.

```
ADD     #1, R0    ;T bit not modified by ADD operation
CMP/EQ  R1, R0    ;T bit set to 1 when R0 = 0
BT      TRGET     ;branch taken to TRGET when T bit = 1 (R0 = 0)
```

Literals: Byte-length literals are inserted directly into the instruction code as immediate data. To maintain the 16-bit fixed-length instruction code, word or longword literals are stored in a table in main memory rather than inserted directly into the instruction code. The memory table is accessed by the MOV instruction using PC-relative addressing with displacement, as follows:

```
MOV.W   @(disp, PC), R0
```

Absolute Addresses: As with word and longword literals, absolute addresses must also be stored in a table in main memory. The value of the absolute address is transferred to a register and the operand access is specified by indexed register-indirect addressing, with the absolute address loaded (like word and longword immediate data) during instruction execution.

16-Bit and 32-Bit Displacements: In the same way, 16-bit and 32-bit displacements also must be stored in a table in main memory. Exactly like absolute addresses, the displacement value is transferred to a register and the operand access is specified by indexed register-indirect addressing, loading the displacement (like word and longword immediate data) during instruction execution.

2.3.2 Addressing Modes

Addressing modes and effective address calculation methods are shown in table 2.2.

Table 2.2 Addressing Modes and Effective Addresses

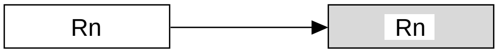
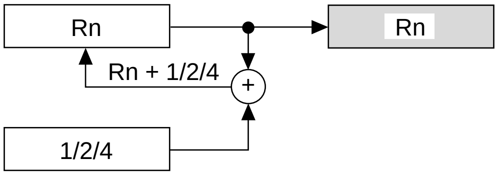
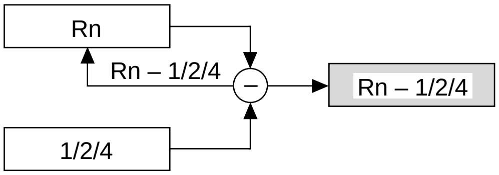
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective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

Table 2.2 Addressing Modes and Effective Addresses (cont)

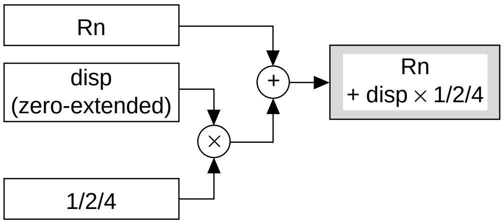
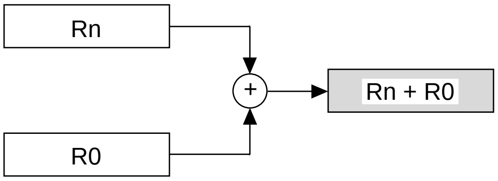
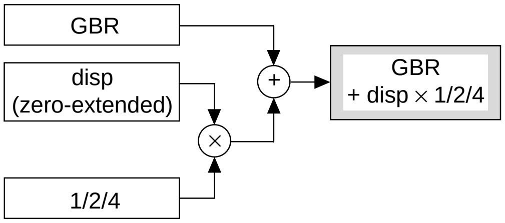
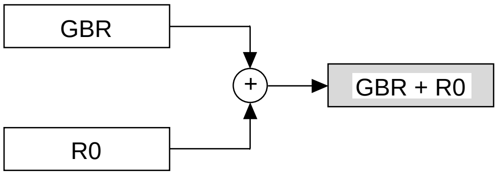
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword),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

Table 2.2 Addressing Modes and Effective Addresses (cont)

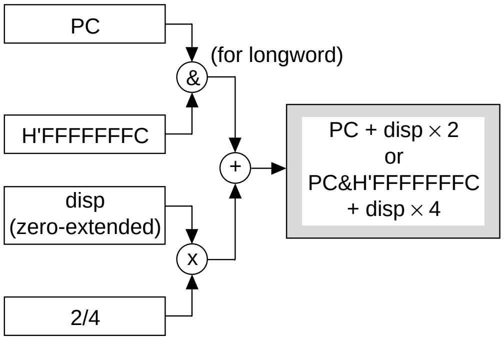
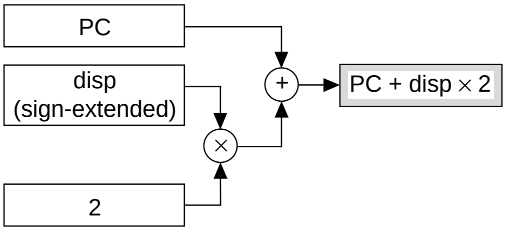
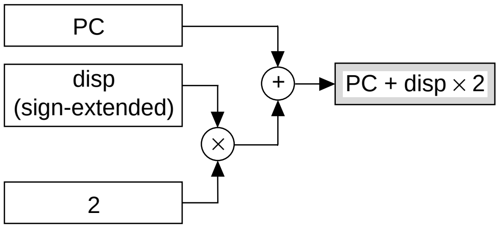
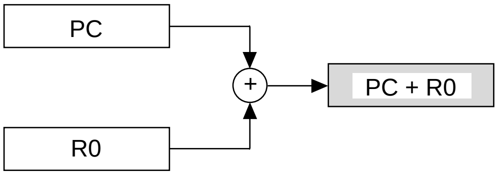
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is register PC contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked.  PC (for longword) & H'FFFFFFFC + disp (zero-extended) x 2/4 PC + disp × 2 or PC&H'FFFFFFFC + disp × 4	Word: PC + disp × 2 Longword: PC & H'FFFF FFFC + disp × 4
PC-relative	disp:8	Effective address is register PC contents with 8-bit displacement disp added after being sign-extended and multiplied by 2.  	PC + disp × 2
	disp:12	Effective address is register PC contents with 12-bit displacement disp added after being sign-extended and multiplied by 2.  	PC + disp × 2

Table 2.2 Addressing Modes and Effective Addresses (cont)

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

Note: For the addressing modes below that use a displacement (disp), the assembler descriptions in this manual show the value before scaling (x1, x2, or x4) is performed according to the operand size. This is done to clarify the operation of the IC. Refer to the relevant assembler notation rules for the actual assembler descriptions.

@ (disp:4, Rn) ; Register indirect with displacement

@ (disp:8, Rn) ; GBR indirect with displacement

@ (disp:8, PC) ; PC-relative with displacement

disp:8, disp:12; PC-relative

2.3.3 Instruction Formats

Table 2.3 explains the meaning of instruction formats and source and destination operands. The meaning of the operands depends on the operation code. The following symbols are used.

xxxx:	Operation code
mmmm:	Source register
nnnn:	Destination register
iiii:	Immediate data
dddd:	Displacement

Table 2.3 Instruction Formats

Instruction Format		Source Operand	Destination Operand	Instruction Example
0 format	15 … 0 xxxx xxxx xxxx xxxx	—	—	NOP
n format	15 … 0 xxxx nnnn xxxx xxxx	—	nnnn: register direct	MOVT Rn
		Control register or system register	nnnn: register direct	STS MACH,Rn
		Control register or system register	nnnn: register indirect with pre-decrement	STC.L SR,@–Rn
m format	15 … 0 xxxx mmmm xxxx xxxx	mmmm: register direct	Control register or system register	LDC Rm,SR
		mmmm: register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
		mmmm: register indirect	—	JMP @Rm
		mmmm: PC-relative using Rm	—	BRAF Rm

Table 2.3 Instruction Formats (cont)

Instruction Format	Source Operand	Destination Operand	Instruction Example
nm format 15 0 xxxx nnnn mmmm xxxx	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: register indirect with post-increment (multiply-and-accumulate operation) nnnn: * register indirect with post-increment (multiply-and-accumulate operation)	MACH,MACL	MAC.W @Rm+,@Rn+
	mmmm: register indirect with post-increment	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: register indirect with pre-decrement	MOV.L Rm,@–Rn
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 15 0 xxxx xxxx mmmm dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 format 15 0 xxxx xxxx nnnn dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)

Table 2.3 Instruction Formats (cont)

Instruction Format		Source Operand	Destination Operand	Instruction Example
nmd format	15 0 xxxx nnnn mmmm dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
		mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn
d format	15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
		R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
		dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
		dddddddd: PC-relative	—	BF label
d12 format	15 0 xxxx dddd dddd dddd	dddddddddddd: PC-relative	—	BRA label (label = disp + PC)
nd8 format	15 0 xxxx nnnn dddd dddd	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i format	15 0 xxxx xxxx iiii iiii	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm, @(R0,GBR)
		iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
		iiiiiiii: immediate	—	TRAPA #imm
ni format	15 0 xxxx nnnn iiii iiii	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

Note: In a multiply-and-accumulate instruction, nnnn is the source register.

2.4 Instruction Set

2.4.1 Instruction Set Classified by Function

The SH7709 instruction set includes 68 basic instruction types, as listed in table 2.4.

Table 2.4 Classification of Instructions

Classificatio n	Types	Operation Code	Function	No. of Instructions
Data transfer	5	MOV	Data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate operation, double-precision multiply-and-accumulate operation	

Table 2.4 Classification of Instructions (cont)

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	21	MUL	Double-precision multiplication (32 × 32 bits)	33
(cont)		MULS	Signed multiplication (16 × 16 bits)	
		MULU	Unsigned multiplication (16 × 16 bits)	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow check	
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
		SHAD	Dynamic arithmetic shift	
		SHLD	Dynamic logical shift	

Table 2.4 Classification of Instructions (cont)

Classification	Types	Operation Code	Function	No. of Instructions
Branch	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control	15	CLRT	T bit clear	75
		CLRMAC	MAC register clear	
		CLRS	Clear S bit	
		LDC	Load to control register	
		LDS	Load to system register	
		LDTLB	Load PTE to TLB	
		NOP	No operation	
		PREF	Prefetch data to cache	
		RTE	Return from exception handling	
		SETS	Set S bit	
		SETT	Set T bit	
		SLEEP	Shift to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			188

Table 2.5 lists the SH7709 instruction code formats.

Table 2.5 Instruction Code Format

Item	Format	Explanation
Instruction mnemonic	OP.Sz SRC,DEST	OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 .... 1111: R15 iiii: Immediate data dddd: Displacement
Operation summary	→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Direction of transfer Memory operand Flag bits in SR Logical AND of each bit Logical OR of each bit Exclusive OR of each bit Logical NOT of each bit n-bit shift
Privileged mode		Indicates whether privileged mode applies
Execution cycles		Value when no wait states are inserted The execution cycles listed in the table are minimums. The actual number of cycles may be increased in cases such as the followsing: 1. When contention occurs between instruction fetches and data access 2. When the destination register of the load instruction (memory → register) and the register used by the next instruction are the same
T bit		Value of T bit after instruction is executed —: No change

Note: Scaling (×1, ×2, ×4) is performed according to the instruction operand size.

Table 2.6 lists the SH7709 data transfer instructions

Table 2.6 Data Transfer Instructions

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

Table 2.6 Data Transfer Instructions (cont)

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	1	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	1	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	—	1	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	—	1	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	1	—
MOV.B	R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1	—
MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1	—
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1	—
MOV.B	@(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1	—
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1	—
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1	—
MOVA	@(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1	—
MOVT	Rn	T → Rn	0000nnnn00101001	—	1	—
SWAP.B	Rm,Rn	Rm → Swap the bottom two bytes → REG	0110nnnnmmmm1000	—	1	—
SWAP.W	Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	—	1	—
XTRCT	Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	—	1	—

Table 2.7 lists the SH7709 arithmetic instructions.

Table 2.7 Arithmetic Instructions

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

Table 2.7 Arithmetic Instructions (cont)

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

Table 2.7 Arithmetic Instructions (cont)

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
NEG	Rm,Rn	0–Rm → Rn	0110nnnnmmmm1011	—	1	—
NEGC	Rm,Rn	0–Rm–T → Rn, Borrow → T	0110nnnnmmmm1010	—	1	Borrow
SUB	Rm,Rn	Rn–Rm → Rn	0011nnnnmmmm1000	—	1	—
SUBC	Rm,Rn	Rn–Rm–T → Rn, Borrow → T	0011nnnnmmmm1010	—	1	Borrow
SUBV	Rm,Rn	Rn–Rm → Rn, Underflow → T	0011nnnnmmmm1011	—	1	Underflow

Note: The normal number of execution cycles is shown. The value in parentheses is the number of cycles required in case of contention with the preceding or following instruction.

Table 2.8 lists the SH7709 logic operation instructions.

Table 2.8 Logic Operation Instructions

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

Table 2.9 lists the SH7709 shift instructions.

Table 2.9 Shift Instructions

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

Table 2.10 lists the SH7709 branch instructions.

Table 2.10 Branch Instructions

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

Note: One state when there is no branch.

Table 2.11 lists the SH7709 system control instructions.

Table 2.11 System Control Instructions

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
`CLRMAC`	0 → MACH, MACL	`0000000000101000`	—	1	—
`CLRS`	0 → S	`0000000001001000`	—	1	—
`CLRT`	0 → T	`0000000000001000`	—	1	0
`LDC Rm,SR`	Rm → SR	`0100mmmm00001110`	Ã	5	LSB
`LDC Rm,GBR`	Rm → GBR	`0100mmmm00011110`	—	1	—
`LDC Rm,VBR`	Rm → VBR	`0100mmmm00101110`	Ã	1	—
`LDC Rm,SSR`	Rm → SSR	`0100mmmm00111110`	Ã	1	—
`LDC Rm,SPC`	Rm → SPC	`0100mmmm01001110`	Ã	1	—
`LDC Rm,R0_BANK`	Rm → R0_BANK	`0100mmmm10001110`	Ã	1	—
`LDC Rm,R1_BANK`	Rm → R1_BANK	`0100mmmm10011110`	Ã	1	—
`LDC Rm,R2_BANK`	Rm → R2_BANK	`0100mmmm10101110`	Ã	1	—
`LDC Rm,R3_BANK`	Rm → R3_BANK	`0100mmmm10111110`	Ã	1	—
`LDC Rm,R4_BANK`	Rm → R4_BANK	`0100mmmm11001110`	Ã	1	—
`LDC Rm,R5_BANK`	Rm → R5_BANK	`0100mmmm11011110`	Ã	1	—
`LDC Rm,R6_BANK`	Rm → R6_BANK	`0100mmmm11101110`	Ã	1	—
`LDC Rm,R7_BANK`	Rm → R7_BANK	`0100mmmm11111110`	Ã	1	—
`LDC.L @Rm+,SR`	(Rm) → SR, Rm + 4 → Rm	`0100mmmm00000111`	Ã	7	LSB
`LDC.L @Rm+,GBR`	(Rm) → GBR, Rm + 4 → Rm	`0100mmmm00010111`	—	1	—
`LDC.L @Rm+,VBR`	(Rm) → VBR, Rm + 4 → Rm	`0100mmmm00100111`	Ã	1	—
`LDC.L @Rm+,SSR`	(Rm) → SSR, Rm + 4 → Rm	`0100mmmm00110111`	Ã	1	—
`LDC.L @Rm+,SPC`	(Rm) → SPC, Rm + 4 → Rm	`0100mmmm01000111`	Ã	1	—
`LDC.L @Rm+, R0_BANK`	(Rm) → R0_BANK, Rm + 4 → Rm	`0100mmmm10000111`	Ã	1	—
`LDC.L @Rm+, R1_BANK`	(Rm) → R1_BANK, Rm + 4 → Rm	`0100mmmm10010111`	Ã	1	—
`LDC.L @Rm+, R2_BANK`	(Rm) → R2_BANK, Rm + 4 → Rm	`0100mmmm10100111`	Ã	1	—
`LDC.L @Rm+, R3_BANK`	(Rm) → R3_BANK, Rm + 4 → Rm	`0100mmmm10110111`	Ã	1	—

Table 2.11 System Control Instructions (cont)

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

Note: The number of cycles until the sleep state is entered.

Table 2.11 System Control Instructions (cont)

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
`STC R4_BANK,Rn`	R4_BANK→ Rn	`0000nnnn11000010`	Ã	1	—
`STC R5_BANK,Rn`	R5_BANK→ Rn	`0000nnnn11010010`	Ã	1	—
`STC R6_BANK,Rn`	R6_BANK→ Rn	`0000nnnn11100010`	Ã	1	—
`STC R7_BANK,Rn`	R7_BANK→ Rn	`0000nnnn11110010`	Ã	1	—
`STC.L SR,@–Rn`	Rn–4 → Rn, SR → (Rn)	`0100nnnn00000011`	Ã	1	—
`STC.L GBR,@–Rn`	Rn–4 → Rn, GBR → (Rn)	`0100nnnn00010011`	—	1	—
`STC.L VBR,@–Rn`	Rn–4 → Rn, VBR → (Rn)	`0100nnnn00100011`	Ã	1	—
`STC.L SSR,@–Rn`	Rn–4 → Rn, SSR → (Rn)	`0100nnnn00110011`	Ã	1	—
`STC.L SPC,@–Rn`	Rn–4 → Rn, SPC → (Rn)	`0100nnnn01000011`	Ã	1	—
`STC.L R0_BANK, @–Rn`	Rn–4 → Rn, R0_BANK → (Rn)	`0100nnnn10000011`	Ã	2	—
`STC.L R1_BANK, @–Rn`	Rn–4 → Rn, R1_BANK → (Rn)	`0100nnnn10010011`	Ã	2	—
`STC.L R2_BANK, @–Rn`	Rn–4 → Rn, R2_BANK → (Rn)	`0100nnnn10100011`	Ã	2	—
`STC.L R3_BANK, @–Rn`	Rn–4 → Rn, R3_BANK → (Rn)	`0100nnnn10110011`	Ã	2	—
`STC.L R4_BANK, @–Rn`	Rn–4 → Rn, R4_BANK → (Rn)	`0100nnnn11000011`	Ã	2	—
`STC.L R5_BANK, @–Rn`	Rn–4 → Rn, R5_BANK → (Rn)	`0100nnnn11010011`	Ã	2	—
`STC.L R6_BANK, @–Rn`	Rn–4 → Rn, R6_BANK → (Rn)	`0100nnnn11100011`	Ã	2	—
`STC.L R7_BANK, @–Rn`	Rn–4 → Rn, R7_BANK → (Rn)	`0100nnnn11110011`	Ã	2	—
`STS MACH,Rn`	MACH → Rn	`0000nnnn00001010`	—	1	—
`STS MACL,Rn`	MACL → Rn	`0000nnnn00011010`	—	1	—
`STS PR,Rn`	PR → Rn	`0000nnnn00101010`	—	1	—
`STS.L MACH,@–Rn`	Rn–4 → Rn, MACH → (Rn)	`0100nnnn00000010`	—	1	—
`STS.L MACL,@–Rn`	Rn–4 → Rn, MACL → (Rn)	`0100nnnn00010010`	—	1	—
`STS.L PR,@–Rn`	Rn–4 → Rn, PR → (Rn)	`0100nnnn00100010`	—	1	—
`TRAPA #imm`	PC → SPC, SR → SSR, imm → TRA	`11000011iiiiiiii`	—	6	—

Notes: 1. The table shows the minimum number of execution cycles. The actual number of instruction execution cycles will increase in cases such as the following:

- When there is contention between an instruction fetch and data access
- When the destination register in a load (memory-to-register) instruction is also used by the next instruction

2. With the addressing modes using displacement (disp) listed below, the assembler descriptions in this manual show the value before scaling (×1, ×2, or ×4) is performed. This is done to clarify the operation of the chip. For the actual assembler descriptions, refer to the individual assembler notation rules.

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

2.4.2 Instruction Code Map

Table 2.12 shows the instruction code map.

Table 2.12 Instruction Code Map

Instruction Code				Fx: 0000		Fx: 0001		Fx: 0010		Fx: 0011 to 1111	
MSB			LSB	MD: 00		MD: 01		MD: 10		MD: 11	
0000	Rn	Fx	0000								
0000	Rn	Fx	0001								
0000	Rn	00MD	0010	STC	SR,Rn	STC GBR,Rn		STC VBR,Rn		STC SSR,Rn	
0000	Rn	01MD	0010	STC	SPC,Rn						
0000	Rn	10MD	0010	STC	R0_BANK,Rn	STC	R1_BANK,Rn	STC	R2_BANK,Rn	STC	R3_BANK,Rn
0000	Rn	11MD	0010	STC	R4_BANK,Rn	STC	R5_BANK,Rn	STC	R6_BANK,Rn	STC	R7_BANK,Rn
0000	Rm	00MD	0011	BSRF	Rm			BRAF	Rm		
0000	Rn	10MD	0011	PREF	@Rn						
0000	Rn	Rm	01MD	MOV.B	Rm,@(R0,Rn)	MOV.W	Rm,@(R0,Rn)	MOV.L	Rm,@(R0,Rn)	MUL.L	Rm,Rn
0000	0000	00MD	1000	CLRT		SETT		CLRMAC		LDTLB	
0000	0000	01MD	1000	CLRS		SETS					
0000	0000	Fx	1001	NOP		DIV0U					
0000	0000	Fx	1010								
0000	0000	Fx	1011	RTS		SLEEP		RTE			
0000	Rn	Fx	1000								
0000	Rn	Fx	1001					MOVT	Rn		
0000	Rn	Fx	1010	STS	MACH,Rn	STS	MACL,Rn	STS	PR,Rn		
0000	Rn	Fx	1011								
0000	Rn	Rm	11MD	MOV.B	@(R0,Rm),Rn	MOV.W	@(R0,Rm),Rn	MOV.L	@(R0,Rm),Rn	MAC.L	@Rm+,@Rn+
0001	Rn	Rm	disp								
0010	Rn	Rm	00MD	MOV.B	Rm,@Rn	MOV.W	Rm,@Rn	MOV.L	Rm,@Rn		
0010	Rn	Rm	01MD	MOV.B	Rm,@-Rn	MOV.W	Rm,@-Rn	MOV.L	Rm,@-Rn	DIV0S	Rm,Rn
0010	Rn	Rm	10MD	TST	Rm,Rn	AND	Rm,Rn	XOR	Rm,Rn	OR	Rm,Rn
0010	Rn	Rm	11MD	CMP/STR	Rm,Rn	XTRCT	Rm,Rn	MULU.W	Rm,Rn	MULSW	Rm,Rn
0011	Rn	Rm	00MD	CMP/EQ	Rm,Rn			CMP/HS	Rm,Rn	CMP/GE	Rm,Rn
0011	Rn	Rm	01MD	DIV1	Rm,Rn	DMULU.L	Rm,Rn	CMP/HI	Rm,Rn	CMP/GT	Rm,Rn
0011	Rn	Rm	10MD	SUB	Rm,Rn			SUBC	Rm,Rn	SUBV	Rm,Rn
0011	Rn	Rm	11MD	ADD	Rm,Rn	DMULS.L	Rm,Rn	ADDC	Rm,Rn	ADDV	Rm,Rn

Table 2.12 Instruction Code Map (cont)

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
0100	Rn	Fx	0000	SHLL Rn	DT Rn	SHAL Rn	
0100	Rn	Fx	0001	SHLR Rn	CMP/PZ Rn	SHAR Rn	
0100	Rn	Fx	0010	STS.L MACH,@-Rn	STS.L MACL,@-Rn	STS.L PR,@-Rn	
0100	Rn	00MD	0011	STC.L SR,@-Rn	STC.L GBR,@-Rn	STC.L VBR,@-Rn	STC.L SSR,@-Rn
0100	Rn	01MD	0011	STC.L SPC,@-Rn			
0100	Rn	10MD	0011	STC.L R0_BANK,@-Rn	STC.L R1_BANK,@-Rn	STC.L R2_BANK,@-Rn	STC.L R3_BANK,@-Rn
0100	Rn	11MD	0011	STC.L R4_BANK,@-Rn	STC.L R5_BANK,@-Rn	STC.L R6_BANK,@-Rn	STC.L R7_BANK,@-Rn
0100	Rn	Fx	0100	ROTL Rn		ROTCL Rn	
0100	Rn	Fx	0101	ROTR Rn	CMP/PL Rn	ROTCR Rn	
0100	Rm	Fx	0110	LDS.L @Rm+,MACH	LDS.L @Rm+,MACL	LDS.L @Rm+,PR	
0100	Rm	00MD	0111	LDC.L @Rm+,SR	LDC.L @Rm+,GBR	LDC.L @Rm+,VBR	LDC.L @Rm+,SSR
0100	Rm	01MD	0111	LDC.L @Rm+,SPC			
0100	Rm	10MD	0111	LDC.L @Rm+,R0_BANK	LDC.L @Rm+,R1_BANK	LDC.L @Rm+,R2_BANK	LDC.L @Rm+,R3_BANK
0100	Rm	11MD	0111	LDC.L @Rm+,R4_BANK	LDC.L @Rm+,R5_BANK	LDC.L @Rm+,R6_BANK	LDC.L @Rm+,R7_BANK
0100	Rn	Fx	1000	SHLL2 Rn	SHLL8 Rn	SHLL16 Rn	
0100	Rn	Fx	1001	SHLR2 Rn	SHLR8 Rn	SHLR16 Rn	
0100	Rm	Fx	1010	LDS Rm,MACH	LDS Rm,MACL	LDS Rm,PR	
0100	Rm/Rn	Fx	1011	JSR @Rm	TAS.B @Rn	JMP @Rm	
0100	Rn	Rm	1100	SHAD Rm,Rn			
0100	Rn	Rm	1101	SHLD Rm,Rn			
0100	Rm	00MD	1110	LDC Rm,SR	LDC Rm,GBR	LDC Rm,VBR	LDC Rm,SSR
0100	Rm	01MD	1110	LDC Rm,SPC			
0100	Rm	10MD	1110	LDC Rm,R0_BANK	LDC Rm,R1_BANK	LDC Rm,R2_BANK	LDC Rm,R3_BANK
0100	Rm	11MD	1110	LDC Rm,R4_BANK	LDC Rm,R5_BANK	LDC Rm,R6_BANK	LDC Rm,R7_BANK
0100	Rn	Rm	1111	MAC.W @Rm+,@Rn+			
0101	Rn	Rm	disp	MOV.L @(disp:4,Rm),Rn			
0110	Rn	Rm	00MD	MOV.B @Rm,Rn	MOV.W @Rm,Rn	MOV.L @Rm,Rn	MOV Rm,Rn
0110	Rn	Rm	01MD	MOV.B @Rm+,Rn	MOV.W @Rm+,Rn	MOV.L @Rm+,Rn	NOT Rm,Rn
0110	Rn	Rm	10MD	SWAP.B Rm,Rn	SWAP.W Rm,Rn	NEGC Rm,Rn	NEG Rm,Rn
0110	Rn	Rm	11MD	EXTU.B Rm,Rn	EXTU.W Rm,Rn	EXTS.B Rm,Rn	EXTS.W Rm,Rn
0111	Rn	imm		ADD #imm:8,Rn			

Table 2.12 Instruction Code Map (cont)

Instruction Code				Fx: 0000	Fx: 0001	Fx: 0010	Fx: 0011 to 1111
MSB			LSB	MD: 00	MD: 01	MD: 10	MD: 11
1000	00MD	Rn	disp	MOV.B R0,@(disp:4,Rn)	MOV.W R0,@(disp:4,Rn)		
1000	01MD	Rm	disp	MOV.B @(disp:4,Rm),R0	MOV.W @(disp:4,Rm),R0		
1000	10MD	imm/disp		CMP/EQ #imm:8,R0	BT label:8		BF label:8
1000	11MD	imm/disp			BT/S label:8		BF/S label:8
1001	Rn	disp		MOV.W @(DISP:8,PC),RN			
1010	disp			BRA label:12			
1011	disp			BSR label:12			
1100	00MD	imm/disp		MOV.B R0,@(disp:8,GBR)	MOV.W R0,@(disp:8,GBR)	MOV.L R0,@(disp:8,GBR)	TRAPA #imm:8
1100	01MD	disp		MOV.B @(disp:8,GBR),R0	MOV.W @(disp:8,GBR),R0	MOV.L @(disp:8,GBR),R0	MOVA @(disp:8,PC),R0
1100	10MD	imm		TST #imm:8,R0	AND #imm:8,R0	XOR #imm:8,R0	OR #imm:8,R0
1100	11MD	imm		TST.B #imm:8,@(R0,GBR)	AND.B #imm:8,@(R0,GBR)	XOR.B #imm:8,@(R0,GBR)	OR.B #imm:8,@(R0,GBR)
1101	Rn	disp		MOV.L @(disp:8,PC),Rn			
1110	Rn	imm		MOV #imm:8,Rn			
1111	************						

Note: See the SH-3/SH-3E Programming Manual for details.

2.5 Processor States and Processor Modes

2.5.1 Processor States

The SH7709 has five processor states: the reset state, exception-handling state, bus-released state, program execution state, and power-down state.

Reset State: In this state the CPU is reset. The reset state is entered when the $\overline{\text{RESET}}$ pin goes low. The CPU enters the power-on reset state if the $\overline{\text{RESETP}}$ pin is low, or the manual reset state if the $\overline{\text{RESETM}}$ pin is low. See section 4, Exception Handling, for more information on resets.

In the power-on reset state, the internal states of the CPU and the on-chip supporting module registers are initialized. In the manual reset state, the internal states of the CPU and registers of on-chip supporting modules other than the bus state controller (BSC) are initialized. Since the BSC is not initialized in the manual reset state, refreshing operations continue. Refer to the register configurations in the relevant sections for further details.

Exception-Handling State: This is a transient state during which the CPU’s processor state flow is altered by a reset, general exception, or interrupt exception handling.

In the case of a reset, the CPU branches to address H'A0000000 and starts executing the user-coded exception handling program.

In the case of a general exception or interrupt, the program counter (PC) contents are saved in the saved program counter (SPC) and the status register (SR) contents are saved in the saved status register (SSR). The CPU branches to the start address of the user-coded exception service routine found from the sum of the contents of the vector base address and the vector offset. See section 4, Exception Processing, for more information on resets, general exceptions, and interrupts.

Program Execution State: In this state the CPU executes program instructions in sequence.

Power-Down State: In the power-down state, CPU operation halts and power consumption is reduced. There are three modes in the power-down state: sleep mode, standby mode and hardware standby mode. See section 8, Power-Down Modes, for more information.

Bus-Released State: In this state the CPU has released the bus to a device that requested it.

Transitions between the states are shown in figure 2.8.

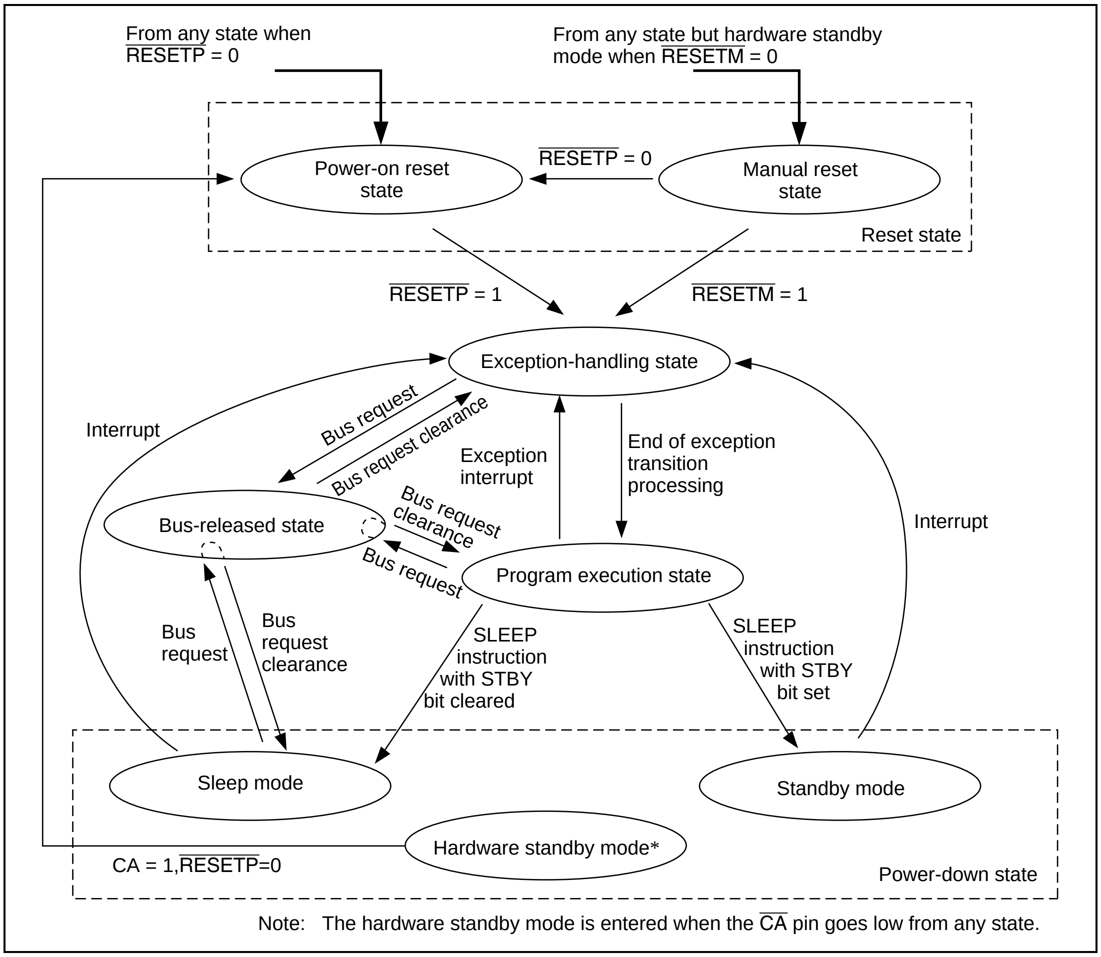

From any state when
$\overline{\text{RESETP}}$ = 0
From any state but hardware standby
mode when $\overline{\text{RESETM}}$ = 0
Power-on reset state
$\overline{\text{RESETP}}$ = 0
Manual reset state
Reset state
$\overline{\text{RESETP}}$ = 1
$\overline{\text{RESETM}}$ = 1
Exception-handling state
Interrupt
Bus request
Bus request clearance
Exception interrupt
End of exception transition processing
Bus-released state
Bus request clearance
Bus request
Program execution state
Interrupt
Bus request
Bus request clearance
SLEEP instruction with STBY bit cleared
SLEEP instruction with STBY bit set
Sleep mode
Standby mode
Hardware standby mode*
CA = 1,$\overline{\text{RESETP}}$=0
Power-down state
Note: The hardware standby mode is entered when the $\overline{\text{CA}}$ pin goes low from any state.

Figure 2.8 Processor State Transitions

2.5.2 Processor Modes

There are two processor modes: privileged mode and user mode. The processor mode is determined by the processor mode bit (MD) in the status register (SR). User mode is selected when the MD bit is 0, and privileged mode when the MD bit is 1. When the reset state or exception state is entered, the MD bit is set to 1. When exception handling ends, the MD bit is cleared to 0 and user mode is entered. There are certain registers and bits which can only be accessed in privileged mode.

Section 3 Memory Management Unit (MMU)

3.1 Overview

3.1.1 Features

The SH7709 has an on-chip memory management unit (MMU) that implements address translation. The SH7709 features a resident translation lookaside buffer (TLB) that caches information for user-created address translation tables located in external memory. It enables high-speed translation of virtual addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 kbyte and 4 kbytes). The access right to virtual address space can be set for privileged and user modes to provide memory protection.

3.1.2 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown in figure 3.1, if a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory. However, if the process increases in size to the extent that it no longer fits into physical memory, it becomes necessary to partition the process and to map those parts requiring execution onto memory as occasion demands ((1)). Having the process itself consider this mapping onto physical memory would impose a large burden on the process. To lighten this burden, the idea of virtual memory was born as a means of performing en bloc mapping onto physical memory ((2)). In a virtual memory system, substantially more virtual memory than physical memory is provided, and the process is mapped onto this virtual memory. Thus a process only has to consider operation in virtual memory. Mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally controlled by the operating system, switching physical memory to allow the virtual memory required by a process to be mapped onto physical memory in a smooth fashion. Switching of physical memory is carried out via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in a time-sharing system (TSS) in which a number of processes are running simultaneously ((3)). If processes running in a TSS had to take mapping onto virtual memory into consideration while running, it would not be possible to increase efficiency. Virtual memory is thus used to reduce this load on the individual processes and so improve efficiency ((4)). In the virtual memory system, virtual memory is allocated to each process. The task of the MMU is to perform efficient mapping of these virtual memory areas onto physical memory. It also has a memory protection feature that prevents one process from inadvertently accessing another process’s physical memory.

When address translation from virtual memory to physical memory is performed using the MMU, it may happen that the relevant translation information is not recorded in the MMU, with the result that one process may inadvertently access the virtual memory allocated to another process. In this case, the MMU will generate an exception, change the physical memory mapping, and record the new address translation information.

Although the functions of the MMU could also be implemented by software alone, the need for translation to be performed by software each time a process accesses physical memory would result in poor efficiency. For this reason, a buffer for address translation (translation lookaside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails—that is, if an exception is generated—switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

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

In the following text, SH7709 address space in virtual memory is referred to as virtual address space, and address space in physical memory as physical memory space.

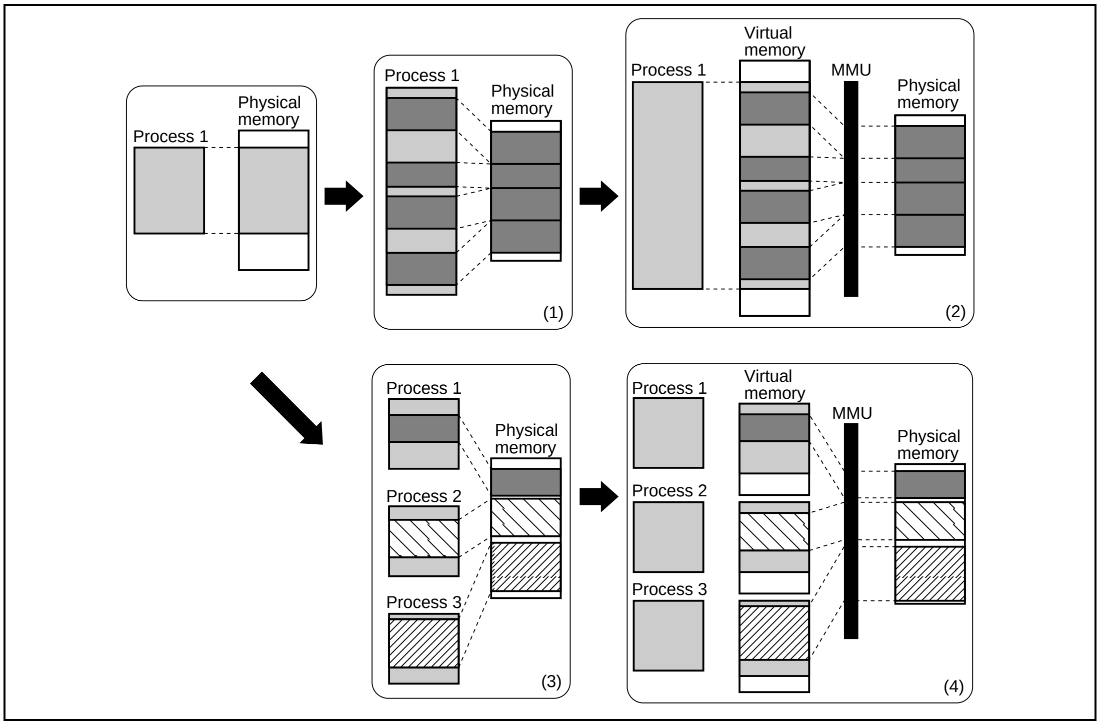


Figure 3.1 MMU Functions

3.1.3 Virtual Address Space

Virtual Address Map: The SH7709 uses 32-bit virtual addresses to access a 4-Gbyte virtual address space that is divided into several areas. Address space mapping is shown in figure 3.2.

In privileged mode, there are five areas, P0–P4. The P0 and P3 areas are mapped onto physical address space in page units, in accordance with address translation table information. Addresses H'7F000000–H'7FFFFFFF in the P0 area can be used as on-chip RAM space by making a setting in the cache control register (CCR) (see section 5, Cache). In this case, mapping by means of the address translation table is not performed for the on-chip RAM space. Copy-back or write-through can be selected for write access by means of a CCR setting.

Mapping of the P1 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P1 area, setting a virtual address MSBs (bit 31) to 0 generates the corresponding physical address. P1 area access can be cached, and the cache control register (CCR) is set to indicate whether to cache or not. Write access is processed as write-through.

Mapping of the P2 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P2 area, setting the top three virtual address bits (bits 31, 30, and 29) to 0 generates the corresponding physical address. P2 area access cannot be cached.

The P1 and P2 areas are not mapped by the address translation table, so the TLB is not used and no exceptions like TLB misses occur. Initialization of MMU-related registers, exception processing handling, and the like are located in the P1 and P2 areas. Because the P1 area is cached, handlers that require high-speed processing are placed there.

Some supporting module control registers are located in a physical address space area 1. If these addresses are not subject to address translation, they should be located in the P2 area. When they are subject to address translation, non-caching should be set.

The P4 area is used for mapping on-chip control register addresses.

In user mode, the 2 Gbytes of virtual address space from H'00000000 to H'7FFFFFFF (area U0) can be accessed. U0 is mapped onto physical address space in page units. The 2 Gbytes of virtual address space from H'80000000 to H'FFFFFFFF cannot be accessed in user mode. Attempting to do so creates an address error. Copy-back or write-through mode can be selected for write accesses by means of a CCR setting.

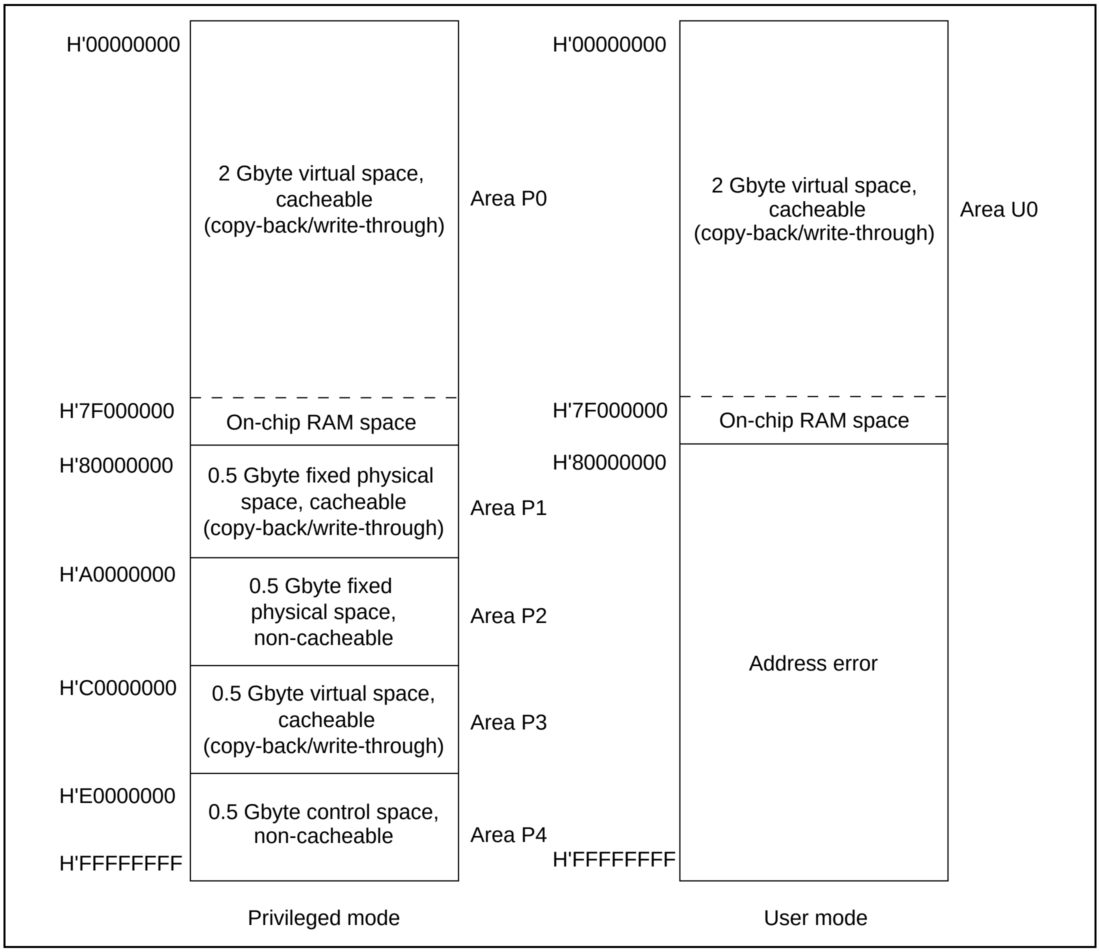

H'00000000
2 Gbyte virtual space, cacheable (copy-back/write-through)
Area P0
H'7F000000
On-chip RAM space
H'80000000
0.5 Gbyte fixed physical space, cacheable (copy-back/write-through)
Area P1
H'A0000000
0.5 Gbyte fixed physical space, non-cacheable
Area P2
H'C0000000
0.5 Gbyte virtual space, cacheable (copy-back/write-through)
Area P3
H'E0000000
0.5 Gbyte control space, non-cacheable
Area P4
H'FFFFFFFF
Privileged mode
H'00000000
2 Gbyte virtual space, cacheable (copy-back/write-through)
Area U0
H'7F000000
On-chip RAM space
H'80000000
Address error
H'FFFFFFFF
User mode

Figure 3.2 Virtual Address Space Mapping

Physical Address Space: The SH7709 supports a 32-bit physical address space, but the upper 3 bits are actually ignored and treated as a shadow. See section 10, Bus State Controller, for details.

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

If the virtual address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that results in a physical address space of H'80000000–H'FFFFFFFF should not be registered in the TLB.

When the MMU is disabled, the virtual address is used directly as the physical address. As the SH7709 supports a 29-bit address space as the physical address space, the top 3 bits of the physical address are ignored, and constitute a shadow space (see section 10, Bus State Controller (BSC)). For example, addresses H'00001000 in the P0 area, H'80001000 in the P1 area, H'A0001000 in the P2 area, and H'C0001000 in the P3 area are all mapped onto the same physical address. When access to these addresses is performed with the cache enabled, an address with the top 3 bits of the physical address masked to 0 is stored in the cache address array to ensure data congruity.

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

Address Space Identifier (ASID): When multiple processes run in parallel sharing the same virtual address space and the processes have unique address translation tables, the virtual space can be multiplexed. When this is done, a given virtual address may be mapped to a different physical address depending on the process. This means that virtual addresses are expanded by using an address space identifier (ASID) and virtual addresses can be distinguished by ASID. The ASID is 8 bits in length and is held in PTEH within the MMU indicating the current process. With ASID, the TLB need not be purged when the process is switched.

When multiple processes run in parallel using the virtual address space exclusively, the physical address corresponding to a given virtual address is specified uniquely. For this kind of single virtual memory, the ASID becomes a key to protect memory (see section 3.4.2).

3.1.4 Register Configuration

A register that has an undefined initial value must be initialized by the software. Table 3.1 shows the configuration of the MMU control registers.

Table 3.1 Register Configuration

Register Name	Abbr.	R/W	Size	Initial Value*1	Address
Page table entry register high	PTEH	R/W	Longword	Undefined	H'FFFFFFF0
Page table entry register low	PTEL	R/W	Longword	Undefined	H'FFFFFFF4
Translation table base register	TTB	R/W	Longword	Undefined	H'FFFFFFF8
TLB exception address register	TEA	R/W	Longword	Undefined	H'FFFFFFFC
MMU control register	MMUCR	R/W	Longword	*2	H'FFFFFFE0

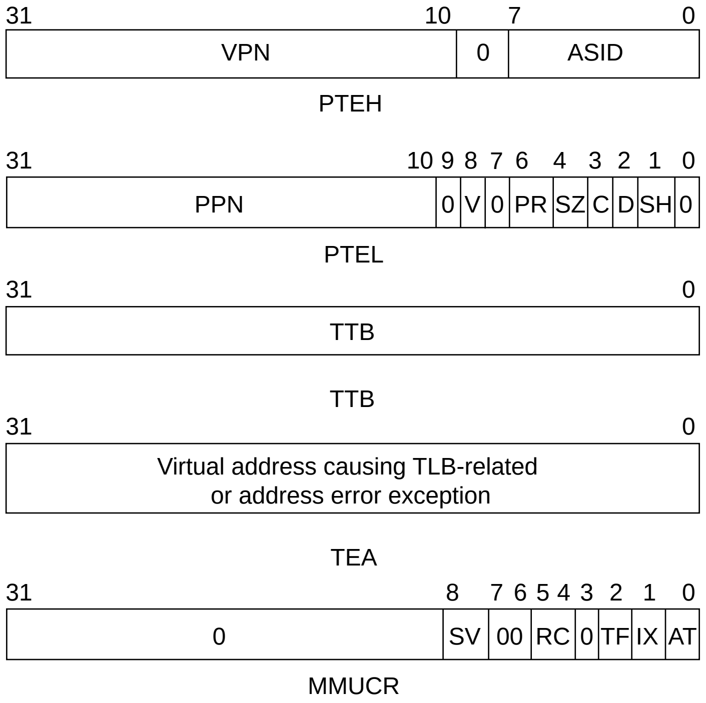
Notes: 1. Initialized by a power-on reset or manual reset.
2. SV bit: undefined
Other bits: 0

3.2 Register Description

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

1. The page table entry register high (PTEH) register residing at address H'FFFFFFF0, which consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the virtual address at which the exception is generated in the case of an MMU exception or address error exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtual address, but in this case the upper 22 bits of the virtual address are set. The VPN can also be modified by software. As the ASID, software sets the number of the currently executing process. The VPN and ASID are recorded in the TLB by the LDTLB instruction.
2. The page table entry register low (PTEL) register residing at address H'FFFFFFF4, and used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified in response to a software command.
3. The translation table base register (TTB) residing at address H'FFFFFFF8, which points to the base address of the current page table. The hardware does not set any value in TTB automatically. TTB is available to software for general purposes.
4. The TLB exception address register (TEA) register residing at address H'FFFFFFFC, which stores the virtual address corresponding to a TLB or address error exception. This value remains valid until the next exception or interrupt.
5. The MMU control register (MMUCR) residing at address H'FFFFFFE0, which makes the MMU settings described in figure 3.3. Any program that modifies MMUCR should reside in the P1 or P2 area.

The MMU registers are shown in figure 3.3.



0: Reserved bits. Always read as 0. Writing is ignored. However, 0 should also be specified in a write to MMUCR only.

SV: Single virtual memory mode bit. Set to 1 for single virtual memory mode, cleared to 0 for multiple virtual memory mode.

RC: A 2-bit random counter, automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all TLB entry ways corresponding to the virtual address at which the exception occurred are checked, and if all ways are valid, 1 is added to RC; if there is one or more invalid way, they are set by priority from way 0, in the order: way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.

TF: TLB flush bit. Write 1 to flush the TLB (clear all valid bits of the TLB to 0). Always reads 0.

IX: Index mode bit. When 0, VPN bits 16–12 are used as the TLB index number. When 1, the value obtained by EX-ORing ASID bits 4–0 in PTEH and VPN bits 16–12 are used as the TLB index number.

AT: Address translation bit. Enables/disables the MMU.
0: MMU disabled
1: MMU enabled

Figure 3.3 MMU Register Contents

3.3 TLB Functions

3.3.1 Configuration of the TLB

The TLB caches address translation table information located in external memory. The address translation table stores the physical page number translated from the virtual page number and the control information for the page, which is the unit of address translation. Figure 3.4 shows the overall TLB configuration. The TLB is 4-way set associative with 128 entries. There are 32 entries for each way. Figure 3.5 shows the configuration of virtual addresses and TLB entries.

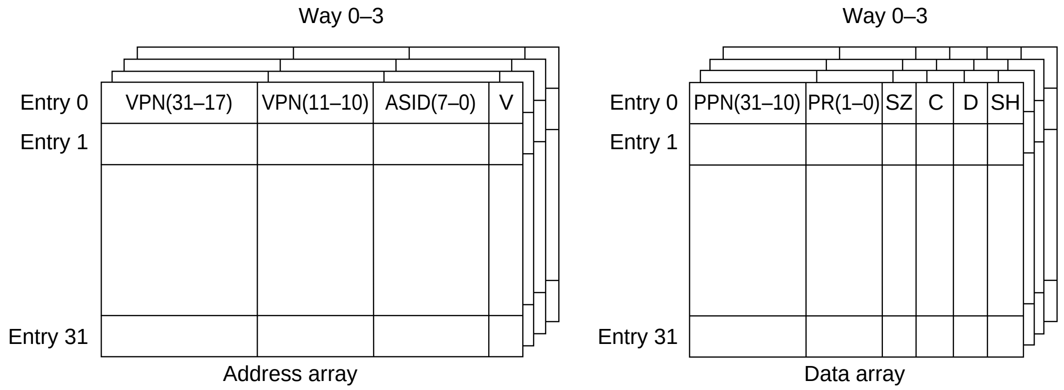


Figure 3.4 Overall Configuration of the TLB

31 10	9 0
VPN	Offset

Virtual address (1-kbyte page)

31 12	11 0
VPN	Offset

Virtual address (4-kbyte page)

(15)	(2)	(8)	(1)	(1)	(1)	(22)	(2)	(1)	(1)
VPN (31–17)	VPN (11–10)	ASID	SH	SZ	V	PPN	PR	C	D

TLB entry

VPN: Virtual page number. Top 22 bits of virtual address for a 1-kbyte page, or top 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16-12 are used as the index number, they are not stored in the TLB entry.

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

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

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

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

PPN: Physical page number. Top 22 bits of physical address. PPN bits 11-10 are not used in the case of a 4-kbyte page. Attention must be paid to the synonym problem in the case of a 1-kbyte page (see section 3.4.4).
Set the most significant bit to 0.

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

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

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

Figure 3.5 Virtual Address and TLB Structure

3.3.2 TLB Indexing

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

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

The second method is used to prevent lowered TLB efficiency that results when multiple processes run simultaneously in the same virtual address space and a specific entry is selected by indexing of each process. Figures 3.6 and 3.7 show the indexing schemes.

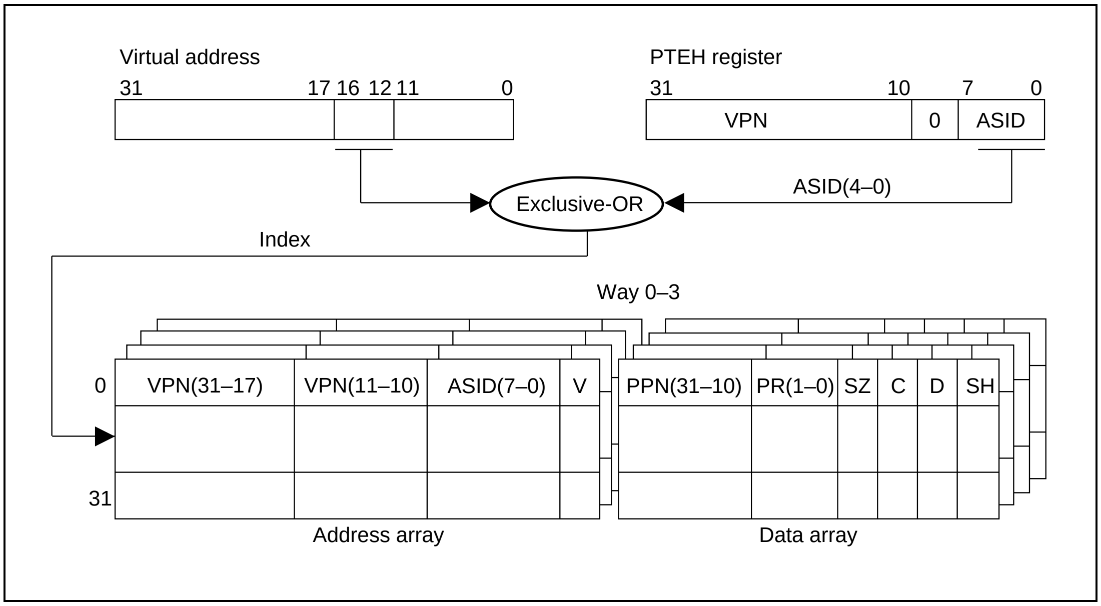


Figure 3.6 TLB Indexing (IX = 1)

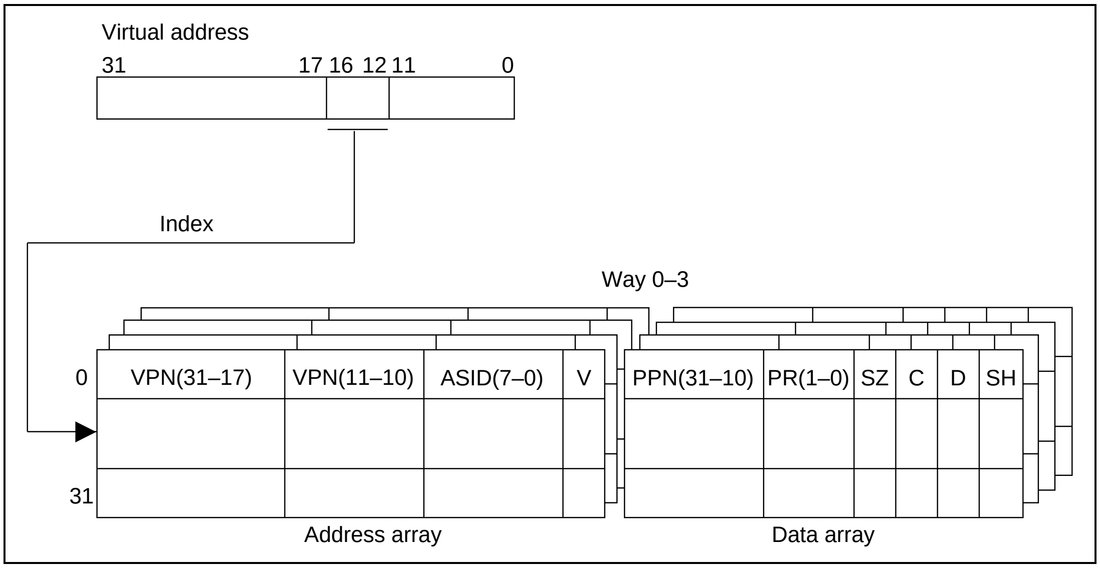


Figure 3.7 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

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

It is necessary to have the software ensure that TLB hits do not occur simultaneously in more than one way, as hardware operation is not guaranteed if this happens. For example, if there are two identical TLB entries with the same VPN and a setting is made such that a TLB hit is made only by a process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the non-shared state (SH = 0), then if the ASID in PHE is set to H'FF, there is a possibility of simultaneous TLB hits in both these ways. It is therefore necessary to ensure that this kind of setting is not made by the software.

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

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

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

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is engaged (SR.MD = 1), all process resources can be accessed. This means that ASIDs are not compared when single virtual memory is supported and privileged mode is engaged. The objects of address comparison are shown in figure 3.8.

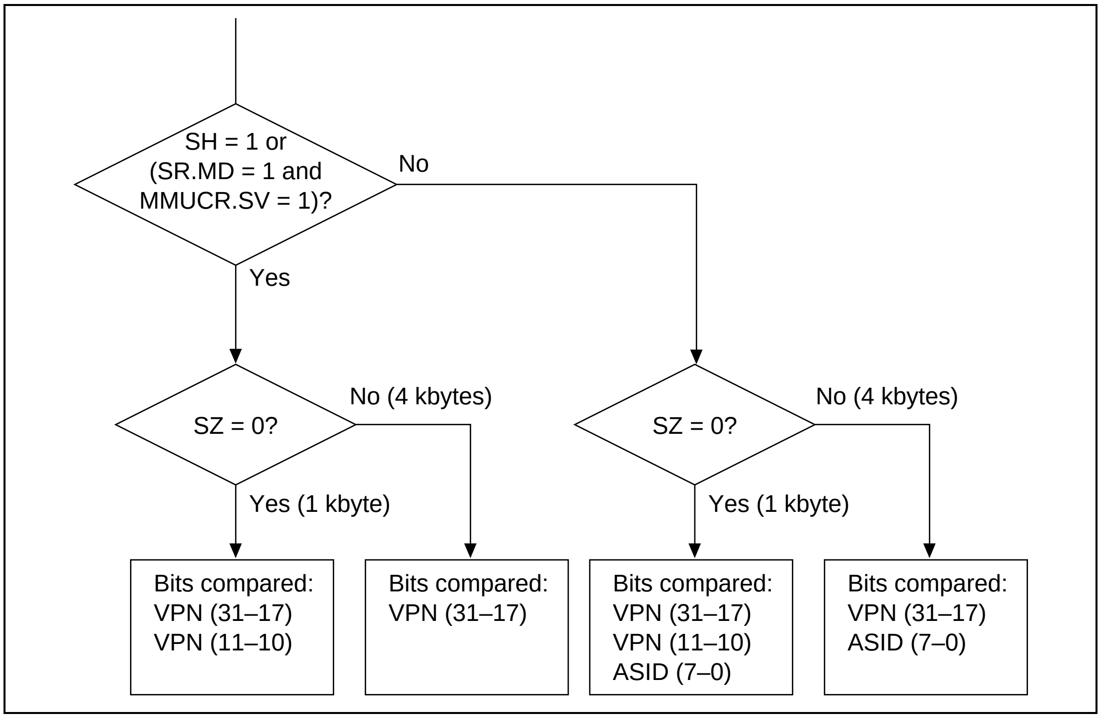


Figure 3.8 Objects of Address Comparison

3.3.4 Page Management Information

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

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

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

Access states designated by the D, C, and PR bits are shown in table 3.2.

Table 3.2 Access States Designated by D, C, and PR Bits

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

3.4 MMU Functions

3.4.1 MMU Hardware Management

MMU hardware management is of the following two kinds.

1. The MMU decodes the virtual address accessed by a process and performs address translation by controlling the TLB in accordance with the MMUCR settings.
2. In address translation, the MMU receives page management information from the TLB, and determines the MMU exception and whether the cache is to be accessed (using the C bit). For details of the determination method and the hardware processing, see section 3.5, MMU Exceptions.

3.4.2 MMU Software Management

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

1. MMU register setting. MMUCR setting, in particular, should be performed in areas P1 and P2 for which address translation is not performed. Also, since SV and IX bit changes constitute address translation system changes, in this case, TLB flushing should be performed by simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generated in the MMU disabled state with the AT bit cleared to 0, use in the disabled state must be avoided with software that does not use the MMU.
2. TLB entry recording, deletion, and reading. TLB entry recording can be done in two ways—by using the LDTLB instruction, or by writing directly to the memory-mapped TLB. For TLB entry deletion and reading, the memory allocation TLB can be accessed. See section 3.4.3, MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 3.6, Memory-Mapped TLB Configuration, for details of the memory-mapped TLB.
3. MMU exception handling. When an MMU exception is generated, it is handled on the basis of information set from the hardware side. See section 3.5, MMU Exceptions, for details.

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

Recording a 1-kbyte page TLB entry may result in a synonym problem. See section 3.4.4, Avoiding Synonym Problems.

3.4.3 MMU Instruction (LDLTB)

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

Figure 3.9 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the virtual address that caused the exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for each exception according to the rules shown in figure 3.9. Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU exception handling routine, TLB entry recording is possible. Any TLB entry can be updated by software rewriting of PTEH and the RC bits in MMUCR.

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

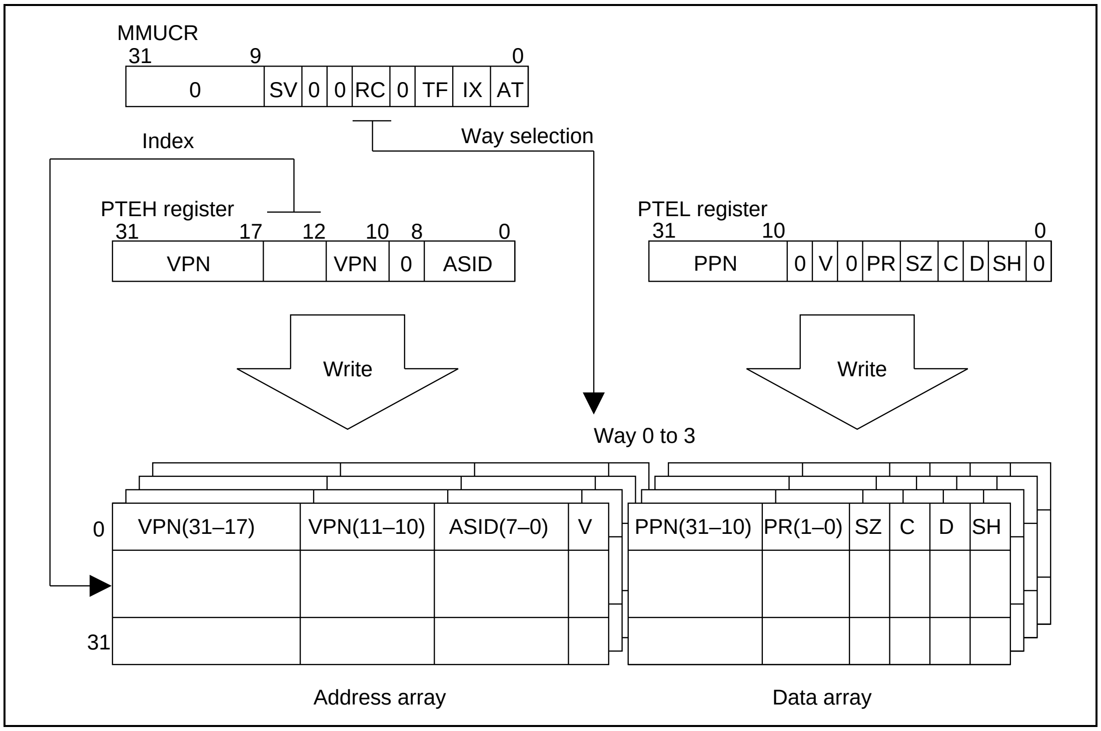


Figure 3.9 Operation of LDTLB Instruction

3.4.4 Avoiding Synonym Problems

When a 1-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a number of virtual addresses are mapped onto a single physical address, the same physical address data will be recorded in a number of cache entries, and it will not be possible to guarantee data congruity. The reason why this problem only occurs when using a 1-kbyte page is explained below with reference to figure 3.10.

To achieve high-speed operation of the SH7709 cache, an index number is created using virtual address bits 10–4. When a 4-kbyte page is used, virtual address bits 10–4 are included in the offset, and since they are not subject to address translation, they are the same as physical address bits 10–4. In cache-based address comparison and recording in the address array, since the cache tag address is a physical address, physical address bits 31–10 are recorded.

When a 1-kbyte page is used, also, a cache index number is created using virtual address bits 10-4. However, in the case of a 1-kbyte page, virtual address bit 10 is subject to address translation and therefore may not be the same as physical address bit 10. Consequently, the physical address is recorded in a different entry from that of the index number indicated by the physical address in the cache address array.

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

Virtual address 1 H'00000000 → physical address H'00000400
Virtual address 2 H'00000400 → physical address H'00000400

Virtual address 1 is recorded in cache entry H'00, and virtual address 2 in cache entry H'40. Since the two virtual addresses are recorded in different cache entries despite the fact that the physical addresses are the same, memory inconsistency will occur as soon as a write is performed to either virtual address. Therefore, when recording a 1-kbyte TLB entry, if the physical address is the same as a physical address already used in another TLB entry, it should be recorded in such a way that physical address bit 10 is the same.

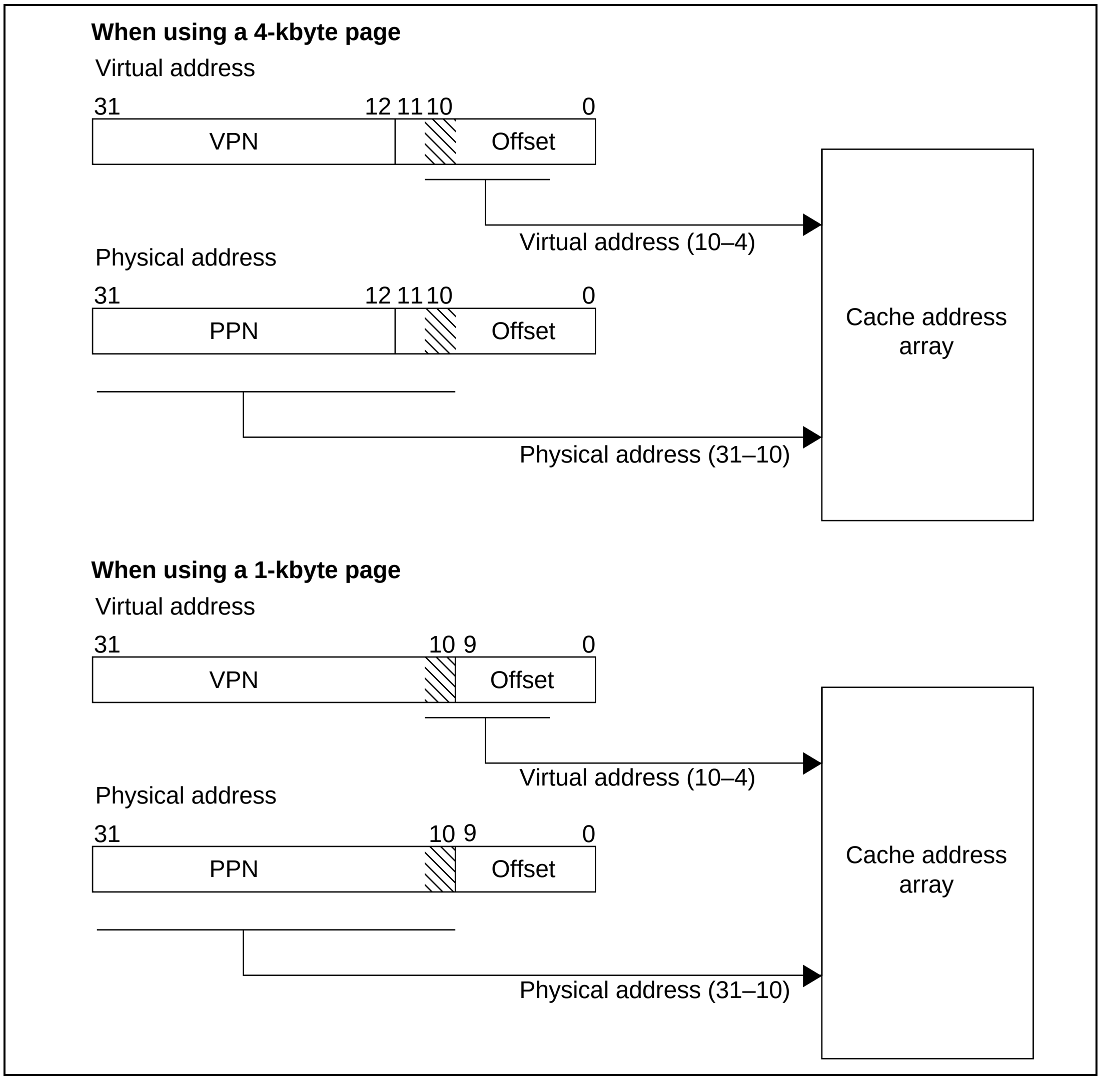


Figure 3.10 Synonym Problem

3.5 MMU Exceptions

There are four MMU exceptions: TLB miss, TLB protection violation, TLB invalid, and initial page write.

3.5.1 TLB Miss Exception

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

Hardware Operations: In a TLB miss, the SH7709 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written to the save program counter (SPC). If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
5. The contents of the status register (SR) at the time of the exception are written to the save status register (SSR).
6. The mode (MD) bit in SR is set to 1 to place the SH7709 in privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The RC field in the MMU control register (MMUCR) is incremented by 1 when all entries indexed are valid. When some entries indexed are invalid, the smallest way number of them is set in RC.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000400 to invoke the user-written TLB miss exception handler.

Software (TLB Miss Handler) Operations: The software searches the page tables in external memory and allocates the required page table entry. Upon retrieving the required page table entry, the software must execute the following operations:

1. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the address translation table in external memory into the PTEL register in the SH7709.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the return from exception handler (RTE) instruction to terminate the handler routine and return to the instruction stream.

3.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the virtual address and the address array of the selected TLB entry are compared and a valid entry is found to match, but the type of access is not permitted by the access rights specified in the PR field. TLB protection violation exception handling includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, the SH7709 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written into SPC (if the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7709 in privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100 to invoke the TLB protection violation exception handler.

Software (TLB Protection Violation Handler) Operations: The software resolves the TLB protection violation and issues the RTE (return from exception handler) instruction to terminate the handler and return to the instruction stream.

3.5.3 TLB Invalid Exception

A TLB invalid exception results when the virtual address is compared to a selected TLB entry address array and a match is found but the entry is not valid (the V bit is 0). TLB invalid exception handling includes both hardware and software operations.

Hardware Operations: In a TLB invalid exception, the SH7709 hardware executes a set of prescribed operations, as follows:

1. The VPN number of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. The way number causing the exception is written to RC in MMUCR.
4. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
5. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the delayed branch instruction is written to the SPC.
6. The contents of SR at the time of the exception are written into SSR.
7. The mode (MD) bit in SR is set to 1 to place the SH7709 in privileged mode.
8. The block (BL) bit in SR is set to 1 to mask any further exception requests.
9. The register bank (RB) bit in SR is set to 1.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100, and the TLB protection violation exception handler starts.

Software (TLB Invalid Exception Handler) Operations: The software searches the page tables in external memory and assigns the required page table entry. Upon retrieving the required page table entry, the software must execute the following operations:

1. Write the values of the physical page number (PPN) field and the values of the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

3.5.4 Initial Page Write Exception

An initial page write exception results in a write access when the virtual address and the address array of the selected TLB entry are compared and a valid entry with the appropriate access rights is found to match, but the D (dirty) bit of the entry is 0 (the page has not been written to). Initial page write exception handling includes both hardware and software operations.

Hardware Operations: In an initial page write exception, the SH7709 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7709 in privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100 to invoke the user-written initial page write exception handler.

Software (Initial Page Write Handler) Operations: The software must execute the following operations:

1. Retrieve the required page table entry from external memory.
2. Set the D bit of the page table entry in external memory to 1.
3. Write the value of the PPN field and the PR, SZ, C, D, SH, and V bits of the page table entry in external memory to the PTEL register.
4. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
5. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
6. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

Figure 3.11 shows the flowchart for MMU exceptions.

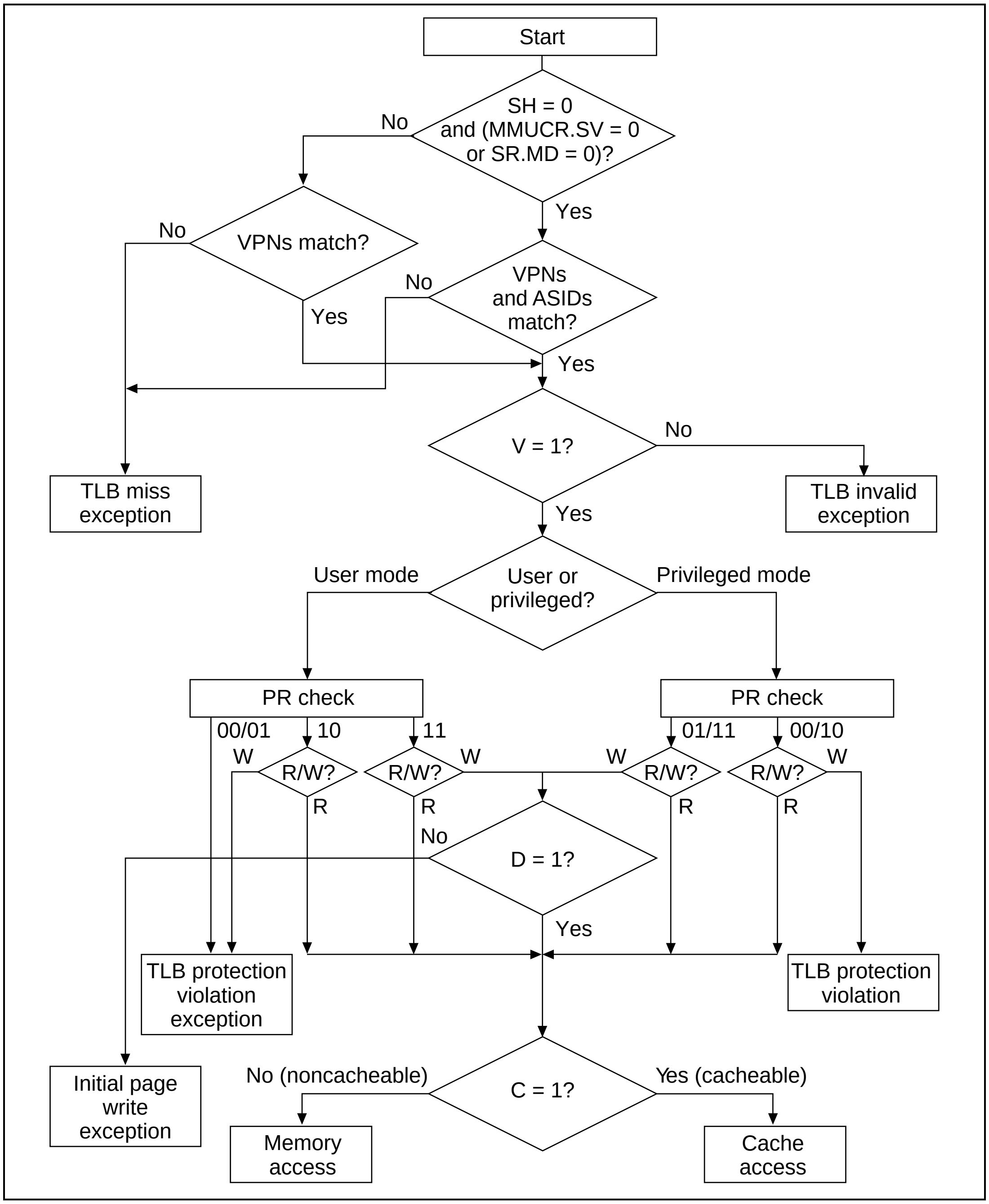


Figure 3.11   MMU Exception Generation Flowchart

3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error)

Figure 3.12 shows the MMU exception signals in instruction fetch mode.

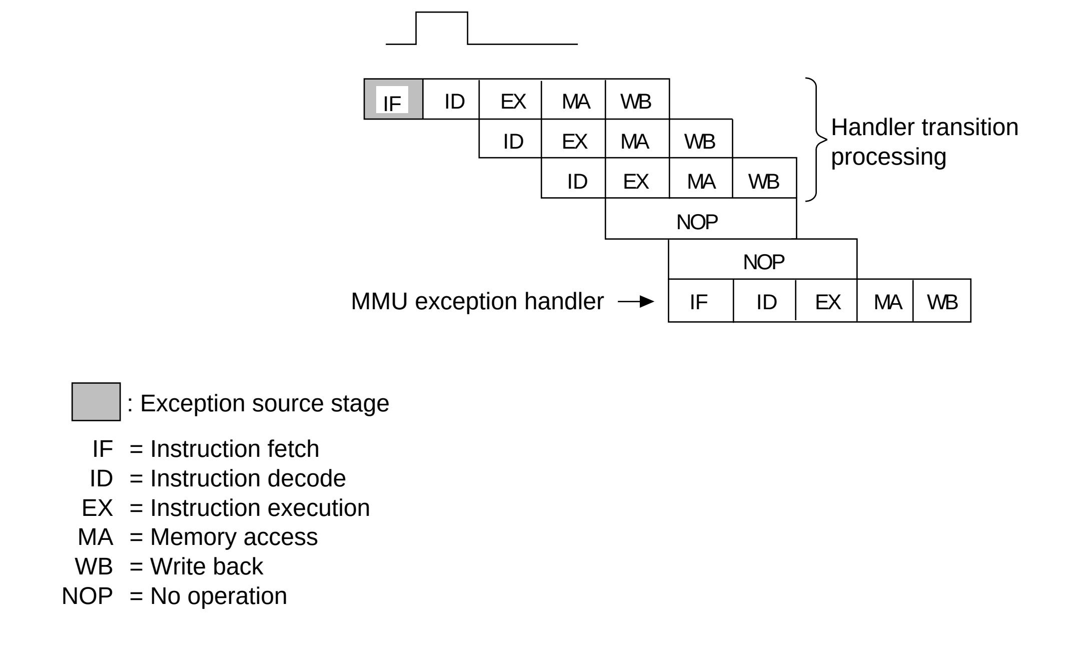


Figure 3.12 MMU Exception Signals in Instruction Fetch

Figure 3.13 shows the MMU exception signals in data access mode.

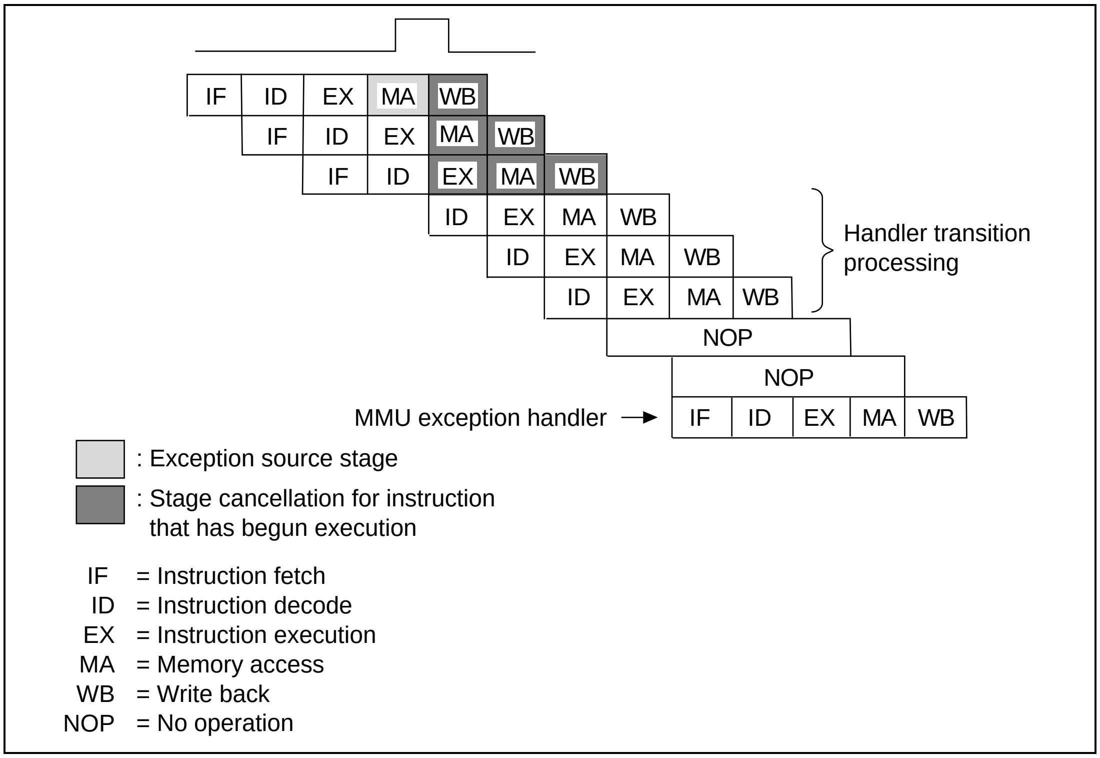


Figure 3.13 MMU Exception Signals in Data Access

3.6 Memory-Mapped TLB

In order for TLB operations to be managed by software, TLB contents can be read or written to in privileged mode using the MOV instruction. The TLB is assigned to the P4 area in virtual address space. The TLB address array (VPN, V bit, and ASID) is assigned to H'F2000000–H'F2FFFFFF, and the data array (PPN, PR, SZ, C, D, and SH bits) to H'F3000000–H'F3FFFFFF. The V bit in the address array can also be accessed from the data array. Only longword access is possible for both the address array and the data array.

3.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address array, the 32-bit address field (for read/write operations) and 32-bit data field (for write operations) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the VPN, V bit and ASID to be written to the address array (figure 3.14 (1)).

In the address field, specify the entry address for selecting the entry (bits 16–12), W for selecting the way (bits 9–8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3) and H'F2 to indicate address array access (bits 31–24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

When writing, specify bit 7 as the A bit. The A bit indicates whether addresses are compared during writing. When the A bit is 1, the VPNs of the four entries selected by the entry addresses are compared to the VPN to be written into the address array specified in the data field. Writing takes place to the way that has a hit. When a miss occurs, nothing is written to the address array and no operation occurs. The way number specified in bits 9–8 is not used. The item compared is determined by the SZ and SH bits of the entry selected by the entry address, the SV bit in MMUCR and the MD bit in SR, just as in ordinary operations (see section 3.3.3).

When the A bit is 0, it is written to the entry selected with the entry address and way number without comparing addresses.

When reading, the VPN (31–17, 11–10), V bit, and ASID of the entry specified by the entry address and way number are read in the format of the data field in figure 3.14 without comparing addresses.

To invalidate a specific entry, specify the entry and write 0 to its V bit. When 1 is specified for the A bit, only the required VPN entry is invalidated.

3.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. These are specified in the general register. The address section specifies information for selecting the entry to be accessed; the data section specifies the longword data to be written to the data array (figure 3.14 (2)).

In the address section, specify the entry address for selecting the entry (bits 16–12), W for selecting the way (bits 9–8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3), and H'F3 to indicate data array access (bits 31–24). The IX bit in MMUCR indicates whether an EX-OR is taken of the entry address and ASID.

Both reading and writing use the longword of the data array specified by the entry address and way number. The access size of the data array is fixed at longword.

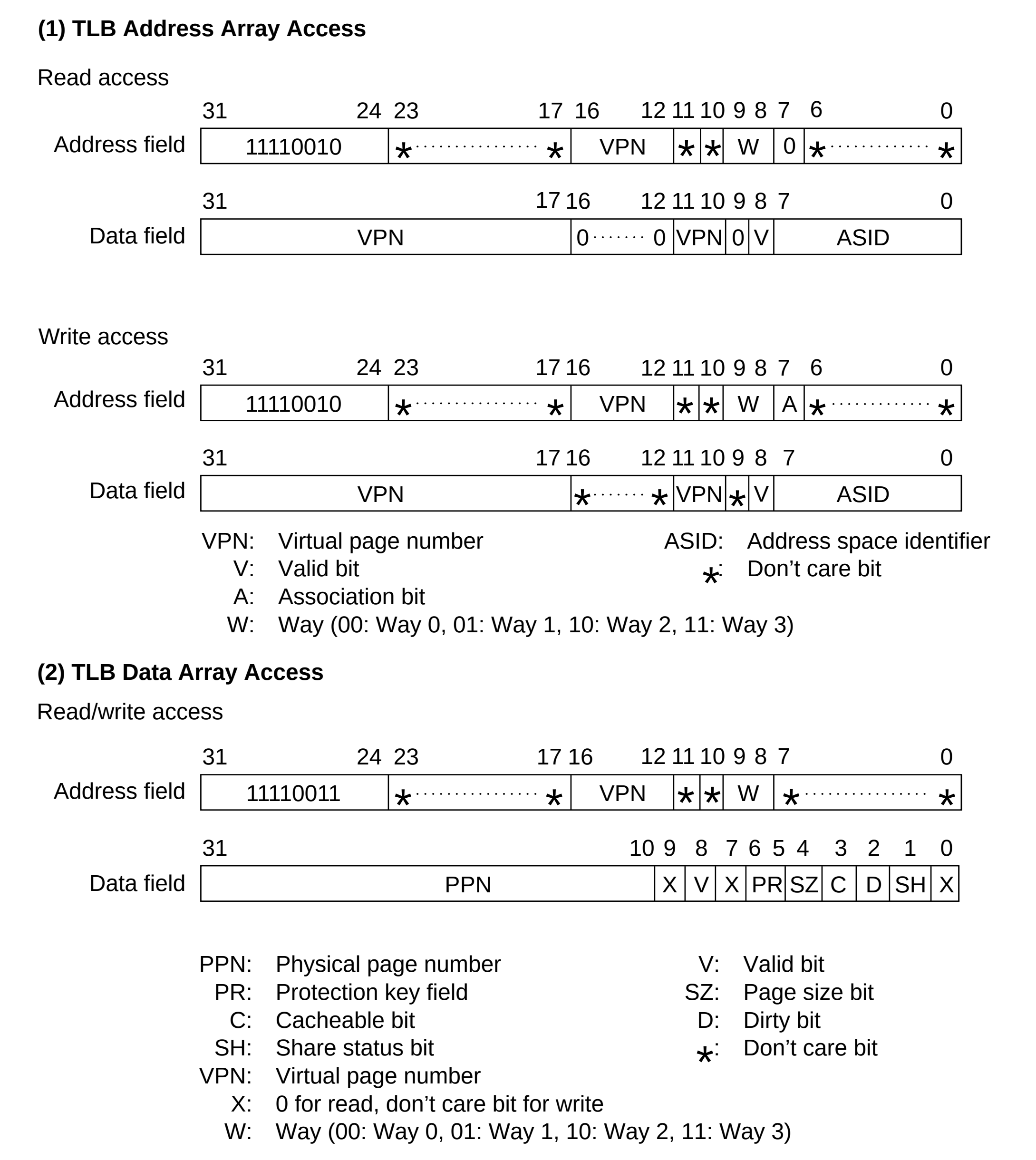


Figure 3.14 Specifying Address and Data for Memory-Mapped TLB Access

3.6.3 Usage Examples

Invalidating Specific Entries: Specific TLB entries can be invalidated by writing 0 to the entry’s V bit. When the A bit is 1, the VPN and ASID specified by the write data is compared to the VPN and ASID within the TLB entry selected by the entry address and data is written to the matching way. If no match is found, there is no operation. R0 specifies the write data and R1 specifies the address.

```
; R0=H'1547 381C  R1=H'F201 3080
; MMUCR.IX=0
; VPN(31–17)=B'0001 0101 0100 011   VPN(11–10)=B'10   ASID=B'0001 1100
; corresponding entry association is made from the entry selected by
; the VPN(16–12)=B'1 0011 index, the V bit of the hit way is cleared
to
; 0,achieving invalidation.

MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a specific TLB entry. The bit order indicated in the data field in figure 3.14 (2) is read. R0 specifies the address and the data section of a selected entry is read to R1.

```
; R1=H'F300 4300   VPN(16-12)=B'00100   Way 3

; MOV.L @R0,R1
```

3.7 Usage Note

Instructions that manipulate the MD or BL bit in register SR (the LDC Rm, SR instruction, LDC @Rm+, SR instruction, and RTE instruction) and the following instruction, or the LDTLB instruction, should be used with the TLB disabled or in a fixed physical address space (the P1 or P2 space).

Section 4 Exception Handling

4.1 Overview

4.1.1 Features

Exceptions are deviations from normal program execution that require special handling. The processor responds to an exception by aborting execution of the current instruction (execution is allowed to continue to completion in all interrupt requests) and passing control from the instruction stream to the appropriate user-written exception handling routine. Here, all exceptions other than resets and interrupts will be called general exceptions. There are thus three types of exceptions: resets, general exceptions, and interrupts.

4.1.2 Register Configuration

A register with an undefined initial value should be initialized by software. Table 4.1 lists the registers used for exception handling.

Table 4.1 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
TRAPA exception register	TRA	R/W	Undefined	H'FFFFFFD0	Longword
Exception event register	EXPEVT	R/W	Power-on reset: H'000 Manual reset: H'020	H'FFFFFFD4	Longword
Interrupt event register	INTEVT	R/W	Undefined	H'FFFFFFD8	Longword
Interrupt event register2	INTEVT2	R	Undefined	H'04000000	Longword

4.2 Exception Handling Function

4.2.1 Exception Handling Flow

Usually the contents of the program counter (PC) and status register (SR) are saved in the saved program counter (SPC) and saved status register (SSR), respectively, and execution of the exception handler is invoked from a vector address. The return from exception handler (RTE) instruction is issued by the exception handler routine at the completion of the routine, restoring the contents of the PC and SR to return to the processor status at the point of interruption and the address where the exception occurred.

A basic exception processing sequence consists of the following operations:

- The contents of the PC and SR are saved in the SPC and SSR, respectively.
- The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
- The mode (MD) bit in SR is set to 1 to place the SH7709 in privileged mode.
- The register bank (RB) bit in SR is set to 1.
- An encoded value identifying the exception event is written to bits 11–0 of the exception event (EXPEVT) or interrupt event (INTEVT and INTEVT2) register.
- Instruction execution jumps to the designated exception processing vector address to invoke the handler routine.

4.2.2 Exception Handling Vector Addresses

The reset vector address is fixed at H'A0000000. The other three events are assigned offsets from the vector base address by software. Translation lookaside buffer (TLB) miss exceptions have an offset from the vector base address of H'00000400. The vector address offset for general exception events other than TLB miss exceptions is H'00000100. The interrupt vector address offset is H'00000600. The vector base address is loaded into the vector base register (VBR) by software. The vector base address should reside in P1 or P2 fixed physical address space. Figure 4.1 shows the relationship between the vector base address, the vector offset, and the vector table.

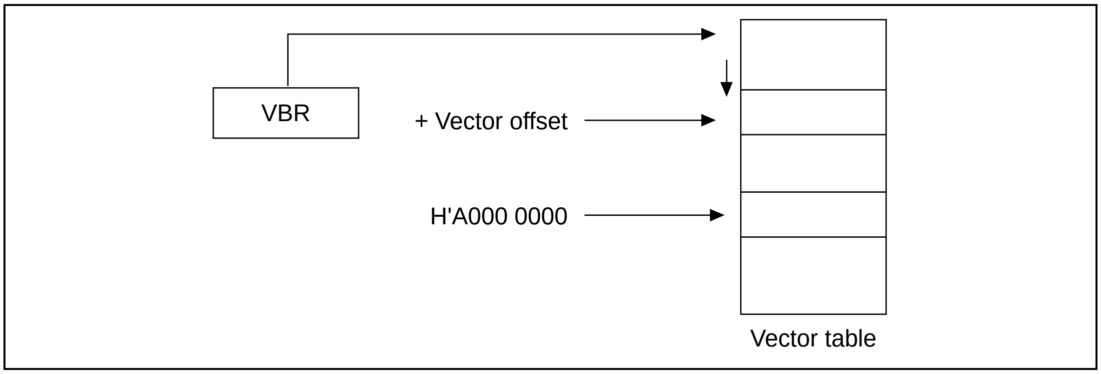


Figure 4.1 Vector Table

In table 4.2, exceptions and their vector addresses are listed by exception type, instruction completion status, relative acceptance priority, relative order of occurrence within an instruction execution sequence and vector address for exceptions and their vector addresses.

Table 4.2 Vectored Exception Events

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Vector Address	Vector Offset
Reset	Aborted	Power-on	1	—	H'A00000000	—
		Manual reset	1	—	H'A00000000	—
General exception events	Aborted and retried	Address error (instruction access)	2	1	—	H'00000100
		TLB miss (instruction access)	2	2	—	H'00000400
		TLB invalid (instruction access)	2	3	—	H'00000100
		TLB protection violation (instruction access)	2	4	—	H'00000100
		Reserved instruction code exception	2	5	—	H'00000100
		Illegal slot instruction exception	2	5	—	H'00000100
		Address error (data access)	2	6	—	H'00000100
		TLB miss (data access)	2	7	—	H'00000400
		TLB invalid (data access)	2	8	—	H'00000100
		TLB protection violation (data access)	2	9	—	H'00000100
		Initial page write	2	10	—	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	5	—	H'00000100
		User breakpoint trap	2	n*[2]	—	H'00000100

Table 4.2 Vectored Exception Events (cont)

Exception Type	Current Instruction	Exception Event	Priority*[1]	Exception Order	Vector Address	Vector Offset
General interrupt requests	Completed	Nonmaskable interrupt	3	—	—	H'00000600
		External hardware interrupt	4*[3]	—	—	H'00000600
		Peripheral module interrupt	4*[3]	—	—	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being highest and 4 being lowest.

2. The user defines the break point traps. 1 is a break point before instruction execution and 11 is a break point after instruction execution. For an operand break point, use 11.

3. Use software to specify relative priorities of external hardware interrupts and peripheral module interrupts (see section 6, Interrupt Controller (INTC)).

4.2.3 Acceptance of Exceptions

Processor resets and interrupts are asynchronous events unrelated to the instruction stream. All exception events are prioritized to establish an acceptance order whenever two or more exception events occur simultaneously. The power-on reset and manual reset may not occur simultaneously, so they have the same priority.

All general exception events occur in a relative order in the execution sequence of an instruction (i.e., execution order), but are handled at priority level 2 in instruction-stream order (i.e., program order), where an exception detected in a preceding instruction is accepted prior to an exception detected in a subsequent instruction.

Three general exception events (reserved instruction code exception, unconditional trap, and illegal slot instruction exception) are detected in the decode stage of different instructions and are mutually exclusive events in the instruction pipeline. They have the same execution priority. Figure 4.2 shows the order of general exception acceptance.

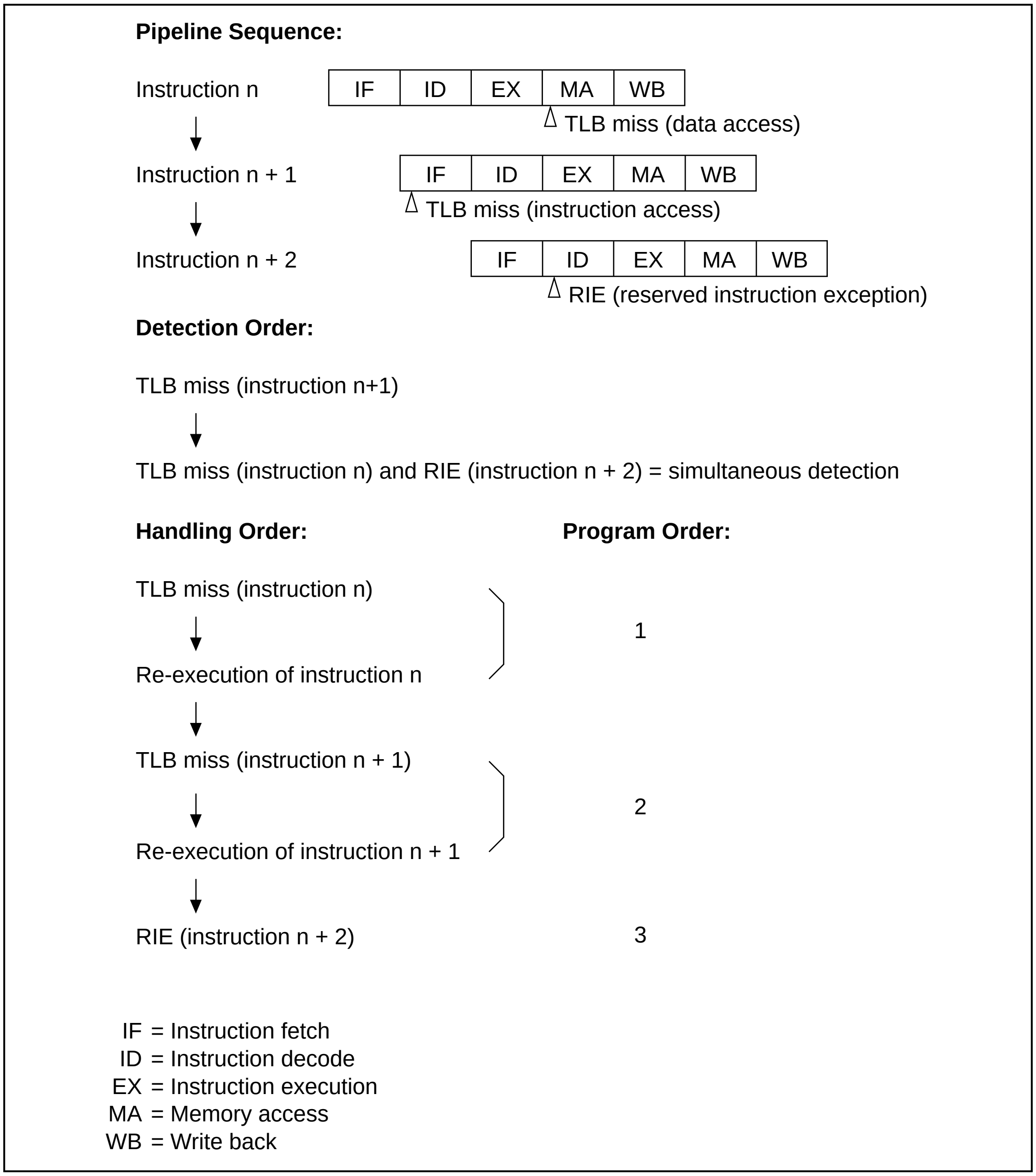


Figure 4.2 Example of Acceptance Order of General Exceptions

All exceptions other than a reset are detected in the pipeline ID stage, and accepted on instruction boundaries. However, an exception is not accepted between a delayed branch instruction and the delay slot. A re-execution type exception detected in a delay slot is accepted before execution of the delayed branch instruction. A completion type exception detected in a delayed branch instruction or delay slot is accepted after execution of the delayed branch instruction. The delay slot here refers to the next instruction after a delayed unconditional branch instruction, or the next instruction when a delayed conditional branch instruction is true.

4.2.4 Exception Codes

Table 4.3 lists the exception codes written to bits 11–0 of the EXPEVT register (for reset or general exceptions) or the INTEVT and INTEVT2 registers (for general interrupt requests) to identify each specific exception event. An additional exception register, the TRAPA (TRA) register, is used to hold the 8-bit immediate data in an unconditional trap (TRAPA instruction).

Table 4.3 Exception Codes

Exception Type	Exception Event	Exception Code
Reset	Power-on	H'000
	Manual reset	H'020
General exception events	TLB miss/invalid (load)	H'040
	TLB miss/invalid (store)	H'060
	Initial page write	H'080
	TLB protection violation (load)	H'0A0
	TLB protection violation (store)	H'0C0
	Address error (load)	H'0E0
	Address error (store)	H'100
	Unconditional trap (TRAPA instruction)	H'160
	Reserved instruction code exception	H'180
	Illegal slot instruction exception	H'1A0
	User breakpoint trap	H'1E0
General interrupt requests	Nonmaskable interrupt	H'1C0
	External hardware interrupts:	
	IRL3–IRL0 = 0000	H'200
	IRL3–IRL0 = 0001	H'220
	IRL3–IRL0 = 0010	H'240
	IRL3–IRL0 = 0011	H'260
	IRL3–IRL0 = 0100	H'280
	IRL3–IRL0 = 0101	H'2A0
	IRL3–IRL0 = 0110	H'2C0

Table 4.3 Exception Codes (cont)

Exception Type	Exception Event	Exception Code
General interrupt requests (cont)	External hardware interrupts (cont):	
	IRL3–IRL0 = 0111	H'2E0
	IRL3–IRL0 = 1000	H'300
	IRL3–IRL0 = 1001	H'320
	IRL3–IRL0 = 1010	H'340
	IRL3–IRL0 = 1011	H'360
	IRL3–IRL0 = 1100	H'380
	IRL3–IRL0 = 1101	H'3A0
	IRL3–IRL0 = 1110	H'3C0

Note: Exception codes H'120, H'140, and H'3E0 are reserved.

4.2.5 Exception Request Masks

If a general exception event occurs when the BL bit in SR is 1, the CPU’s internal registers are set to their post-reset state, other module registers retain their contents prior to the general exception, and a branch is made to the same address (H'A0000000) as for a reset.

If a general interrupt occurs when BL = 1, the request is masked (held pending) and not accepted until the BL bit is cleared to 0 by software. For reentrant exception handling, the SPC and SSR must be saved and the BL bit in SR cleared to 0.

4.2.6 Returning from Exception Handling

The RTE instruction is used to return from exception handling. When RTE is executed, the SPC value is set in the PC, and the SSR value in SR, and the return from exception handling is performed by branching to the SPC address.

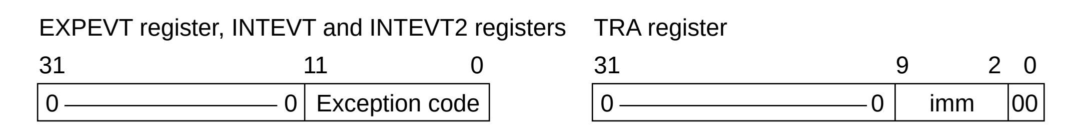
If the SPC and SSR have been saved in external memory, set the BL bit in SR to 1, then restore the SPC and SSR, and issue an RTE instruction.

4.3 Register Description

There are three registers related to exception handling. These are peripheral module registers, and therefore reside in area P4. They can be accessed by specifying the address in privileged mode only.

1. The exception event register (EXPEVT) resides at address H'FFFFFFD4, and contains 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.
2. The interrupt event register (INTEVT) resides at address H'FFFFFFD8, and contains a 12-bit exception code. The exception code set in EXPEVT is that for an interrupt request. The exception code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.
3. The TRAPA exception register (TRA) resides at address H'FFFFFFD0, and contains 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.

The bit configurations of the EXPEVT, INTEVT, and TRA registers are shown in figure 4.3.



0: Reserved bits, always read as zero

imm: 8-bit immediate data in TRAPA instruction

Figure 4.3 Bit Configurations of EXPEVT, INTEVT, INTEVT2, and TRA Registers

4.4 Exception Handler Operation

4.4.1 Reset

The reset sequence is used to power up or restart the SH7709 from the initialization state. The $\overline{\text{RESET}}$ signal is sampled every clock cycle, and in the case of a power-on reset, all processing being executed (excluding the RTC) is suspended, all unfinished events are canceled, and reset processing is executed immediately. In the case of a manual reset, however, processing to retain external memory contents is continued. The reset sequence consists of the following operations:

- The MD bit in SR is set to 1 to place the SH7709 in privileged mode.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The RB bit in SR is set to 1.
- An encoded value of H'000 in a power-on reset or H'020 in a manual reset is written to bits 11–0 of the EXPEVT register to identify the exception event.
- Instruction execution jumps to the user-written exception handler at address H'A0000000.

4.4.2 Interrupts

An interrupt processing request is accepted on completion of the current instruction. The interrupt acceptance sequence consists of the following operations:

- The contents of the PC and SR are saved in SPC and SSR, respectively.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The MD bit in SR is set to 1 to place the SH7709 in privileged mode.
- The RB bit in SR is set to 1.
- An encoded value identifying the exception event is written to bits 11–0 of the INTEVT and INTEVT2 registers.
- Instruction execution jumps to the vector location designated by the sum of the value of the contents of the vector base register (VBR) and H'00000600 to invoke the exception handler.

4.4.3 General Exceptions

When the SH7709 encounters any exception condition other than a reset or interrupt request, it executes the following operations:

- The contents of the PC and SR are saved in the SPC and SSR, respectively.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The MD bit in SR is set to 1 to place the SH7709 in privileged mode.
- The RB bit in SR is set to 1.
- An encoded value identifying the exception event is written to bits 11–0 of the EXPEVT register.
- Instruction execution jumps to the vector location designated by either the sum of the vector base address and offset H'00000400 in the vector table in a TLB miss trap, or by the sum of the vector base address and offset H'00000100 for exceptions other than TLB miss traps, to invoke the exception handler.

4.5 Individual Exception Operations

This section describes the conditions for specific exception handling, and the processor operations.

4.5.1 Resets

- Power-On Reset
  - Conditions: $\overline{\text{RESETP}}$ low
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bits are set to 1 and the IMASK field is set to B'1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details. A power-on reset must always be performed when powering on.
- Manual Reset
  - Conditions: $\overline{\text{RESETM}}$ low
  - Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB, and BL bits are set to 1 and the IMASK field is set to B'1111. The CPU and on-chip supporting modules are initialized. See the egister descriptions in the relevant sections for details.

Table 4.4 Types of Reset

		Internal State	
Type	**Conditions for Transition to Reset State**	**CPU**	**On-Chip Supporting Modules**
Power-on reset	$\overline{\text{RESETP}}$ = Low	Initialized	(See register configuration in relevant sections)
Manual reset	$\overline{\text{RESETM}}$ = Low	Initialized	

4.5.2 General Exceptions

- TLB miss exception
  - Conditions: Comparison of TLB addresses shows no address match
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The RC bit in MMUCR is incremented by one for replacement.

  The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset differs from other exceptions.

- TLB invalid exception
  - Conditions: Comparison of TLB addresses shows address match but V = 0
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved in the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Initial page write exception
  - Conditions: A hit occurred to the TLB for a store access, but D = 0
    This occurs for initial writes to the page registered by the load.
  - Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in MMUCR.RC.

The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'080 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs in PC = VBR + H'0100.

- TLB protection exception

— Conditions: When a hit access violates the TLB protection information (PR bits) shown below:

PR	Privileged mode	User mode
00	Only read enabled	No access
01	Read/write enabled	No access
10	Only read enabled	Only read enabled
11	Read/write enabled	Read/write enabled

— Operations: The virtual address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31–10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0A0 is set in EXPEVT; if the exception occurred during a write, H'0C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Address error

— Conditions:

a. Instruction fetch from odd address (4n + 1, 4n + 3)

b. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)

c. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)

d. Virtual space accessed in user mode in the area H'80000000 to H'FFFFFFFF.

— Operations: The virtual address (32 bits) that caused the exception is set in TEA. The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0E0 is set in EXPEVT; if the exception occurred during a write, H'100 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Unconditional trap

— Conditions: TRAPA instruction executed

— Operations: The exception is a processing-completion type, so the PC of the instruction after the TRAPA instruction is saved to the SPC. SR from the time when the TRAPA instruction was executing is saved to SSR. The 8-bit immediate value in the TRAPA instruction is quadrupled and set in TRA(9–0). H'160 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Reserved instruction exception
  - Conditions:

    a. When undefined code not in a delay slot is decoded

    Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instructions: H'Fxxx

    b. When a privileged instruction not in a delay slot is decoded in user mode

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions
  - Operations: The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'180 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.
- Illegal slot instruction
  - Conditions:

    a. When undefined code in a delay slot is decoded

    Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
    Undefined instructions: H'Fxxx

    b. When an instruction that rewrites the PC in a delay slot is decoded

    Instructions that rewrite the PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L, @Rm+, SR

    c. When a privileged instruction in a delay slot is decoded in user mode

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions
  - Operations: The PC of the previous delay branch instruction is saved to the SPC. SR of the instruction that generated the exception is saved to SSR. H'1A0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.
- User break point trap
  - Conditions: When a break condition set in the user break point controller is satisfied
  - Operations: When a post-execution break occurs, the PC of the instruction immediately after the instruction that set the break point is set in the SPC. If a pre-execution break occurs, the PC of the instruction that set the break point is set in the SPC. SR when the break occurs is set in SSR. H'1E0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. See section 7, User Break Controller, for more information.

4.5.3 Interrupts

1. NMI

   Conditions: NMI pin assert

   Operations: The PC and SR after the instruction that receives the interrupt are saved to the SPC and SSR, respectively. H'01C0 is set to INTEVT. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to PC = VBR + H'0600. This interrupt is not masked by SR.IMASK and received with top priority when the SR’s BL bit is 0. When the BL bit is 1, the interrupt is masked when BLMSK in ICRI is a logic zero and not masked when BLMSK in ICRI is a logic one. See section 6, Interrupt Controller, for more information.

2. IRL Interrupts

   Conditions: The value of the interrupt mask bits in SR is lower than the IRL3–IRL0 level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the IRL3–IRL0 level is set in INTEVT. The corresponding code is given as H'200 + B'(IRL3–IRL0) × H'20. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set in SR.IMASK. See section 6, Interrupt Controller, for more information.

3. IRQ Pin Interrupts

   Conditions: IRQ pin is asserted and SR.IMASK is lower than the IRQ priority level and the SR’s BL bit is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set to SR.IMASK. See section 6, Interrupt Controller, for more information.

4. INT Pin Interrupts

   Conditions: INT pin is asserted and SR.IMASK is lower than the INT priority level and the SR’s BL bit is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set to SR.IMASK. See section 6, Interrupt Controller, for more information.

5. On-Chip Module Interrupts

   Conditions: SR.IMASK is lower than the on-chip module (TMU, RTC, SCI0, SCI1, SCI2, A/D, DMAC, CPG, REF) interrupt level and the SR’s BL bit is 0. The interrupt is accepted at an instruction boundary.

Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. B'0000 to B'1111 are set to the interrupt priority level registers (IRPA - IRPF) within the interrupt controller. See section 6, Interrupt Controller, for more information.

4.6 Cautions

- Return from exception handling
  - Check the BL bit in SR with software. When the SPC and SSR have been saved to external memory, set the BL bit in SR to 1 before restoring them.
  - Issue an RTE instruction. Set the SPC in the PC and SSR in SR with the RTE instruction, branch to the SPC address, and return from exception handling.
- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is set to 0 by software. If there is a request and the reception conditions are satisfied, the interrupt is accepted after the execution of the instruction that sets the BL bit in SR to 0. During the sleep or standby mode, however, the interrupt will be accepted even when the BL bit in SR is 1.

    NMI is accepted when BLMSK in ICRI is a logic 1.
  - Exception: No user break point trap will occur even when the break conditions are met. When one of the other exceptions occurs, a branch is made to the fixed address of the reset (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR registers are undefined.
- SPC when an Exception Occurs: The PC saved to the SPC when an exception occurs is as shown below:
  - Re-executing-type exceptions: The PC of the instruction that caused the exception is set in the SPC and re-executed after return from exception handling. If the exception occurred in a delay slot, however, the PC of the immediately prior delayed branch instruction is set in the SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.
  - Completed-type exceptions and interrupts: The PC of the instruction after the one that caused the exception is set in the SPC. If the exception was caused by a delayed conditional instruction, however, the branch destination PC is set in SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.

- Initial register values after reset
  - Undefined registers

    R0_BANK0/1–R7_BANK0/1, R8–R15, GBR, SPC, SSR, MACH, MACL, PR
  - Initialized registers

    VBR = H'00000000

    SR.MD = 1, SR.BL = 1, SR.RB = 1, SR.I3–SR.I0 = H'F. Other SR bits are undefined.

    PC = H'A0000000
- Ensure that an exception is not generated at an RTE instruction delay slot, as operation is not guaranteed in this case.

Section 5 Cache

5.1 Overview

5.1.1 Features

The cache specifications are listed in table 5.1.

Table 5.1 Cache Specifications

Parameter	Specification
Capacity	Selectable: Normal mode: 8 kbytes RAM mode: 4 kbytes cache and 4 kbytes RAM
Structure	Instruction/data mixed, 4-way set associative (2-way set associative in RAM mode)
Line size	16 bytes
Number of entries	128 entries/way
Write system	P0, P1, P3, U0: Write-back/write-through selectable
Replacement method	Least-recently-used (LRU) algorithm

5.1.2 Cache Structure

The cache mixes data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section. Each of the address and data sections is divided into 128 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 2 kbytes (16 bytes × 128 entries), with a total of 8 kbytes in the cache as a whole (4 ways). Figure 5.1 shows the cache structure.

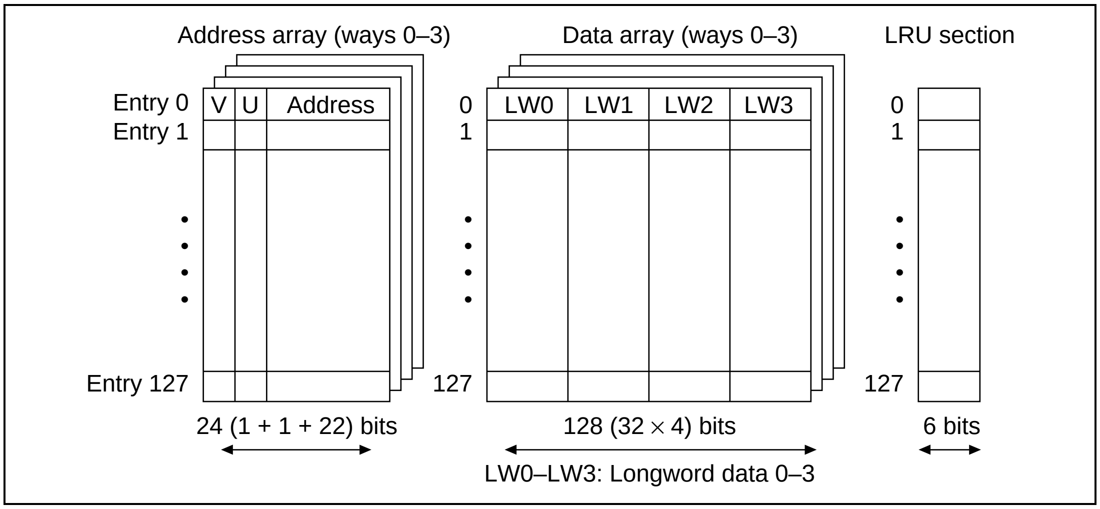


Figure 5.1 Cache Structure

Address Array: The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid. The U bit indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not. The address tag holds the physical address used in the external memory access. It is composed of 22 bits (address bits 31–10) used for comparison during cache searches.

In the SH7709, the top three of the 32 physical address bits are used as shadow bits (see section 10), and therefore in a normal replace operation the top three bits of the vector address are cleared to 0.

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

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

LRU: With the 4-way set associative system, up to four instructions or data with the same entry address (address bits 10–4) can be registered in the cache. When an entry is registered, the LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way.

In normal mode, four ways are used as cache and six LRU bits indicate the way to be replaced (table 5.2). If a bit pattern other than those listed in table 5.2 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 5.2.

In RAM mode, two ways are used as cache (way 0 and way 1). Bit 5 of the LRU bits indicates which way is to be replaced. When bit 5 is 0, way 1 is to be replaced. When bit 5 is 1, way 0 is to be replaced.

The LRU bits are initialized to 0 by a power-on reset, but are not initialized by a manual reset.

Table 5.2 LRU and Way Replacement in Normal Mode

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

5.1.3 Register Configuration

Table 5.3 shows details of the cache control register.

Table 5.3 Register Configuration

Register Name	Abbr.	R/W	Size	Initial Value	Address
Cache control register	CCR	R/W	Longword	H'00000000	H'FFFFFFEC

5.2 Register Description

5.2.1 Cache Control Register (CCR)

The cache is enabled or disabled using the CE bit of the cache control register (CCR). CCR also has an RA bit (which switches the cache operation mode between RAM mode and normal mode), a CF bit (which invalidates all cache entries), a CB bit (which selects either write-through mode or write-back mode of P1), and a WT bit (which selects either write-through mode or write-back mode of P0, U0 and P3). Programs that change the contents of the CCR register should be placed in address space that is not cached. When updating the contents of the CCR register, always set bit 4 to 0. Figure 5.2 shows the configuration of the CCR register.

31	6	5	4	3	2	1	0
— ···	··· ··· ··· ··· ··· ··· ··· ···	RA	0	CF	CB	WT	CE

RA: RAM bit. Indicates the cache operation mode.
1 = 4 kbytes cache/4 kbytes cache (RAM mode)
0 = 8 kbytes cache (normal mode)
0: Always set to 0 when setting the register.
CF: Cache flush bit. Invalidates all cache entries. 1 = flush (clears the V, U, and LRU bits of all entries to 0).
Always reads 0. Write-back to external memory is not performed when the cache is flushed.
CB: Cache write-back bit. Indicates the cache's operating mode for area P1.
1 = write-back mode, 0 = write-through mode.
WT: Write-through bit. Indicates the cache’s operating mode for areas P0, U0, and P3.
1 = write-through mode, 0 = write-back mode.
CE: Cache enable bit. Indicates whether the cache function is used.
1 = cache used, 0 = cache not used.

Figure 5.2 CCR Register Configuration

5.3 Cache Operation

5.3.1 Searching the Cache

If the cache is enabled, whenever instructions or data in memory are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 5.3 illustrates the method by which the cache is searched. The cache is a physical cache and holds physical addresses in its address section.

Entries are selected using bits 10–4 of the address (virtual) of the access to memory and the address tag of that entry is read. In parallel to reading of the address tag, the virtual address is translated to a physical address in the MMU. The physical address after translation and the physical address read from the address section are compared. The address comparison uses all four ways in normal mode. In RAM mode, two ways (way 0 and way 1) are used in the address comparison. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 5.3 shows a hit on way 1.

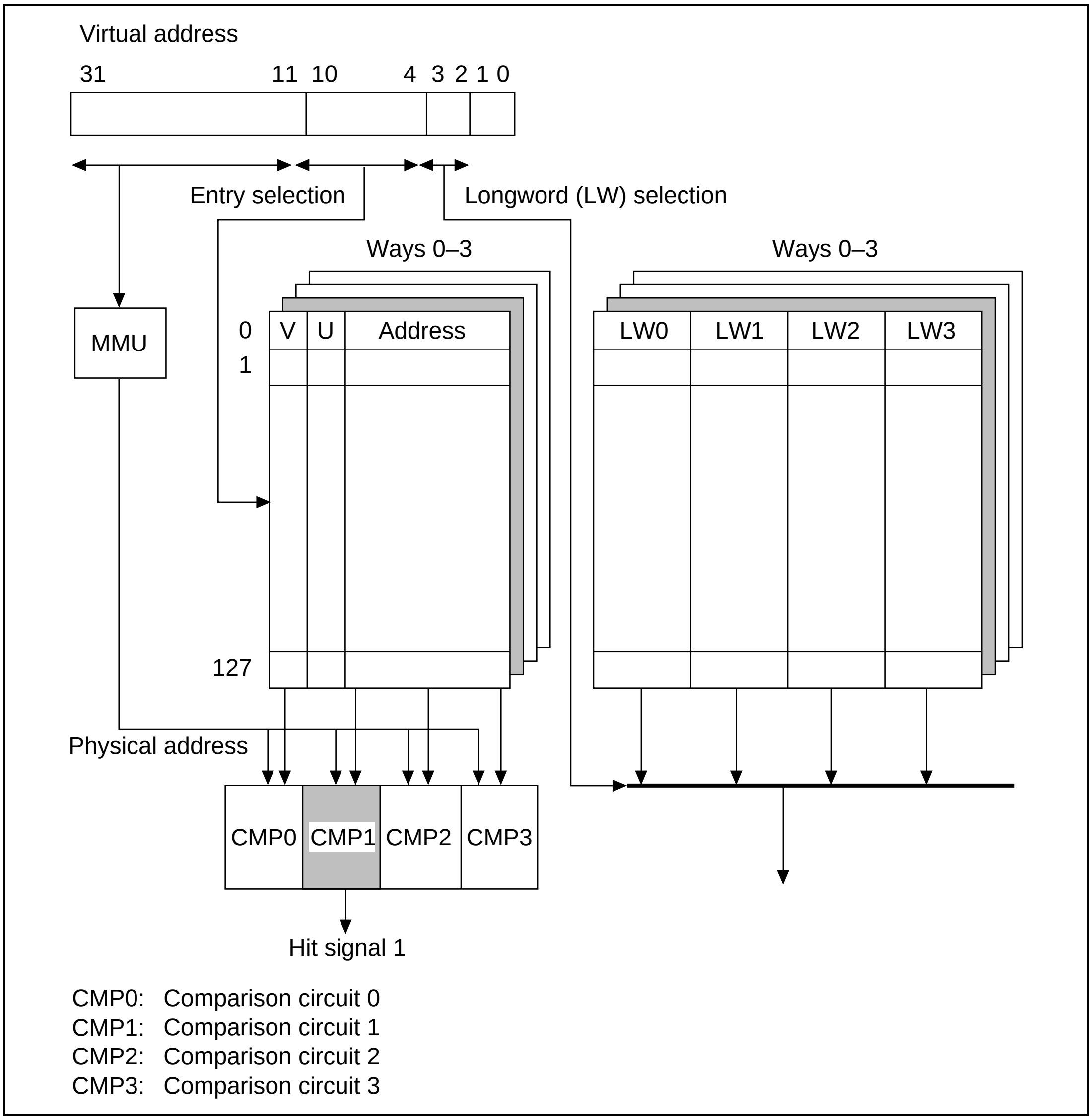


Figure 5.3 Cache Search Scheme (Normal Mode)

5.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. The transfer unit is 32 bits. The LRU is updated.

Read Miss: An external bus cycle starts and the entry is updated. The way replaced is the one least recently used. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the U bit is cleared to 0 and the V bit is set to 1. When the U bit of a replaced entry in write-back mode is 1, the cache fill cycle starts after the entry is transferred to the write-back buffer. After the cache completes its fill cycle, the write-back buffer writes back the entry to memory. The write-back unit is 16 bytes.

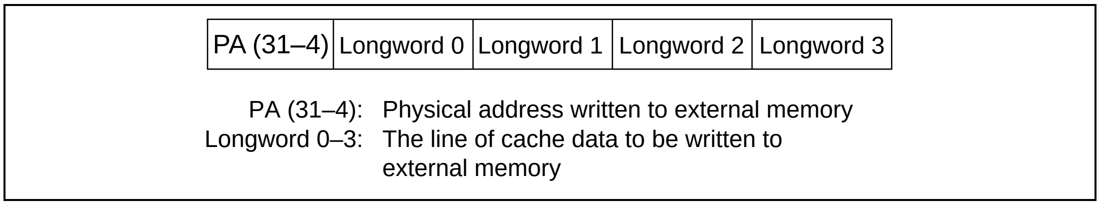
5.3.3 Write Access

Write Hit: In a write access in write-back mode, data is written to the cache and the U bit of the entry written is set to 1. Writing occurs only to the cache; no external memory write cycle is issued. In write-through mode, data is written to the cache and an external memory write cycle is issued.

Write Miss: In write-back mode, an external bus cycle starts when a write miss occurs and an entry with its U bit set to 1 is replaced. The way to be replaced is the one least recently used. When the U bit of the entry to be replaced is 1, the cache fill cycle starts after the entry is transferred to the write-back buffer. After the cache completes its fill cycle, the write-back buffer writes back the entry to memory. The write-back unit is 16 bytes. Data is written to the cache and the U bit is set to 1. In write-through mode, no write to cache occurs in a write miss; the write is only to external memory.

5.3.4 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, it must be written back to external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to external memory. During the write back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 5.4 shows the configuration of the write-back buffer.



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

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

Figure 5.4 Write-Back Buffer Configuration

5.3.5 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by the SH7709 and another device is accessed, the latest data may be in a write-back mode cache, so invalidate the entry that includes the latest data in the cache, generate a write back, and update the data in memory before using it. When the caching area is updated by a device other than the SH7709, invalidate the entry that includes the updated data in the cache.

5.3.6 RAM Mode

In RAM mode, way 0 and way 1 function as a 4-kbyte two-way set associative cache, while way 2 and way 3 function as 4-kbyte internal RAM. The internal RAM is mapped onto H'7F000000 to H'7F000FFF with 4-kbyte shadow areas from H'7F001000 to H'7FFFFFFF. In RAM mode with the MMU enabled, a virtual address from H'7F000000 to H'7FFFFFFF is not translated to an external physical address. The internal RAM can be accessed in both privileged and user mode by setting its address as source or destination address in the instructions. Before changing the RA bit in the CCR register to change the cache operation mode, all entries in the cache should be invalidated.

5.4 Memory-Mapped Cache

To allow software management of the cache, it is mapped onto virtual address space P4. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF and the data array onto addresses H'F1000000 to H'F1FFFFFF. In privileged mode, the cache contents can be read or written using the MOV instruction.

5.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the address, V bit, U bit, and LRU bits to be written to the address array (figure 5.5 (1)).

In the address field, specify the entry address for selecting the entry (bits 10–4), W for selecting the way (bits 12–11: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3 in normal mode (8-kbyte cache); 00 and 10 are way 0, and 01 and 11 are way 1 in RAM mode), and H'F0 to indicate address array access (bits 31–24).

When writing, specify bit 3 as the A bit. The A bit indicates whether addresses are compared during writing. When the A bit is 1, the addresses of the four entries selected by the entry addresses are compared to the addresses to be written into the address array specified in the data field. Writing takes place to the way that has a hit. When a miss occurs, nothing is written to the address array and no operation occurs. The way number (W) specified in bits 12–11 is not used. When the A bit is 0, it is written to the entry selected with the entry address and way number without comparing addresses. The address specified by bits 31–10 in the data specification in figure 5.5 (1), address array access, is a virtual address. When the MMU is enabled, the address is translated into a physical address, then the physical address is used in comparing addresses when the A bit is 1. The physical address is written into the address array.

When reading, the address tag, V bit, U bit, and LRU bits of the entry specified by the entry address and way number (W) are read using the data format shown in figure 5.5 without comparing addresses. To invalidate a specific entry, specify the entry by its entry address and way number, and write 0 to its V bit. To invalidate only an entry for an address to be invalidated, specify 1 for the A bit.

When an entry for which 0 is written to the V bit has a U bit set to 1, if it is a valid entry it will be written back. This allows coherency to be achieved between the external memory and cache by invalidating the entry. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

In the SH7709, the top 3 bits of the 32-bit physical address are treated as a shadow (see section 10, Bus State Controller (BSC)). Therefore, in the event of a cache miss, 0 is registered in the top 3 bits of the address array address tag.

When directly changing the address array using the MOV instruction, also, a value other than 0 must not be set in the top 3 bits of the address tag.

5.4.2 Data Array

The data array is mapped onto H'F1000000 to H'F1FFFFFF. To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array (figure 5.5 (2)).

In the address field, specify the entry address for selecting the entry (bits 10–4), L for indicating the longword position within the (16 byte) line (bits 3–2: 00 is longword 0, 01 is longword 1, 10 is longword 2, 11 is longword 3), W for selecting the way (bits 12–11: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3 in normal mode; 00 and 10 are way 0, and 01 and 11 are way 1 in RAM mode), and H'F1 to indicate data array access (bits 31–24).

Both reading and writing use the longword of the data array specified by the entry address, way number and longword address. The access size of the data array is fixed at longword.

(1) Address array access

Address specification

Read access

31 24	23 13	12 11	10 4	3	2 0
11110000	*——*	W	Entry	0	* * *

Write access

31 24	23 13	12 11	10 4	3	2 0
11110000	*——*	W	Entry	A	* * *

Data specification (both read and write accesses)

31 10	9 4	3 2	1	0
Address tag (31–10)	LRU	X X	U	V

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

Address specification

31 24	23 13	12 11	10 4	3 2	1 0
11110001	*——*	W	Entry	L	* *

Data specification

31 0
Longword

X: 0 for read, don’t care bit for write
*: Don’t care bit

Figure 5.5 Specifying Address and Data for Memory-Mapped Cache Access

5.5 Usage Examples

5.5.1 Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit. When the A bit is 1, the address tag specified by the write data is compared to the address tag within the cache selected by the entry address, and data is written when a match is found. If no match is found, there is no operation. R0 specifies the write data in R0 and R1 specifies the address. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1.

```
; R0=H'01100010; VPN=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F0000088; address array access, entry=B'0001000, A=1
;
MOV.L R0,@R1
```

5.5.2 Reading the Data of a Specific Entry

This example reads the data section of a specific cache entry. The longword indicated in the data field of the data array in figure 5.5 is read to the register. R0 specifies the address and R1 is read.

```
; R1=H'F100 004C; data array access, entry=B'0000100, Way = 0,
longword ; address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 6 Interrupt Controller (INTC)

6.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

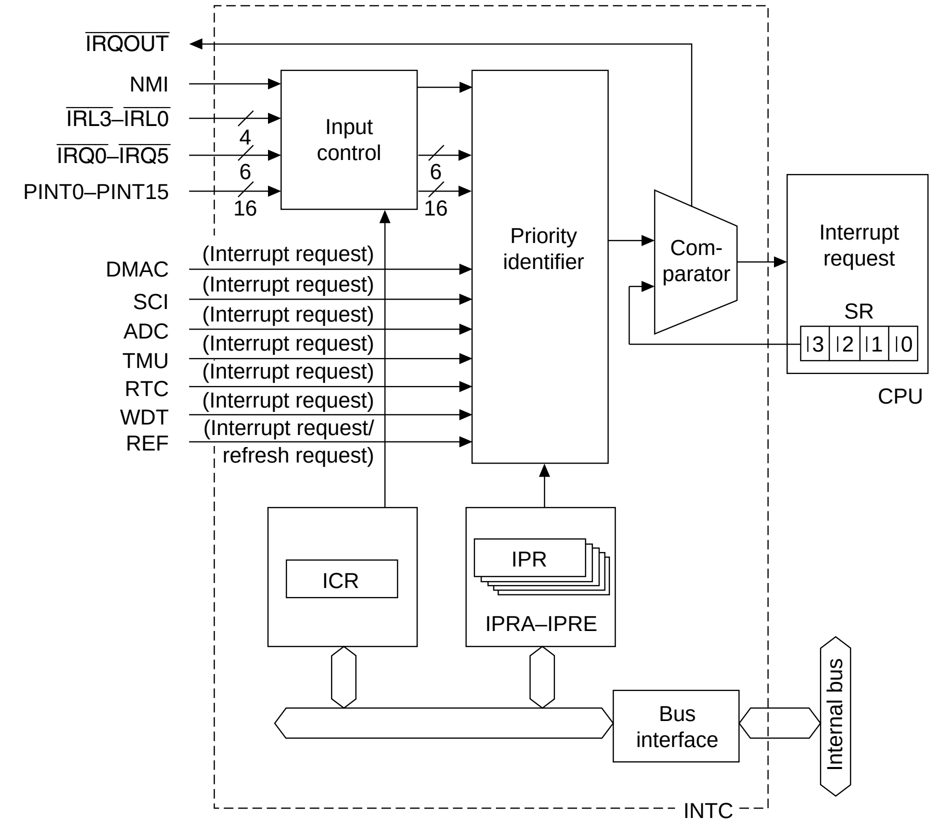
6.1.1 Features

INTC has the following features:

- 16 levels of interrupt priority can be set: By setting the six interrupt-priority registers, the priorities of on chip peripheral module interrupts can be selected from 16 levels for different request sources.
- NMI noise canceler function: NMI input level bit indicates NMI pin status. By reading this bit in the interrupt exception service routine, the pin status can be checked, enabling it to be used as a noise canceler.
- External devices can be notified that an interrupt has been received (IRQOUT): For example, when the SH7709 has released the bus right, the external bus master can be notified the fact that an external interrupt, an on-chip peripheral module interrupt or a memory refresh request has occurred, enabling this LSI to request the bus right.

6.1.2 Block Diagram

Figure 6.1 is a block diagram of the INTC.



TMU:	Timer unit
RTC:	Realtime clock unit
SCI:	Serial communication interface
WDT:	Watch dog timer
REF:	Memory refresh controller section of the bus state controller
ICR:	Interrupt control register
IPRA-IPRE:	Registers A-E for setting the interrupt proprity levels
SR:	Status register
DMAC:	Direct memory access controller
ADC:	Analog-to-digital converter

Figure 6.1 INTC Block Diagram

6.1.3 Pin Configuration

Table 6.1 lists the INTC pin configuration.

Table 6.1 Pin Configuration

Pin Name	Symbol	I/O	Description
Nonmaskable interrupt input pin	NMI	I	Input of interrupt request signal, which is nonmaskable by SR.IMASK
Interrupt input pins	IRQ0–IRQ5 $\overline{\text{IRL0}}$–$\overline{\text{IRL3}}$	I	Input of interrupt request signals, which is maskable by SR.IMASK
Port interrupt input pins	PINT0–PINT15	I	Port input of interrupt request signals, which is maskable by SR.IMASK
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of signal that notifies external devices that an interrupt source or memory refresh has occurred

6.1.4 Register Configuration

The INTC has the 15 registers listed in table 6.2.

Table 6.2 Register Configuration

Register Name	Abbr.	R/W	Initial Value*1	Address	Access Size
Interrupt control register 0	ICR0	R/W	*2	H'FFFFFEE0	16
Interrupt control register 1	ICR1	R/W	H'0000	H'4000010	16
Interrupt control register 2	ICR2	R/W	H'0000	H'4000012	16
INT interrupt enable register	INTER	R/W	H'0000	H'4000014	16
Interrupt priority level setting register A	IPRA	R/W	H'0000	H'FFFFFEE2	16
Interrupt priority level setting register B	IPRB	R/W	H'0000	H'FFFFFEE4	16
Interrupt priority level setting register C	IPRC	R/W	H'0000	H'4000016	16
Interrupt priority level setting register D	IPRD	R/W	H'0000	H'4000018	16
Interrupt priority level setting register E	IPRE	R/W	H'0000	H'400001A	16
Interrupt request register 0	IRR0	R	H'00	H'4000004	8
Interrupt request register 1	IRR1	R	H'00	H'4000006	8
Interrupt request register 2	IRR2	R	H'00	H'4000008	8

Notes: 1. Initialized by a power-on or manual reset.

2. H'8000 when the NMI pin is at high level. H'0000 when the NMI pin is at low level.

6.2 Interrupt Sources

There are four types of interrupt sources: NMI, IRQ, PINT, and on-chip peripheral modules. Each interrupt has priority levels (0–16) with 0 the lowest and 16 the highest. Priority level 0 masks an interrupt.

6.2.1 NMI Interrupts

The NMI interrupt has the highest priority level of 16. When the BLMSK bit of the interrupt control register (ICR1) is 1 or the BL bit of the status register (SR) is 0, NMI interrupts are accepted when the MAI bit of the ICR1 register is 0. NMI interrupts are edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL. The NMI edge select bit (NMIE) in the interrupt control register0 (ICR0) is used to select either the rising or falling edge. When the NMIE bit of the ICR0 register is changed, the NMI interrupt is not detected for 20 cycles after changing the ICR.NMIE to avoid a false detection of the NMI interrupt. NMI interrupt exception processing does not affect the interrupt mask level bits (I3–I0) in the status register (SR).

When the BLMSK bit of the ICR1 register is set to 1 and only NMI interrupts are accepted, the SPC register and SSR register are updated by the NMI interrupt handler, making it impossible to return to the original processing from exception handling initiated prior to the NMI. Use should therefore be restricted to cases where return is not necessary.

It is possible to wake the chip up from the standby state with an NMI interrupt (except when the MAI bit of the ICR1 register is set to 1).

6.2.2 IRQ Interrupt

IRQ interrupts are input by priority from pins IRQ0–IRQ5 with a level or an edge. The priority level can be set by priority setting registers C–D (IPRC–IPRD) in a range from levels 0–15.

When edge-sensing is used for an IRQ interrupt, clear the corresponding bit in IRR0 to 0 by software to clear the interrupt source.

When the ICR1 register is rewritten, IRQ interrupts may be mistakenly detected, depending on the pin states. To prevent this, rewrite the register while interrupts are masked, then release the mask after clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

It is necessary for an edge input interruopt detection to input a pulse width more than 2 cycle width by P clock basis.

The interrupt mask bits (I3–I0) of the status register (SR) are not affected by IRQ interrupt processing.

Interrupts IRQ4–IRQ0 and port interrupts (PINT0/1) can wake the chip up from the standby state when the relevant interrupt level is higher than I3–I0 in the SR register (but only when the RTC 32 kHz oscillator is used).

6.2.3 IRL Interrupts

IRL interrupts are input by level at pins $\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$. The priority level is the level indicated by pins $\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$. An $\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ value of 0 (0000) indicates the highest-level interrupt request (interrupt priority level 15). A value of 15 (1111) indicates no interrupt request (interrupt priority level 0). Figure 6.2 shows an examples of an IRL interrupt connection. Table 6.3 shows $\overline{\mathrm{IRL}}$ pins and interrupt levels.

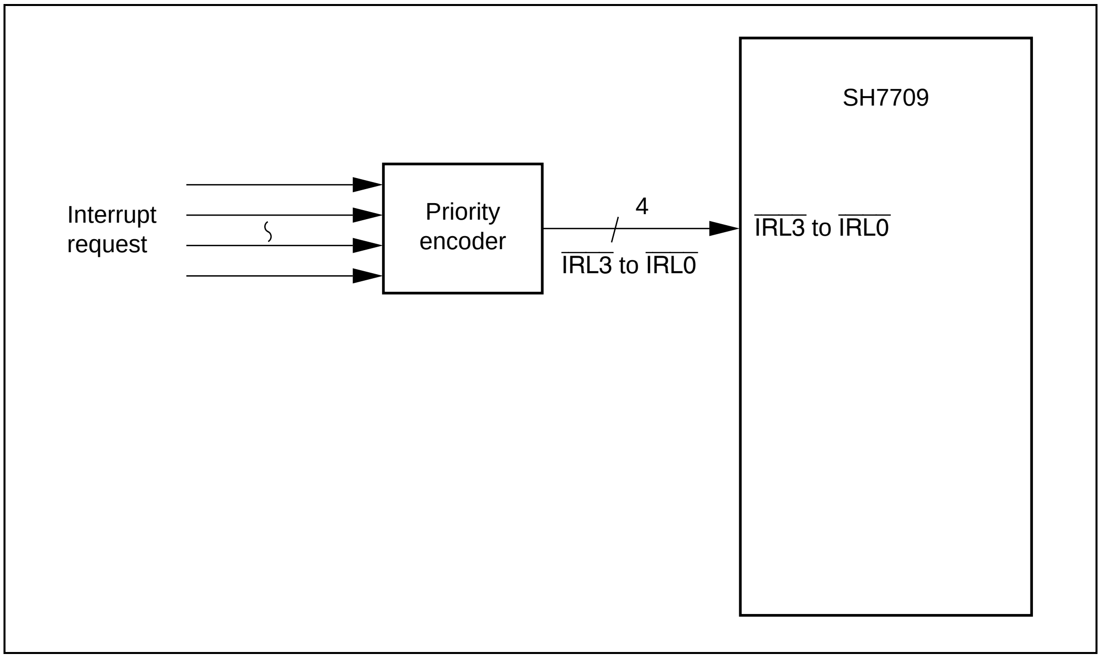

SH7709
Interrupt request
Priority encoder
4
$\overline{\mathrm{IRL3}}$ to $\overline{\mathrm{IRL0}}$
$\overline{\mathrm{IRL3}}$ to $\overline{\mathrm{IRL0}}$

Figure 6.2 Example of IRL Interrupt Connection

Table 6.3 $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ Pins and Interrupt Levels

$\overline{\text{IRL3}}$	$\overline{\text{IRL2}}$	$\overline{\text{IRL1}}$	$\overline{\text{IRL0}}$	**Interrupt Priority Level**	**Interrupt Request**
0	0	0	0	15	Level 15 interrupt request
0	0	0	1	14	Level 14 interrupt request
0	0	1	0	13	Level 13 interrupt request
0	0	1	1	12	Level 12 interrupt request
0	1	0	0	11	Level 11 interrupt request
0	1	0	1	10	Level 10 interrupt request
0	1	1	0	9	Level 9 interrupt request
0	1	1	1	8	Level 8 interrupt request
1	0	0	0	7	Level 7 interrupt request
1	0	0	1	6	Level 6 interrupt request
1	0	1	0	5	Level 5 interrupt request
1	0	1	1	4	Level 4 interrupt request
1	1	0	0	3	Level 3 interrupt request
1	1	0	1	2	Level 2 interrupt request
1	1	1	0	1	Level 1 interrupt request
1	1	1	1	0	No interrupt request

A noise-cancellation feature is built in, and the IRL interrupt is not detected unless the levels sampled at every supporting module cycle remain unchanged for two consecutive cycles, so that no transient level on the $\overline{\text{IRL}}$ pin change is detected. In standby mode, as the peripheral clock is stopped, noise cancellation is performed using the 32.768 kHz clock for the RTC instead. Therefore when the RTC is not used, interruption by means of IRL interrupts cannot be performed in standby mode.

The priority level of the IRL interrupt must not be lowered unless the interrupt is accepted and the interrupt handling starts. However, the priority level can be changed to a higher one.

The interrupt mask bits (I3–I0) in the status register (SR) are not affected by $\overline{\text{IRL}}$ interrupt handling.

6.2.4 PINT Interrupt

PINT interrupts are input by priority from pins INT0–INT5 with a level. The priority level can be set by priority setting registers D (IPRD) in a range from levels 0–15, in the unit of PINT0–PINT7 or PINT8–PINT15.

With a PINT interrupt, the level should be held until the interrupt is accepted and interrupt processing is started.

The interrupt mask bits (I3–I0) of the status register (SR) are not affected by INT interrupt processing.

6.2.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following nine modules:

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interface (SCI)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- Analog-to-digital converter (ADC)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected on the interrupt event register (INTEVT and INTEVT2). It is easy to identify sources by using the values of the INTEVT or INTEVT2 register as branch offsets (in the exception service routine).

The priority level (from 0–15) can be set for each module by writing to the interrupt priority setting registers A–B and E (IPRA–IPRB and IPRE).

The interrupt mask bits (I3–I0) of the status register are not affected by the on-chip peripheral module interrupt processing.

TMU and RTC interrupts can wake the chip up from the standby state when the relevant interrupt level is higher than I3—I0 in the SR register (but only when the RTC 32 kHz oscillator is used).

6.2.6 Interrupt Exception Processing and Priority

Table 6.4 lists the codes for the interrupt event register (INTEVT and INTEVT2), and the order of interrupt priority. Each interrupt source is assigned unique code. The start address of the interrupt service routine is common to each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as offset at the start of the interrupt service routine and branched to identify the interrupt source.

The order of priority of the on-chip peripheral module is set within the priority levels 0–15 at will by using the interrupt priority level set to registers A–E (IPRA–IPRE). The order of priority of the on-chip peripheral module is set to zero by RESET.

When the order of priorities for multiple interrupt sources are set to the same level and such interrupts are generated at the same time, they are processed according to the default order listed in table 6.4.

Table 6.4 Interrupt Exception Vectors and Rankings (IRQ Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		0x1C0 (0x1C0)	16	—	—	High
IRQ	IRQ0	0x200–3C0 (0x600)*1	0–15 (0)	IPRC (3–0)	—	↓
	IRQ1	0x200–3C0 (0x620)*1	0–15 (0)	IPRC (7–4)	—	
	IRQ2	0x200–3C0 (0x640)*1	0–15 (0)	IPRC (11–8)	—	
	IRQ3	0x200–3C0 (0x660)*1	0–15 (0)	IPRC (15–12)	—	
	IRQ4	0x200–3C0 (0x680)*1	0–15 (0)	IPRD (3–0)	—	
	IRQ5	0x200–3C0 (0x6A0)*1	0–15 (0)	IPRD (7-4)	—	
INT	PINT0	0x200–3C0 (0x700)*1	0–15 (0)	IPRD (15–12)	—	
	PINT1					
	PINT2					
	PINT3					
	PINT4					
	PINT5					
	PINT6					
	PINT7					
	PINT8	0x200–3C0 (0x720)*1	0–15 (0)	IPRD (11–8)	—	
	PINT9					
	PINT10					
	PINT11					
	PINT12					
	PINT13					
	PINT14					
	PINT15					
DMAC	DEI0	0x200–3C0 (0x800)*1	0–15 (0)	IPRE (15–12)	High	
	DEI1	0x200–3C0 (0x820)*1				
	DEI2	0x200–3C0 (0x840)*1				
	DEI3	0x200–3C0 (0x860)*1			Low	

Table 6.4 Interrupt Exception Vectors and Rankings (IRQ Mode) (cont)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
IrDA	ERI1	0x200–3C0 (0x880)*[1]	0–15 (0)	IPRE (11–8)	High	
	RXI1	0x200–3C0 (0x8A0)*[1]				
	BRI1	0x200–3C0 (0x8C0)*[1]				
	TXI1	0x200–3C0 (0x8E0)*[1]			Low	
SCIF	ERI2	0x200–3C0 (0x900)*[1]	0–15 (0)	IPRE (7–4)	High	
	RXI2	0x200–3C0 (0x920)*[1]				
	BRI2	0x200–3C0 (0x940)*[1]				
	TXI2	0x200–3C0 (0x960)*[1]			Low	
A/D	ADI	0x200–3C0 (0x980)*[1]	0–15 (0)	IPRE (3–0)	—	
TMU0	TUNI0	0x400 (0x400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	0x420 (0x420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	0x440 (0x440)	0–15 (0)	IPRA (7–4)	High	
	TICPI2	0x460 (0x460)			Low	
RTC	ATI	0x480 (0x480)	0–15 (0)	IPRA (3–0)	High	
	PRI	0x4A0 (0x4A0)			↓	
	CUI	0x4C0 (0x4C0)			Low	
SCI0	ERI	0x4E0 (0x4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	0x500 (0x500)			↓	
	TXI	0x520 (0x520)			↓	
	TEI	0x540 (0x540)			Low	
WDT	ITI	0x560 (0x560)	0–15 (0)	IPRB (15–12)	—	
REF	RCMI	0x580 (0x580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	0x5A0 (0x5A0)			Low	Low

Note: The code corresponding to an interrupt level shown in table 6.6 is set.

Table 6.5 Interrupt Exception Vectors and Rankings (IRL Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		0x1C0 (0x1C0)	16	—	—	High
IRL	$\overline{\text{IRL(3:0)}}$ = 0000	0x200 (0x200)	15	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0001	0x220 (0x220)	14	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0010	0x240 (0x240)	13	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0011	0x260 (0x260)	12	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0100	0x280 (0x280)	11	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0101	0x2A0 (0x2A0)	10	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0110	0x2C0 (0x2C0)	9	—	—	
	$\overline{\text{IRL(3:0)}}$ = 0111	0x2E0 (0x2E0)	8	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1000	0x300 (0x300)	7	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1001	0x320 (0x320)	6	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1010	0x340 (0x340)	5	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1011	0x360 (0x360)	4	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1100	0x380 (0x380)	3	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1101	0x3A0 (0x3A0)	2	—	—	
	$\overline{\text{IRL(3:0)}}$ = 1110	0x3C0 (0x3C0)	1	—	—	
IRQ	IRQ4	0x200–3C0 (0x680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	0x200–3C0 (0x6A0)	0–15 (0)	IPRD (7–4)	—	Low

Table 6.4 Interrupt Exception Vectors and Rankings (cont)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
INT	PINT0	0x200–3C0 (0x700)	0–15 (0)	IPRD (15–12)	—	High
	PINT1					
	PINT2					
	PINT3					
	PINT4					
	PINT5					
	PINT6					
	PINT7					
	PINT8	0x200–3C0 (0x720)	0–15 (0)	IPRD (11–8)	—	
	PINT9					
	PINT10					
	PINT11					
	PINT12					
	PINT13					
	PINT14					
	PINT15					
DMAC	DEI0	0x200–3C0 (0x800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	0x200–3C0 (0x820)				
	DEI2	0x200–3C0 (0x840)				
	DEI3	0x200–3C0 (0x860)			Low	
IrDA	ERI1	0x200–3C0 (0x880)	0–15 (0)	IPRE (11–8)	High	
	RXI1	0x200–3C0 (0x8A0)				
	BRI1	0x200–3C0 (0x8C0)				
	TXI1	0x200–3C0 (0x8E0)			Low	Low

Table 6.4 Interrupt Exception Vectors and Rankings (cont)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
SCIF	ERI2	0x200–3C0 (0x900)	0–15 (0)	IPRE (7–4)	High	High
	RXI2	0x200–3C0 (0x920)				
	BRI2	0x200–3C0 (0x940)				
	TXI2	0x200–3C0 (0x960)			Low	
A/D	ADI	0x200–3C0 (0x980)	0–15 (0)	IPRE (3–0)	—	
TMU0	TUNI0	0x400 (0x400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	0x420 (0x420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	0x440 (0x440)	0–15 (0)	IPRA (7–4)	High	
	TICPI2	0x460 (0x460)			Low	
RTC	ATI	0x480 (0x480)	0–15 (0)	IPRA (3–0)	High	
	PRI	0x4A0 (0x4A0)			↓	
	CUI	0x4C0 (0x4C0)			Low	
SCI0	ERI	0x4E0 (0x4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	0x500 (0x500)			↓	
	TXI	0x520 (0x520)			↓	
	TEI	0x540 (0x540)			Low	
WDT	ITI	0x560 (0x560)	0–15 (0)	IPRB (15–12)	—	
REF	RCMI	0x580 (0x580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	0x5A0 (0x5A0)			Low	Low

Table 6.6 Interrupt Exception Vectors and Rankings (IRL Mode)

Interrupt	INTEVT Code
15	H’200
14	H’220
13	H’240
12	H’260
11	H’280
10	H’2A0
9	H’2C0
8	H’2E0
7	H’300
6	H’320
5	H’340
4	H’360
3	H’380
2	H’3A0
1	H’3C0

6.3 INTC Registers

6.3.1 Interrupt Priority Registers A to E (IPRA–IPRE)

Interrupt priority registers A to E (IPRA to IPRE) are 16-bit read/write registers that set priority levels from 0 to 15 for on-chip peripheral module interrupts. These registers are initialized to H'0000 at power-on reset, manual reset, or in hardware standby mode, but is not initialized in standby mode.

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

Table 6.5 lists the relationship between the interrupt sources and the IPRA and IPRE bits.

Table 6.5 Interrupt Request Sources and IPRA–IPRE

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SCI0	Reserved *
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	PINT0 to PINT7	PINT8 to PINT15	IRQ5	IRQ4
IPRE	DMAC	IrDA	SCIF	A/D

Notes Reserved bits: Always read as 0. Only 0 should be written in.

As listed in table 6.5, four sets of on-chip peripheral modules are assigned to each register. 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested); H'F is priority level 15 (the highest level). A reset initializes IPRA–IPRE to H'0000.

H'0000 should be set into bits corresponding to an unused interrupt.

6.3.2 Interrupt Control Register 0 (ICR0)

The ICR0 is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and indicates the input signal level to the NMI pin. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	NMIL	—	—	—	—	—	—	NMIE
Initial value:	0/1*	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Note: When NMI input is high: 1; when NMI input is low: 0.

Bit 15—NMI Input Level (NMIL): Sets 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.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

Bit 8—NMI Edge Select (NMIE): Selects whether the falling or rising edge of the interrupt request signal to the NMI is detected.

Bit 8: NMIE	Description
0	Interrupt request is detected on the falling edge of NMI input
1	Interrupt request is detected on rising edge of NMI input

Bits 14–9 and 7–0—Reserved: Writing is invalid. Always read as 0.

6.3.3 Interrupt Control Register 1 (ICR1)

The ICR1 is a 16-bit register that specifies the detection mode to external interrupt input pins, IRQ0 to IRQ5 indivisually: rising edge, falling edge, or low level. This register, initialized to H'0000 at power-on reset or manual reset, is not initialized in the standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
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 15—Mask All Interrupts (MAI): Masks NMI interrupts when set to 1. Also selects whether or not all interrupt requests are masked when a low level is being input to the NMI pin.

Bit 15: MAI	Description
0	All interrupt requests are not masked (initial value)
1	All interrupt requests are masked

Bit 14—Interrupt Request Level Detect (IRQLVL): Selects whether the IRQ3–IRQ0 pins are used as four independent interrupt pins or as 15-level interrupt pins encoded as $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$.

Bit 14: IRQLVL	Description
0	Used as four independent interrupt request pins IRQ3–IRQ0
1	Used as encoded 15-level interrupt pins as $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ (initial value)

Bit 13—BL Bit Mask (BLMSK): Specifies whether NMI interrupts are masked when the BL bit of the SR register is 1.

Bit 13: BLMSK	Description
0	NMI interrupts are masked when the BL bit is 1 (initial value)
1	NMI interrupts are accepted regardless of the BL bit setting

Bit 12—Reserved. Bit 12 cannot be modified. It always reads 0.

Bits 11 and 10—IRQ5 Sense Select (IRQ51S and IRQ50S): Selects whether the interrupt signal to the IRQ5 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 11: IRQ51S	Bit 10: IRQ50S	Description
0	0	An interrupt request is detected at IRQ5 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ5 input rising edge
1	0	An interrupt request is detected at IRQ5 input low level
1	1	Reserved

Bits 9 and 8—IRQ4 Sense Select (IRQ41S and IRQ40S): Selects whether the interrupt signal to the IRQ4 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 9: IRQ41S	Bit 8: IRQ40S	Description
0	0	An interrupt request is detected at IRQ4 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ4 input rising edge
1	0	An interrupt request is detected at IRQ4 input low level
1	1	Reserved

Bits 7 and 6—IRQ3 Sense Select (IRQ31S and IRQ30S): Selects whether the interrupt signal to the IRQ3 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 7: IRQ31S	Bit 6: IRQ30S	Description
0	0	An interrupt request is detected at IRQ3 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ3 input rising edge
1	0	An interrupt request is detected at IRQ3 input low level
1	1	Reserved

Bits 5 and 4—IRQ2 Sense Select (IRQ21S and IRQ20S): Selects whether the interrupt signal to the IRQ2 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 5: IRQ21S	Bit 4: IRQ20S	Description
0	0	An interrupt request is detected at IRQ2 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ2 input rising edge
1	0	An interrupt request is detected at IRQ2 input low level
1	1	Reserved

Bits 3 and 2—IRQ1 Sense Select (IRQ11S and IRQ10S): Selects whether the interrupt signal to the IRQ1 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 3: IRQ11S	Bit 2: IRQ10S	Description
0	0	An interrupt request is detected at IRQ1 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ1 input rising edge
1	0	An interrupt request is detected at IRQ1 input low level
1	1	Reserved

Bits 1 and 0—IRQ0 Sense Select (IRQ01S and IRQ00S): Selects whether the interrupt signal to the IRQ0 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 1: IRQ01S	Bit 0: IRQ00S	Description
0	0	An interrupt request is detected at IRQ0 input falling edge (initial value)
0	1	An interrupt request is detected at IRQ0 input rising edge
1	0	An interrupt request is detected at IRQ0 input low level
1	1	Reserved

6.3.4 Interrupt Control Register 2 (ICR2)

The ICR2 is a 16-bit read/write register that sets the detection mode to external interrupt input pins PINT0 to PINT15. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in software standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	PINT15S	PINT14S	PINT13S	PINT14S	PINT11S	PINT10S	PINT9S	PINT8S
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:	7	6	5	4	3	2	1	0
Bit name:	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
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

Bits 15–0—PINT15 to PINT0 Sense Select (PINT15S to PINT0S): Selects whether interrupt request signals to PINT15 to PINT0 are detected at low levels or high levels. .

Bit 15–0: PINT15S to PINT0S	**Description**
0	Interrupt requests are detected at low level input to the PINT pins (initial value)
1	Interrupt requests are detected at high level input to the PINT pins

6.3.5 PINT Interrupt Enable Register (PINTER)

The PINTER is a 16-bit read/write register that enables interrupt requests input to external interrupt input pins PINT0 to PINT15. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	PINT15E	PINT14E	PINT13E	PINT14E	PINT11E	PINT10E	PINT9E	PINT8E
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:	7	6	5	4	3	2	1	0
Bit name:	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
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

Bits 15–0—PINT15 to PINT0 interrupt enable (PINT15E to PINT0E): Enables whether the interrupt requests input to the PINT15 to PINT0 pins.

Bit 15–0: PINT15E to PINT0E	**Description**
0	Disables PINT input interrupt requests (initial value)
1	Enables PINT input interrupt requests

When all or some of these pins, PINT0 - PINT15 are not used as an interrupt input, a bit corresponding to an terminal unused as an interrupt request should be set to 0.

6.3.6 Interrupt Request Register 0 (IRR0)

The IRR0 is an 8-bit register that indicates interrupt requests from external input pins IRQ0 to IRQ5 and PINT0 to PINT15. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PINT0R	PINT1R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R
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

When clearing IRQnR bit to 0, 0 should be written to the bit after reading the content of 1 after it has been set to 1. Only 0 can be written to IRQnR.

Bit 7—PINT0 to PINT7 interrupt request (PINT0R): Indicates whether interrupt requests are input to PINT0 to PINT7 pins.

Bit 7: PINT0R	Description
0	Interrupt requests are not input to PINT0 to PINT7 pins (initial value)
1	Interrupt requests are input to PINT0 to PINT7 pins

Bit 6—PINT8 to PINT15 interrupt request (PINT1R): Indicates whether interrupt requests are input to PINT8 to PINT15 pins.

Bit 6: PINT1R	Description
0	Interrupt requests are not input to PINT8 to PINT15 pins (initial value)
1	Interrupt requests are input to PINT8 to PINT15 pins

Bit 5—IRQ5 interrupt request (IRQ5R): Indicates whether an interrupt request is input to IRQ5 pin. When edge detection mode is set for IRQ5, an interrupt request is cleared by clearing the IRQ5R bit.

Bit 5: IRQ5R	Description
0	An interrupt request is not input to IRQ5 pin (initial value)
1	An interrupt request is input to IRQ5 pin

Bit 4—IRQ4 interrupt request (IRQ4R): Indicates whether an interrupt request is input to IRQ4 pin. When edge detection mode is set for IRQ4, an interrupt request is cleared by clearing the IRQ4R bit.

Bit 4: IRQ4R	Description
0	An interrupt request is not input to IRQ4 pin (initial value)
1	An interrupt request is input to IRQ4 pin

Bit 3—IRQ3 interrupt request (IRQ3R): Indicates whether an interrupt request is input to IRQ3 pin. When edge detection mode is set for IRQ3, an interrupt request is cleared by clearing the IRQ3R bit.

Bit 3: IRQ3R	Description
0	An interrupt request is not input to IRQ3 pin (initial value)
1	An interrupt request is input to IRQ3 pin

Bit 2—IRQ2 interrupt request (IRQ2R): Indicates whether an interrupt request is input to IRQ2 pin. When edge detection mode is set for IRQ2, an interrupt request is cleared by clearing the IRQ2R bit.

Bit 2: IRQ2R	Description
0	An interrupt request is not input to IRQ2 pin (initial value)
1	An interrupt request is input to IRQ2 pin

Bit 1—IRQ1 interrupt request (IRQ1R): Indicates whether an interrupt request is input to IRQ1 pin. When edge detection mode is set for IRQ1, an interrupt request is cleared by clearing the IRQ1R bit.

Bit 1: IRQ1R	Description
0	An interrupt request is not input to IRQ1 pin (initial value)
1	An interrupt request is input to IRQ1 pin

Bit 0—IRQ0 interrupt request (IRQ0R): Indicates whether an interrupt request is input to IRQ0 pin. When edge detection mode is set for IRQ0, an interrupt request is cleared by clearing the IRQ0R bit.

Bit 0: IRQ0R	Description
0	An interrupt request is not input to IRQ0 pin (initial value)
1	An interrupt request is input to IRQ0 pin

6.3.7 Interrupt Request Register 1 (IRR1)

The IRR1 is an 8-bit read-only register that indicates whether DMAC or IrDA interrupt requests are generated. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in software mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TXI1R	BRI1R	RXI1R	ERI1R	DEI3R	DEI2R	DEI1R	DEI0R
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit 7—TXI1 interrupt request (TXI1R): Indicates whether a TXI1 (IrDA) interrupt request is generated.

Bit 7: TXI1	Description
0	A TXI1 interrupt request is not generated (initial value)
1	A TXI1 interrupt request is generated

Bit 6—BRI1 interrupt request (BRI1R): Indicates whether a BRI1 (IrDA) interrupt request is generated.

Bit 6: BRI1R	Description
0	A BRI1 interrupt request is not generated (initial value)
1	A BRI1 interrupt request is generated

Bit 5—RXI1 interrupt request (RXI1R): Indicates whether an RXI1 (IrDA) interrupt request is generated.

Bit 5: RXI1R	Description
0	An RXI1 interrupt request is not generated (initial value)
1	An RXI1 interrupt request is generated

Bit 4—ERI1 interrupt request (ERI1R): Indicates whether an ERI1 (IrDA) interrupt request is generated.

Bit 4: ERI1R	Description
0	An ERI1 interrupt request is not generated (initial value)
1	An ERI1 interrupt request is generated

Bit 3—DEI3 interrupt request (DEI3R): Indicates whether a DEI3 (DMAC) interrupt request is generated.

Bit 3: DEI3R	Description
0	A DEI3 interrupt request is not generated (initial value)
1	A DEI3 interrupt request is generated

Bit 2—DEI2 interrupt request (DEI2R): Indicates whether a DEI2 (DMAC) interrupt request is generated.

Bit 2: DEI2R	Description
0	A DEI2 interrupt request is not generated (initial value)
1	A DEI2 interrupt request is generated

Bit 1—DEI1 interrupt request (DEI1R): Indicates whether a DEI1 (DMAC) interrupt request is generated.

Bit 1: DEI1R	Description
0	A DEI1 interrupt request is not generated (initial value)
1	A DEI1 interrupt request is generated

Bit 0—DEI0 interrupt request (DEI0R): Indicates whether a DEI0 (DMAC) interrupt request is generated.

Bit 0: DEI0R	Description
0	A DEI0 interrupt request is not generated (initial value)
1	A DEI0 interrupt request is generated

6.3.8 Interrupt Request Register 2 (IRR2)

The IRR2 is an 8-bit read-only register that indicates whether A/D converter, or SCIF interrupt requests are generated. This register is initialized to H'00 at power-on reset, manual reset, or in hardware standby mode, but is not initialized in software standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	ADIR	TXI2R	BRI2R	RXI2R	ERI2R
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 to 5—Reserved bits: Writing is invalid. Always read as 0.

Bit 4—ADI interrupt request (ADIR): Indicates whether an ADI (ADC) interrupt request is generated.

Bit 4: ADIR	Description
0	An ADI interrupt request is not generated (initial value)
1	An ADI interrupt request is generated

Bit 3—TXI2 interrupt request (TXI2R): Indicates whether a TXI2 (SCIF) interrupt request is generated.

Bit 3: TXI2R	Description
0	A TXI2 interrupt request is not generated (initial value)
1	A TXI2 interrupt request is generated

Bit 2—BRI2 interrupt request (BRI2R): Indicates whether a BRI2 (SCIF) interrupt request is generated.

Bit 2: BRI2R	Description
0	A BRI2 interrupt request is not generated (initial value)
1	A BRI2 interrupt request is generated

Bit 1—RXI2 interrupt request (RXI2R): Indicates whether an RXI2 (SCIF) interrupt request is generated.

Bit 1: RXI2R	Description
0	An RXI2 interrupt request is not generated (initial value)
1	An RXI2 interrupt request is generated

Bit 0—ERI2 interrupt request (ERI2R): Indicates whether an ERI2 (SCIF) interrupt request is generated.

Bit 0: ERI2R	Description
0	An ERI2 interrupt request is not generated (initial value)
1	An ERI2 interrupt request is generated

6.4 INTC Operation

6.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers A to E (IPRA to IPRE). 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 default priority or the highest priority within its IPR setting unit (as indicated in table 6.4) is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3–I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller receives an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU receives an interrupt at a break in instructions.
6. The interrupt source code is set in the interrupt event register (INTEVT, INTEVT2).
7. The status register (SR) and program counter (PC) are saved to SSR and SPC, respectively.
8. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
9. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). The interrupt handler may branch with the INTEVT register value as its offset in order to identify the interrupt source. This enables it to branch to the handling routine for the individual interrupt source.

Notes: 1. The interrupt mask bits (I3–I0) in the status register (SR) are not changed by acceptance of an interrupt in the SH7709.

2. $\overline{\text{IRQOUT}}$ outputs a low level until the interrupt request is cleared. However, if the interrupt source is masked by an interrupt mask bit, the $\overline{\text{IRQOUT}}$ pin returns to the high level. The level is output without regard to the BL bit.

3. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, then wait for the interval shown in table 6.6 (Time for priority decision and SR mask bit comparison) before clearing the BL bit or executing an RTE instruction.

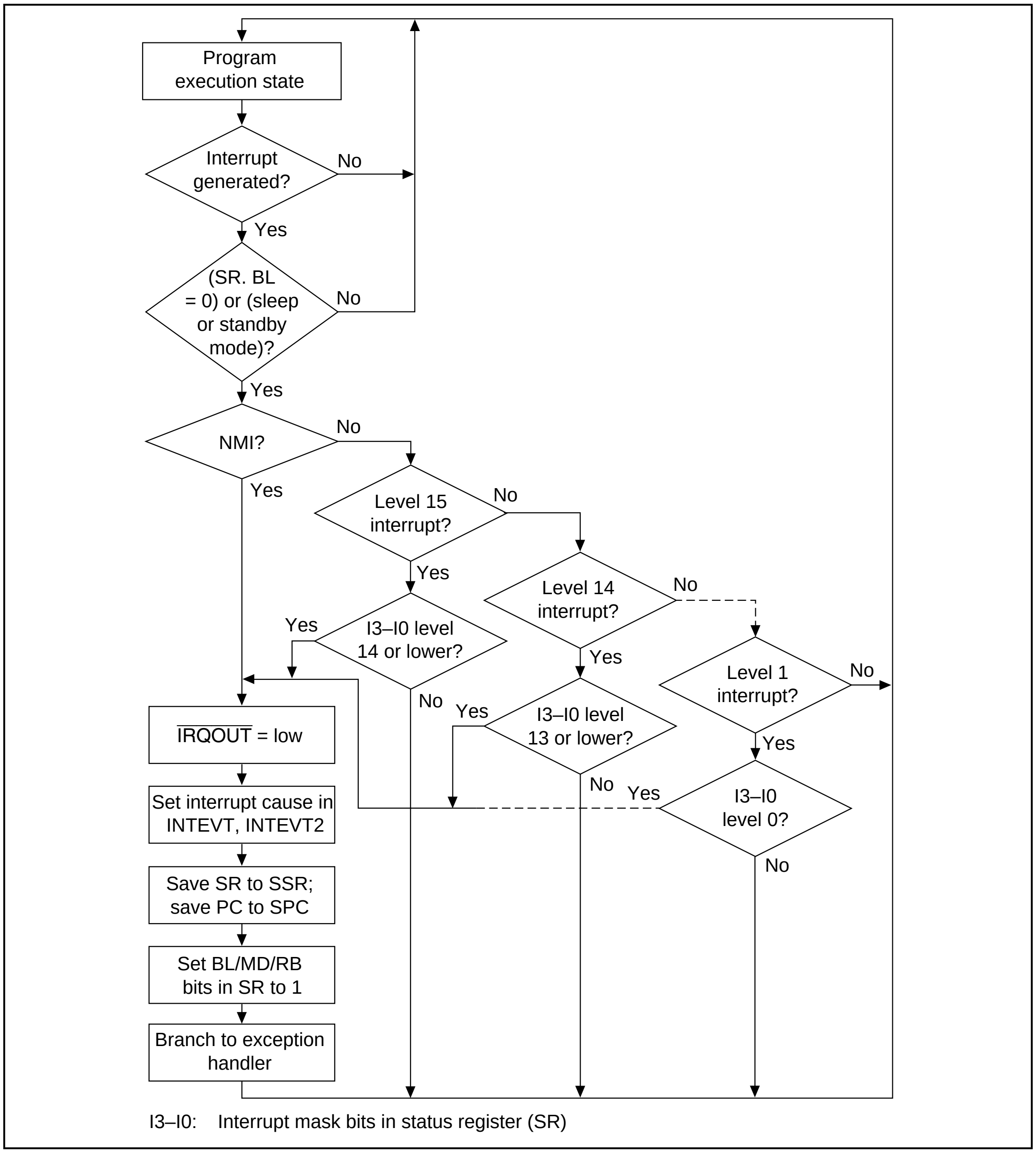


Figure 6.3 Interrupt Operation Flowchart

6.4.2 Multiple Interrupts

When handling multiple interrupts, an interrupt handler should include the following procedures:

1. Branch to a specific interrupt handler corresponding to a code set in INTEVT and INTEVT2. The code in INTEVT and INTEVT2 can be used as a branch-offset for branching to the specific handler.
2. Clear the cause of the interrupt in each specific handler.
3. Save SSR and SPC to memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bits in SR.
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4. Figure 6.3 shows a sample interrupt operation flowchart.

6.5 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception handling is performed and fetching of the first instruction of the exception handler is started (the interrupt response time) is shown in table 6.8. Figure 6.4 shows an example of pipeline operation when an IRL interrupt is accepted. When SR.BL is 1, interrupt exception handling is masked, and is kept waiting until completion of an instruction that clears BL to 0.

Table 6.8 Interrupt Response Time

	Number of States				
Item	NMI	IRQ	PINT	Supporting Modules	Notes
Time for priority decision and SR mask bit comparison	0.5 × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	0.5 × Icyc + 1 × Bcyc + 4.5 × Pcyc	0.5 × Icyc + 3.5 × Pcyc	0.5 × Icyc + 1.5 × Pcyc*[4]	
		0.5 × Icyc + 1 × Bcyc + 2.5 × Pcyc		0.5 × Icyc + 3 × Pcyc*[5]	
Wait time until end of sequence being executed by CPU	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	Interrupt exception handling is kept waiting until the executing instruction ends. If the number of instruction execution states is S*[1], the maximum wait time is: X = S – 1. However, if BL is set to 1 by instruction execution or by an exception, interrupt exception handling is deferred until completion of an instruction that clears BL to 0. If the following instruction masks interrupt exception handling, the handling may be further deferred.
Time from interrupt exception handling (save of SR and PC) until fetch of first instruction of exception handler is started	5 × Icyc	5 × Icyc	5 × Icyc	5 × Icyc	

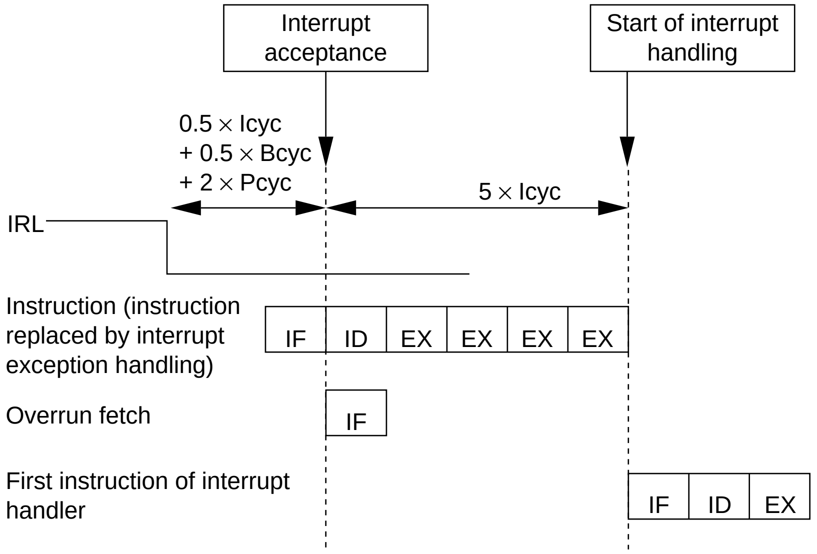
Table 6.8 Interrupt Response Time (cont)

		Number of States				
Item		**NMI**	**IRQ**	**PINT**	**Peripheral Modules**	**Notes**
Response time	Total	(5.5 + X) × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	(5.5 + X) × Icyc + 1 × Bcyc + 4.5 × Pcyc	(5.5 + X) × Icyc + 3.5 × Pcyc	(5.5 + X) × Icyc + 1.5 × Pcyc*4 (5.5 + X) × Icyc + 3 × Pcyc*5	
	Minimum case*2	7.5	16.5	12.5	8.5*4/11.5*5	At 60 (CKIO=30) MHz operation: 0.13–0.28 µs
	Maximum case*3	8.5 + S	26.5 + S	18.5 + S	10.5 + S*4 16.5 + S*5	At 60 (CKIO=15) MHz operation: 0.26–0.56 µs (in case of operand cache-hit) At 60 (CKIO=15) MHz operation: 0.29–0.59 µs (when external memory access is performed with wait = 0)

Icyc: Duration of one cycle of internal clock supplied to CPU, etc.
Bcyc: Duration of one CKIO cycle
Pcyc: Duration of one cycle of peripheral clock supplied to supporting modules

Notes:
1. S also includes the memory access wait time.

   The processing requiring the maximum execution time is LDC.L @Rm+, SR. When the memory access is a cache-hit, this requires 7 instruction execution cycles. When external access is performed, the corresponding number of cycles must be added. There are also instructions that perform two external memory accesses; if external memory access is slow, the number of instruction execution cycles will increase accordingly.
2. The internal clock : CKIO : peripheral clock ratio is 2 : 1 : 1.
3. The internal clock : CKIO : peripheral clock ratio is 4 : 1 : 1.
4. Modules: TMU, RTC, SCI, WDT, REFC
5. Modules: DMAC, ADC, IrDA, SCIF


Interrupt acceptance
Start of interrupt handling
0.5 × Icyc
+ 0.5 × Bcyc
+ 2 × Pcyc
5 × Icyc
IRL
Instruction (instruction replaced by interrupt exception handling)
IF
ID
EX
EX
EX
EX
Overrun fetch
IF
First instruction of interrupt handler
IF
ID
EX

IF: Instruction fetch ... Instruction is fetched from memory in which program is stored.
ID: Instruction decode ... Fetched instruction is decoded.
EX: Instruction execution ... Data operation and address calculation are performed in accordance with result of decoding.

Figure 6.4 Example of Pipeline Operations when IRL Interrupt is Accepted

Section 7 User Break Controller (UBC)

7.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling programs to be debugged in the chip alone, without using an in-circuit emulator. Break conditions that can be set in the UBC are instruction fetch or data read/write, data size, data content, address value, and stop timing during instruction fetches.

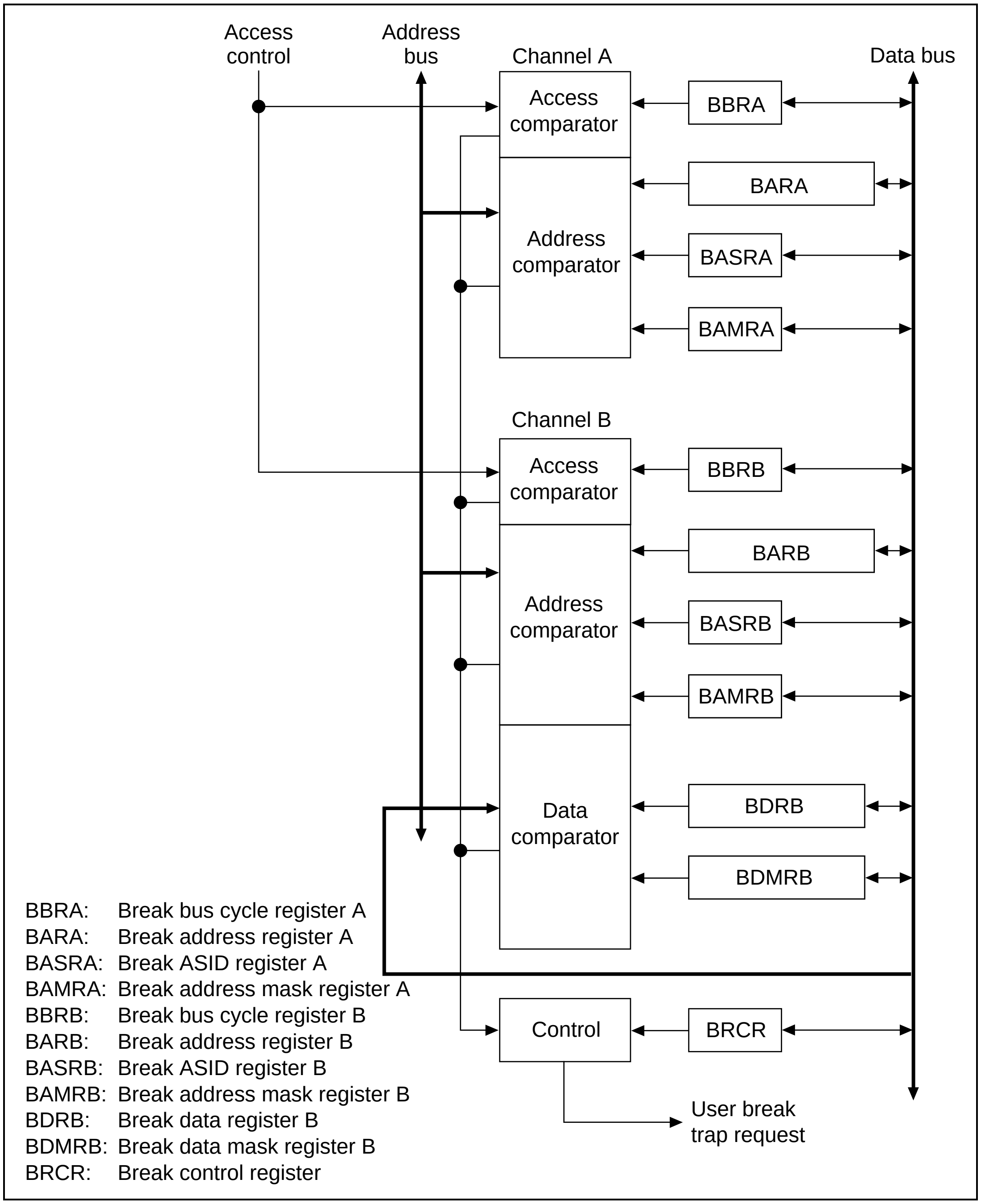
7.1.1 Features

The features of the user break controller are listed below.

- Two break channels (channel A and channel B). User break interrupts can be requested using either independent or sequential condition for the two channels (sequential breaks are channel A, then channel B).
- Selection and setting of the following as break compare conditions:
  - Address
    Selection of 32-bit logical address and ASID to be compared
    Address: Compare all bits, mask bottom 10 bits, mask bottom 12 bits. mask all bits
    ASID: Compare all bits/mask all bits
  - Data (channel B only, 32-bit maskable)
  - Bus cycle: Instruction fetch/data access
  - Read/write
  - Operand size: byte/word/longword
- The instruction fetch cycle break can be performed before or after the instruction is executed.
- User break trap generated when break conditions are satisfied. A user-designed user break trap routine can be run.

7.1.2 Block Diagram

Figure 7.1 shows the logical block diagram of the user break controller.



BBRA:	Break bus cycle register A
BARA:	Break address register A
BASRA:	Break ASID register A
BAMRA:	Break address mask register A
BBRB:	Break bus cycle register B
BARB:	Break address register B
BASRB:	Break ASID register B
BAMRB:	Break address mask register B
BDRB:	Break data register B
BDMRB:	Break data mask register B
BRCR:	Break control register

Figure 7.1 Logical Block Diagram of User Break Controller

7.1.3 Register Configuration

Table 7.1 shows the user break controller registers.

Table 7.1 UBC Registers

Channel	Register	Initial Value*	Access Size	Access Address	R/W
A	BARA	Undefined	Longword	H'FFFFFFB0	R/W
	BASRA	Undefined	Byte	H'FFFFFFE4	R/W
	BAMRA	Undefined	Byte	H'FFFFFFB4	R/W
	BBRA	H'0000	Word	H'FFFFFFB8	R/W
B	BARB	Undefined	Longword	H'FFFFFFA0	R/W
	BAMRB	Undefined	Byte	H'FFFFFFA4	R/W
	BASRB	Undefined	Byte	H'FFFFFFE8	R/W
	BBRB	H'0000*	Word	H'FFFFFFA8	R/W
	BDRB	Undefined	Longword	H'FFFFFF90	R/W
	BDMRB	Undefined	Longword	H'FFFFFF94	R/W
Common	BRCR	H'0000	Word	H'FFFFFF98	R/W

Note: Value is retained in standby mode.

7.1.4 Break Conditions and Register Settings

The relationship between break conditions and register settings is as follows:

1. Break conditions for channel A or B are set in the respective registers.
2. The address is set in the BARA or BARB register. ASID is set in the BASRA or BASRB register. Whether the address is included in the break conditions, or whether or not masking is to be performed, is set in the BAMA or BAMB bit of the BAMRA or BAMRB register. If ASID is included in the conditions, this is set in the BASMA or BASMB bit of the BAMRA or BAMRB register.
3. Bus cycle break conditions are set in the BBRA or BBRB register. Settings are instruction fetch or data access, read or write, and data access size. In the case of an instruction fetch, whether the break is to be made before or after instruction execution is set in the PCBA or PCBB bit of the BRCR register.
4. For channel B, data can be included in the break conditions. Data is set in the BDRB register. If data is to be masked, it is set in the BDMRB register. Data inclusion in or exclusion from break conditions is set in the DBEB bit of the BRCR register.
5. Sequential use of channels A and B is set in the SEQ bit of the BRCR. When sequential use is designated, a user break occurs when the channel B conditions are matched after matching of channel A conditions.

6. When a user break occurs, the CMFA and CMFB bits in the BRCR register are set to 1. If a break is to be generated again, the CMFA and CMFB bits should be cleared to 0.

7.2 UBC Register Functions

7.2.1 Break Address Register A (BARA)

Bit:	31	30	29	28	27	26	25	24
Bit name:	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17	16
Bit name:	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
Initial value:	—	—	—	—	—	—	—	—
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
Bit name:	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address register A (BARA) is a 32-bit read/write register that stores the virtual address of the channel A break condition. It is not initialized by a reset.

Bits 31 to 0—Break Address A31 to 0 (BAA31 to BAA0): These bits store the virtual address of the channel A break condition.

7.2.2 Break Address Register B (BARB)

BARB is the break address register for channel B. The bit configuration is the same as for BARA.

7.2.3 Break ASID Register A (BASRA)

Bit:	7	6	5	4	3	2	1	0
Bit name:	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break ASID register A (BASRA) specifies the ASID that serves as the break condition for channel A. It is compared to the ASID field of the MMU's PTEH register. BASRA is an 8-bit read/write register. It is not initialized by a reset.

Bits 7 to 0—Break ASID A7 to 0 (BASA7 to BASA0): These bits store the ASID (bits 7 to 0) that is the channel A break condition.

7.2.4 Break ASID Register B (BASRB)

BASRB is the break ASID register for channel B. The bit configuration is the same as for BASRA.

7.2.5 Break Address Mask Register A (BAMRA)

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	BASMA	BAMA1	BAMA0
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Break address mask register A (BAMRA) is an 8-bit read/write register that specifies which bits in the break ASID specified in BASRA and which bits in the break address specified in BARA are masked. It is not initialized by a reset.

Bits 7 to 3—Reserved: These bits always read 0. The write value should always be 0.

Bit 2—Break ASID Mask A (BASMA): Indicates whether the bits of the channel A breakpoint ASID7 to ASID0 (BASA7 to BASA0) set in BASRA are masked.

Bit 2: BASMA	Description
0	BASRA not masked; all bits included in break condition.
1	All BASRA bits masked; ASID not included in break condition.

Bits 1 and 0—Break Address Mask A1 and A0 (BAMA1 and BAMA0): These bits indicate which of the channel A break address bits 31–0 (BAA31–BAA0) set in BARA are masked.

Bit 1: BAMA1	Bit 0: BAMA0	Description
0	0	BARA not masked; all bits included in break condition.
0	1	Lowest 10 bits masked and excluded from break condition.
1	0	Lowest 12 bits masked and excluded from break condition.
1	1	All BARA bits masked; address not included in break condition.

7.2.6 Break Address Mask Register B (BAMRB)

BAMRB is the break address mask register for channel B. The bit configuration is the same as for BAMRA.

7.2.7 Break Bus Cycle Register A (BBRA)

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0
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

The break bus cycle register A (BBRA) is a 16-bit read/write register that sets the following three channel A break conditions for channel A: (1) instruction fetch/data access, (2) read/write, and (3) operand size. A reset initializes BBRA to H'0000.

Bits 15 to 6—Reserved: These bits always read 0. The write value should always be 0.

Bits 5 and 4—Instruction Fetch/Data Access Select A (IDA1 and IDA0): These bits select whether to break channel A on instruction fetch and/or data access cycles.

Bit 5: IDA1	Bit 4: IDA0	Description	
0	0	No conditions compared	(Initial value)
	1	Break on instruction fetch cycle	
1	0	Break on data access cycle	
	1	Break on either instruction fetch or data access cycle	

Bits 3 and 2—Read/Write Select A (RWA1 and RWA0): These bits select whether to break channel A on read and/or write cycles.

Bit 3: RWA1	Bit 2: RWA0	Description	
0	0	No conditions compared	(Initial value)
	1	Break on read cycles	
1	0	Break on write cycles	
	1	Break on both read and write cycles	

Bits 1 and 0—Operand Size Select A (SZA1 and SZA0): These bits select the bus cycle operand size as a channel A break condition.

Bit 1: SZA1	Bit 0: SZA0	Description	
0	0	Operand size is not a break condition	(Initial value)
	1	Break on byte access	
1	0	Break on word access	
	1	Break on longword access	

7.2.8 Break Bus Cycle Register B (BBRB)

BBRB is the break bus cycle register for channel B. The bit configuration is the same as for BBRA.

7.2.9 Break Data Register B (BDRB)

Bit:	31	30	29	28	27	26	25	24
Bit name:	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17	16
Bit name:	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
Initial value:	—	—	—	—	—	—	—	—
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
Bit name:	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data register B (BDRB) is a 32-bit read/write register that specifies the data that is the break condition for channel B data breaks. BDRB is not initialized by a reset.

BDRB Bits 31 to 0—Break Data B31 to B0 (BDB31 to BDB0): These bits store the data that is the break condition for break channel B.

When byte access has been specified by the SZB bit in the BBRB register, set the same byte data in bits BDB15–BDB8 as has been set in bits BDB7–BDB0. Bits BDB31–BDB16 are ignored when either byte or word access is specified. When the instruction fetch cycle is specified as a channel B break condition, or when the data bus is not included in the conditions according to the DBEB bit specification in BRCR (0), the BDRB value is ignored.

7.2.10 Break Data Mask Register B (BDMRB)

Bit:	31	30	29	28	27	26	25	24
Bit name:	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17	16
Bit name:	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
Initial value:	—	—	—	—	—	—	—	—
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
Bit name:	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break data mask register B (BDMRB) is a 32-bit read/write register that determines which of the bits in the break address set in BDRB are masked. BDMRB is not initialized by a reset.

BDMRB Bits 31 to 0—Break Data Mask B31 to B0 (BDMB31 to BDMB0): These bits specify whether bits B31–B0 (BDB31 to BDB0) of the channel B break data set in BDRB are masked. Set the same values in BDMB15–BDMB8 as are set in BDMB7–BDMB 0.

Bits 31–0: BDMBn	Description
0	Channel B break data bit BDBn is included in the break condition.
1	Channel B break data bit BDBn is masked and therefore not included in the break condition.

n = 31 to 0

Notes: 1. When the data bus value is contained in the break conditions, specify the operand size.

2. For byte data, set the same data in bits 0–7 and bits 8–15 of BDRB and BDMRB.

3. Bits 31–16 of BDRB and BDMRB are ignored for byte and word sizes.

7.2.11 Break Control Register (BRCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	CMFA	CMFB	—	—	—	PCBA	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	DBEB	PCBB	—	—	SEQ	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R	R	R

The break control register (BRCR) is a 16-bit read/write register that controls user breaks.

BRCR:

1. Determines whether to use channels A and B as two independent channels or as sequential conditions.
2. Selects whether to break before or after instruction execution during the instruction fetch cycle.
3. Determines whether to include the BDRB register in the channel B comparison conditions.

It also has a condition-match flag. A reset initializes BRCR to H'0000.

Bit 15—Condition Match Flag A (CMFA): Set to 1 when the break conditions set in channel A are met. Not cleared to 0. To check a flag setting after it has been set, clear it by writing 0.

Bit 15: CMFA	Description	
0	Channel A break conditions do not match.	(Initial value)
1	Channel A break conditions match.	

Bit 14—Condition Match Flag B (CMFB): Set to 1 when the break conditions set in channel B are met. Not cleared to 0. To check a flag setting after it has been set, clear it by writing 0.

Bit 14: CMFB	Description	
0	Channel B break conditions do not match.	(Initial value)
1	Channel B break conditions match.	

Bits 13 to 11—Reserved: These bits always read 0. The write value should always be 0.

Bit 10—PC Break Select A (PCBA): Selects whether to place the channel A instruction fetch cycle break before or after instruction execution.

Bit 10: PCBA	Description
0	Places the channel A PC break before instruction execution. (Initial value)
1	Places the channel A PC break after instruction execution.

Bits 9 and 8—Reserved: These bits always read 0. The write value should always be 0.

Bit 7—Data Break Enable B (DBEB): Selects whether to include data bus conditions in the channel B break conditions.

Bit 7: DBEB	Description
0	Do not include data bus conditions in the channel B conditions. (Initial value)
1	Include data bus conditions in the channel B conditions.

Note: When the data bus is not included in the break conditions, the IDB1 and IDB0 bits of break bus cycle register B (BBRB) should be 10 or 11.

Bit 6—PC Break Select B (PCBB): Selects whether to place the channel B instruction fetch cycle break before or after instruction execution

Bit 6: PCBB	Description
0	Places the channel B PC break before instruction execution. (Initial value)
1	Places the channel B PC break after instruction execution.

Bits 5 and 4—Reserved: These bits always read 0. The write value should always be 0.

Bit 3—Sequence Condition Select (SEQ): Selects whether to handle the channel A and B conditions independently or sequentially. When set for sequential, the CMFB flag is set when the channel B condition is matched after the channel A condition has already been matched.

Bit 3: SEQ	Description
0	Compare channel A and B conditions independently. (Initial value)
1	Compare channel A and B conditions sequentially (channel A, then channel B).

Bits 2 to 0—Reserved: These bits always read 0. The write value should always be 0.

7.3 UBC Operation

7.3.1 User Break Operation Flow

The flow from break condition setting to user break trap processing is as follows:

1. In the break conditions, set the break address in the break address register for the relevant channel (BARA or BARB), the ASID corresponding to the break space in the break ASID register (BASRA or BASRB), and the address and ASID masking method in the break address masking register (BAMRA or BAMRB). If the data bus value is included in the break conditions, set the break data in the break data register (BDRB) and the data mask in the break data mask register (BDMRB).
2. Set the break bus conditions in the break bus cycle register (BBRA or BBRB). If 00 is set for even one set out of BBRA/BBRB register instruction fetch/data access select and read/write select, a user trap break will not be generated in the corresponding channel.

   Set such specifications as pre- or post-execution in the case of instruction fetch, whether the data bus value is to be included in the conditions in the case of data access, and independent or sequential conditions for channels A and B, in the break control register (BRCR).

   Set the BBRA and BBRB registers only after all other break-related register settings have been completed. If break enabling is set with the BBRA and BBRB registers while the break address, data, mask, and other registers are still in their initial post-reset state, a break may occur inadvertently.
3. When a condition is matched, the condition match flag for the relevant channel (CMFA or CMFB) is set. This flag is set by a condition match but is not reset. To confirm setting of the same flag again, therefore, it should first be cleared to 0.
4. When sequential conditions are set, a break is made at the instruction matched by channel B when the channel B condition is matched after matching of the channel A condition. No break is made if the channel B set condition is matched before or at the same time as the channel A condition.

   With sequential conditions, the condition match flag is set only for channel B, and not for channel A.

7.3.2 Instruction Fetch Cycle Break

1. Making an instruction fetch/read/word setting is made in the break bus cycle register (BBRA/BBRB) enables an instruction fetch cycle to be set as a break condition. In this case, pre- or post-execution of the instruction can be selected by means of bit PCBA/PCBB in the break control register (BRCR).
2. When instructions are fetched consecutively, 32 bits (two instructions) are fetched in one bus cycle. In this case, although only one bus cycle is generated, breaks can be set for both instructions by setting the start addresses of the respective instructions in the break address registers (BARA and BARB).
3. With an instruction subject to a pre-execution break, the break is executed when it has been confirmed that the instruction has been fetched and is to be executed. Consequently, an overrun-fetched instruction (an instruction fetched but not executed in the event of a branch or exception) cannot be subject to a break. If an exception when an instruction subject to a break is fetched, exception processing is performed first, and the break is executed only when the instruction is re-executed.

   Since a delayed branch instruction and delay slot instruction are executed as a single instruction, if a pre-execution condition is specified for the delay slot instruction, a break is made before execution of the delayed branch instruction. However, a pre-execution break condition cannot be specified for an RTE instruction delay slot instruction.
4. With a post-execution condition, the instruction set as the break condition is executed and a break trap is generated before the next instruction is executed. In the same way, a break cannot be specified for an overrun-fetch instruction. When a post-execution condition is set for a delayed branch instruction, similarly, the break is made after executing the delay slot and before executing the instruction at the branch destination.
5. When an instruction fetch cycle is set for channel B, break data register B (BDRB) is ignored. Therefore, break data need not be set for an instruction fetch cycle break.
6. Instruction fetch cycle breaks cannot be specified consecutively for a delayed branch instruction and its delay slot.

7.3.3 Data Access Cycle Break

1. In the case of a data access cycle break, the bits used for comparison with the address bus depend on the break bus cycle register (BBRA/BBRB) operand size specification, as follows:

Operand size	Compared address
Not included in conditions (00):	For byte address, comparison with address bits A31–A0 For word address, comparison with address bits A31–A1 For longword address, comparison with address bits A31–A2
Byte (01):	Comparison with address bits A31–A0
Word (10):	Comparison with address bits A31–A1
Longword (11):	Comparison with address bits A31–A2

2. When data value is included in break condition in channel B

   When the data value is included in the break conditions, set the DBEB bit in the break control register (BRCR) to 1. In this case, break data register B (BDRB) and break data mask register B (BDMRB) settings are needed in addition to the address condition. A user break trap is generated on a match of the address condition and the data condition.

   Bits IDB1 and IDB0 of break bus cycle register B (BBRB) should be set to 00 or 01.

   When byte data is specified, set the same data in the two bytes comprising bits 15–8 and bits 7–0 in break data register B (BDRB) and break mask register B (BDMRB). If word or byte is designated, bits 31–16 of BDRB and BDMRB are ignored.

7.3.4 Saved Program Counter (PC) Value

1. When instruction fetch (pre-execution) is set as break condition

   The program counter (PC) value saved in the SPC in user break interrupt handling is the address of the instruction for which the break condition matched. In this case, the fetched instruction is not executed, due to the user break interrupt generated prior to its execution. In the fetch cycle of an instruction located in the delay slot of a delayed branch instruction, a break is generated before the branch, so that the SPC value indicates the delayed branch instruction.

2. When instruction fetch (post-execution) is set as break condition

   The program counter (PC) value saved in the SPC in user break interrupt processing is the address of the next instruction to be executed after the instruction for which the break condition matched. In this case, the fetched instruction is executed, and a user break trap occurs before execution of the next instruction. When a delayed branch instruction is designated, the delay slot instruction is executed and a user break occurs before execution of the instruction at the branch destination. In this case, the PC value saved in the SPC is the address of the branch destination instruction.

3. When data access (address only) is set as break condition

   The value saved is the address of the next instruction to be executed after the instruction for which the condition matched. The condition-matching instruction is executed, and a user break trap occurs before execution of the next instruction.

4. When data access (address + data) is set as break condition

   The value saved is the start address of the next instruction after the instruction for which execution has been completed when user break trap processing is initiated. When a data value is set as a break condition, the point at which the break is to be made cannot be specified. A break is executed before execution of the instruction fetched around the time of the break data access.

7.3.5 Examples of Use

Register settings, set conditions, and states in which the set conditions are matched, are as follows:

1. Instruction fetch cycle break condition setting (independent channel A and B conditions)

BRCR = H'0400: Independent channel A and B conditions, post-execution for channel A, pre-execution for channel B

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00000404:	Address H'00000404
	BAMRA = H'00:	Address mask H'00
	BBRA = H'0014:	Bus cycle, instruction fetch (post-execution), read (operand size not included in conditions)

Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'00008010:	Address H'00008010
	BAMRB = H'02:	Address mask H'02
	BBRB = H'0014:	Bus cycle, instruction fetch (pre-execution), read (operand size not included in conditions)
	BDRB = H'00000000:	Data H'00000000
	BDMRB = H'00000000:	Data mask H'00000000

A user break is generated after execution of the instruction at address H'00000404 with ASID = H'80, or before execution of instructions at addresses H'00008000 to H'000083FE with ASID = H'70.

2. Instruction fetch cycle break condition setting (independent channel A and B conditions)

BRCR = H'0080: Channel A → channel B sequential conditions, pre-execution for channel A, pre-execution for channel B

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00037226:	Address H'00037226
	BAMRA = H'00:	Address mask H'00
	BBRA = H'0016:	Bus cycle, instruction fetch (pre-execution), read, word

Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'0003722E:	Address H'0003722E
	BAMRB = H'00:	Address mask H'00
	BBRB = H'0016:	Bus cycle, instruction fetch (pre-execution), read, word
	BDRB = H'00000000:	Data H'00000000
	BDMRB = H'00000000:	Data mask H'00000000

A user break is generated before execution of the instruction at address H'0003722E with ASID = H'70, after execution of the instruction at address H'00037226 with ASID = H'80.

3. Data access cycle break condition setting

BRCR = H'0080: Independent channel A and B conditions, data break enable

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00123456:	Address H'00123456
	BAMRA = H'00:	Address mask H'00
	BBRA = H'0024:	Bus cycle, data access, read (operand size not included in conditions)

Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'000ABCDE:	Address H'000ABCDE
	BAMRB = H'02:	Address mask H'02
	BBRB = H'002A:	Bus cycle, data access, write, word
	BDRB = H'0000A512:	Data H'0000A512, (data break enable)
	BDMRB = H'00000000:	Data mask H'00000000

For channel A, a user break trap occurs when ASID = H'80 and a longword read is performed at address H'00123454, a word read is performed at address H'00123456, or a byte read is performed at address H'00123456.

For channel B, a user break trap occurs when ASID = H'70 and H'A512 is written anywhere in addresses H'000AB000 to H'000ABFFE.

4. Instruction fetch cycle break condition setting (example of setting error)

BRCR = H'0000: Independent channel A and B conditions, pre-execution for channel A, pre-execution for channel B

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00027128	Address H'00027128
	BAMRA = H'00:	Address mask H'00
	BBRA = H'001A:	Bus cycle, instruction fetch (pre-execution), write, word

Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'00031415	Address H'00031415
	BAMRB = H'00:	Address mask H'00
	BBRB = H'0014:	Bus cycle, instruction fetch (pre-execution), read, (operand size not included in conditions)
	BDRB = H'00000000:	Data H'00000000
	BDMRB = H'00000000:	Data mask H'00000000

For channel A, a user break trap does not occur since an instruction fetch is not a write cycle.

For channel B, a user break trap does not occur since an instruction fetch is performed on an even address.

7.3.6 Cautions

1. If pre-execution is specified for one channel and post-execution for the other for the same address, a pre-execution break will be generated but the condition match flag will be set for both channels.
2. Do not set consecutive PC breaks for a delayed branch instruction and a delay slot instruction.
3. If a PC break (post-execution condition) is set for the TRAPA instruction, the condition match flag will be set but a break will not be executed. The TRAP instruction will be processed correctly.
4. If data access (address + data) is set as a break condition, and an exception is generated by the instruction (including the delay slot for a delayed branch instruction) following that at which that break condition was matched, the condition match flag will be set but a break will not be executed. The exception generated after the break will be processed correctly.
5. If data access (address + data) is set as a break condition, and the instruction following that at which that break condition was matched is a SLEEP instruction, the condition match flag will be set but a break will not be executed. The SLEEP instruction will be processed correctly.
6. If an instruction fetch (halt after execution) is set as a break condition, and a nonmaskable interrupt is detected at the instruction following that at which that break condition was matched, the condition match flag will be set but a break will not be executed. The nonmaskable interrupt will be processed correctly.
7. When a sequential break setting is made, a condition match occurs on a channel B match in a bus cycle after that in which a channel A match occurred. Therefore, a condition match will not be recognized if bus cycles occurring simultaneously in channel A and B are designated. Also, since the CPU has a pipeline structure, the order of instruction fetch and data access cycles is determined by the pipeline. With sequential conditions, therefore, the sequential conditions will be taken as being matched as long as the respective channel conditions match in the order in which the bus cycles occur.
8. With an emulator, the UBC is used on the emulator system side in order to implement the emulator’s break functions. Consequently, no UBC functions can be used when an emulator is used.

Section 8 Power-Down Modes

8.1 Overview

In the power-down modes, all CPU and some on-chip supporting module functions are halted. This lowers power consumption.

8.1.1 Power-Down Modes

The SH7709 has three power-down modes:

1. Sleep mode
2. Standby mode
3. Hardware standby mode
4. Module standby function (TMU, RTC, and SCI on-chip supporting modules)

Table 8.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and supporting module states in each mode and the procedures for canceling each mode.

Table 8.1 Power-Down Modes

		State							
Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts	Held	Held	Run	Held	Refresh	1. Interrupt 2. Reset
Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Held	Halt[*1]	Held	Self-refresh	1. Interrupt 2. Reset
Hardware standby mode	Drive CA pin low	Halts	Halts	Held	Held	Halts	Held	Self-refresh	Power-on reset
Module standby function	Set MSTP bit of STBCR to 1	Runs	Runs	Held	Held	Specified module halts	[*2]	Refresh	1. Clear MSTP bit to 0 2. Reset

Notes: 1. The RTC still runs if the START bit in RCR2 is set to 1 (see section 12, Realtime Clock (RTC)). TMU still runs when output of the RTC is used as input to its counter (see section 11, Timer (TMU)).

2. Depends on the on-chip supporting module.
TMU external pin: Held
SCI external pin: Reset

8.1.2 Pin Configuration

Table 8.2 lists the pins used for the power-down modes.

Table 8.2 Pin Configuration

Processing Status 1 Pin (STATUS1)	Processing Status 0 Pin (STATUS0)	I/O	Processor Operating Status
High	High	O	Reset
	Low		Sleep mode
Low	High		Standby mode
	Low		Normal operation

Pin Name	Symbol	I/O	Description
Wakeup from standby mode	WAKEUP	O	Low active assert after accepting wakeup interrupt in standby mode until returning to normal operation with WDT overflow

8.1.3 Register Configuration

Table 8.3 shows the configuration of the control register for the power-down modes.

Table 8.3 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'00	H'FFFFFF82	Byte
Standby control register 2	STBCR2	R/W	H'00	H'FFFFFF88	Byte

8.2 Register Description

8.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit read/write register that sets the power-down mode. STBCR is initialized to H'00 by a power-on reset. Always set bits 6–3 to 0 when writing to the STBCR register.

Bit:	7	6	5	4	3	2	1	0
Bit name:	STBY	—	—	—	—	MSTP2	MSTP1	MSTP0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit 7—Standby (STBY): Specifies transition to standby mode.

Bit 7: STBY	Description
0	Executing SLEEP instruction puts the chip into sleep mode. (Initial value)
1	Executing SLEEP instruction puts the chip into standby mode.

Bits 6 to 3—Reserved: These bits always read 0. The write value should always as 0.

Bit 2—Module Standby 2 (MSTP2): Specifies halting the clock supply to the timer unit TMU (an on-chip supporting module). When the MSTP2 bit is set to 1, the supply of the clock to the TMU is halted.

Bit 2: MSTP2	Description
s0	TMU runs. (Initial value)
1	Clock supply to TMU is halted.

Bit 1—Module Standby 1 (MSTP1): Specifies halting the clock supply to the realtime clock RTC (an on-chip supporting module). When the MSTP1 bit is set to 1, the supply of the clock to RTC is halted. When the clock halts, all RTC registers become inaccessible, but the counter keeps running.

Bit 1: MSTP1	Description
0	RTC runs. (Initial value)
1	Clock supply to RTC is halted.

Bit 0—Module Standby 0 (MSTP0): Specifies halting the clock supply to the serial communication interface SCI (an on-chip supporting module). When the MSTP0 bit is set to 1, the supply of the clock to the SCI is halted.

Bit 0: MSTP0	Description	
0	SCI operates.	(Initial value)
1	Clock supply to SCI is halted.	

8.2.2 Standby Control Register 2 (STBCR2)

The standby control register 2 (STBCR2) is a read/write 8-bit register that sets the power-down mode. The STBCR2 is initialized to H'00 during a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	MSTP SLP0	—	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3
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 7—Sleep Mode Module Stop (MSTPSLP0): Specifies halting of clock supply to the bus controller BSC (an on-chip peripheral module) in the sleep mode, when power consumption can be reduced by operating only the bus controller with lower power (BSCP) since the BSC only performs memory refresh and data transfer by the DMAC. The MSTPSLP0 bit does not affect the BSC operation in the normal mode. When the MSTPSLP0 bit is set to 1, the supply of the clock to the BSC is halted.

Bit 7: MSTPSLP0	Description	
0	BSC runs	(Initial value)
1	Clock supply to BSC is halted	

Bit 5— Module Standby 8 (MSTP8): Specifies halting the clock supply to the user break controler UBC (an on-chip supporting module). When the MSTP8 bit is set to 1, the supply of the clock to the UBC is halted.

Bit 5: MSTP8	Description	
0	UBC runs	(Initial value)
1	Clock supply to UBC is halted	

Bit 4—Module Stop 7 (MSTP7): Specifies halting of clock supply to the DMAC (an on-chip peripheral module). When the MSTP7 bit is set to 1, the supply of the clock to the DMAC is halted.

Bit 4: MSTP7	Description	
0	DMAC runs	(Initial value)
1	Clock supply to DMAC halted	

Bit 3—Module Stop 6 (MSTP6): Specifies halting of clock supply to the DAC (an on-chip peripheral module). When the MSTP6 bit is set to 1, the supply of the clock to the DAC is halted.

Bit 3: MSTP6	Description	
0	DAC runs	(Initial value)
1	Clock supply to DAC halted	

Bit.2—Module Stop 5 (MSTP5): Specifies halting of clock supply to the ADC (an on-chip peripheral module). When the MSTP5 bit is set to 1, the supply of the clock to the ADC is halted.

Bit 2: MSTP5	Description	
0	ADC runs	(Initial value)
1	Clock supply to ADC halted	

Bit.1—Module Stop 4 (MSTP4): Specifies halting the clock supply to the serial communication interface with FIFO (an on-chip peripheral module). When the MSTP1 bit is set to 1, the supply of the clock to the SCIF is halted.

Bit 1: MSTP4	Description	
0	SCIF runs	(Initial value)
1	Clock supply to SCIF halted	

Bit.0—Module Stop 3 (MSTP3): Specifies halting the clock supply to the infrared data association interface with FIFO (an on-chip peripheral module). When the MSTP1 bit is set to 1, the supply of the clock to the IrDA is halted.

Bit 0: MSTP3	Description	
0	IrDA runs	(Initial value)
1	Clock supply to IrDA halted	

8.3 Sleep Mode

8.3.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip supporting modules continue to run during sleep mode and the clock continues to be output to the CKIO pin. In sleep mode, the STATUS1 pin is set high and the STATUS0 pin low. However, during a refresh cycle, the STATUS1 pin and STATUS0 pin are both set low.

8.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRL, on-chip supporting module) or reset. Interrupts are accepted during sleep mode even when the BL bit in the SR register is 1.

Canceling with an Interrupt: When an NMI, IRL or on-chip supporting module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in the INTEVT register.

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

8.4 Standby Mode

8.4.1 Transition to Standby Mode

To enter standby mode, set the STBY bit to 1 in STBCR, then execute the SLEEP instruction. The chip moves from the program execution state to standby mode. In standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip supporting modules as well. The clock output from the CKIO pin also halts. CPU and cache register contents are held, but some on-chip supporting modules are initialized. Table 8.4 lists the states of registers in standby mode.

Table 8.4 Register States in Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller	—	All registers
Break controller	—	All registers
Bus state controller	—	All registers
On-chip clock pulse generator	—	All registers
Timer unit	TSTR register	Registers other than TSTR
Realtime clock	—	All registers

The procedure for moving to standby mode is as follows:

1. Clear the TME bit in the WDT’s timer control register (WTCSR) to 0 to stop the WDT. Set the WDT’s timer counter (WTCNT) and the CKS2–CKS0 bits of the WTCSR register to appropriate values to secure the specified oscillation settling time.
2. When PLL circuit 1 is running in clock modes 3–6, clear the PSTBY and PLLEN bits in the frequency control register (FRQCR) to 0 to stop PLL circuit 1.
3. After the STBY bit in the STBCR register is set to 1, a SLEEP instruction is executed.
4. Standby mode is entered and the clocks within the chip are halted. The STATUS1 pin output goes low and the STATUS0 pin output goes high.

8.4.2 Canceling Standby Mode

Standby mode is canceled by an interrupt (NMI, IRL, or on-chip supporting module) or a reset.

Exit by Interrupt: A hot start can be performed by means of the on-chip WDT. When an NMI, IRQ, IRL, PINT*[1], or on-chip peripheral module (except interval timer)*[2] interrupt is detected, after the elapse of the time set in the WDT's timer control/status register, clocks are supplied to the entire LSI, standby mode is exited, and the STATUS1 and STATUS0 pins both go low. Interrupt exception processing is then executed, and the code corresponding to the interrupt source is set in INTEVT and INTEVT2. After the branch to the interrupt processing routine, clear the STBY bit in the STBCR register. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT will continue operating and standby mode*[3] will be entered again when the count reaches H'00. This function prevents data corruption due to a rise in voltage when the power supply is unstable, for instance. In standby mode, interrupts are accepted even if the BL bit in the SR register is 1, and so, if necessary, SPC and SSR should be saved to the stack before executing the SLEEP instruction.

The phase of the CKIO pin clock output may be unstable immediately after an interrupt is detected, until standby mode is exited. A condition for exiting standby mode is that the interrupt request level (IRQ, IRL, on-chip supporting module) must be higher than the interrupt mask level set in bits I3—I0 in the SR register.

Notes: 1. Only when the RTC is used, standby mode can be exited by means of IRL3—IRL0, IRQ4—IRQ0, and PINT0/1.

2. Standby mode can be exited by means of an RTC interrupt or a TMU interrupt (when operating on the RTC clock).

3. This standby mode can only be exited by means of a power-on reset.

Canceling with a Reset: Standby mode can be canceled with a reset (power-on or manual). Keep the RESET pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

8.4.3 Clock Pause Function

In standby mode, the clock input from the EXTAL pin or CKIO pin can be halted and the frequency can be changed. This function is used as follows:

1. Enter standby mode using the appropriate procedures.
2. Once standby mode is entered and the clock stopped within the chip, the STATUS1 pin output is low and the STATUS0 pin output is high.
3. Once the STATUS1 pin goes low and the STATUS0 pin goes high, the input clock is stopped or the frequency is changed.
4. When the frequency is changed, an NMI, IRL, IRQ, PINT or on chip supporthing mudule (except the internal timer) interrupt is input after the change. When the clock is stopped, the same interrupts are input after the clock is applied.
5. After the time set in the WDT has elapsed, the clock starts being applied internally within the chip, the STATUS1–STATUS0 pins both go low, interrupts are handled, and operation resumes.

8.5 Module Standby Function

8.5.1 Transition to Module Standby Function

Setting the standby control register MSTP2–MSTP0 bits to 1 halts the supply of clocks to the corresponding on-chip supporting modules. This function can be used to reduce the power consumption in sleep mode. The module standby function holds the status prior to halt of the external pins of the on-chip supporting modules. TMU external pins hold their status prior to the halt. SCI external pins go to the reset state. With a few exceptions, all registers hold their values.

Bit	Value	Description
MSTPSLO	0	BSC runs
	1	Supply of clock to BSC halted
MSTP8	0	UBC runs.
	1	Supply of clock to UBC halted.
MSTP7	0	DMAC runs.
	1	Supply of clock to DMAC halted.
MSTP6	0	DAC runs.
	1	Supply of clock to DAC halted.
MSTP5	0	ADC runs.
	1	Supply of clock to ADC halted.
MSTP4	0	SCIF2 runs.
	1	Supply of clock to SCIF2 halted.
MSTP3	0	SCIF1 runs.
	1	Supply of clock to SCIF1 halted.
MSTP2	0	TMU runs.
	1	Supply of clock to TMU halted. Registers initialized.*1
MSTP1	0	RTC runs.
	1	Supply of clock to RTC halted. Register access prohibited.*2
MSTP0	0	SCI operates.
	1	Supply of clock to SCI halted.

Notes: 1. The registers initialized are the same as in standby mode (table 8.4).
2. The counter runs.

8.5.2 Clearing the Module Standby Function

The module standby function can be cleared by clearing the MSTPSLP0 MSTP8–MSTP0 bits to 0, or by a power-on reset or manual reset.

8.6 Timing of STATUS Pin Changes

The timing of STATUS1 and STATUS0 pin changes is shown in figures 8.1 through 8.9.

The meaning of the STATUS descriptions is as follows:

Reset:	HH (STATUS1 high, STATUS0 high)
Sleep:	HL (STATUS1 high, STATUS0 low)
Standby:	LH (STATUS1 low, STATUS0 high)
Normal:	LL (STATUS1 low, STATUS0 low)

The meaning of the clock units is as follows:

Bcyc:	Bus clock cycle
Pcyc:	Peripheral clock cycle

8.6.1 Timing for Resets

Power-On Reset (Clock Modes 0, 1, 2, and 7):

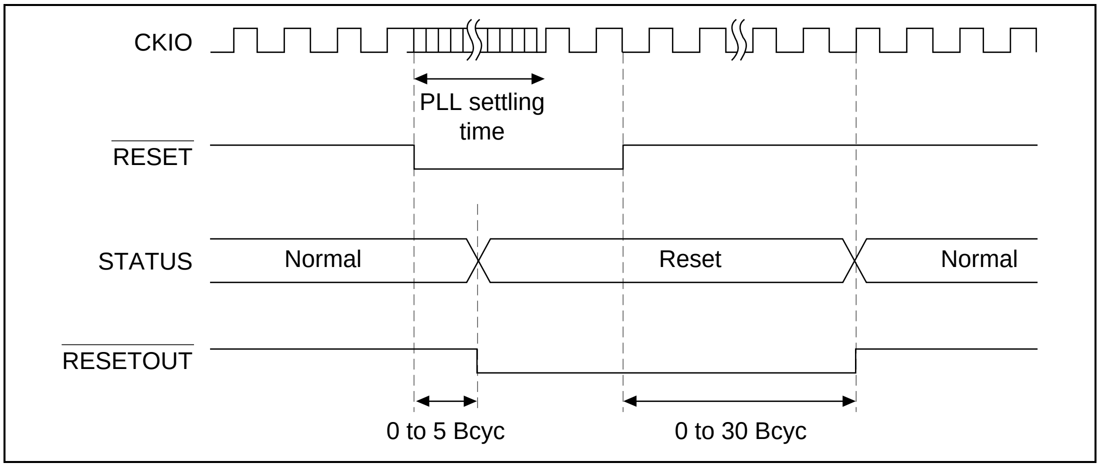


Figure 8.1 Power-On Reset (Clock Mode 0, 1, 2, and 7) STATUS Output

Power-On Reset (Clock Modes 3 and 4):

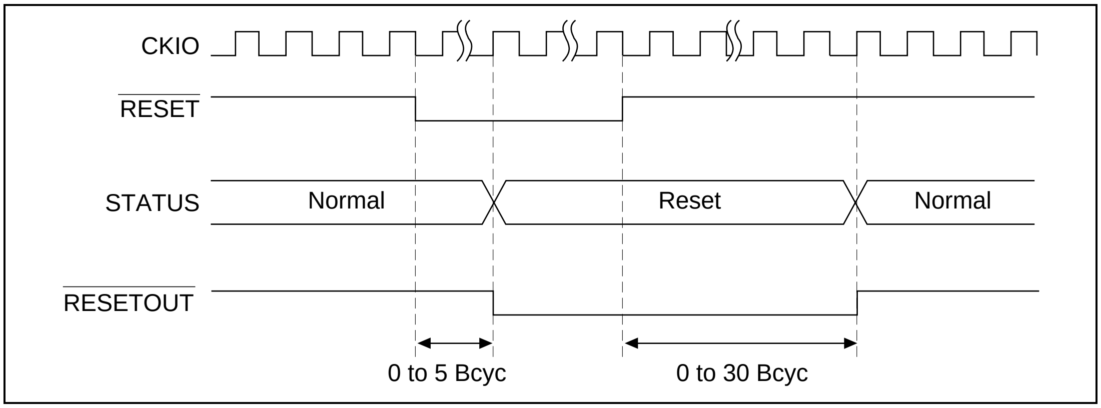
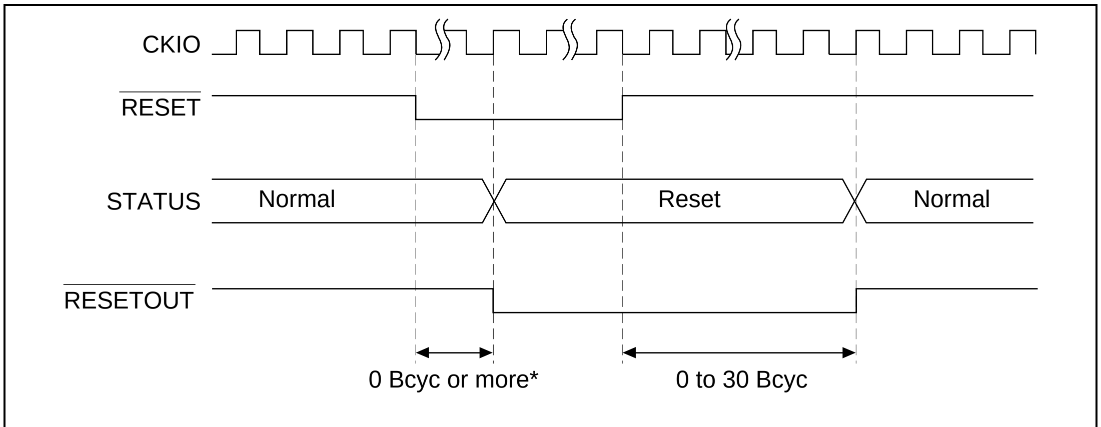


Figure 8.2 Power-On Reset (Clock Mode 3 and 4) STATUS Output

Manual Reset:



Note: During manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting for the executing bus cycle to end.

Figure 8.3 Manual Reset STATUS Output

8.6.2 Timing for Canceling Standbys

Standby to Interrupt:

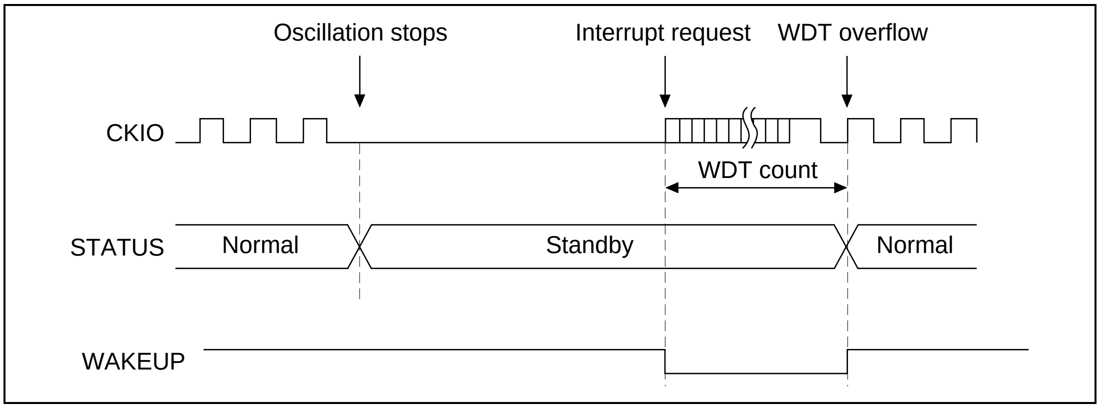
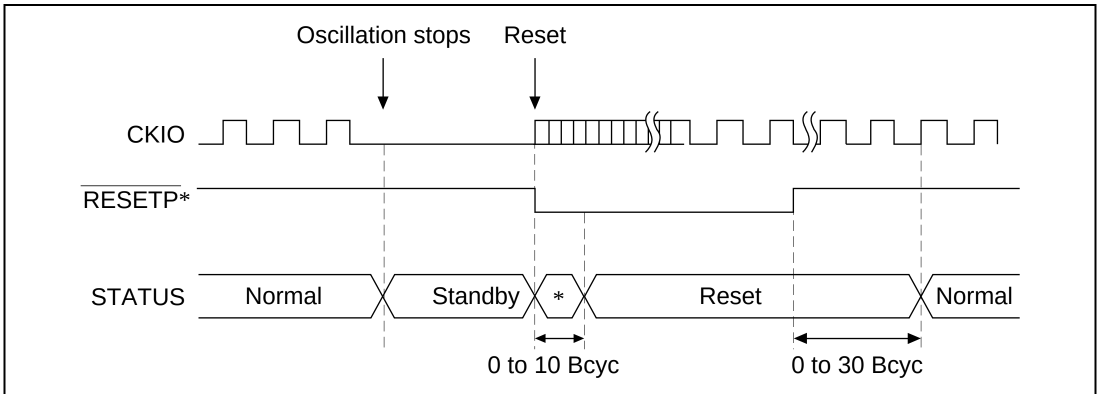


Figure 8.4 Standby to Interrupt STATUS Output

Standby to Power-On Reset:

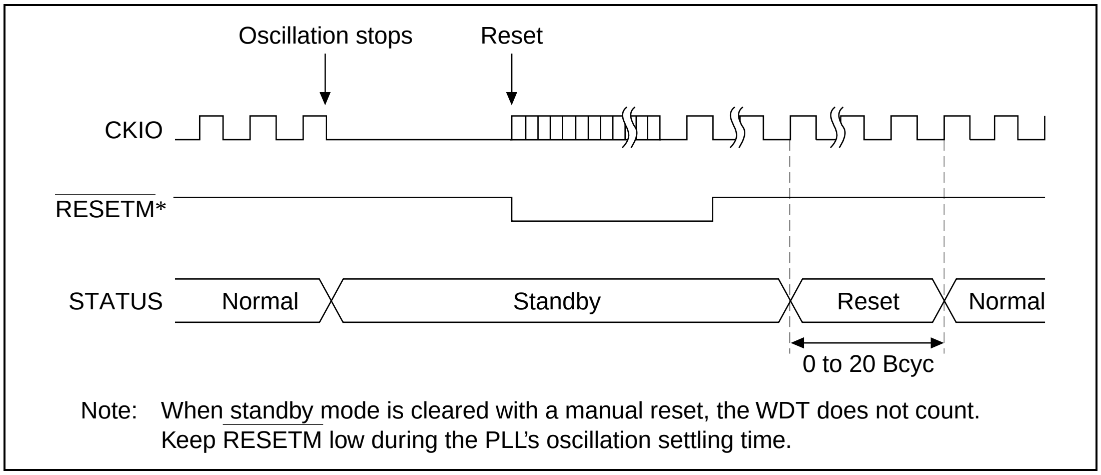


Note: When standby mode is cleared with a power-on reset, the WDT does not count. Keep $\overline{\text{RESETP}}$ low during the PLL's oscillation settling time.

*: Undefined

Figure 8.5 Standby to Power-On Reset STATUS Output

Standby to Manual Reset:



Note: When standby mode is cleared with a manual reset, the WDT does not count. Keep RESETM low during the PLL's oscillation settling time.

Figure 8.6 Standby to Manual Reset STATUS Output

8.6.3 Timing for Canceling Sleep Mode

Sleep to Interrupt:

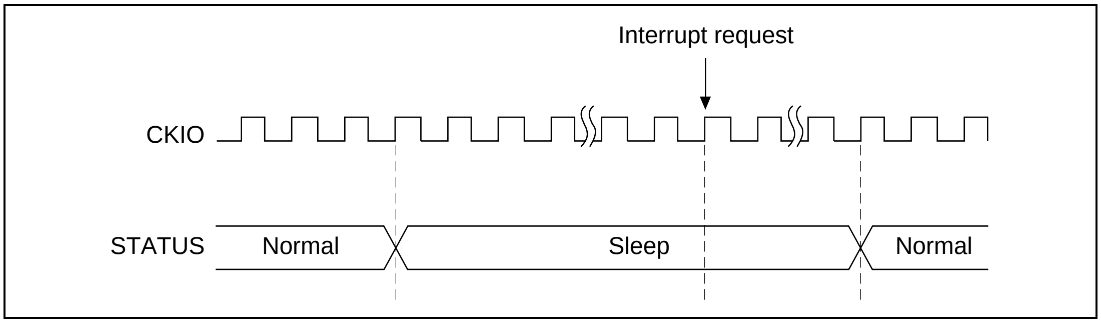
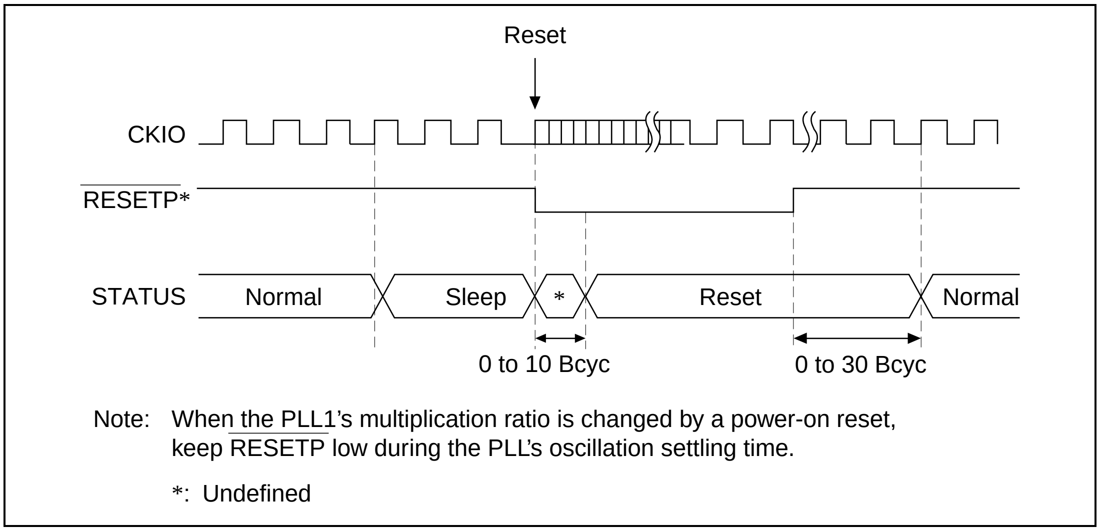


Figure 8.7 Sleep to Interrupt STATUS Output

Sleep to Power-On Reset:

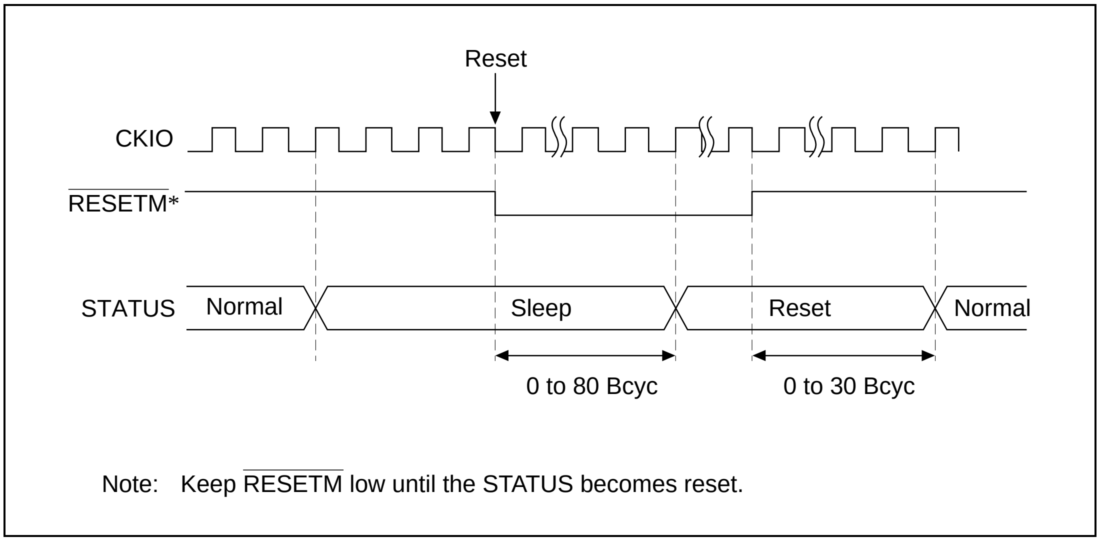


Note: When the PLL1's multiplication ratio is changed by a power-on reset, keep RESETP low during the PLL's oscillation settling time.

*: Undefined

Figure 8.8 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset:



Note: Keep RESETM low until the STATUS becomes reset.

Figure 8.9 Sleep to Manual Reset STATUS Output

8.7 Hardware Standby Function

8.7.1 Transition to Hardware Standby Mode

Driving the CA pin low causes a transition to hardware standby mode. In hardware standby mode, all modules except those operating on an RTC clock are halted, as in the standby mode entered on execution of a SLEEP instruction ((software) standby mode).

Hardware standby mode differs from (software) standby mode as follows.

1. Interrupts and manual resets are not accepted.
2. The TMU does not operate.

Operation when a low-level signal is input at the CA pin depends on the CPG state, as follows.

1. In standby mode
   The clock remains stopped and the chip enters the hardware standby state. Acceptance of interrupts and manual resets is disabled, TCLK output is fixed low, and the TMU halts.
2. During WDT operation when standby mode is canceled by an interrupt
   The chip enters hardware standby mode after standby mode is canceled and the CPU resumes operation.
3. In sleep mode
   The chip enters hardware standby mode after sleep mode is canceled and the CPU resumes operation.
4. During PLL standby (see section 9.6 for the PLL standby function)
   The chip enters hardware standby mode after the PLL turned off.

Hold the CA pin low in hardware standby mode.

8.7.2 Canceling Hardware Standby Mode

Hardware standby mode can only be canceled by a power-on reset.

When the CA pin is driven high while the $\overline{\text{RESETP}}$ pin is low, clock oscillation is started. Hold the $\overline{\text{RESETP}}$ pin low until clock oscillation stabilizes. When the $\overline{\text{RESETP}}$ pin is driven high, the CPU begins power-on reset processing.

Hardware standby mode cannot be canceled by an interrupt or manual reset.

8.7.3 Hardware Standby Mode Timing

Figures 8.10 and 8.11 show examples of pin timing in hardware standby mode.

The CA pin is sampled using EXTAL2 (32.768 kHz), and a hardware standby request is only recognized when the pin is low for two consecutive clock cycles.

The CA pin must be held low while the chip is in hardware standby mode.

Clock oscillation starts when the CA pin is driven high after the $\overline{\text{RESETP}}$ pin is driven low.

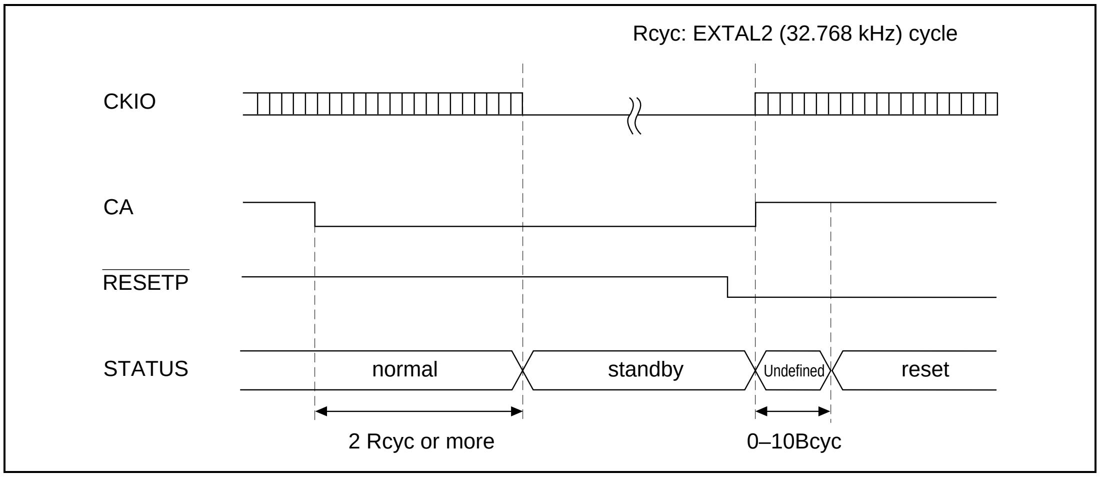


Figure 8.10 Hardware Standby Mode (When CA Goes Low in Normal Operation)

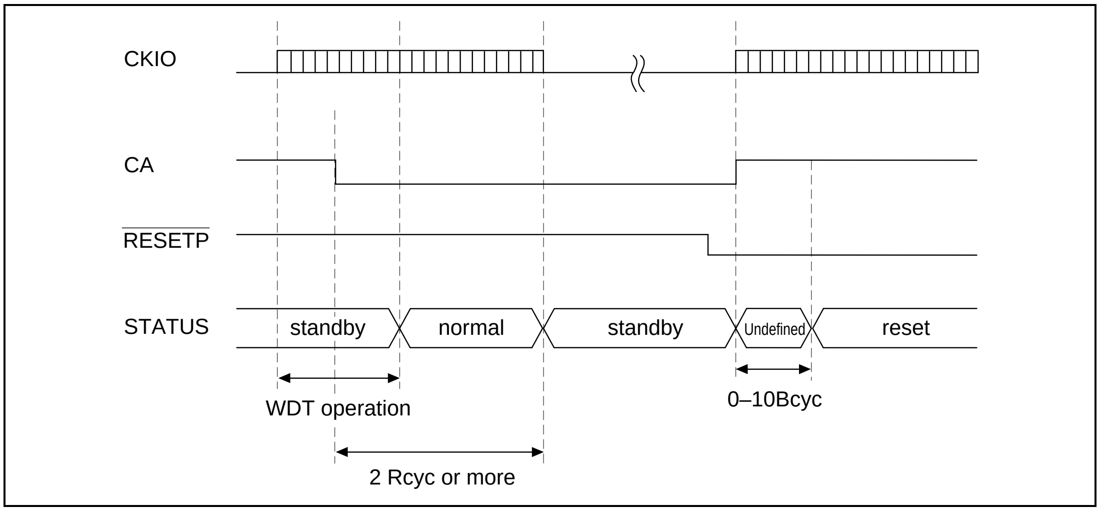


Figure 8.11 Hardware Standby Mode Timing
(When CA Goes Low during WDT Operation on Standby Mode Cancellation)

Section 9 On-Chip Oscillation Circuits

9.1 Overview

The clock pulse generator (CPG) supplies all clocks to the processor and controls the power-down modes. The watchdog timer (WDT) is a single-channel timer that counts the clock settling time and is used when clearing standby mode and temporary standbys, such as frequency changes. It can also be used as an ordinary watchdog timer or interval timer.

9.1.1 Features

The CPG has the following features:

- Six clock modes: Selection of eight clock modes for different frequency ranges, power consumption, direct crystal input, and external clock input.
- Three clocks generated independently: An internal clock for the CPU, cache, and TLB (Iϕ); a peripheral clock (Pϕ) for the on-chip supporting modules; and a bus clock (CKI0) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- PLL on/off function: Power consumption can be decreased by stopping the PLL circuit when operating at low frequencies.
- Power-down mode control: The clock can be stopped for sleep mode and standby mode and specific modules can be stopped using the module standby function.

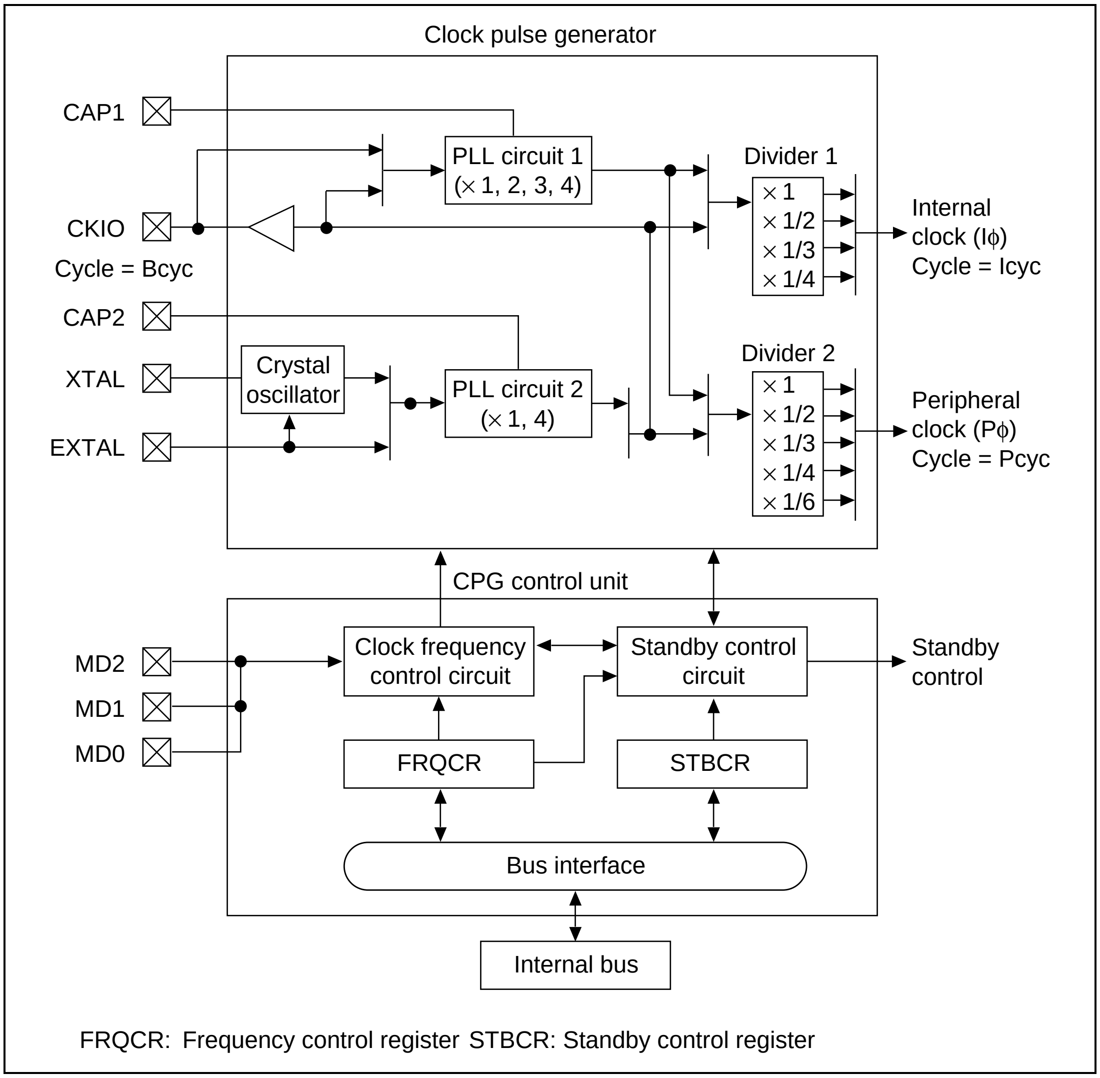
The WDT has the following features:

- Can be used to ensure the clock settling time: Use the WDT to cancel standby mode and the temporary standbys which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter overflow. Selection of power-on reset or manual reset.
- Generates interrupts in interval timer mode: Internal timer interrupts occur after counter overflow.
- Selection of eight counter input clocks. Eight clocks (×1 to ×1/4096) can be obtained by dividing the peripheral clock.

9.2 Overview of the CPG

9.2.1 CPG Block Diagram

A block diagram of the on-chip clock pulse generator is shown in figure 9.1.



FRQCR: Frequency control register STBCR: Standby control register

Figure 9.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

1. PLL Circuit 1: PLL circuit 1 doubles, triples, quadruples, or leaves unchanged the input clock frequency from the CKIO terminal. The multiplication rate is set by the frequency control register. When this is done, the phase of the leading edge of the internal clock is controlled so that it will agree with the phase of the leading edge of the CKIO pin.
2. PLL Circuit 2: PLL circuit 2 leaves unchanged or quadruples the frequency of the crystal oscillator or the input clock frequency coming from the EXTAL pin. The multiplication ratio is fixed by the clock operation mode. The clock operation mode is set by pins MD0, MD1, and MD2. See table 9.3 for more information on clock operation modes.
3. Crystal Oscillator: This oscillator is used when a crystal oscillator element is connected to the XTAL and EXTAL pins. It operates according to the clock operating mode setting.
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal clock. The operating frequency can be 1, 1/2, 1/3, 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
5. Divider 2: Divider 2 generates a clock at the operating frequency used by the peripheral clock. The operating frequencies can be 1, 1/2, 1/3, 1/4, or 1/6 times the output frequency of PLL Circuit 1 or the clock frequency of the CKIO pin, as long as it stays at or below the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
6. Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD pin and the frequency control register.
7. Standby Control Circuit: The standby control circuit controls the status of the clock pulse generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, on/off control of PLL circuit 1, PLL standby, the frequency multiplication ratio of PLL 1, and the frequency division ratio of the internal clock and the peripheral clock.
9. Standby Control Register: The standby control register has bits for controlling the power-down modes. See section 8, Power-Down Modes, for more information.

9.2.2 CPG Pin Configuration

Table 9.1 lists the CPG pins and their functions.

Table 9.1 Clock Pulse Generator Pins and Functions

Pin Name	Symbol	I/O	Description
Mode control pins	MD0	I	Set the clock operating mode.
	MD1	I	
	MD2	I	
Crystal I/O pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used to input an external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock. Level can be fixed during output.
Capacitor connection pins for PLL	CAP1	I	Connects capacitor for PLL circuit 1 operation (recommended value 470 pF).
	CAP2	I	Connects capacitor for PLL circuit 2 operation (recommended value 470 pF).

9.2.3 CPG Register Configuration

Table 9.2 shows the CPG register configuration.

Table 9.2 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0102	H'FFFFFF80	16

9.3 Clock Operating Modes

Table 9.3 shows the relationship between the mode control pin (MD2–MD0) combinations and the clock operating modes. Table 9.4 shows the usable frequency ranges in the clock operating modes.

Table 9.3 Clock Operating Modes

	Pin Values			Clock I/O		PLL2	PLL1	Divider 1	Divider 2	CKIO
Mode	**MD2**	**MD1**	**MD0**	**Source**	**Output**	**On/Off**	**On/Off**	**Input**	**Input**	**Frequency**
0	0	0	0	EXTAL	CKIO	On multi-plication ratio: 1	ON	PLL1 output	PLL1	(EXTAL)
1	0	0	1	EXTAL	CKIO	On multi-plication ratio: 4	ON	PLL1 output	PLL1	(EXTAL) × 4
2	0	1	0	Crystal oscillator	CKIO	On multi-plication ratio: 4	On	PLL1 output	PLL1	(Crystal) × 4
3	0	1	1	EXTAL	CKIO	On multi-plication ratio: 1	Off (initial value)	PLL2 output	PLL2	(EXTAL) × 1
							On	PLL1 output		
4	1	0	0	Crystal oscillator	CKIO	On multi-plication ratio: 1	Off (initial value)	PLL2 output	PLL2	(Crystal) × 1
							On	PLL1 output		
7	1	1	1	CKIO	–	Off	On	PLL1 output	PLL1	(CKIO)

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7709. PLL circuit 1 is constantly on, and there are no frequency range restrictions compared to mode 3. An input clock frequency of 16 MHz to 40 MHz can be used, and the CKIO frequency range is 10 MHz to 40 MHz.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4 by PLL circuit 2 before being supplied inside the SH7709, allowing a low-frequency external clock to be used. An input clock frequency of 5 MHz to 10 MHz can be used, and the CKIO frequency range is 20 MHz to 40 MHz.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside the SH7709, allowing a low crystal frequency to be used. A crystal oscillation frequency of 5 MHz to 10 MHz can be used, and the CKIO frequency range is 20 MHz to 40 MHz.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Mode 3: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7709. PLL circuit 1 is off in the default state at power-on reset, and PLL circuit 1 can be selected as on or off, enabling power consumption to be kept lower than in mode 0. An input clock frequency of 16 MHz to 25 MHz can be used, and the CKIO frequency range is 16 MHz to 25 MHz.

Mode 4: The on-chip crystal oscillator operates, with its output supplied inside the SH7709 as a square waveform by PLL circuit 2. PLL circuit 1 is off in the default state at power-on reset, and PLL circuit 1 can be selected as on or off, enabling power consumption to be reduced accordingly. A crystal oscillation frequency of 16 MHz to 25 MHz can be used, and the CKIO frequency range is 16 MHz to 25 MHz.

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and undergoes waveform shaping, and also frequency multiplication according to the setting, by PLL circuit 1 before being supplied to the SH7709. In modes 0 to 4, the system clock is generated from the output of the SH7709’s CKIO pin. Consequently, if a large number of ICs are operating on the clock cycle, the CKIO pin load will be large. This mode, however, assumes a comparatively large-scale system. If a large number of ICs are operating on the clock cycle, a clock generator with a number of low-skew clock outputs can be provided, so that the ICs can operate synchronously by distributing the clocks to each one.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Table 9.4 Range of Usable Frequencies for Each Clock Operating Mode

Mode	FRQCR Register Value	PLL1	PLL2	Clock Ratio* (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
0	0100	ON (× 1)	ON (× 1)	1:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0101	ON (× 1)	ON (× 1)	1:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0102	ON (× 1)	ON (× 1)	1:1:1/4	16 MHz to 40 MHz	16 MHz to 40 MHz
	0111	ON (× 2)	ON (× 1)	2:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0112	ON (× 2)	ON (× 1)	2:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0115	ON (× 2)	ON (× 1)	1:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0116	ON (× 2)	ON (× 1)	1:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0122	ON (× 4)	ON (× 1)	4:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	0126	ON (× 4)	ON (× 1)	2:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	012a	ON (× 4)	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	a100	ON (× 3)	ON (× 1)	3:1:1	25 MHz to 26.7MHz	25 MHz to 26.7MHz
	a101	ON (× 3)	ON (× 1)	3:1:1/2	25 MHz to 26.7MHz	25 MHz to 26.7MHz
	e100	ON (× 3)	ON (× 1)	1:1:1	25 MHz to 26.7MHz	25 MHz to 26.7MHz
	e101	ON (× 3)	ON (× 1)	1:1:1/2	25 MHz to 26.7MHz	25 MHz to 26.7MHz
1, 2	0100	ON (× 1)	ON (× 4)	4:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0101	ON (× 1)	ON (× 4)	4:4:2	5 MHz to 10 MHz	20 MHz to 40 MHz
	0102	ON (× 1)	ON (× 4)	4:4:1	5 MHz to 10 MHz	20 MHz to 40 MHz
	0111	ON (× 2)	ON (× 4)	8:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0112	ON (× 2)	ON (× 4)	8:4:2	5 MHz to 10 MHz	20 MHz to 40 MHz
	0115	ON (× 2)	ON (× 4)	4:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0116	ON (× 2)	ON (× 4)	4:4:2	5 MHz to 10 MHz	20 MHz to 40 MHz
	a100	ON (× 3)	ON (× 4)	12:4:4	5 MHz to 6.7MHz	20 MHz to 26.7MHz
	a101	ON (× 3)	ON (× 4)	12:4:2	5 MHz to 6.7MHz	20 MHz to 26.7MHz
	e100	ON (× 3)	ON (× 4)	4:4:4	5 MHz to 6.7MHz	20 MHz to 26.7MHz
	e101	ON (× 3)	ON (× 4)	4:4:2	5 MHz to 6.7MHz	20 MHz to 26.7MHz

Table 9.4 Range of Usable Frequencies for Each Clock Operating Mode (cont)

Mode	FRQCR Register Value	PLL1	PLL2	Clock Ratio* (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
3, 4	0100	OFF	ON (× 1)	1:1:1	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	0101	OFF	ON (× 1)	1:1:1/2	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	0102	OFF	ON (× 1)	1:1:1/4	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01d1	ON (× 2)	ON (× 1)	2:1:1/2	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01d0	ON (× 2)	ON (× 1)	2:1:1	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01d4	ON (× 2)	ON (× 1)	1:1:1	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01d5	ON (× 2)	ON (× 1)	1:1:1/2	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01d6	ON (× 2)	ON (× 1)	1:1:1/4	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	8IC0	ON (× 3)	ON (× 1)	3:1:1	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	8IC1	ON (× 3)	ON (× 1)	3:1:1/2	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	CIC0	ON (× 3)	ON (× 1)	1:1:1	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	CIC1	ON (× 3)	ON (× 1)	1:1:1/2	16 MHz to 26.7 MHz	16 MHz to 26.7 MHz
	01e0	ON (× 4)	ON (× 1)	4:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	01e4	ON (× 4)	ON (× 1)	2:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	01e5	ON (× 4)	ON (× 1)	2:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	01e6	ON (× 4)	ON (× 1)	2:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz
	01e8	ON (× 4)	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	01e9	ON (× 4)	ON (× 1)	1:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	01ea	ON (× 4)	ON (× 1)	1:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz
7	0100	ON (× 1)	OFF	1:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0101	ON (× 1)	OFF	1:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0102	ON (× 1)	OFF	1:1:1/4	16 MHz to 40 MHz	16 MHz to 40 MHz
	0111	ON (× 2)	OFF	2:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0112	ON (× 2)	OFF	2:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0115	ON (× 2)	OFF	1:1:1	16 MHz to 30 MHz	16 MHz to 30 MHz
	0116	ON (× 2)	OFF	1:1:1/2	16 MHz to 40 MHz	16 MHz to 40 MHz
	0122	ON (× 4)	OFF	4:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	0126	ON (× 4)	OFF	2:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	012a	ON (× 4)	OFF	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	a100	ON (× 3)	OFF	3:1:1	25 MHz to 26.7MHz	25 MHz to 26.7MHz
	e100	ON (× 3)	OFF	1:1:1	25 MHz to 26.7MHz	25 MHz to 26.7MHz
	e101	ON (× 3)	OFF	1:1:1/2	25 MHz to 26.7MHz	25 MHz to 26.7MHz

Note: Taking input clock as 1

Cautions:

1. When clock operating modes 3, 4 are used:
   - The on/off state of PLL circuit 1 is set by the frequency control register.
   - PLL circuit 1 is initialized to the off state by a power-on reset.
   - Always turn PLL circuit 1 off before going into standby mode.
2. The input to divider 1 becomes the output of:
   - PLL circuit 1 when PLL circuit 1 is on.
   - PLL circuit 2 when PLL circuit 1 is off and PLL circuit 2 is on.
3. The input of divider 2 becomes the output of:
   - PLL circuit 1 when the clock operating mode is 0–2 or 7.
   - PLL circuit 2 when the clock operating mode is 3, 4 and PLL circuit 2 is on.
4. The frequency of the internal clock (Iϕ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on.
   - Equal to the frequency of CKIO pin when PLL circuit 1 is off.
   - Do not set the internal clock frequency lower than the CKIO pin frequency.
5. The frequency of the peripheral clock (Pϕ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2 when the clock operating mode is 0–2 or 7.
   - The product of the frequency of the CKIO pin and the division ratio of divider 2 when the clock operating mode is 3, 4.
   - The peripheral clock frequency should not be set higher than the frequency of the CKIO pin, higher than 30 MHz, or lower than 1/4 the internal clock (Iϕ).
6. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1. This frequency should be equal to or lower than 80 MHz.
7. × 1, × 2, × 3, or × 4 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, × 1/3, and × 1/4 can be selected as the division ratios of dividers 1 and 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 is determined by the mode.

9.4 Register Descriptions

9.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit read/write register used to specify whether a clock is output from the CKIO pin, the on/off state of PLL circuit 1, PLL standby, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register. FRQCR is initialized to H'0102 by a power-on reset, but retains its value in a manual reset and in standby mode.

FRQCR:

Bit:	15	14	13	12	11	10	9	8
Bit name:	STC2	IFC2	PFC2	—	—	—	SLPFRQ	CKOEN
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	PLLEN	PSTBY	STC1	STC0	IFC1	IFC0	PFC1	PFC0
Initial value:	0	0	0	0	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 amd 5 and 4—Frequency Multiplication Ratio (STC2, STC1, STC0): These bits specify the frequency multiplication ratio of PLL circuit 1.

Bit 15: STC2	Bit 5: STC1	Bit 4: STC0	Description	
0	0	0	× 1	(Initial value)
0	0	1	× 2	
0	1	0	× 3	
1	0	0	× 4	

Note: Do not set the output frequency of PLL circuit 1 higher than 80 MHz.

Bits 14 and 3 and 2—Internal Clock Frequency Division Ratio (IFC2, IFC1, IFC0): These bits specify the frequency division ratio (frequency divider 1) of the internal clock with respect to the output frequency of PLL circuit 1. When PLL circuit 1 is off or in standby mode, set × 1.

Bit 14: IFC2	Bit 3: IFC1	Bit 2: IFC0	Description	
0	0	0	× 1	(Initial value)
0	0	1	× 1/2	
0	1	0	× 1/3	
1	0	0	× 1/4	

Note: Do not set the internal clock frequency lower than the CKIO frequency.

Bits 13 and 1 and 0—Peripheral Clock Frequency Division Ratio (PFC2, PFC1, PFC0): These bits specify the frequency division ratio (frequency divider 2) of the peripheral clock frequency with respect to the frequency of the output frequency of PLL circuit 1 or the frequency of the CKIO pin.

Bit 13: PFC2	Bit 1: PFC1	Bit 0: PFC0	Description	
0	0	0	× 1	
0	0	1	× 1/2	
0	1	0	× 1/3	(Initial value)
1	0	0	× 1/4	
1	0	1	× 1/6	

Note: Do not set the peripheral clock frequency higher than the frequency of the CKIO pin.

Bit 9—Divide Ratio of the External Bus Clock in the Sleep Mode (SLPFRQ): SLPFRQ specifies the divide ratio of the external bus clock in the sleep mode in clock modes 3 to 7. The clock frequency output from the CKIO is not changed. As refresh cycles are temporarily maintained by the bus state controller (BSC), it is not necessary to change RTSCR register or other settings.

Bit 9: SLPFRQ	Description	
0	External bus clock is not changed in the sleep mode.	(Initial value)
1	External bus clock is multiplied by 1/4 in the sleep mode.	

Bit 8—Clock Enable (CKOEN): Used to output a clock from the CKIO pin or to fix the level of the CKIO pin. Even when the level is fixed, the SH7709 will operate internally at the frequency before the level was fixed. When using clock operating modes 0 to 2, this bit should be set to 1. In case of clock operating mode 7, the CKIO pin becomes an input pin irrespective of the value of this bit.

Bit 8: CKOEN	Description	
0	Fixes the level of CKIO terminal.	
1	Outputs a clock from the CKIO pin.	(Initial value)

Bit 7—PLL Circuit Enable (PLLEN): Specifies the on/off state of PLL circuit 1. This bit is valid in clock operating modes 3, 4. PLL circuit 1 goes on when the clock operating mode is 0–2 or 7 irrespective of the value of PLLEN.

Bit 7: PLLEN	Description	
0	PLL circuit 1 is not used.	(Initial value)
1	PLL circuit 1 is used.	

Bit 6—PLL Standby (PSTBY): Specifies PLL standby. When PLL standby is active, PLL circuit 1 will be in standby mode at the frequency specified by the STC bit. This function is valid in clock operating modes 3, 4.

Bit 6: PSTBY	Description	
0	PLL is not in standby mode.	(Initial value)
1	PLL is in standby mode.	

9.5 Changing the Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of dividers 1 and 2. All of these are controlled by software through the frequency control register. The methods are described below. In modes 3, 4 the frequency can also be changed by turning PLL circuit 1 on and off, as described in section 9.6, PLL Standby Function.

9.5.1 Changing the Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:
   WTCSR register TME bit = 0: WDT stops
   WTCSR register CKS2–CKS0 bits: Division ratio of WDT count clock
   WTCNT counter: Initial counter value
3. Set the desired value in the STC2, STC1 and STC0 bits. The division ratio can also be set in the IFC2–IFC0 bits and PFC2–PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. In clock modes 0–2 and 7, the internal and peripheral clocks both stop. In clock modes 3, 4 only the internal clock stops. The clock will continue to be output at the CKIO pin as long as the CKOEN bit in the FRQCR register is set to 1.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

9.5.2 Changing the Division Ratio

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

1. In the initial state, IFC2–IFC0 = 000 and PFC2–PFC0 = 010.
2. Set the IFC2, IFC1, IFC0, PFC2, PFC1, and PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

9.6 PLL Standby Function

9.6.1 Overview of the PLL Standby Function

When operating in clock modes 3, 4 the internal clock can be controlled by turning the PLL1 circuit on and off. A long oscillation settling time is required, however, when the PLL circuit is started up from a complete halt. During this time, processor operation halts. To enable fast on/off switching of the PLL1 circuit, the PLL standby function is provided. This function is controlled by software using the frequency control register. The use of the PLL standby function is described below.

9.6.2 Usage

From Off to On:

1. Initially, PSTBY = 0, PLLEN = 0, and PLL circuit 1 is stopped. The output of PLL circuit 2 or divider 3 is used for divider 1 input.
2. When the multiplication rate of PLL circuit 1 is set in the STC2–STC0 bits and PSTBY is set to 1, PLL circuit 1 begins oscillating at the specified multiplication rate. The input to divider 1 is still the output of PLL circuit 2 or divider 3 at this point.
3. After PLL circuit 1 oscillation has stabilized, the input of divider 1 switches when PLLEN is set to 1 and the oscillation output of PLL circuit 1 is divided and becomes the internal clock. At this time, the division ratio can be changed by changing the settings of IFC2–IFC0 and PFC2–PFC0. For several cycles before and after the clock switches, the internal clock will be stopped, but the peripheral clock and CKIO output do not stop.

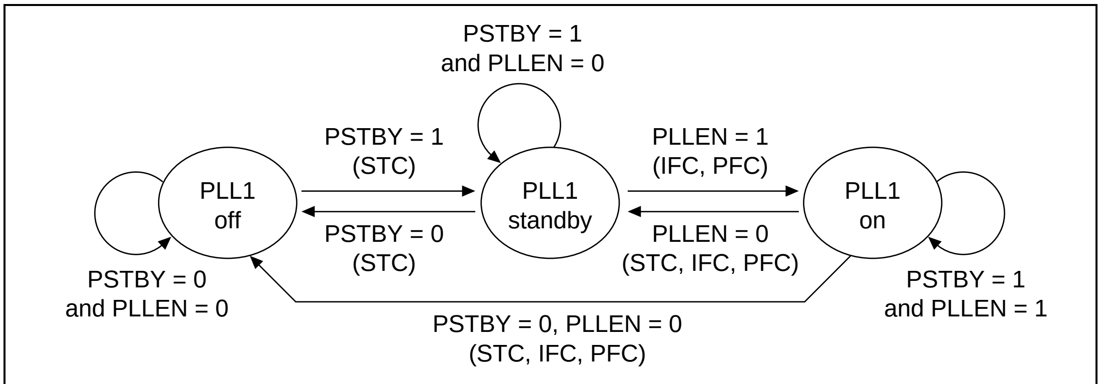
From On to Off:

1. When PLLEN is set to 0, the input of divider 1 switches to the output of PLL circuit 2. At this time, the division ratio can be changed by changing the settings of IFC2–IFC0 and PFC2–PFC0.
2. When PSTBY is set to 0, PLL circuit 1 stops. This setting can be performed simultaneously (and with the same instruction as) the setting in 1 above.

Notes: 1. There are some restrictions on the PLL standby state (PSTBY = 1, PLLEN = 0) as follows: The settings of the frequency control register's CKOEN, STC2–STC0, IFC2–IFC0 and PFC2–PFC0 bits generally cannot be changed. In some cases, however, they can be changed if the PSTBY and PLLEN bit settings are also changed simultaneously (figure 9.2). The SLEEP instruction cannot be executed.

2. It is the responsibility of software to ensure the oscillation settling time. If PLLEN is set to 1 before the oscillation has settled, malfunctions may be caused by an unstable clock.

3. In clock modes 3, 4 the SH7709 cannot go to standby mode while PLL circuit 1 is on. Always set PSTBY and PLLEN to 0 to stop PLL circuit 1 before going to standby mode.
4. When PSTBY and PLLEN are both changed from 0 to 1 together, the WDT will automatically start counting and the clock will switch when the WDT overflows. See section 9.5, Changing the Frequency, for setting the WDT.



Note: Bits in parentheses can be changed simultaneously.

Figure 9.2 State Transitions for the PLL Standby Function

9.7 Controlling Clock Output

The CKOEN bit in the FRQCR register can be used to switch between outputting a clock to the CKIO pin or having the level fixed.

9.7.1 Clock Modes 0–2

The CKIO pin level cannot be fixed. Always set the CKOEN bit in FRQCR to 1 (clock output).

9.7.2 Clock Modes 3, 4

The CKIO output changes as soon as the CKOEN bit is changed. When the WDT is started by simultaneously changing the multiplication rate of PLL circuit 1 or switching PLL circuit 1 on or off, the WDT starts running after the CKIO output is switched, and then the internal clock changes.

9.8 Overview of the WDT

9.8.1 Block Diagram of the WDT

Figure 9.3 shows a block diagram of the WDT.

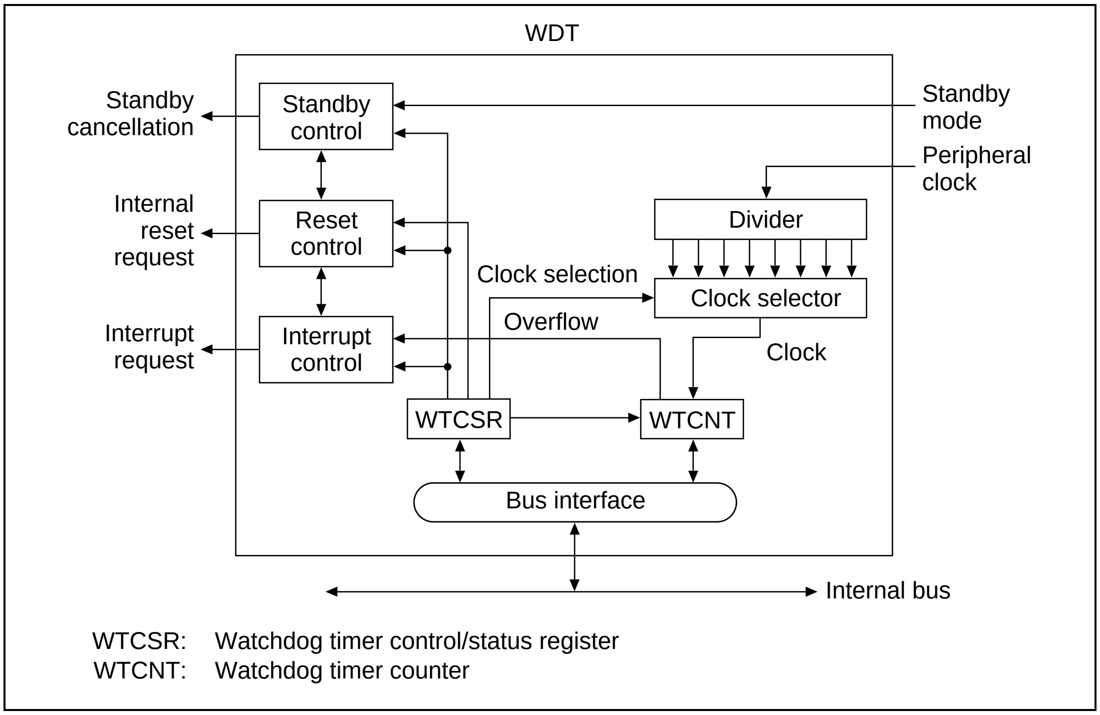


Figure 9.3 Block Diagram of the WDT

9.8.2 Register Configurations

The WDT has two registers that select the clock, switch the timer mode, and perform other functions. Table 9.5 shows the WDT register.

Table 9.5 Register Configuration

Name	Abbreviation	R/W	Size	Initial Value	Address
Watchdog timer counter	WTCNT	R/W*	R: byte; W: word*	H'00	H'FFFFFF84
Watchdog timer control/status register	WTCSR	R/W*	R: byte; W: word*	H'00	H'FFFFFF86

Note: Write with a word access. Write H'5A and H'A5, respectively, in the upper bytes. Byte or longword writes are not possible. Read with a byte access.

9.9 WDT Registers

9.9.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit read/write counter that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. Its address is H'FFFFFF84. The WTCNT counter is initialized to H'00 only by a power-on reset through the RESET pin. Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to read WTCNT.

Bit:	7	6	5	4	3	2	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

9.9.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags. Its address is H'FFFFFF86. The WTCSR register is initialized to H'00 only by a power-on reset through the RESET pin. When a WDT overflow causes an internal reset, the WTCSR retains its value. When used to count the clock settling time for canceling a standby, it retains its value after counter overflow. Use a word access to write to the WTCSR counter, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Bit:	7	6	5	4	3	2	1	0
	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
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 7—Timer Enable (TME): Starts and stops timer operation. Clear this bit to 0 when using the WDT in standby mode or when changing the clock frequency.

Bit 7: TME	Description
0	Timer disabled: Count-up stops and WTCNT value is retained
1	Timer enabled

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether to use the WDT as a watchdog timer or an interval timer.

Bit 6: WT/$\overline{\text{IT}}$	**Description**	
0	Use as interval timer	(Initial value)
1	Use as watchdog timer	

Note: If WT/$\overline{\text{IT}}$ is modified when the WDT is running, the up-count may not be performed correctly.

Bit 5—Reset Select (RSTS): Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored.

Bit 5: RSTS	**Description**	
0	Power-on reset	(Initial value)
1	Manual reset	

Note: When WTCNT overflows and a reset is effected, RESETOUT is output (low). STATUS0 and STATUS1 go to the reset state.

Bit 4—Watchdog Timer Overflow (WOVF): Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode.

Bit 4: WOVF	**Description**	
0	No overflow	(Initial value)
1	WTCNT has overflowed in watchdog timer mode	

Bit 3—Interval Timer Overflow (IOVF): Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode.

Bit 3: IOVF	**Description**	
0	No overflow	(Initial value)
1	WTCNT has overflowed in interval timer mode	

Bits 2 to 0—Clock Select 2–0 (CKS2–CKS0): These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock. The overflow period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Division Ratio		Overflow Period (when Pϕ = 15 MHz)
0	0	0	1	(Initial value)	17 µs
		1	1/4		68 µs
	1	0	1/16		273 µs
		1	1/32		546 µs
1	0	0	1/64		1.09 ms
		1	1/256		4.36 ms
	1	0	1/1024		17.46 ms
		1	1/4096		69.84 ms

Note: If bits CKS2–CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

9.9.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 9.4. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

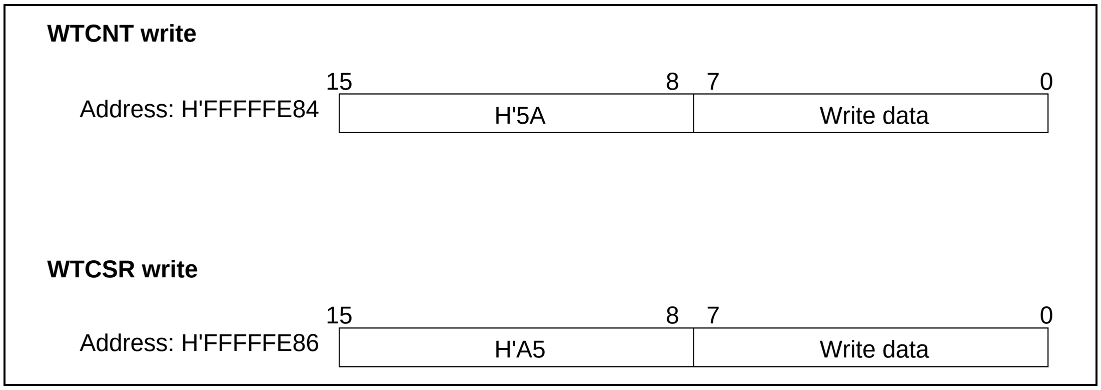


Figure 9.4 Writing to WTCNT and WTCSR

9.10 Using the WDT

9.10.1 Canceling Standbys

The WDT can be used to cancel standby mode with an NMI or other interrupts. The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the RESET pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal or detecting interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. As the WDT counts continuously from H'00, the WDT count should be stopped by clearing the STBY bit in the STBCR register to 0 in the interrupt processing program. If the STBY bit is left set to 1, standby mode will be entered again when the WDT count reaches H'80. This standby mode can only be exited by means of a power-on reset.

9.10.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits of WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the clock stops and the processor enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00–H'01. The stop value depends on the clock ratio.

9.10.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2–CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

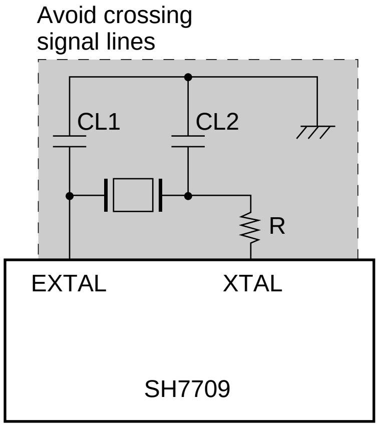
9.10.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 0, set the type of count clock in the CKS2–CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to INTC. The counter then resumes counting.

9.11 Notes on Board Design

When Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



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

Figure 9.5 Points for Attention when Using Crystal Resonator

Decoupling Capacitors: As far as possible, insert a laminated ceramic capacitor of 0.01 to 0.1 µF as a passive capacitor for each V_{SS}/V_{CC} pair. Mount the passive capacitors as close as possible to the SH7709 power supply pins, and use components with a frequency characteristic suitable for the SH7709 operating frequency, as well as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 19-21, 27-29, 33-35, 45-47, 57-59, 69-71, 79-81, 83-85, 95-97, 109-111, 132-134, 141-163, 173-176, 181-183

On-chip oscillator V_{SS}/V_{CC} pairs: 147-145, 6-3, (148,152)-150, 153-154

Analog V_{SS}/V_{CC} pairs: (198, 208)-205

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{SS} (PLL1) and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and do not locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{CC} or V_{SS} and leave the XTAL pin open.

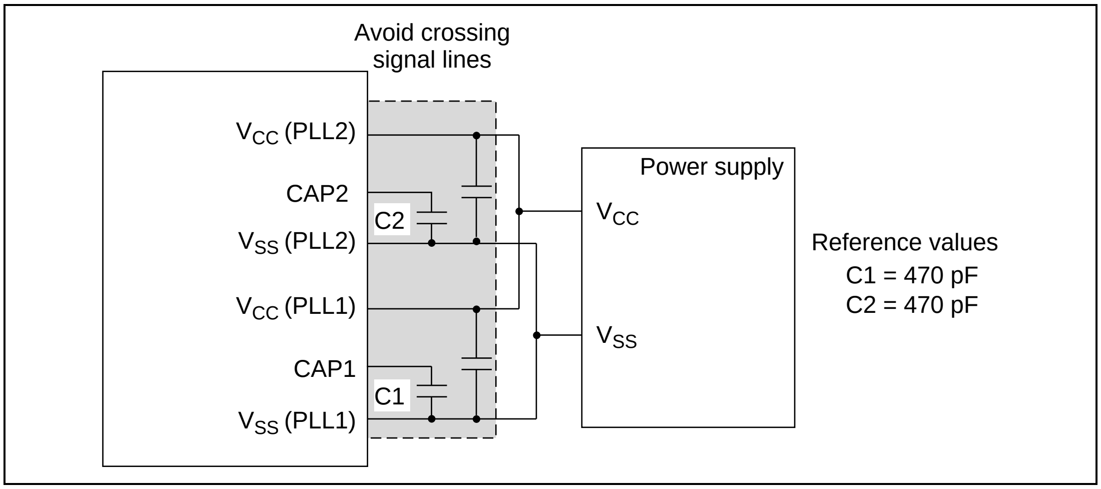


Figure 9.6 Points for Attention when Using PLL Oscillator Circuit

Section 10 Bus State Controller (BSC)

10.1 Overview

This LSI has two bus state controllers (hereafter referred to as BSC and BSCP) for the CPU / cache memory and for the DMAC. The two bus state controllers function in cooperation with each other. The bus state controllers (BSC and BSCP) divide physical address space and output control signals for various types of memory and bus interface specifications. BSC and BSCP functions enable this LSI to link directly with DRAM, SDRAM, SRAM, ROM, and other memory storage devices without an external circuit. The BSC also allows direct connection to PCMCIA interfaces, simplifying system design and allowing high-speed data transfers in a compact system.

10.1.1 Features

The BSC has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2–6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the WAIT pin
  - Wait state insertion can be controlled through software. Register settings can be used to specify the insertion of 1–10 cycles independently for each area
  - The type of memory connected can be specified for each area, and control signals are output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous memory accesses to different areas or writes directly following reads of the same area
- Direct interface to DRAM
  - Multiplexes row/column addresses according to DRAM capacity
  - Supports burst operation (high-speed page mode, hyper page mode)
  - Supports CAS-before-RAS refresh and self-refresh
  - Controls timing of DRAM direct-connection control signals according to register settings
- Direct interface to SDRAM
  - Multiplexes row/column addresses according to SDRAM capacity
  - Supports burst operation
  - Has both auto-refresh and self-refresh functions
  - Controls timing of SDRAM direct-connection control signals according to register setting
- ROM burst interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface

  - Insertion of wait states controllable through software
  - Burst operation (page mode)
  - Bus sizing function for I/O bus width
- Disables the output of clock signals to the BSC in the sleep mode to reduce power consumption

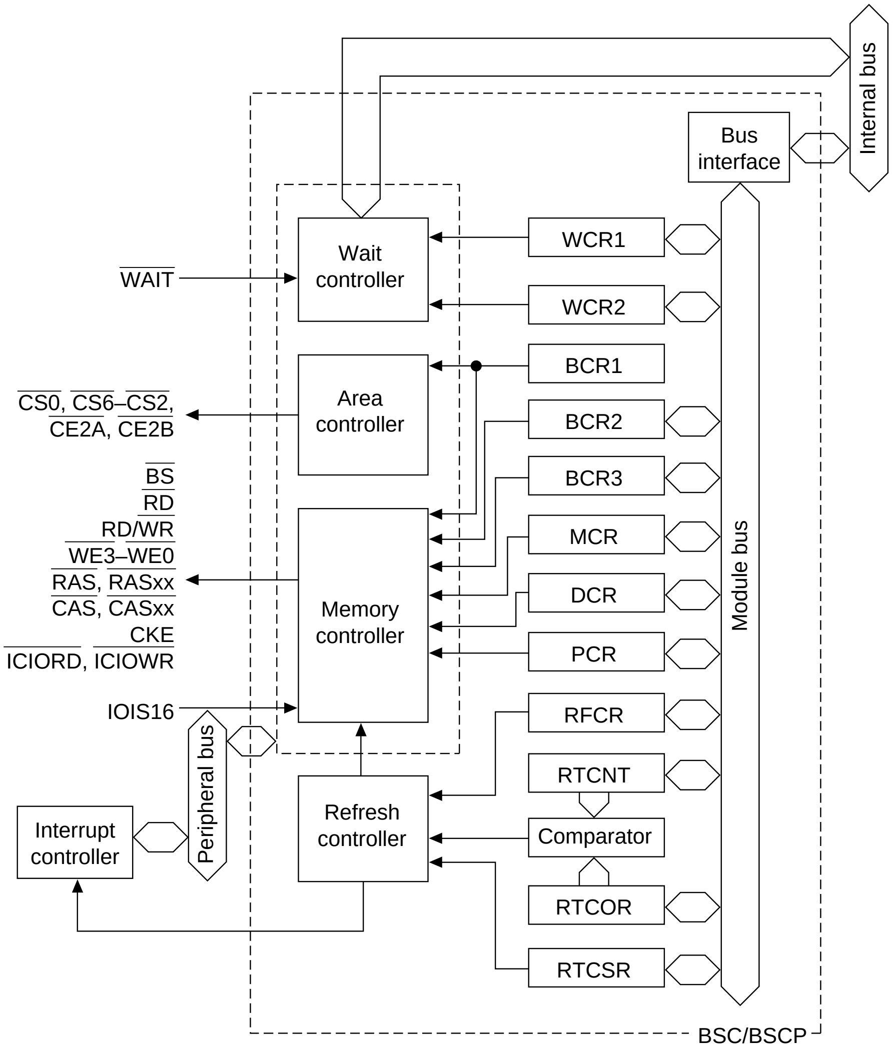
The BSCP has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2–6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the WAIT pin
  - Wait state insertion can be controlled through software. Register settings can be used to specify the insertion of 1–10 cycles independently for each area
  - The type of memory connected can be specified for each area, and control signals are output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous memory accesses to different areas or writes directly following reads of the same area
- Direct interface to DRAM
  - Multiplexes row/column addresses according to DRAM capacity
  - Supports burst operation (high-speed page mode, hyper page mode)
  - Supports CAS-before-RAS refresh and self-refresh
  - Supports byte control using CAS4 corresponding to low power consumption
  - Controls timing of DRAM direct-connection control signals according to register settings
- Direct interface to SDRAM
  - Multiplexes row/column addresses according to SDRAM capacity
  - Supports burst operation
  - Has both auto-refresh and self-refresh functions
  - Controls timing of SDRAM direct-connection control signals according to register setting
- ROM burst interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface
  - Insertion of wait states controllable through software
  - Burst operation (page mode)
  - Bus sizing function for I/O bus width
- Refresh function
  - Refresh cycles will be automatically maintained in the sleep mode even after the external bus frequency is reduced to 1/4 of its normal operating frequency

- Short refresh cycle control
  - The overflow interrupt function of the refresh counter enables the refresh function immediately after the self-refresh operation using the low power-consumption DRAM
- The refresh counter can be used as an interval timer
  - Outputs an interrupt request signal using the compare-matching function
  - Outputs an interrupt request signal when the refresh counter overflows
- Automatically disables the output of clock signals to anywhere but the refresh counter, except during execution of external bus cycles

10.1.2 Block Diagram

Figures 10.1 (a) and (b) shows the functional block diagram of the bus state controller.



WCR:	Wait state control register	RFCR:	Refresh count register
BCR:	Bus control register	RTCNT:	Refresh timer count register
MCR:	Memory control register	RTCOR:	Refresh time constant register
DCR:	DRAM control register	RTCSR:	Refresh timer control/status register
PCR:	PCMCIA control register		

Figure 10.1 BSC / BSCP Functional Block Diagram (a)

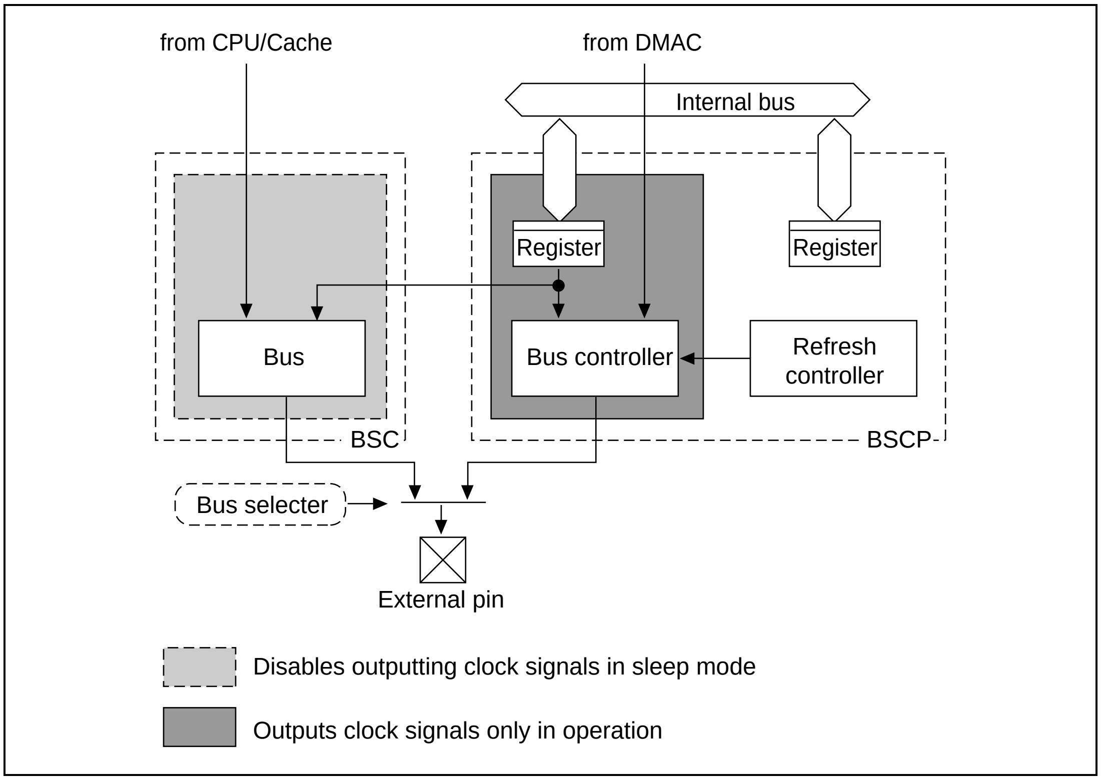


Figure 10.1 BSC / BSCP Functional Block Diagram (b)

10.1.3 Pin Configuration

Table 10.1 lists the BSC pin configuration.

Table 10.1 Pin Configuration (Preliminary)

Pin Name	Symbol	I/O	Description
Address bus	A25–A0	O	Address output
Data bus	D15–D0	I/O	Data I/O
	D31–D16	I/O	When 32-bit bus width, data I/O
Bus cycle start	$\overline{\text{BS}}$	O	Shows start of bus cycle. During burst transfers, asserts every data cycle.
Chip select 0, 2–4	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$–$\overline{\text{CS4}}$	O	Chip select signal to indicate area being accessed.
Chip select 5, 6	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$, $\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O	Chip select signal to indicate area being accessed. CS5/CE1A and CS6/CE1B can also be used as CE1A and CE1B of PCMCIA.
Read/write	RDWR	O	Data bus direction indicator signal. DRAM/SDRAM/PCMCIA write indicator signal.
Row address strobe 3L	$\overline{\text{RAS3L}}$	O	When DRAM or SDRAM is used in area 3, RAS3L for lower 32Mbyte address.
Row address strobe 3U	$\overline{\text{RAS3U}}$	O	When DRAM or SDRAM is used in area 3, RAS3U for upper 32Mbyte address.
Column address strobe	$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$	O	When DRAM is used, CASLL signal for D7–D0. When SDRAM is used, CAS signal.
Column address strobe LH	$\overline{\text{CASLH}}$	O	When DRAM is used, CASLH signal for D15–D8
Column address strobe HL	$\overline{\text{CASHL}}$	O	When DRAM is used, CASHL signal for D23–D16.
Column address strobe HH	$\overline{\text{CASHH}}$	O	When DRAM is used, CASHH signal for D31–D24.
Column address strobe 2L	$\overline{\text{CAS2L}}$		When the area 2 DRAM is used, CAS2L signal for D7–D0.
Column address strobe 2H	$\overline{\text{CAS2H}}$		When the area 2 DRAM is used, CAS2H signal for D15–D8.

Table 10.1 Pin Configuration (Preliminary) (cont)

Pin Name	Symbol	I/O	Description
Data enable 0	$\overline{\text{WE0}}$/DQMLL	O	When memory other than SDRAM is used, selects D7–D0 write strobe signal. When SDRAM is used, selects D7–D0.
Data enable 1	$\overline{\text{WE1}}$/DQMLU/ $\overline{\text{WE}}$	O	When memory other than SDRAM and PCMCIA is used, selects D15–D8 write strobe signal. When SDRAM is used, selects D15–D8. When PCMCIA is used, strobe signal that indicates the write cycle.
Data enable 2	$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$	O	When memory other than SDRAM and PCMCIA is used, selects D23–D16 write strobe signal. When SDRAM is used, selects D23–D16. When PCMCIA is used, strobe signal indicating I/O read.
Data enable 3	$\overline{\text{WE3}}$/DQMUU/ $\overline{\text{ICIOWR}}$	O	When memory other than SDRAM and PCMCIA is used, selects D31–D24 write strobe signal. When SDRAM is used, selects D31–D24. When PCMCIA is used, strobe signal indicating I/O write.
Read	$\overline{\text{RD}}$	O	Strobe signal indicating read cycle
Wait	$\overline{\text{WAIT}}$	I	Wait state request signal
IOIS16	$\overline{\text{IOIS16}}$	I	Signal indicating PCMCIA 16 bit I/O. Valid only in little-endian mode.
Clock enable	CKE	O	Clock enable control signal of SDRAM
Bus release request	$\overline{\text{BREQ}}$	I	Bus release request signal
Bus release acknowledgment	$\overline{\text{BACK}}$	O	Bus release acknowledge signal
PCMCIA card select	$\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$	O	When PCMCIA is used, CE2A and CE2B
Row address strobe 2L	$\overline{\text{RAS2L}}$	O	When DRAM is used in area 2, RAS2L signal for lower 32Mbyte address
Row address strobe 2U	$\overline{\text{RAS2U}}$	O	When DRAM is used in area 2, RAS2U signal for upper 32Mbyte address

10.1.4 Register Configuration

The BSC has ten registers (table 10.2). The SDRAM also has a built-in SDRAM mode register. These registers control direct connection interfaces to memory, wait states, refreshes, and PCMCIA devices.

Table 10.2 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Bus Width
Bus control register 1	BCR1	R/W	H'0000	H'FFFFFF60	16
Bus control register 2	BCR2	R/W	H'3FF0	H'FFFFFF62	16
Wait state control register 1	WCR1	R/W	H'3FF3	H'FFFFFF64	16
Wait state control register 2	WCR2	R/W	H'FFFF	H'FFFFFF66	16
Individual memory control register	MCR	R/W	H'0000	H'FFFFFF68	16
DRAM control register	DCR	R/W	H'0000	H'FFFFFF6A	16
PCMCIA control register	PCR	R/W	H'0000	H'FFFFFF6C	16
Refresh timer control/status register	RTCSR	R/W	H'0000	H'FFFFFF6E	16
Refresh timer counter	RTCNT	R/W	H'0000	H'FFFFFF70	16
Refresh time constant register	RTCOR	R/W	H'0000	H'FFFFFF72	16
Refresh count register	RFCR	R/W	H'0000	H'FFFFFF74	16
Bus control register 3	BCR3	R/W	H'0000	H'FFFFFF7E	16
SDRAM mode register, area 2	SDMR	W	—	H'FFFFD000– H'FFFFDFFF	8*1
SDRAM mode register, area 3			—	H'FFFFE000– H'FFFFEFFF	

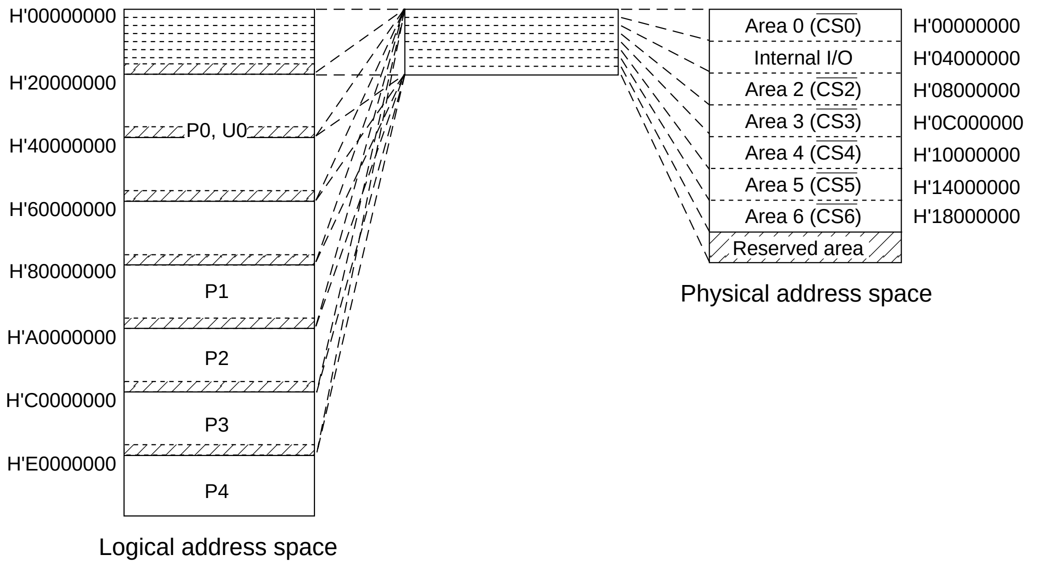
Notes: 1. For details see section 10.2.8, SDRAM Mode Register.

10.1.5 Area Overview

Space Allocation: In the SH-3 architecture, both logical spaces and physical spaces have 32-bit address spaces. The logical space is divided into five areas by the value of the upper bits of the address. The physical space is divided into eight areas.

Logical space can be allocated at will to physical spaces using a memory management unit (MMU). For details, refer to section 3, Memory Management Unit, which describes area allocation for physical spaces.

As listed in table 10.4, the SH-3 can be connected directly to seven areas of memory/PCMCIA interface, and it outputs chip select signals (CS0, CS2–CS6, CE2A, CE2B) for each of them. CS0 is asserted during area 0 access; CS6 is asserted during area 6 access. When DRAM and SDRAM are connected to areas 2 or 3, signals such as RAS, CAS, RDWR, and DQM are also asserted. When PCMCIA interface is selected in areas 5 or 6, in addition to CS5/CS6, CE2A/CE2B are asserted for the corresponding bytes accessed.



Note: For logical address spaces P0 and P3, when the memory management unit (MMU) is on, it can optionally generate a physical address for the logical address. It can be applied when the MMU is off and when the MMU is on and each physical address for the logical address is equal except for upper three bits.

Figure 10.2 Corresponding to Logical Address Space and Physical Address Space

Table 10.3 Physical Address Space Map

Area	Physical Address	Connectable Memory	Capacity	Access Size
0	H'00000000 to H'03FFFFFF	Ordinary memory*[1], burst ROM	64 Mbytes	8, 16, 32*[2]
	H'00000000 + H'20000000 × n to H'03FFFFFF + H'20000000 × n		Shadow	n: 1–6
1	H'04000000 to H'07FFFFFF	Internal I/O registers*[8]	64 Mbytes	8, 16, 32*[3]
	H'04000000 + H'20000000 × n to H'07FFFFFF + H'20000000 × n		Shadow	n: 1–6
2	H'08000000 to H'0BFFFFFF	Ordinary memory*[1], SDRAM, DRAM	64 Mbytes	8, 16, 32*[3], *[4]
	H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n		Shadow	n: 1–6
3	H'0C000000 to H'0FFFFFFF	Ordinary memory*[1], SDRAM, DRAM	64 Mbytes	8, 16, 32*[3], *[5]
	H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n		Shadow	n: 1–6
4	H'10000000 to H'13FFFFFF	Ordinary memory*[1]	64 Mbytes	8, 16, 32*[3]
	H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n		Shadow	n: 1–6
5*[5]	H'14000000 to H'15FFFFFF	Ordinary memory*[1], PCMCIA, burst ROM	32 Mbytes	8, 16, 32*[3], *[6]
	H'16000000 to H'17FFFFFF	Ordinary memory*[1], burst ROM	32 Mbytes	
	H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n		Shadow	n: 1–6
6	H'18000000 to H'19FFFFFF	Ordinary memory*[1], PCMCIA, burst ROM	32 Mbytes	8, 16, 32 *[3], *[6]
	H'1A000000 to H'1BFFFFFF			
	H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n		Shadow	n: 1–6
7	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n	Reserved area*[7]		n: 0–7

Notes: 1. Memory with an interface, such as SRAM and ROM

2. Memory bus width is specified by external pins.

3. Memory bus width is specified by a register setting.

4. With the synchronous DRAM interface, the bus width is 32 bits only.
With the DRAM interface, the bus width is 16 bits only.

5. With the synchronous DRAM interface, the bus width is 32 bits only.
With the DRAM interface, the bus width is 16 or 32 bits only.
When the DRAM interface is specified for both area 2 and area 3, the bus width is 16 bits only.

6. With the PCMCIA interface, the bus width is 8 or 16 bits only.

7. Do not access a reserved area, as operation cannot be guaranteed in this case.

8. If an area 1 control register is not to be subject to address translation by the MMU, set the first 3 bits of the logical address to 101 to locate the register in the P2 area.

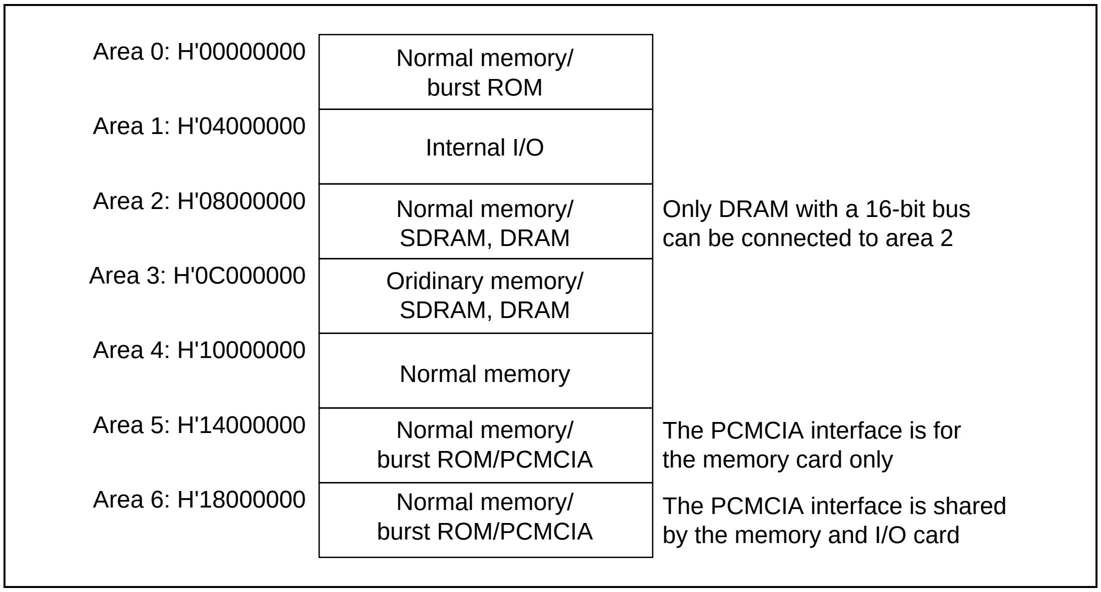


Figure 10.3 Physical Space Allocation

Memory Size: The memory size in this LSI can be set for each area. In area 0, an external pin can be used to select byte (8 bits), word (16 bits), or longword (32 bits) on power-on reset. The correspondence between the external pins (MD4 and MD3) and memory size is listed in table below.

Table 10.4 Correspondence between External Pins (MD4 and MD3) and Memory Size

MD4	MD3	Memory Size
0	0	Reserved (Do not set)
0	1	8 bits
1	0	16 bits
1	1	32 bits

For areas 2–6, byte, word, and longword may be chosen for the bus width using bus control register 2 (BCR2) whenever ordinary memory, ROM, or burst ROM are used. When the DRAM interface are used, word or longword can be chosen as the bus width using the individual memory control register (MCR). Set the bus width to longword with MCR for SDRAM interface.

When area 2 is used as a DRAM area, set the bus widths of areas 2 to word. When areas 5 and 6 are used as PCMCIA interfaces, set the bus width to byte or word. When using the port function, set each of the bus widths to byte or word for all areas. For more information, see section 10.2.2, Bus Control Register 2 (BCR2), and section 10.2.5, Individual Memory Control Register (MCR).

Shadow Space: Areas 0, 2–6 are decoded by physical addresses A28–A26, which correspond to areas 000 to 110. Address bits 31–29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space obtained by adding to it H'20000000 × n (n = 1–6). The address range for area 7, which is on-chip I/O space, is H'FC000000 to H'FFFFFFFF. The address space H'1C000000 + H'20000000 × n–H'1FFFFFFF + H'20000000 × n (n = 0–6) corresponding to the area 7 shadow space is reserved, so do not use it.

10.1.6 PCMCIA Support

The SH-3 supports PCMCIA standard interface specifications in physical space areas 5 and 6.

The interface supported with BSC is basically the IC memory card interface and I/O card interface defined by PCMCIA Specifications Version 2.1. In addition, burst access is supported to enable high-speed access.

Physical space area 5 supports IC memory card Interface only; area 6 supports both IC memory card interface and the I/O card interface.

Table 10.5 PCMCIA Interface Characteristics

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Mask 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 of I/O bus width* The PCMCIA interface can be accessed from the address conversion region non-address conversion region.

Note*: Dynamic bus sizing of I/O bus width is supported only in the little endian mode.

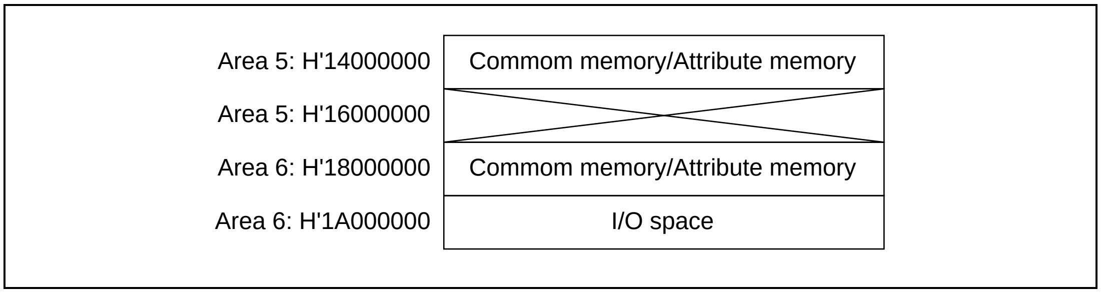


Figure 10.4 PCMCIA Space Allocation

Table 10.6 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**SH7709 Pin**
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	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CE1A}}$ or $\overline{\text{CE1B}}$
8	A10	I	Address	A10	I	Address	A10
9	OE	I	Output enable	$\overline{\text{OE}}$	I	Output enable	$\overline{\text{RD}}$
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	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE}}$
16	+RDY/$\overline{\text{BSY}}$	O	Ready/Busy	$\overline{\text{IREQ}}$	O	Ready/Busy	—
17	V_{CC}		Operation power	V_{CC}		Operation power	—
18	VPP1		Program power	VPP1		Program/ peripheral power	—

Table 10.6 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**SH7709 Pin**
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
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	$\overline{\text{IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	—
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	CE2	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	$\overline{\text{VS1}}$	I	Voltage sense 1	VS1	I	Voltage sense 1	—
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{ICIOWR}}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19

Table 10.6 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**SH7709 Pin**
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	V_{CC}		Power supply	V_{CC}		Power supply	—
52	VPP2		Program power	VPP2		Program/ peripheral power	—
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	$\overline{\text{VS2}}$	I	Voltage sense 2	VS2	I	Voltage sense 2	—
58	RESET	I	Reset	+RESET	I	Reset	—
59	$\overline{\text{WAIT}}$	O	Wait request	WAIT	O	Wait request	—
60	RFU		Reserved	INPACK	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	REG	I	Attribute memory space select	—
62	BVD2	O	Battery voltage detection	SPKR	O	Digital voice signal	—
63	BVD1	O	Battery voltage detection	STSCHG	O	Card status change	—
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	CD2	O	Card detection	—
68	GND		Ground	GND		Ground	—

10.2 BSC, BSCP Registers

10.2.1 Bus Control Register 1 (BCR1)

"BCR1" can be used for BSC and BSCP. The bus control register 1 (BCR1) is a 16-bit read/write register that sets the functions and bus cycle status for each area. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR1 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
Initial value:	—	—	0	0	0/1*	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	A5BST0	A6BST1	A6BST0	DRAM TP2	DRAM TP1	DRAM TP0	A5 PCM	A6 PCM
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

Note: Samples the value of the external pin designating endian at power-on reset.

Bits 15–14—Reserved: Bits 15–14 cannot be modified and always read as 0.

Bit 13—Hi-Z memory control (HIZMEM): Specifies the status of A25-0, $\overline{BS}$, $\overline{CS}$, RD/$\overline{WR}$, $\overline{WE}$/DQM, $\overline{RD}$, $\overline{CE2A}$/PTE[4], $\overline{CE2B}$/PTE[5] and DRAK0/1 in standby mode.

Bit 13: HIZMEM	Description
0	A25-0, $\overline{BS}$, $\overline{CS}$, RD/$\overline{WR}$, $\overline{WE}$/DQM, $\overline{RD}$, $\overline{CE2A}$/PTE[4], $\overline{CE2B}$/PTE[5] and DRAK0/1 are Hi-Z in standby mode.
1	A25-0 is Low in standby mode. $\overline{BS}$, $\overline{CS}$, RD/$\overline{WR}$, $\overline{WE}$/DQM, $\overline{RD}$, $\overline{CE2A}$/PTE[4], $\overline{CE2B}$/PTE[5] and DRAK0/1 are High in standby mode.

Bit 12—High-Z Control (HIZCNT): Specifies the status of the RAS and the CAS signals at standby and bus right release.

Bit 12: HIZCNT	Description
0	The RAS and the CAS signals are high-impedance status (High-Z) at standby and bus right release. (Initial value)
1	The RAS and the CAS signals are driven at standby and bus right release.

Bit 11—Endian Flag (ENDIAN): Samples the value of the external pin designating endian upon a power on reset. Endian for all physical spaces is decided by this bit, which is read-only.

Bit 11: ENDIAN	Description
0	(On reset) Endian setting external pin (MD5) is low. Indicates the SH-3 is set as big endian.
1	(On reset) Endian setting external pin (MD5) is high. Indicates the SH-3 is set as little endian.

Bits 10, 9—Area 0 Burst ROM Control (A0BST1–A0BST0): Specify whether to use burst ROM in physical space area 0. When burst ROM is used, set the number of burst transfers.

Bit 10: A0BST1	Bit 9: A0BST0	Description
0	0	Access area 0 as ordinary memory (initial value)
	1	Access area 0 as burst ROM (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Access area 0 as burst ROM (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Access area 0 as burst ROM (16 consecutive accesses). Can be used only when bus width is 8.

Bits 8, 7—Area 5 Burst Enable (A5BST1–A5BST0): Specify whether to use burst ROM and PCMCIA burst mode in physical space area 5. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers.

Bit 8: A5BST1	Bit 7: A5BST0	Description
0	0	Access area 5 as ordinary memory (initial value)
	1	Burst access of area 5 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 5 (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Burst access of area 5 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 6, 5—Area 6 Burst Enable (A6BST1–A6BST0): Specify whether to use burst ROM and PCMCIA burst mode in physical space area 6. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers.

Bit 6: A6BST1	Bit 5: A6BST0	Description
0	0	Access area 6 as ordinary memory (initial value)
	1	Burst access of area 6 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 6 (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Burst access of area 6 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 4–2—Area 2, Area 3 Memory Type (DRAMTP2, DRAMTP1, DRAMTP0): Designate the types of memory connected to physical space areas 2 and 3. Normal memory, such as ROM, SRAM, or flash RAM, can be directly connected. DRAM, and SDRAM can also be directly connected.

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Areas 2 and 3 are normal memory
		1	Reserved
	1	0	Area 2: normal memory; area 3: SDRAM
		1	Areas 2 and 3 are SDRAM
1	0	0	Area 2: normal memory; area 3: DRAM
		1	Areas 2 and 3 are DRAM *
	1	0	Reserved
		1	Reserved

Note: When selecting these bits, set the area 2 bus widths as word.

Bit 1—Area 5 Bus Type (A5PCM): Designates whether to access physical space area 5 as PCMCIA space.

Bit 1: A5PCM	Description
0	Access physical space area 5 as normal memory
1	Access physical space area 5 as PCMCIA space

Bit 0—Area 6 Bus Type (A6PCM): Designates whether to access physical space area 6 as PCMCIA space.

Bit 0: A6PCM	Description
0	Access physical space area 6 as normal memory
1	Access physical space area 6 as PCMCIA space

10.2.2 Bus Control Register 2 (BCR2)

"BCR2" can be used for BSC and BSCP. The bus control register 2 (BCR2) is a 16-bit read/write register that selects the bus size width of each area. It is initialized to H'3FF0 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR2 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0
Initial value:	0	0	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—
Initial value:	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bits 15, 14, 3–0—Reserved: These bits cannot be modified and always read as 0.

Bits 2n + 1, 2n—Area n (2–6) Bus Size Specification (AnSZ1, AnSZ0): Specify the bus sizes of physical space area n (n = 2 to 6).

Bit 2n + 1: AnSZ1	Bit 2n: AnSZ0	Port A / B	Description
0	0	Unused	Reserved (Do not set)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Longword (32-bit) size
0	0	Used	Reserved (Do not set)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Reserved (Do not set)

10.2.3 Bus Control Register 3 (BCR3)

Bus control register 3 (BCR3) is a 16-bit read/write register that specifies RAS and CAS timing for the DRAM (areas 2 and 3). This enables a large amount of data to be transferred efficiently, for example, when transferring image data. The BCR3 is initialized to H'0000 by power-on resets, but is not initialized by manual resets or in the standby mode. The bits EXTEND, TPC31-30, RCD31-30, TRAS31-30, TPC21-20, RCD21-20, and TRAS21-20 are written to during the initialization after a power-on reset and are not modified again.

Bit:	15	14	13	12	11	10	9	8
Bit name:	EXT END	—	TPC31	TPC30	RCD31	RCD30	TRAS31	TRAS30
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:	7	6	5	4	3	2	1	0
Bit name:	—	—	TPC21	TPC20	RCD21	RCD20	TRAS21	TRAS20
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 15—BSCP DRAM/PSRAM Access Mode Extended Control (EXTEND): The EXTEND bit specifies the short pitch access mode to the DRAM (areas 2 and 3) for the BSCP (DMAC bus state controller). The EXTEND bit does not affect the BSC (CPU/cache bus controller). Setting the EXTEND bit to 1 enables a single access to be performed in a minimum of two cycles (RCDn(1-0)=00) to the DRAM, and a burst access to be performed in a pitch of one cycle for the second and later data. In addition, by setting the EXTEND bit to 1, other bits of the BCR3 register (TPC3(1-0), SCD3(1-0), TRAS3(1-0), TPC2 (1-0), RCD2 (1-0), and TRAS2 (1-0) become valid.

Bit 15: EXTEND	Description
0	The BSCP is set to normal mode. The BSCP setting is the same as the BSC. (Initial value)
1	The BSCP is set to extended mode. DRAM is accessed in a short pitch.

Bits 13 and 12—RAS Precharge Time (TPC31, TPC30): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the TPC (31-30) bits in place of the TPC (1-0) bits in the MCR. When the DRAM interface is selected as the connected memory of area 3, the TPC bits set the minimum number of cycles for RAS precharge until the next RAS assertion after RAS negation. Set the same value in BCR3. TPC3(1-0), TPC2(1-0), MCR.TPC(1-0), and DCR.TPC(1-0).

Bit 13	Bit 12	
TPC31	**TPC30**	**Function**
0	0	1 cycle (Initial value)
0	1	2 cycles
1	0	Reserved
1	1	Reserved

Bits 11 and 10—RAS-CAS Delay (RCD31, RCD30): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the RCD (31-30) bits in place of the RCD (1-0) bits in the MCR. When the DRAM interface is selected as the connected memory of area 3, these bits set the RAS-CAS assert delay.

Bit 11	Bit 10	
RCD31	**RCD30**	**Function**
0	0	1 cycle (Initial value)
0	1	2 cycles
1	0	3 cycles
1	1	4 cycles

Bits 9 and 8—CAS-Before-RAS Refresh RAS Assert Time (TRAS31, TRAS30): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the TRAS(31-30) bits in place of TRAS (1-0) bits in the MCR. When the DRAM interface is selected as the connected memory of area 3, these bits set the RAS assert period for CAS-before-RAS refreshes. Set the same value in BCR3.TRAS3(1-0), TRAS2(1-0), MCR.TRAS(1-0), and DCR. TRAS(1-0).

Bit 9	Bit 8	
TRAS31	**TRAS30**	**Function**
0	0	2 cycle (Initial value)
0	1	3 cycles
1	0	4 cycles
1	1	5 cycles

Bits 5 and 4—RAS Precharge Time (TPC21, TPC20): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the TPC (21-20) bits in place of the TPC (1-0) bits in the DCR. When the DRAM interface is selected as the connected memory of area 2, the TPC bits set the minimum number of cycles for RAS precharge until the next RAS assertion after RAS negation.

Bit 5	**Bit 4**	**Function**	
TPC21	**TPC20**	**Normal**	**After self refresh**
0	0	1 cycle (Initial value)	2 cycles (Initial value)
0	1	2 cycles	5 cycles
1	0	3 cycles	8 cycles
1	1	4 cycles	11 cycles

Bits 3 and 2—RAS-CAS Delay (RCD21, RCD20): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the RCD (21-20) bits in place of the RCD (1-0) bits in the DCR. The RAS-CAS delay is set for the DRAM interface connected to area 2.

Bit 3	**Bit 2**	
RCD21	**RCD20**	**Function**
0	0	1 cycle (Initial value)
0	1	2 cycles
1	0	3 cycles
1	1	4 cycles

Bits 1 and 0—CAS-Before-RAS Refresh RAS Assert Time (TRAS21, TRAS20): When the EXTEND bit in the BCR3 is set to 1, the BSCP uses the TRAS (21-20) bits in place of the TRAS (1-0) bits in the DCR. The RAS assert time is set for the DRAM interface connected to area 2 when CAS-before-RAS refreshes.

Bit 1	**Bit 0**	
TRAS21	**TRAS20**	**Function**
0	0	2 cycles (Initial value)
0	1	3 cycles
1	0	4 cycles
1	1	5 cycles

Bits 14, 7 and 6—Reserved: Always read 0. In addition, 0 is always written to these bits.

10.2.4 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register shared between the BSC and BSCP that specifies the number of idle (wait) state cycles inserted for each area. For some memories, the drive of the data bus may not be turned off quickly even when the read signal from the external device is turned off. This can result in conflicts between data buses when consecutive memory accesses are to different memories or when a write immediately follows a memory read. The SH-3 automatically inserts idle states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0
Initial value:	0	0	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0
Initial value:	1	1	1	1	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bits 15, 14, 3, 2 —Reserved: Bits 15, 14, 3 and 2 cannot be modified and always read as 0.

Bits 2n + 1, 2n—Area n (6–0) Intercycle Idle Specification (AnIW1, AnIW0): Specify the number of idles inserted between bus cycles when switching between physical space area n (6–0) to another space or between a read access to a write access in the same physical space.

Bit 2n + 1: AnIW1	Bit 2n: AnIW0	Description
0	0	1 idle cycle inserted
	1	1 idle cycle inserted
1	0	2 idle cycles inserted
	1	3 idle cycles inserted

10.2.5 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit read/write register shared between the BSC and BSCP that specifies the number of wait state cycles inserted for each area. It also specifies the pitch of data access for burst memory accesses. This allows direct connection of even low-speed memories without an external circuit. WCR2 is initialized to H'FFFF by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	A6 W2	A6 W1	A6 W0	A5 W2	A5 W1	A5 W0	A4 W2	A4 W1
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:	7	6	5	4	3	2	1	0
Bit name:	A4 W0	A3 W1	A3 W0	A2 W1	A2 W0	A0 W2	A0 W1	A0 W0
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

Bits 15–13—Area 6 Wait Control (A6W2, A6W1, A6W0): Specify the number of wait states inserted into physical space area 6. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 15: A6W2	Bit 14: A6W1	Bit 13: A6W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Disable	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable (Initial value)

Bits 12–10—Area 5 Wait Control (A5W2, A5W1, A5W0): Specify the number of wait states inserted into physical space area 5. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 12: A5W2	Bit 11: A5W1	Bit 10: A5W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Disable	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable (Initial value)

Bits 9–7—Area 4 Wait Control (A4W2, A4W1, A4W0): Specify the number of wait states inserted into physical space area 4.

			Description	
Bit 9: A4W2	Bit 8: A4W1	Bit 7: A4W0	Inserted Wait Status	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Ignored
		1	1	Enable
	1	0	2	Enable
		1	3	Enable
1	0	0	4	Enable
		1	6	Enable
	1	0	8	Enable
		1	10	Enable (Initial value)

Bits 6, 5—Area 3 Wait Control (A3W1, A3W0): Specify the number of wait states inserted into physical space area 3. When the DRAM is used for area 3 and the EXTEND bit of the BCR3 is set, the BSCP ignores bits 6 and 5.

- For Normal Memory

		Description	
Bit 6: A3W1	**Bit 5: A3W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enable
1	0	2	Enable
	1	3	Enable (Initial value)

- For DRAM, SDRAM

		Description	
Bit 6: A3W1	**Bit 5: A3W0**	**DRAM: CAS Assert Period**	**SDRAM: CAS Latency**
0	0	1	1
	1	1	1
1	0	2	2
	1	3	3 (Initial value)

Bits 4, 3—Area 2 Wait Control (A2W1, A2W0): Specify the number of wait states inserted into physical space area 2. When the DRAM is used for area 2 and the EXTEND bit of the BCR3 is set, the BSCP ignores bits 4 and 3.

- For Normal Memory

		Description	
Bit 4: A2W0	**Bit 3: A2W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enable
1	0	2	Enable
	1	3	Enable (Initial value)

- For DRAM, SDRAM

		Description	
Bit 4: A2W0	**Bit 3: A2W0**	**DRAM: CAS Assert Period**	**SDRAM: CAS Latency**
0	0	1*	1*
	1	1	1
1	0	2	2
	1	3	3 (Initial value)

Bits 2–0—Area 0 Wait Control (A0W2, A0W1, A0W0): Specify the number of wait states inserted into physical space area 0. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 2: A0W2	**Bit 1: A0W1**	**Bit 0: A0W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	0	Ignored	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10	Enable	10	Enable (Initial value)

10.2.6 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit read/write register shared between the BSP and BSCP that specifies RAS and CAS timing and burst control for DRAM(area 3 only), SDRAM(area 2 and 3), specifies address multiplexing, and controls refresh. This enables direct connection of DRAM, and SDRAM without external circuits.

The MCR is initialized to H'0000 by power-on resets, but is not initialized by manual resets or standby mode. The bits TPC1–TPC0, RCD1–RCD0, TRWL1–TRWL0, TRAS1–TRAS0, BE, AMX1–AMX0, and EDOMODE are written to at the initialization after a power-on reset and are not then modified again. When RFSH and RMODE are written to, write the same values to the other bits. When using DRAM, and SDRAM, do not access areas 2 and 3 until this register is initialized.

Bit:	15	14	13	12	11	10	9	8
Bit name:	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0
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:	7	6	5	4	3	2	1	0
Bit name:	—	BE	—	AMX1	AMX0	RFSH	RMODE	EDO MODE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—RAS Precharge Time (TPC1, TPC0): When DRAM interface is selected as connected memory, the TPC bits set the minimum number of cycles until the next RAS assertion after RAS negation. When SDRAM interface is selected, they set the minimum number of cycles until output of the next bank-active command after precharge. Set the same value in BCR3. TPC3(1-0), TPC2(1-0), MCR.TPC(1-0), and DCR.TPC(1-0).

Bit 15: TPC1	Bit 14: TPC0	Description
0	0	1 cycle
	1	2 cycles
1	0	Reserved
	1	Reserved

Bits 13 and 12—RAS–CAS Delay (RCD1, RCD0): The RCD bits set the RAS–CAS assert delay time for the connected memory when DRAM interface is selected. When SDRAM interface is selected, sets the bank active read/write command delay time. When the EXTEND bit of the BCR3 is set, the BSCP ignores the RCD bits.

Bit 13: RCD1	Bit 12: RCD0	Description
0	0	1 cycle (Initial value)
	1	2 cycles
1	0	3 cycles
	1	4 cycles

Bits 11 and 10—Write-Precharge Delay (TRWL1, TRWL0): The TRWL bits set the SDRAM write-precharge delay time. This designates the time between the end of a write cycle and the next bank-active command. This is valid only when SDRAM is connected. After the write cycle, the next bank-active command is not issued for the period Tpc + T_{RWL}.

Bit 11: TRWL1	Bit 10: TRWL0	Description
0	0	1 cycle (Initial value)
	1	2 cycles
1	0	3 cycles
	1	Reserved

Bits 9 and 8—CAS-Before-RAS Refresh RAS Assert Time (TRAS1, TRAS0): When DRAM interface is selected as connected memory, the TRAS bits set the RAS assertion period for CAS-before-RAS refreshes. When SDRAM interface is selected, no bank-active command is issues during the period TPC + TRAS after an auto-refresh command. Set the same value in BCR3. TRAS3(1-0), TRAS2(1-0), MCR.TRAS(1-0), and DCR.TRAS(1-0).

Bit 9: TRAS1	Bit 8: TRAS0	Description
0	0	2 cycles (Initial value)
	1	3 cycles
1	0	4 cycles
	1	5 cycles

Bit 7—Reserved: Bit 7 cannot be modified. It always reads 0.

Bit 6—Burst Enable (BE): The BE bit specifies whether to conduct a burst access of DRAM. When accessing SDRAM, burst access is always carried out, regardless of this bit's designation.

Bit 6: BE	Description
0	Burst disabled (Initial value)
1	When DRAM interface, do a high-speed page mode access

Bit 5—Reserved: Bit 5 cannot be modified.

Bits 4 and 3—Address Multiplex (AMX1, AMX0): The AMX bits specify address multiplexing for DRAM and SDRAM. The actual address shift value differs between DRAM interface and SDRAM interface.

For DRAM Interface:

Bit 4: AMX1	Bit 3: AMX0	Description
0	0	The row address begins with A9. (The A9 value is output at A1 when the row address is output.) (Initial value)
	1	The row address begins with A10. (The A10 value is output at A1 when the row address is output.)
1	0	The row address begins with A11. (The A11 value is output at A1 when the row address is output.)
	1	The row address begins with A12. (The A12 value is output at A1 when the row address is output.)

For SDRAM Interface:

Bit 4: AMX1	Bit 3: AMX0	Description
0	0	The row address begins with A9. (The A9 value is output at A1 when the row address is output. 1M × 16-bit products) (Initial value)
	1	The row address begins with A10. (The A10 value is output at A1 when the row address is output. 2M × 8-bit products)
1	0	The row address begins with A11. (The A11 value is output at A1 when the row address is output. 4M × 4-bit products)
	1	The row address begins with A9. (The A9 value is output at A1 when the row address is output. 256K × 16-bit products)

Bit 2—Refresh Control (RFSH): The RFSH bit determines whether or not the refresh operation of the DRAM and SDRAM is performed. The timer for generation of the refresh request frequency can also be used as an internal timer.

Bit 2: RFSH	Description
0	No refresh (Initial value)
1	Refresh

Bit 1—Refresh Mode (RMODE): The RMODE bit selects whether to perform an ordinary refresh or a self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 0, a CAS-before-RAS refresh or an auto-refresh is performed on DRAM or SDRAM at the period set by the refresh-related registers RTCNT, RTCOR and RTCSR. When a refresh request occurs during an external bus cycle, the bus cycle will be ended and the refresh cycle performed. When the RFSH bit is 1 and this bit is also 1, the DRAM or SDRAM will wait for the end of any executing external bus cycle before going into a self-refresh. All refresh requests to memory that is in the self-refresh state are ignored.

Bit 1: RMODE	Description
0	CAS-before-RAS refresh (RFSH must be 1) (Initial value)
1	Self-refresh (RFSH must be 1)

Bit 0—EDO Mode (EDOMODE): The EDOMODE bit specifies the data sampling timing during data reads when using DRAM in EDO mode. Operating timing of memory other than DRAM does not change even if this bit is set. This bit is only valid for DRAM connected to area 3. Do not set this bit to 1 when using SDRAM.

Bit 0: EDOMODE	Description
0	Set when using normal DRAM. Data sampling timing during read cycle is on the falling edge of BCLK. (Initial value)
1	Set when using hyper page mode DRAM. Data sampling timing during read cycle is on the rising edge of BCLK. Also, RAS signal negate timing is delayed 1/2 machine cycle.

10.2.7 DRAM Control Register (DCR)

The DRAM area control register (DCR) is a 16-bit read/write register shared by the BSC and the BSCP that specifies RAS and CAS timing and burst control for DRAM connected to area 2. It also specifies address multiplexing and controls refresh. When DRAM is connected to area 2, the bus width is fixed at 16 bits. The DCR is initialized to H'0000 by power-on resets, but is not initialized by manual resets or standby mode. Do not access external memory outside area 2 until this register's initialization is complete.

Bit:	15	14	13	12	11	10	9	8
Bit name:	TPC1	TPC0	RCD1	RCD0	—	—	TRAS1	TRAS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	BE	—	AMX1	AMX0	RFSH	RMODE	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R

Bits 15 and 14—RAS Precharge Time (TPC1, TPC0): The TPC bits set the RAS precharge time for the DRAM connected to area 2. Set the same value in BCR3.TPC3(1-0), TPC2(1-0), MCR.TPC(1-0), and DCR.TPC(1-0).

Bit 15: TPC1	Bit 14: TPC0	Description
0	0	1 cycle
	1	2 cycles
1	0	Reserved
	1	Reserved

Bits 13 and 12—RAS–CAS Delay (RCD1, RCD0): The RCD bits set the RAS–CAS delay time for the DRAM connected to area 2. When the BCR3 EXTEND bit is set, the BSCP ignores bits 13 and 12.

Bit 13: RCD1	Bit 12: RCD0	Description
0	0	1 cycle (Initial value)
	1	2 cycles
1	0	3 cycles
	1	4 cycles

Bits 9 and 8—CAS-Before-RAS Refresh RAS Assert Time (TRAS1, TRAS0): The TRAS bits set timing for the DRAM connected to area 2. These bits set the RAS assert period for CAS-before-RAS refreshes. Set the same value in BCR3.TRAS3(1-0), TRAS2(1-0), MCR.TRAS(1-0), and DCR.TRAS(1-0).

Bit 9: TRAS1	Bit 8: TRAS0	Description
0	0	2 cycles (Initial value)
	1	3 cycles
1	0	4 cycles
	1	5 cycles

Bit 6—Burst Enable (BE): The BE bit specifies whether to conduct a burst access of the DRAM connected to area 2.

Bit 6: BE	Description
0	Burst disabled (Initial value)
1	Do a high-speed page mode access

Bits 4 and 3—Address Multiplex (AMX1, AMX0): The AMX bits specify address multiplexing for the DRAM connected to area 2.

Bit 4: AMX1	Bit 3: AMX0	Description
0	0	The row address begins with A9. (The A9 value is output at A1 when the row address is output.) (Initial value)
	1	The row address begins with A10. (The A10 value is output at A1 when the row address is output.)
1	0	The row address begins with A11. (The A11 value is output at A1 when the row address is output.)
	1	The row address begins with A12. (The A12 value is output at A1 when the row address is output.)

Bit 2—Refresh Control (RFSH): The RFSH bit determines whether or not the refresh operation of the DRAM connected to area 2 is performed.

Bit 2: RFSH	Description
0	No refresh (Initial value)
1	Refresh

Bit 1—Refresh Mode (RMODE): The RMODE bit selects the refresh mode for the DRAM connected to area 2.

Bit 1: RMODE	Description
0	CAS-before-RAS refresh (RFSH must be 1) (Initial value)
1	Self-refresh (RFSH must be 1)

Bits 11, 10, 7, 5, and 0—Reserved: Bits 11, 10, 7, 5, and 0 cannot be modified. They always read 0.

10.2.8 PCMCIA Control Register (PCR)

The PCMCIA control register (PCR) is a 16-bit read/write register shared by the BSC and the BSCP that specifies the OE and WE signal assert/negate timing for PCMCIA interfaces connected to areas 5, 6. The OE, WE signal assert pulse widths are designated by the WCR2 wait control bits. This register is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH1	A6TEH0
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

Bits 15–8—Reserved: Bits 15–8 cannot be modified. They always read 0.

Bits 7 and 6—Area 5 Address OE/WE Assert Delay (A5TED1, A5TED0): The A5TED bits specify the address to OE/WE assert delay time for the PCMCIA interface connected to area 5.

Bit 7: A5TED1	Bit 6: A5TED0	Description
0	0	0.5 cycle delay (Initial value)
	1	1.5 cycle delay
1	0	2.5 cycle delay
	1	3.5 cycle delay

Bits 5 and 4—Area 6 Address OE/WE Assert Delay (A6TED1, A6TED0): The A6TED bits specify the address to OE/WE assert delay time for the PCMCIA interface connected to area 6.

Bit 5: A6TED1	Bit 4: A6TED0	Description
0	0	0.5 cycle delay (Initial value)
	1	1.5 cycle delay
1	0	2.5 cycle delay
	1	3.5 cycle delay

Bits 3 and 2—Area 5 OE/WE Negate Address Delay (A5TEH1, A5TEH0): The A5TEH bits specify the OE/WE negate address delay time for the PCMCIA interface connected to area 5.

Bit 3: A5TEH1	Bit 2: A5TEH0	Description
0	0	0.5 cycle delay (Initial value)
	1	1.5 cycle delay
1	0	2.5 cycle delay
	1	3.5 cycle delay

Bits 1 and 0—Area 6 OE/WE Negate Address Delay (A6TEH1, A6TEH0): The A6TEH bits specify the OE/WE negate address delay time for the PCMCIA interface connected to area 6.

Bit 1: A6TEH1	Bit 0: A6TEH0	Description
0	0	0.5 cycle delay (Initial value)
	1	1.5 cycle delay
1	0	2.5 cycle delay
	1	3.5 cycle delay

10.2.9 SDRAM Mode Register (SDMR)

The SDRAM mode register (SDMR) is written to via the SDRAM address bus and is an 8-bit write-only register. It sets SDRAM mode for areas 2 and 3. SDMR is undefined after a power-on reset. The register contents are not initialized by a manual reset or standby mode; values remain unchanged.

Writes to the SDRAM mode register use the address bus rather than the data bus. If the value to be set is X and the SDMR address is Y, the value X is written in the SDRAM mode register by writing in address X + Y. Since A0 of the synchronous DRAM is connected to A2 of the chip and A1 of the Synchronous DRAM is connected to A3 of the chip, the value actually written the the synchronous DRAM is the X value shifted two bits right. For example, when H'0230 is written to the SDMR register of area 2, random data is written to the address H'FFFFD000 (address Y) + H'08C0 (value X), or H'FFFFD8C0. As a result, H'0230 is written to the SDMR register. When H'0230 is written to the SDMR register of area 3, random data is written to the address H'FFFFE000 (address Y) + H'08C0 (value X), or H'FFFFE8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H'0000 to H'0FFC.

Bit:	31		12	11	10	9	8
Bit name:	SDMR address			—	—	—	—
Initial value:	—	....	—	—	—	—	—
R/W:	—	....	—	W*	W*	W	W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	—	—

Note: Depending on the type of SDRAM.

10.2.10 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTSCR) is a 16-bit read/write register that specifies the refresh cycle, whether to generate an interrupt, and that interrupt's cycle. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
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

Bits 15–8—Reserved: Bits 15–8 cannot be modified. They always read 0.

Bit 7—Compare Match Flag (CMF): The CMF status flag indicates that the values of RTCNT and RTCOR match.

Bit 7: CMF	Description
0	The values of RTCNT and RTCOR do not match. Clear condition: When a refresh is performed After 0 has been written in CMF and RFSH = 1 and RMODE = 0 (to perform a CBR refresh). (Initial value)
1	The values of RTCNT and RTCOR match. Set condition: RTCNT = RTCOR *

Note : Contents do not change when 1 is written to CMF.

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt request caused when the CMF of RTCSR is set to 1. Do not set this bit to 1 when using CAS-before-RAS refreshing or auto-refreshing.

Bit 6: CMIE	Description
0	Disables an interrupt request caused by CMF (Initial value)
1	Enables an interrupt request caused by CMF

Bits 5–3—Clock Select Bits (CKS2–CKS0): Select the clock input to RTCNT. The source clock is the external bus clock (BCLK). The RTCNT count clock is CKIO divided by the specified ratio. The specified ratios are shown below in the normal external bus clock and when setting the external bus clock to 1/4 by setting the SLPFRQ bit in FRQCR to 1 in sleep mode.

			Description	
			Normal external bus clock	
Bit 5: CKS2	**Bit 4: CKS1**	**Bit 3: CKS0**		**1/4-bus clock**
0	0	0	Disables clock input	Disables clock input (Initial value)
		1	Bus clock (CKIO)/4	CKIO/1
	1	0	CKIO/16	CKIO/4
		1	CKIO/64	CKIO/16
1	0	0	CKIO/256	CKIO/64
		1	CKIO/1024	CKIO/256
	1	0	CKIO/2048	CKIO/512
		1	CKIO/4096	CKIO/1024

Bit 2—Refresh Count Overflow Flag (OVF): The OVF status flag indicates when the number of refresh requests indicated in the refresh count register (RFCR) exceeds the limit set in the LMTS bit of RTCSR.

Bit 2: OVF	**Description**
0	RFCR has not exceeded the count limit value set in LMTS Clear Conditions: When 0 is written to OVF (Initial value)
1	RFCR has exceeded the count limit value set in LMTS Set Conditions: When the RFCR value has exceeded the count limit value set in LMTS*

Note: Contents don’t change when 1 is written to OVF.

Bit 1—Refresh Count Overflow Interrupt Enable (OVIE): OVIE selects whether to suppress generation of interrupt requests by OVF when the OVF bit of RTCSR is set to 1.

Bit 1: OVIE	**Description**
0	Disables interrupt requests from the OVF (Initial value)
1	Enables interrupt requests from the OVF

Bit 0—Refresh Count Overflow Limit Select (LMTS): Indicates the count limit value to be compared to the number of refreshes indicated in the refresh count register (RFCR). When the value RFCR overflows the value specified by LMTS, the OVF flag is set.

Bit 0: LMTS	**Description**
0	Count limit value is 1024 (Initial value)
1	Count limit value is 512

10.2.11 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit read/write register. RTCNT is an 8-bit counter that counts up with input clocks. The clock select bits (CKS2–CKS0) of RTCSR select the input clock. When RTCNT matches RTCOR, the CMF bit of RTCSR is set and RTCNT is cleared. RTCNT is initialized to H'00 by a power-on reset; it continues incrementing after a manual reset; it is not initialized by standby mode and holds its values unchanged.

Bit:	15	14	13	12	11	10	9	8
Bit name:								
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

10.2.12 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit read/write register. The values of RTCOR and RTCNT (bottom 8 bits) are constantly compared. When the values match, the compare match flag (CMF) of RTCSR is set and RTCNT is cleared to 0. When the refresh bit (RFSH) of the individual memory control register (MCR) is set to 1 and the refresh mode is set to CAS-before-RAS refresh, a memory refresh cycle occurs when the CMF bit is set. RTCOR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset or standby mode, but holds its contents.

Bit:	15	14	13	12	11	10	9	8
Bit name:								
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

10.2.13 Refresh Count Register (RFCR)

The refresh count register (RFCR) is a 16-bit read/write register. It is a 10-bit counter that increments every time RTCOR and RTCNT match. When RFCR exceeds the count limit value set in the LMTS of RTCSR, RTCSR’s OVF bit is set and RFCR clears. RFCR is initialized to H'0000 when a power-on reset is performed. It is not initialized by a manual reset or standby mode, but holds its contents.

Bit:	15	14	13	12	11	10	9	8
Bit name:								
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

10.2.14 Cautions on Accessing Refresh Control Related Registers

RFCR, RTCSR, RTCNT, and RTCOR require that a specific code be appended to the data when it is written to prevent data from being mistakenly overwritten by program overruns or other write operations (figure 10.5). Perform reads and writes using the following methods:

1. When writing to RFCR, RTCSR, RTCNT, and RTCOR, use only word transfer instructions. You cannot write with byte transfer instructions.

   When writing to RTCNT, RTCSR, or RTCOR, place B'10100101 in the upper byte and the write data in the lower byte. When writing to RFCR, place B'101001 in the top 6 bits and the write data in the remaining bits, as shown in figure 10.5.
2. When reading from RFCR, RTCSR, RTCNT, and RTCOR, carry out reads with 16-bit width. 0 is read out from undefined bit sections.

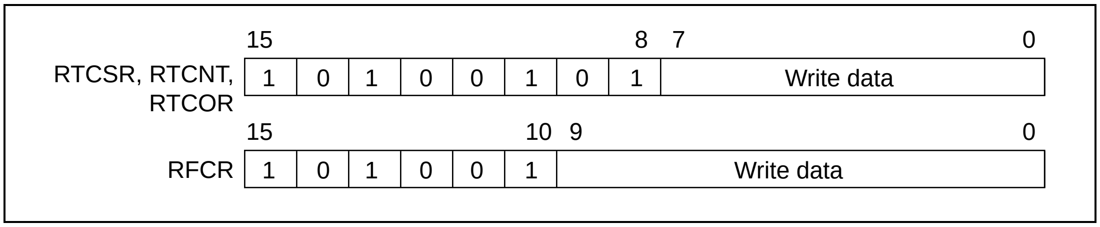


Figure 10.5 Writing to RFCR, RTCSR, RTCNT, and RTCOR

10.3 BSC Operation

10.3.1 Endian/Access Size and Data Alignment

The SH-3 supports both big endian, in which the 0 address is the most significant byte in the byte data, and little endian, in which the 0 address is the least significant byte. This switch over is designated by an external pin (MD5 pin) at the time of a power-on reset. After a power-on reset, big endian is engaged when MD5 is low; little endian is engaged when MD5 is high.

Three data bus widths are available for ordinary memory (byte, word, longword) and two data bus widths (word and longword) for DRAM. Only longword is available for SDRAM. For the PCMCIA interface, choose from byte and word. This means data alignment is done by matching the device's data width and endian. The access unit must also be matched to the device's bus width. This also means that when longword data is read from a byte-width device, the read operation must happen 4 times. In the SH-3, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 10.7 through 10.12 show the relationship between endian, device data width, and access unit.

Table 10.7 32-Bit External Device/Big Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31–D24	D23–D16	D15–D8	D7–D0	Operation
Assert	—	—	—	Data 7–0	—	—	—	Byte access at 0
—	Assert	—	—	—	Data 7–0	—	—	Byte access at 1
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 2
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 3
Assert	Assert	—	—	Data 15–8	Data 7–0	—	—	Word access at 0
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 2
Assert	Assert	Assert	Assert	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Longword access at 0

Table 10.8 16-Bit External Device/Big Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 0	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 1	
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 2	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 3	
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 0	
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 2	
—	—	Assert	Assert	—	—	Data 31–24	Data 23–16	Longword access at 0	1st time at 0
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0		2nd time at 2

Table 10.9 8-Bit External Device/Big Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 0	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 1	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 2	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 3	
—	—	—	Assert	—	—	—	Data 15–8	Word access at 0	1st time at 0
—	—	—	Assert	—	—	—	Data 7–0		2nd time at 1
—	—	—	Assert	—	—	—	Data 15–8	Word access at 2	1st time at 2
—	—	—	Assert	—	—	—	Data 7–0		2nd time at 3
—	—	—	Assert	—	—	—	Data 31–24	Longword access at 0	1st time at 0
—	—	—	Assert	—	—	—	Data 23–16		2nd time at 1
—	—	—	Assert	—	—	—	Data 15–8		3rd time at 2
—	—	—	Assert	—	—	—	Data 7–0		4th time at 3

Table 10.10 32-Bit External Device/Little Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31–D24	D23–D16	D15–D8	D7–D0	Operation
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 0
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 1
—	Assert	—	—	—	Data 7–0	—	—	Byte access at 2
Assert	—	—	—	Data 7–0	—	—	—	Byte access at 3
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 0
Assert	Assert	—	—	Data 15–8	Data 7–0	—	—	Word access at 2
Assert	Assert	Assert	Assert	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Longword access at 0

Table 10.11 16-Bit External Device/Little Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 0	
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 1	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 2	
—	—	Assert	—	—	—	Data 7–0	—	Byte access at 3	
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 0	
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Word access at 2	
—	—	Assert	Assert	—	—	Data 15–8	Data 7–0	Longword access at 0	1st time at 0
—	—	Assert	Assert	—	—	Data 31–24	Data 23–16		2nd time at 2

Table 10.12 8-Bit External Device/Little Endian Access and Data Alignment

WE3, CASHH, DQMUU	WE2, CASHL, DQMUL	WE1, CASLH, CAS2H, DQMLU	WE0, CASLL, CAS2L, DQMLL	D31– D24	D23– D16	D15– D8	D7–D0	Operation	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 0	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 1	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 2	
—	—	—	Assert	—	—	—	Data 7–0	Byte access at 3	
—	—	—	Assert	—	—	—	Data 7–0	Word access at 0	1st time at 0
—	—	—	Assert	—	—	—	Data 15–8		2nd time at 1
—	—	—	Assert	—	—	—	Data 7–0	Word access at 2	1st time at 2
—	—	—	Assert	—	—	—	Data 15–8		2nd time at 3
—	—	—	Assert	—	—	—	Data 7–0	Longword access at 0	1st time at 0
—	—	—	Assert	—	—	—	Data 15–8		2nd time at 1
—	—	—	Assert	—	—	—	Data 23–16		3rd time at 2
—	—	—	Assert	—	—	—	Data 31–24		4th time at 3

10.3.2 Description of Areas

Area 0: Area 0 physical addresses A28–A26 are 000. Addresses A31–A29 are ignored and the address range is H'00000000 + H'20000000 × n – H'03FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Ordinary memories such as SRAM, ROM, and burst ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using external pins. When the Area 0 space is accessed, a CS0 signal is asserted. An RD signal that can be used as OE and the WE0–WE3 signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A0W2–A0W0 bits of WCR2. When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits.

Area 1: Area 1 physical addresses A28–A26 are 001. Addresses A31–A29 are ignored and the address range is H'04000000 + H'20000000 × n – H'07FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Area 1 is the area specifically for the internal peripheral modules. The external memories cannot be connected.

Control registers of peripheral modules shown below are mapped to this area 1. Their addresses are physical address, to which logical addresses can be mapped with the MMU enabled.
⊕\DMAC, PORT, IrDA, SCIF, ADC, DAC, INTC(except INTEVT, IPRA, IPRB)
Those registers must not be cached.

Area 2: Area 2 physical addresses A28–A26 are 010. Addresses A31–A29 are ignored and the address range is H'08000000 + H'20000000 × n – H'0BFFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Ordinary memories like SRAM and ROM, as well as DRAM and SDRAM, can be connected to this space. Byte, word, or longword can be selected as the bus width using the A2SZ1–A2SZ0 bits of BCR2 for ordinary memory. For SDRAM, set longword using the SZ bit of MCR. When DRAM is connected to Area 2, the bus width is fixed at 16 bits.

When the area 2 space is accessed, a CS2 signal is asserted. When ordinary memories are connected, an RD signal that can be used as OE and the WE0–WE3 signals for write control are also asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A12W1 to A12W0 bits of WCR2.

When SDRAM is connected, the RAS signal, CAS signal, RD/WR signal, and byte controls DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed. Control of RAS, CAS, data timing, and address multiplexing is set with MCR.

When DRAM is connected, the RAS2U, RAS2L signal, CAS2H signal, CAS2L signal, and RD/WR signal are all asserted and addresses multiplexed. Control of RAS2, CAS, data timing, and address multiplexing is set with DCR.

Area 3: Area 3 physical addresses A28–A26 are 011. Addresses A31–A29 are ignored and the address range is H'0C000000 + H'20000000 × n – H'0FFFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Ordinary memories like SRAM and ROM, as well as DRAM, and SDRAM, can be connected to this space. Byte, word or longword can be selected as the bus width using the A3SZ1–A3SZ0 bits of BCR2 for ordinary memory. For DRAM, word or longword can be selected using the SZ bit of MCR. When SDRAM is connected, set to longword using the MCR register’s SZ bit.

When area 3 space is accessed, CS3 is asserted.

When ordinary memories are connected, an RD signal that can be used as OE and the WE0–WE3 signals for write control are asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A3W1 to A3W0 bits of WCR2.

When SDRAM is connected, the RAS3U, RAS3L signal, CAS signal, RD/WR signal, and byte controls DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed. When DRAM is connected, the RAS3U, RAS3L signal, CASHH signal, CASHL signal, CASLH signal, CASLL signal, and RD/WR signal are all asserted and addresses multiplexed. For all of these, control of RAS3 , CAS , and data timing and of address multiplexing is set with MCR.

Area 4: Area 4 physical addresses A28–A26 are 100. Addresses A31–A29 are ignored and the address range is H'10000000 + H'20000000 × n – H'13FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Only ordinary memories like SRAM and ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using the A4SZ1–A4SZ0 bits of BCR2. When the area 4 space is accessed, a CS4 signal is asserted. An RD signal that can be used as OE and the WE0–WE3 signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A4W2–A4W0 bits of WCR2.

Area 5: Area 5 physical addresses A28–A26 are 101. Addresses A31–A29 are ignored and the address range is the 64 Mbytes at H'14000000 + H'20000000 × n – H'17FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. PCMCIA interfaces only use their IC memory card interface, so the address range becomes the 32 Mbytes at H'14000000 + H'20000000 × n – H'15FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A5SZ1–A5SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A5SZ1–A5SZ0 bits of BCR2.

When the area 5 space is accessed and ordinary memory is connected, a CS5 signal is asserted. An RD signal that can be used as OE and the WE0–WE3 signals for write control are also asserted. When the PCMCIA interface is used, the CE1 signal, CE2 signal, OE signal, and WE signal are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A5W2–A5W0 bits of WCR2. When a burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits. The setup and hold times of address/CS5 for the read/write strobe signals can be set in the range 0.5–3.5 using A5TED1–A5TED0 and A5TEH1 to A5TEH0 bits of the PCR register.

Area 6: Area 6 physical addresses A28–A26 are 110. Addresses A31–A29 are ignored and the address range is the 64 Mbytes at H'18000000 + H'20000000 × n – H'1BFFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. When the PCMCIA interface is used, the IC memory card interface address range is the 32 Mbytes at H'18000000 + H'20000000 × n – H'19FFFFFF + H'20000000 × n and the I/O card interface address range is the 32 Mbytes at H'1A000000 + H'20000000 × n – H'1BFFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A6SZ1–A6SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A6SZ1–A6SZ0 bits of BCR2.

When the area 6 space is accessed and ordinary memory is connected, a CS6 signal is asserted. An RD signal that can be used as OE and the WE0–WE3 signals for write control are also asserted. When the PCMCIA interface is used, the CE1 signal, CE2 signal, OE signal, and WE , IORD, and IOWR signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A6W2–A6W0 bits of WCR2. The bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits.

10.3.3 Basic Interface

Basic Timing: The basic interface of the SH7709 uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. Figure 10.6 shows the basic timing of normal space accesses. A no-wait normal access is completed in two cycles. The BS signal is asserted for one cycle to indicate the start of a bus cycle. The CSn signal is negated on the T2 clock falling edge to secure the negation period. Therefore, in case of access at minimum pitch, there is a half-cycle negation period.

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in the case of a 32-bit device, and 16 bits in the case of a 16-bit device. When writing, only the WE signal for the byte to be written is asserted. For details, see section 10.3.1, Endian/Access Size and Data Alignment.

Read/write for cache fill or write-back follows the set bus width and transfers a total of 16 bytes continuously. The bus is not released during this transfer. For cache misses that occur during byte or word operand accesses or branching to odd word boundaries, the fill is always performed by longword accesses on the chip-external interface. Write-through area write access and noncacheable read/write access is based on the actual address size.

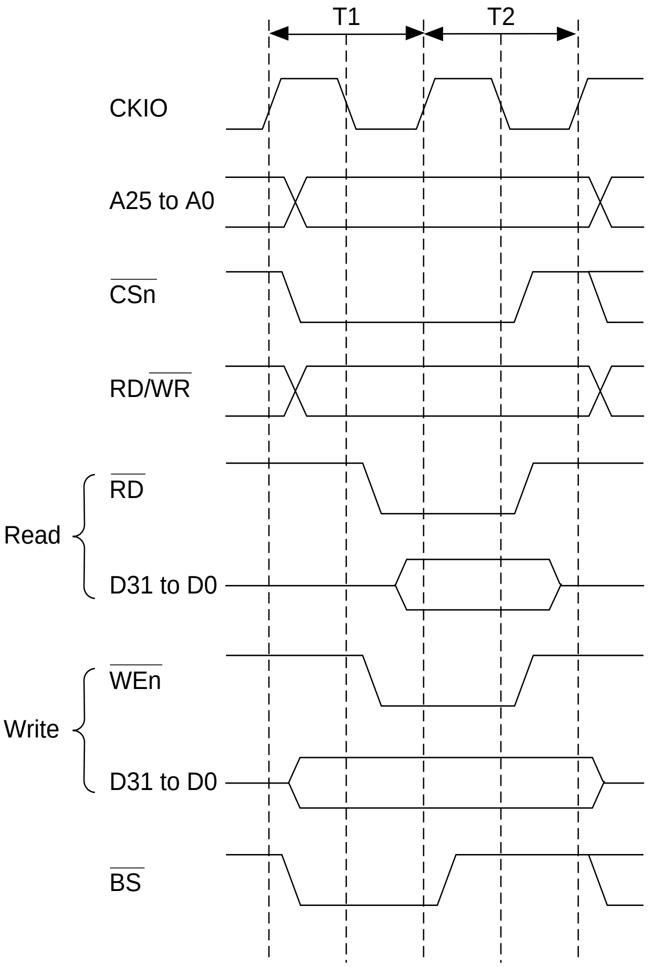


Figure 10.6 Basic Timing of Basic Interface

Figures 10.7, 10,8, and 10.9 show examples of connection to 32, 16, and 8-bit data width static RAM.

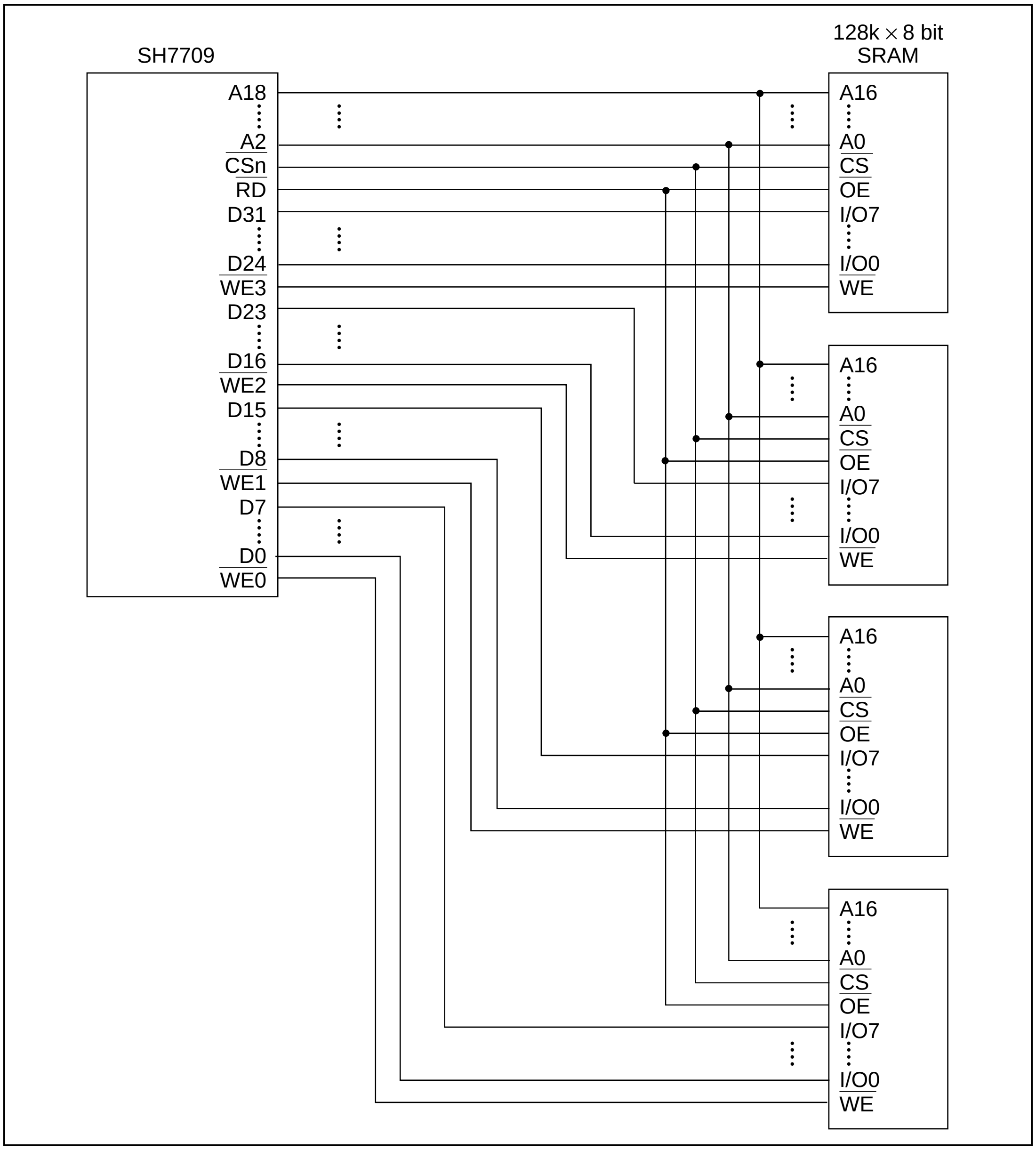


Figure 10.7 Example of 32-Bit Data Width Static RAM Connection

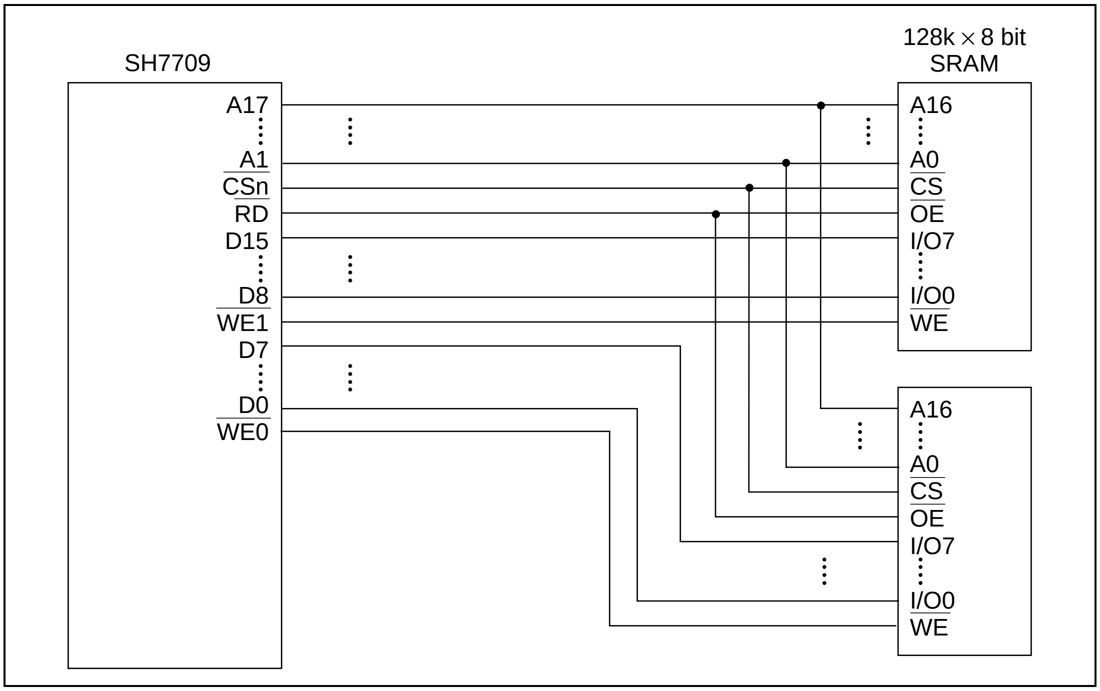


Figure 10.8 Example of 16-Bit Data Width Static RAM Connection

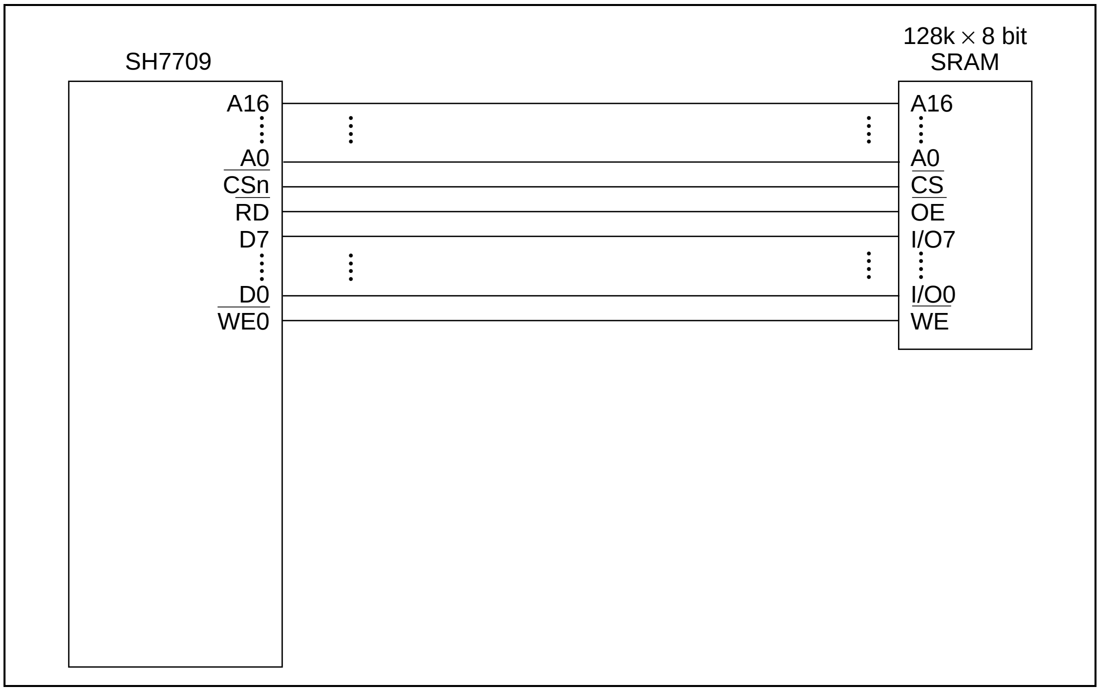


Figure 10.9 Example of 8-Bit Data Width Static RAM Connection

Wait State Control: Wait state insertion on the basic interface can be controlled by the WCR2 settings. If the WCR2 wait specification bits corresponding to a particular area are not zero, a software wait is inserted in accordance with that specification. For details, see section 10.2.4, Wait Control Register 2 (WCR2).

The specified number of Tw cycles are inserted as wait cycles using the basic interface wait timing shown in figure 10.10.

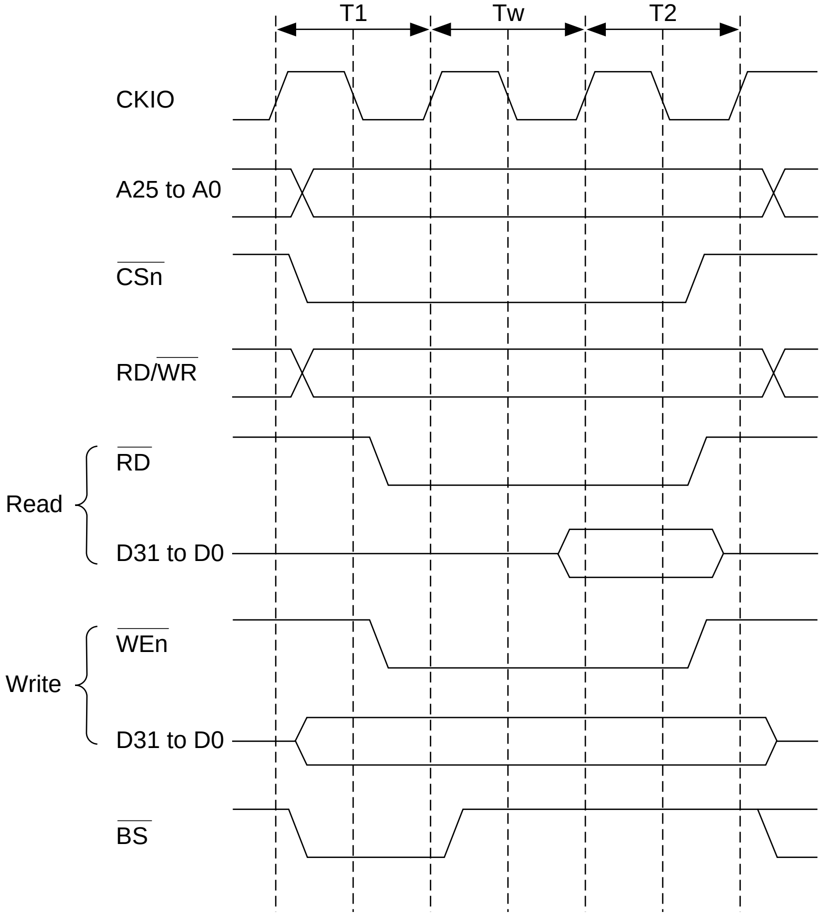


Figure 10.10 Basic Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by WCR2, the external wait input WAIT signal is also sampled. WAIT pin sampling is shown in figure 10.11. A 2-cycle wait is specified as a software wait. Sampling is performed at the transition from the Tw state to the T2 state; therefore, if the WAIT signal has no effect if asserted in the T1 cycle or the first Tw cycle. The WAIT signal is sampled on the rising edge of the clock.

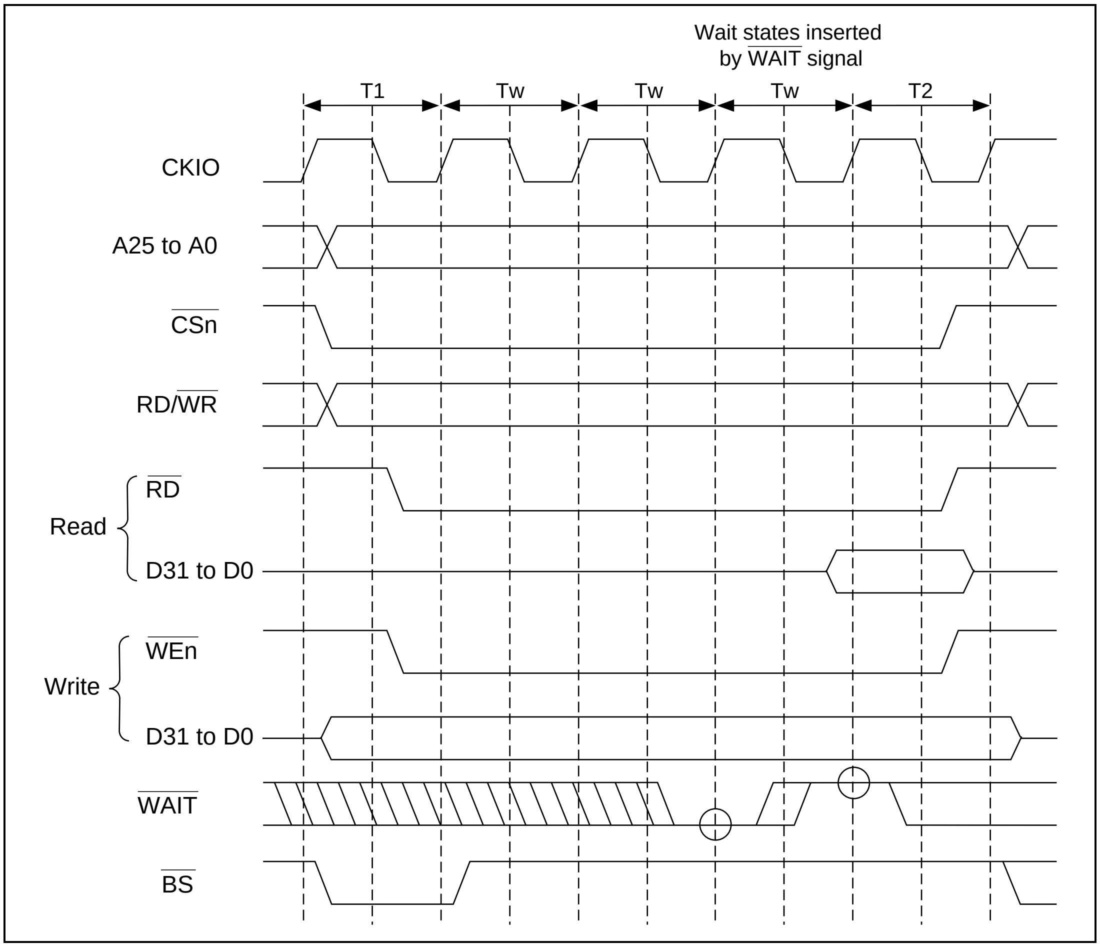


Figure 10.11 Basic Interface Wait State Timing (Wait State Insertion by WAIT Signal)

10.3.4 DRAM Interface

DRAM Connection Method: When the memory type bits (DRAMTP2-0) in BCR1 are set to 100, area 3 becomes DRAM space; when set to 101, area 2 and area 3 become DRAM space. The DRAM interface function can then be used to connect the SH7709 directly to DRAM.

16 or 32 bits can be selected as the interface data width for area 3 when bits DRAMTP2-0 are set to 100, and 16 bits can be used for area 2 and 16 or 32 bits can be used for area 3 when bits DRAMTP2-0 are set to 101.

2-CAS 16-bit DRAMs can be connected, since CAS is used to control byte access.

Signals used for connection when DRAM is connected to area 3 are RAS3, CASHH, CASHL, CASLH, CASLL, and RD/WR. CASHH and CASHL are not used when the data width is 16 bits. When DRAM is connected to areas 2 and 3, the signals for area 2 DRAM connection are RAS2L, RAS2U, CAS2H, CAS2L, and RD/WR, and those for area 3 DRAM connection are RAS3L, RAS3U, CASHH, CASHL, CASLH, CASLL, and RD/WR.

In addition to normal read and write access modes, high-speed page mode is supported for burst access. Also, for DRAM connected to area 3, EDO mode, which enables the DRAM access time to be increased by delaying the data sampling timing by 1/2 clock when reading, is supported in addition to normal read and write access for burst mode.

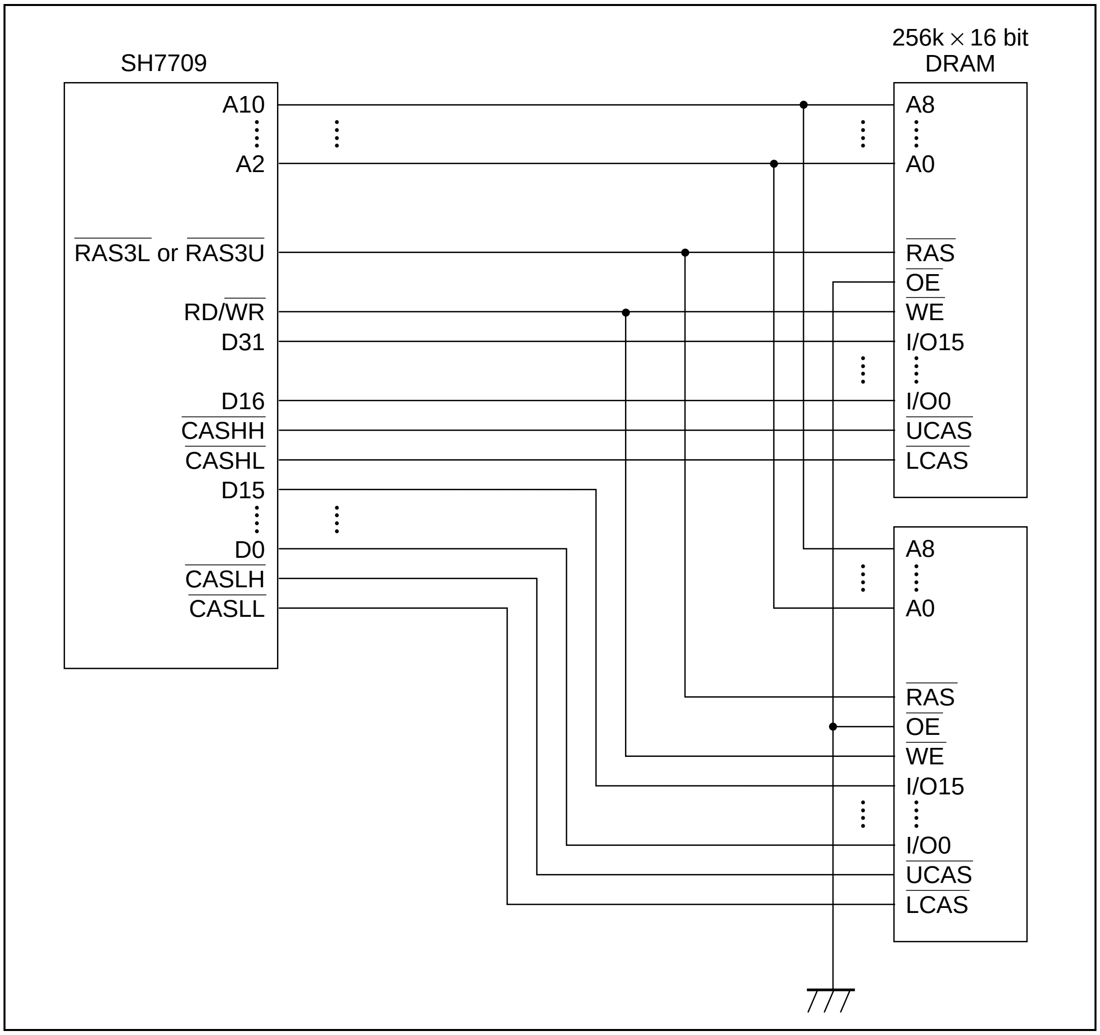


Figure 10.12 Example of DRAM Connection (32-Bit Data Width)

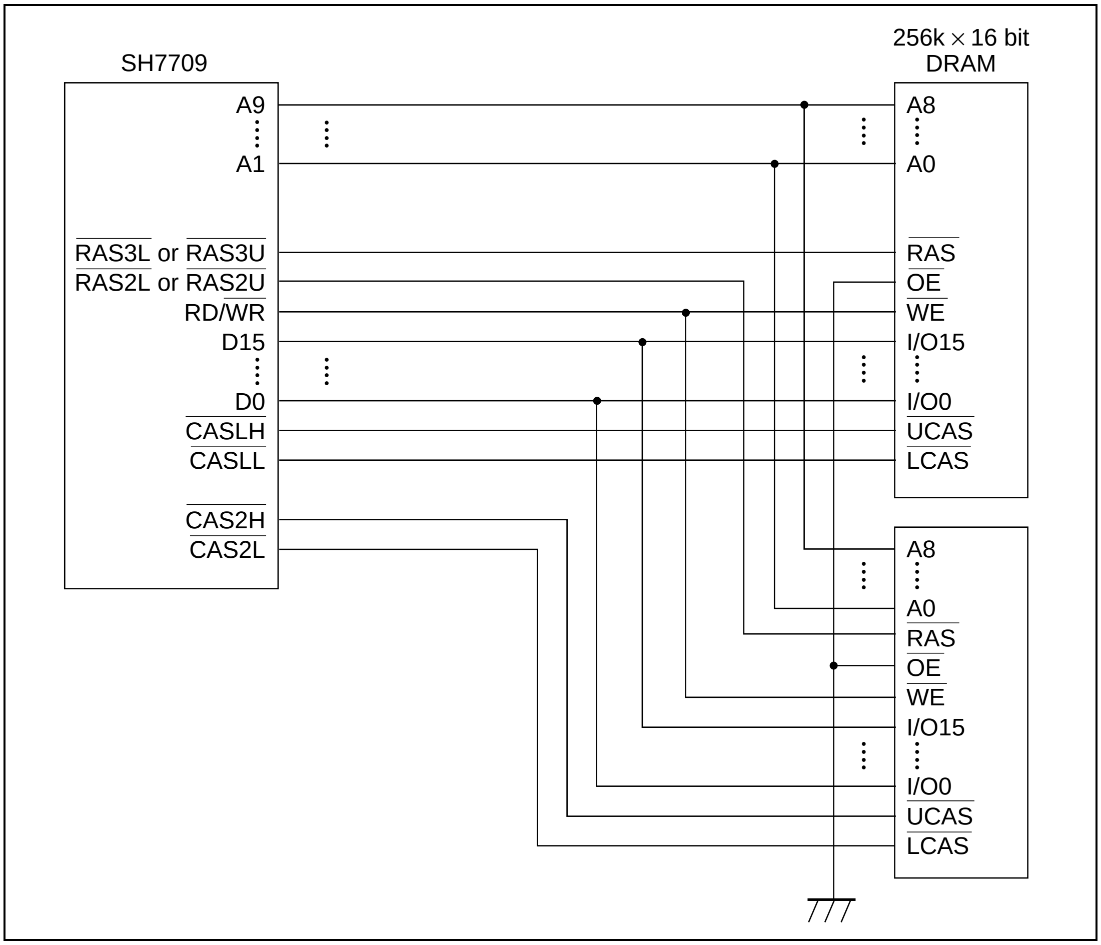


Figure 10.13 Example of DRAM Connection (16-Bit Data Width)

Address Multiplexing: When area 2 and area 3 are designated as DRAM space, address multiplexing is always performed in accesses to DRAM. This enables DRAM, which requires row and column address multiplexing, to be connected directly to the SH7709 without using an external address multiplexer circuit. Any of the four multiplexing methods shown below can be selected, by setting bits AMX1-0 in MCR for area 3 DRAM, or bits AMX1-0 in DCR for area 2 DRAM. The relationship between bits AMX1-0 and address multiplexing is shown in figure 10.12. The address output pins subject to address multiplexing are A15 to A1. Pins A25 to A16 carry the original address.

Table 10.13 Relationship between AMX1-0 and Address Multiplexing

Setting		Number of Column		External Address Pins	
AMX1	**AMX0**	**Address Bits**	**Output Timing**	**A1 to A14**	**A15**
0	0	8 bits	Column address	A1 to A14	A15
			Row address	A9 to A22	A23
0	1	9 bits	Column address	A1 to A14	A15
			Row address	A10 to A23	A24
1	0	10 bits	Column address	A1 to A14	A15
			Row address	A11 to A24	A25
1	1	11 bits	Column address	A1 to A14	A15
			Row address	A12 to A25	A15

Basic Timing: The basic timing for DRAM access is 3 cycles (RCD (1–0) = 00, AnW (1–0) = 01). This basic timing is shown in figure 10.14. Tpc is the precharge cycle, Tr the RAS assert cycle, Tc1 the CAS assert cycle, and Tc2 the read data latch cycle.

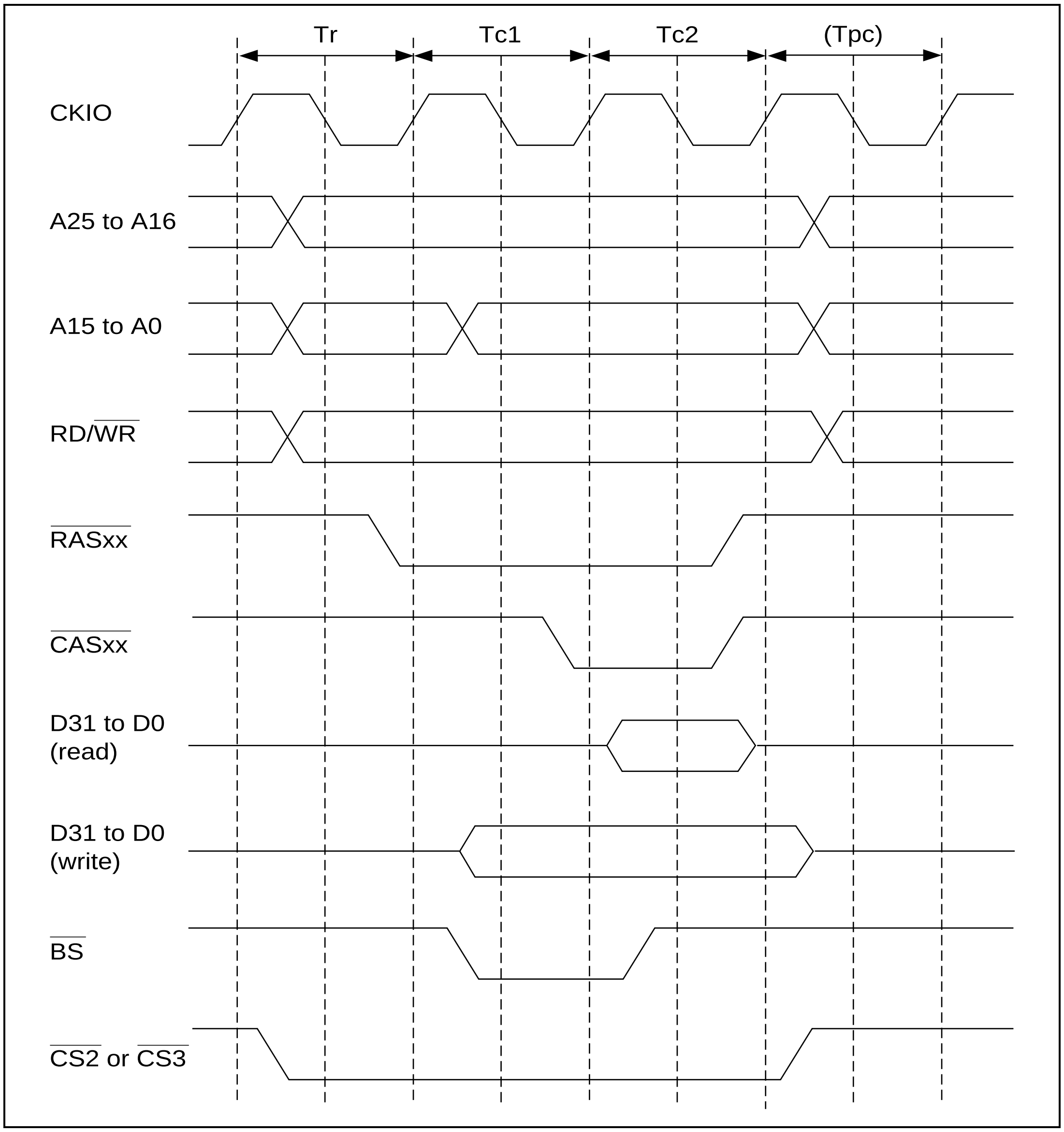

Tr
Tc1
Tc2
(Tpc)
CKIO
A25 to A16
A15 to A0
RD/$\overline{\text{WR}}$
$\overline{\text{RASxx}}$
$\overline{\text{CASxx}}$
D31 to D0
(read)
D31 to D0
(write)
$\overline{\text{BS}}$
$\overline{\text{CS2}}$ or $\overline{\text{CS3}}$

Figure 10.14 Basic Timing for DRAM Access

Wait State Control: As the clock frequency increases, it becomes impossible to complete all states in one cycle as in basic access. Therefore, provision is made for state extension by using the setting bits in WCR2, MCR, and DCR. The timing with state extension using these settings is shown in figure 10.15. Additional Tpc cycles (cycles used to secure the RAS precharge time) can be inserted by means of the TPC bits in MCR and DCR, giving from 1 to 4 cycles. The number of cycles from RAS assertion to CAS assertion can be set to between 1 and 4 by inserting Trw cycles by means of the RCD bits in MCR and DCR. And the number of cycles from CAS assertion to the end of the access can be varied between 1 and 3 according to the setting of A1-2W (1-0) or A3W (1-0) in WCR2.

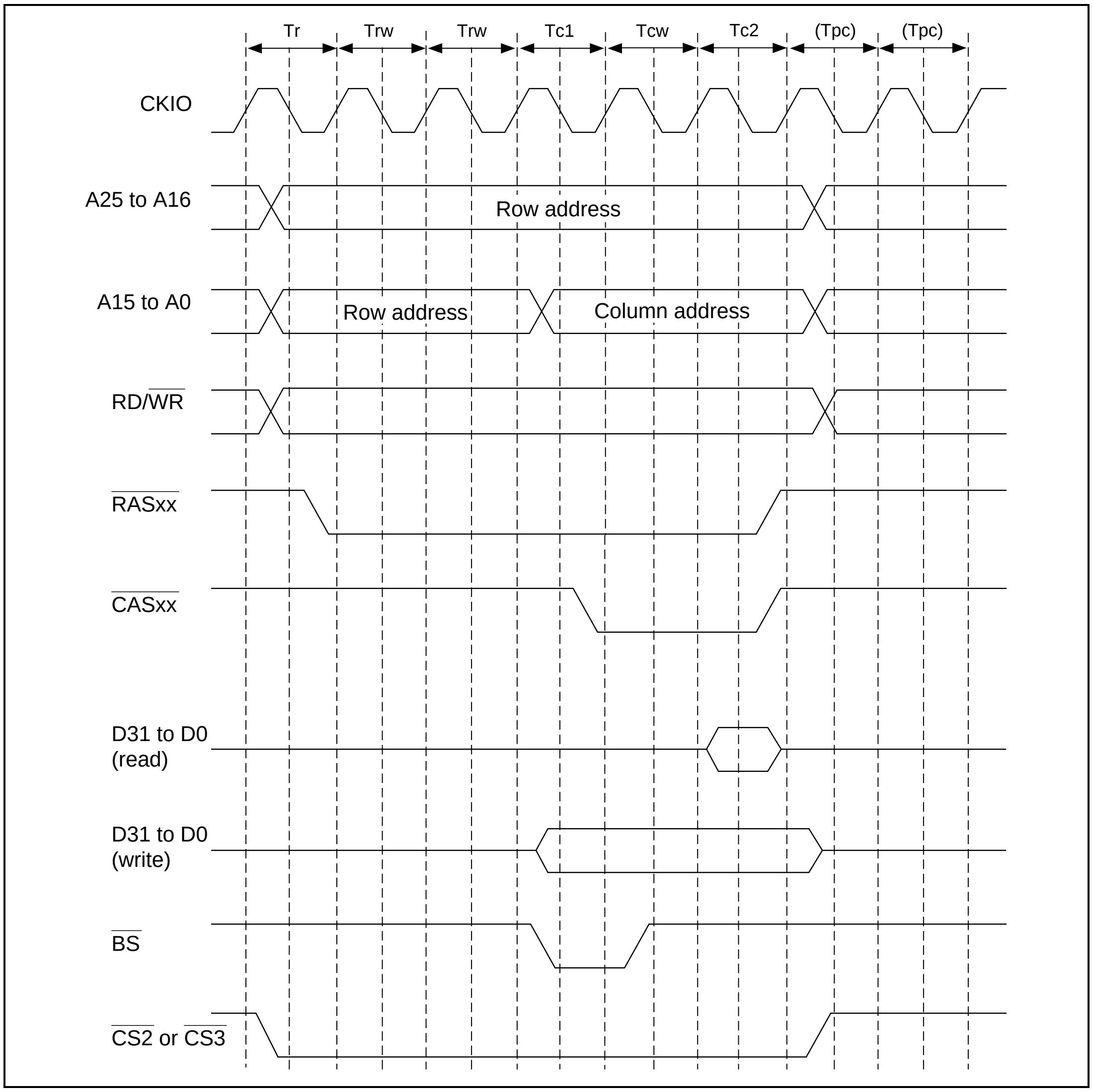


Figure 10.15 DRAM Wait State Timing

Short Pitch Access: The bus state controller (only BSCP) of the SH7709 can perform short-pitch access to DRAM as shown in figure 10.16, when AnW (1–0) is set to 00. Tcs is a CAS assert and data access state.

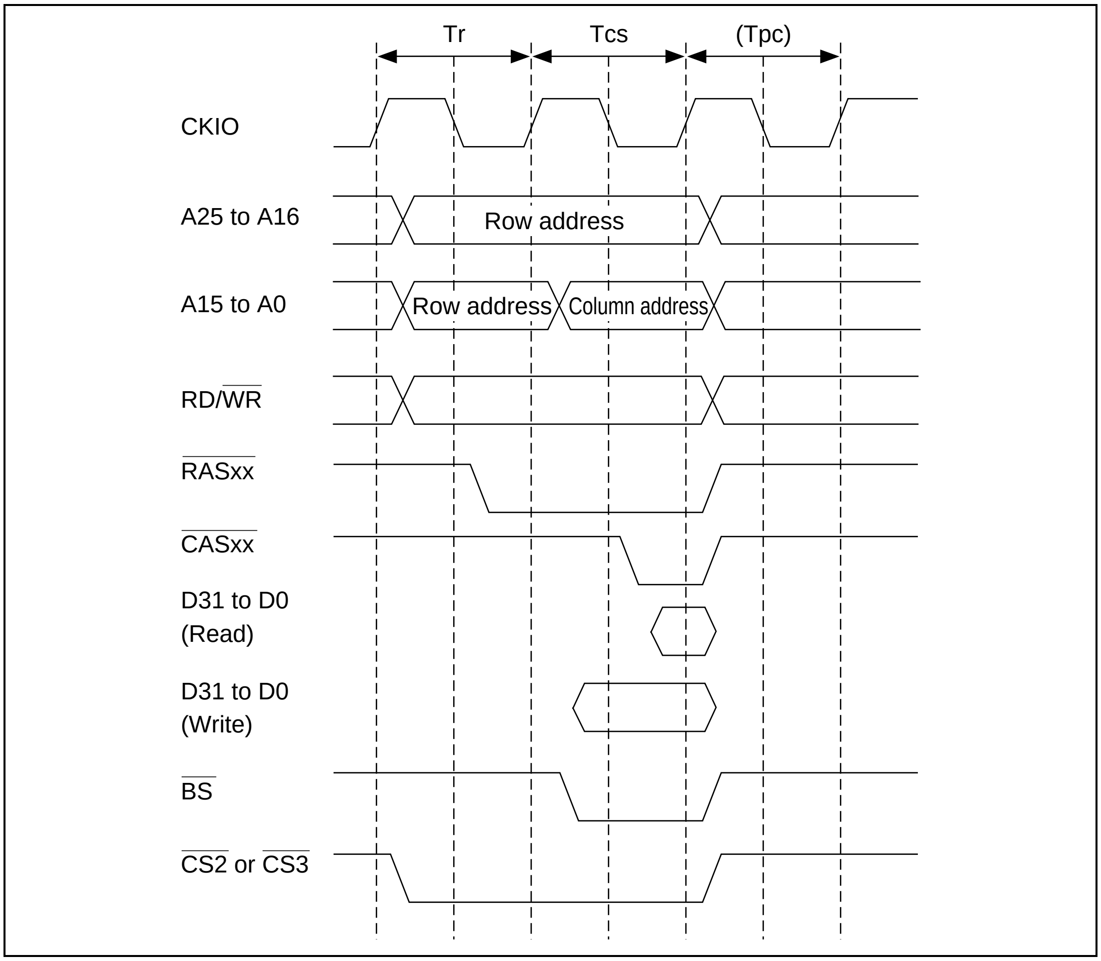


Figure 10.16 DRAM Short-Pitch Access Timing

Burst Access: In addition to the normal DRAM access mode in which a row address is output in each data access, a high-speed page mode is also provided for the case where consecutive accesses are made to the same row. This mode allows fast access to data by outputting the row address only once, then changing only the column address for each subsequent access. Normal access or burst access using high-speed page mode can be selected by means of the burst enable (BE) bit in MCR and DCR. The timing for burst access using high-speed page mode is shown in figure 10.17.

In burst transfer, 4 (longword access) or 16 (cache fill or cache write-back) bytes of data are burst-transferred in the case of a 16-bit bus size. With a 32-bit bus size, 16 bytes of data are burst-transferred (cache fill or cache write-back). In a 16-byte burst transfer (cache fill), the first access comprises a longword that includes the data requiring access. The remaining accesses are performed on 16-byte boundary data that includes the relevant data. In burst transfer (cache write-back), sequential writing is performed from first-to-last order for 16-byte boundary data.

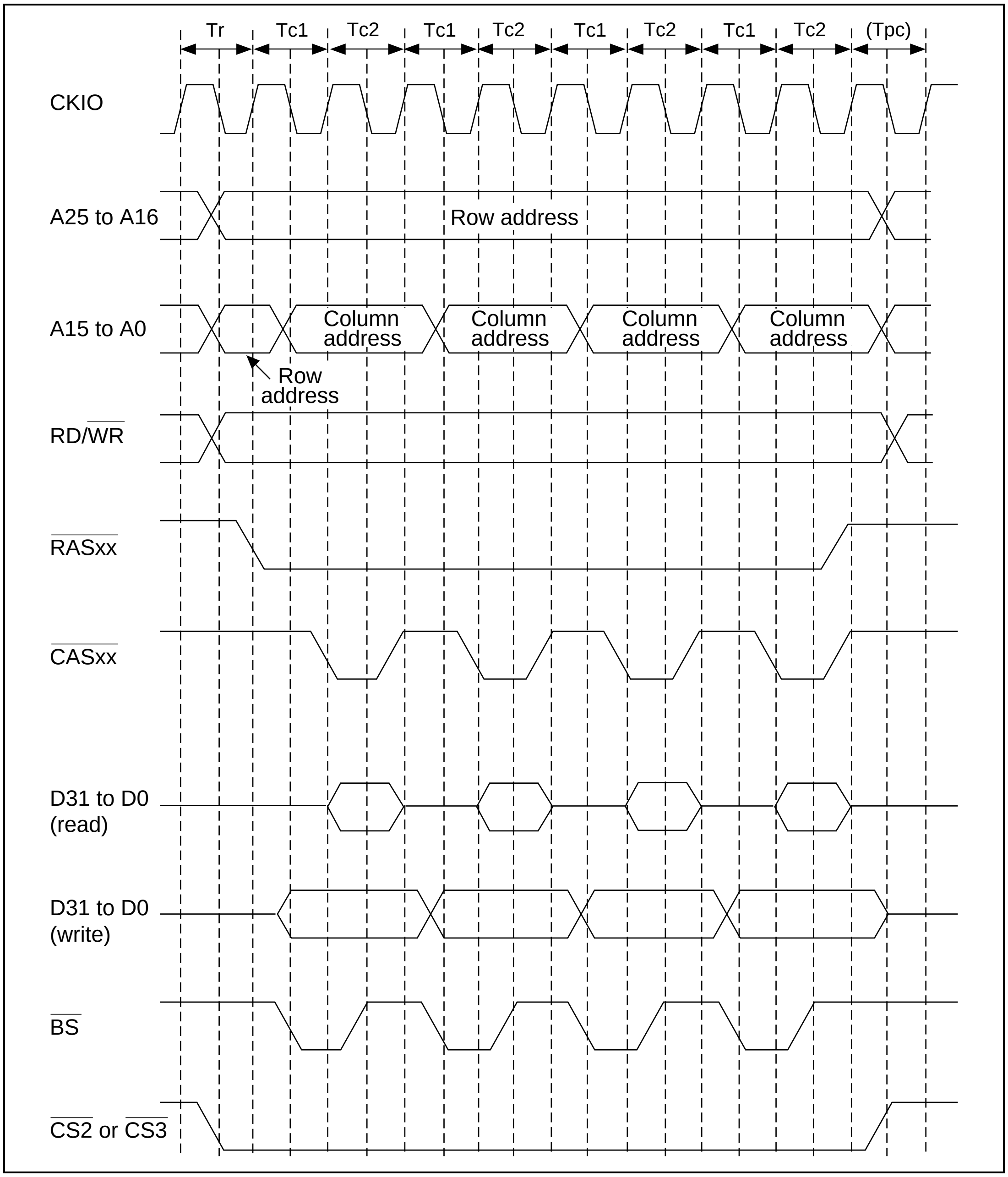


Figure 10.17 DRAM Burst Access Timing

Short-Pitch Burst Access: The bus state controller (only BSCP) of the SH7709 can perform short-pitch burst access to DRAM as shown in figure 10.18, when AnW (1–0) and BE are set to 00 and 1, respectively. Tcs is a CAS assert and data access state.

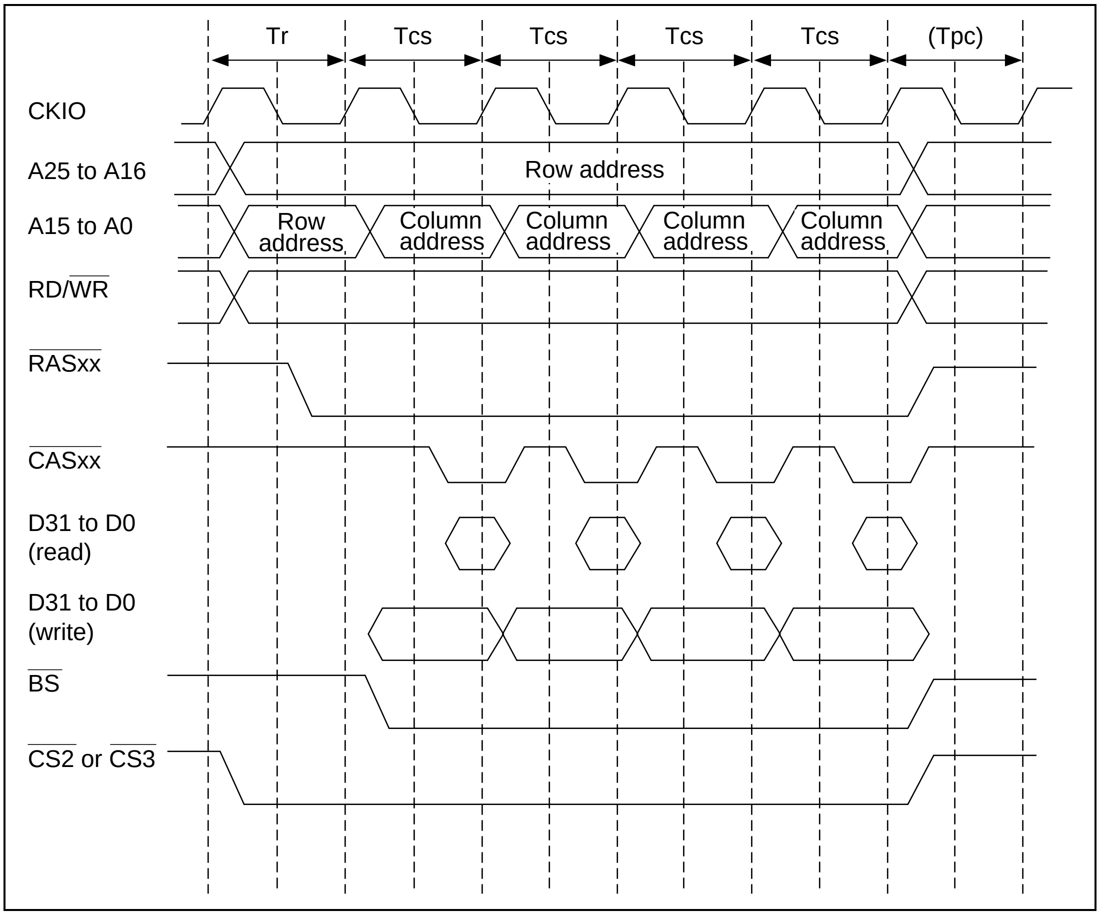


Figure 10.18 DRAM Short-Pitch Burst Access Timing

EDO Mode: With DRAM, in addition to the mode in which data is output to the data bus only while the CAS signal is asserted in a data read cycle, an EDO (extended data out) mode is also provided in which, once the CAS signal is asserted while the RAS signal is asserted, even if the CAS signal is negated, data is output to the data bus until the CAS signal is next asserted. In the SH7709, the EDO mode bit (EDOMODE) in MCR enables selection, for area 3 DRAM only, of either normal access/burst access using high-speed page mode or EDO mode normal access/burst access. EDO mode normal access is shown in figure 10.19, and burst access in figure 10.20.

In EDO mode, the timing for data output to the data bus in a read cycle is extended as far as the next assertion of the CAS signal, so that the data latch timing is delayed by 1/2 cycle and made the rising edge of the CKIO clock, enabling the DRAM access time to be increased.

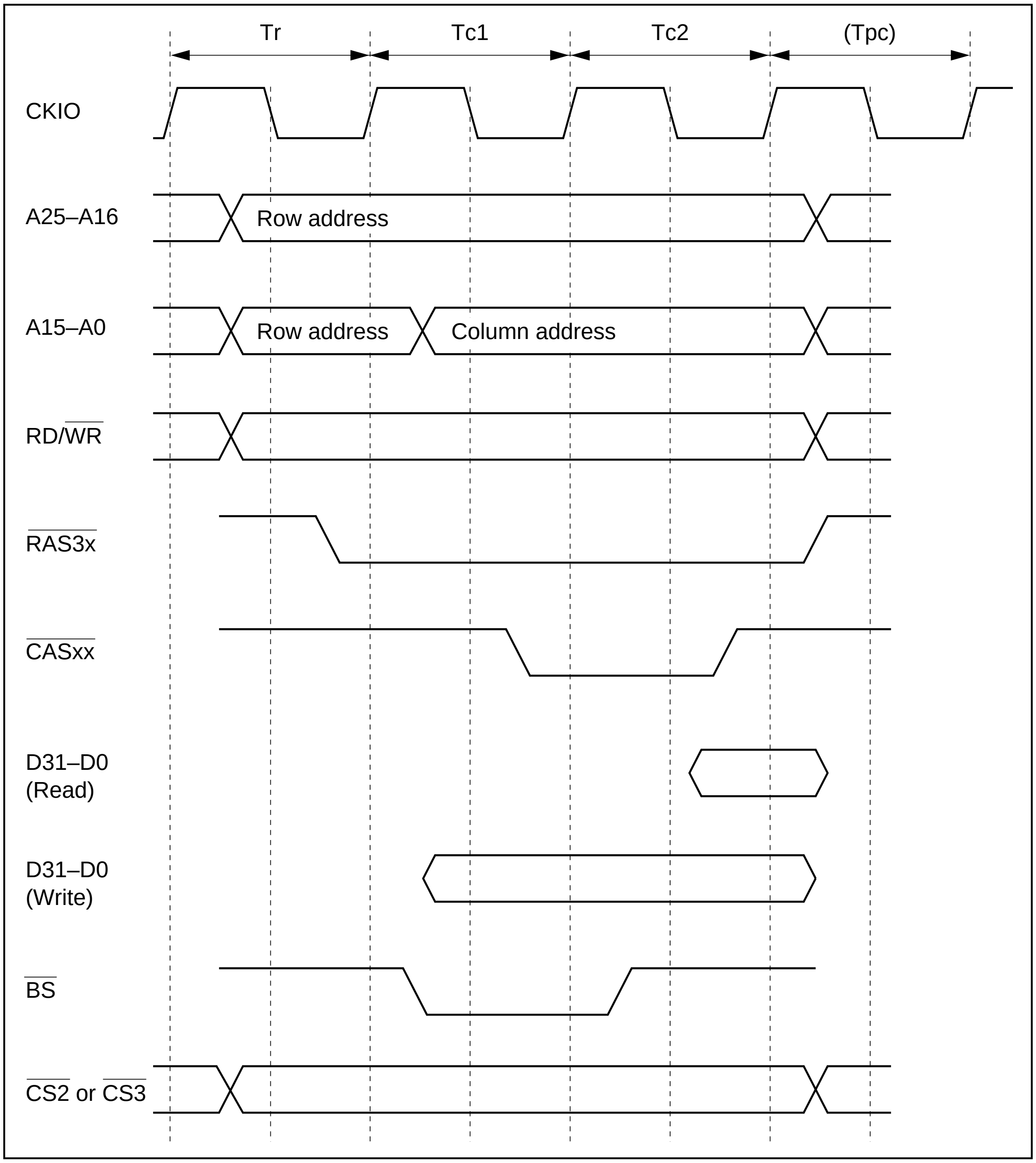


Figure 10.19 Normal Access Timing in DRAM EDO Mode

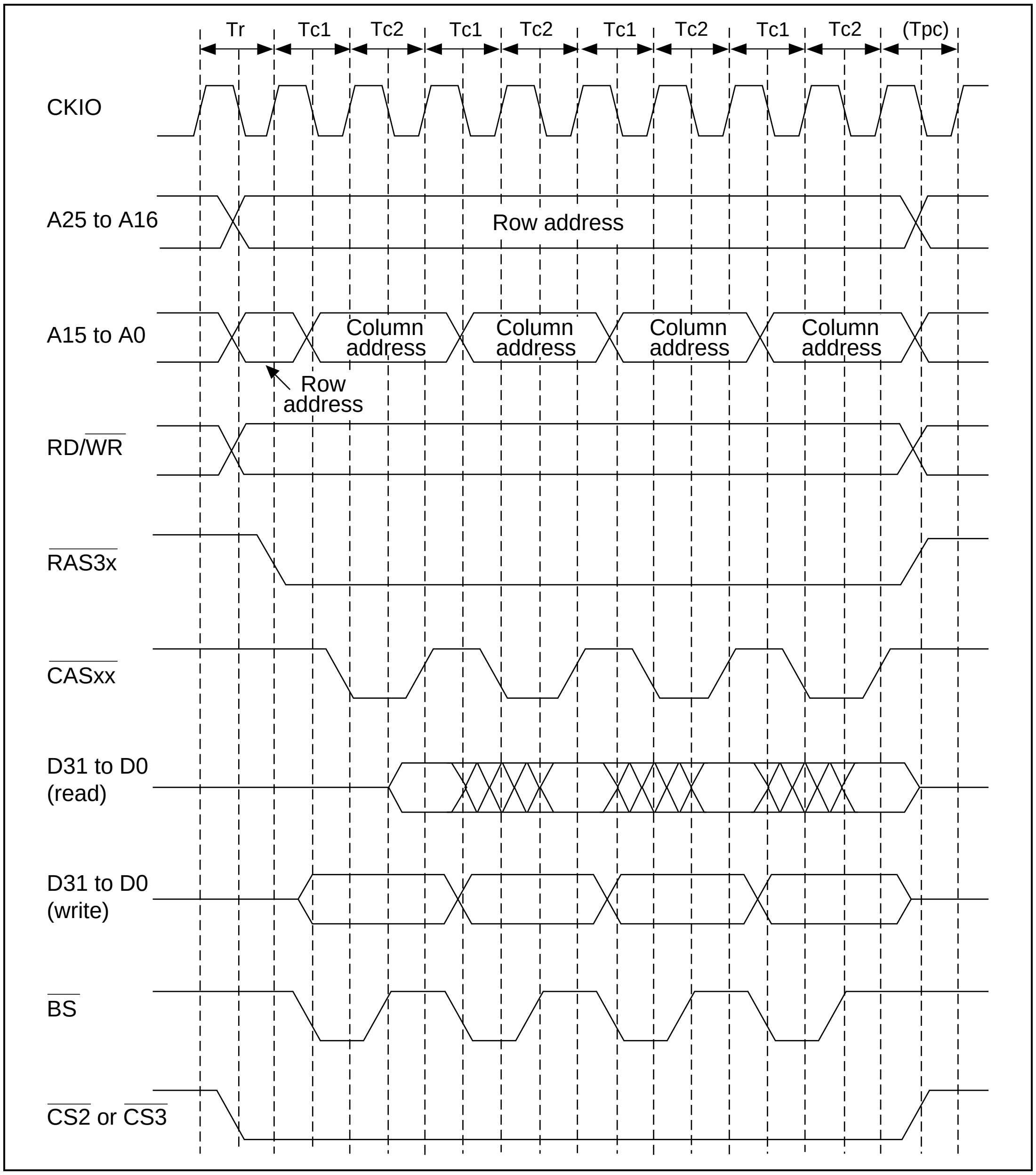


Figure 10.20 Burst Access Timing in DRAM EDO Mode

Refresh Timing: The bus state controller includes a function for controlling DRAM refreshing. Distributed refreshing using a CAS-before-RAS cycle can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR for area 3 DRAM, or by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in DCR for area 2 DRAM.

When CAS-before-RAS refresh cycles are executed, refreshing is performed at intervals determined by the input clock selected by bits CKS2-0 in RTCSR, and the value set in RTCOR. The value of bits CKS2-0 in RTCOR should be set so as to satisfy the stipulation for the DRAM refresh interval. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2-CKS0 setting. When the clock is selected by CKS2-CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and the IRQOUT pin goes low. If the SH7709's external bus can be used, CAS-before-RAS refreshing is performed, and if there is no other interrupt request the IRQOUT pin goes high. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 10.21 shows the operation of CAS-before-RAS refreshing.

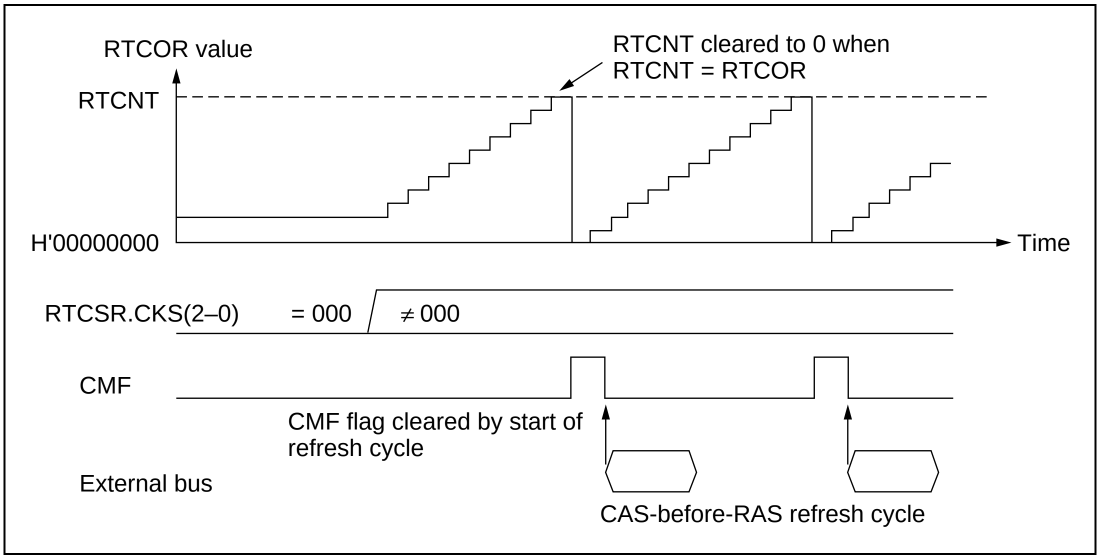


Figure 10.21 CAS-Before-RAS Refresh Operation

Figure 10.22 shows the timing of the CAS-before-RAS refresh cycle.

The number of RAS assert cycles in the refresh cycle is specified by the TRAS bits in MCR and DCR. The specification of the RAS precharge time in the refresh cycle is determined by the setting of the TPC bits in MCR and DCR in the same way as for normal access.

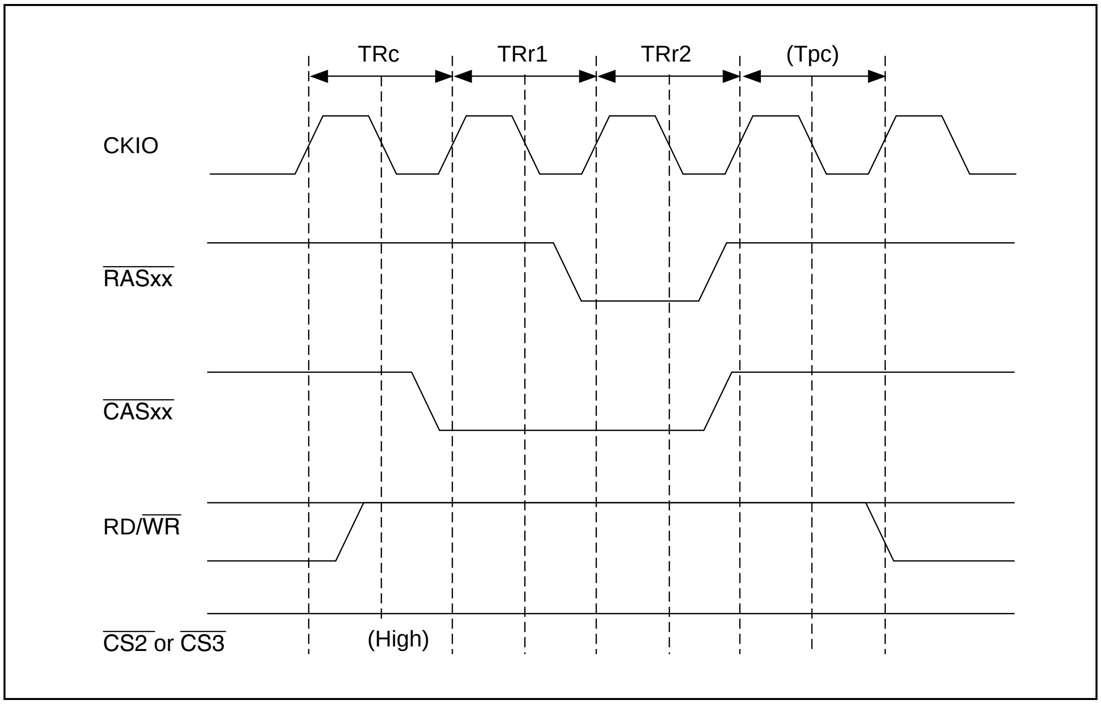


Figure 10.22 DRAM CAS-Before-RAS Refresh Cycle Timing

The self-refreshing supported by the SH7709 is shown in figure 10.23.

After the self-refresh is cleared, the refresh controller immediately generates a refresh request. The RAS precharge time immediately after the end of the self-refreshing can be set by the TPC bits in MCR and DCR.

DRAMs include low-power products (L versions) with a long refresh cycle time (for example, the L version of the HM51W4160AL has a refresh cycle of 1024 cycles/128 ms compared with 1024 cycles/16 ms for the normal version). With these DRAMs, however, the same refresh cycle as for the normal version is requested only ion the case of refreshing immediately following self-refreshing. To ensure efficient DRAM refreshing, therefore, processing is needed to generate an overflow interrupt and restore the refresh cycle to the proper value, after the necessary CAS-before-RAS refreshing has been performed following self-refreshing of an L-version DRAM, using RFCR and the OVF, OVIE, and LMTS bits in RTCSR. The necessary procedure is as follows.

1. Normally, set the refresh counter count value to the optimum value for the L version (e.g. 1024 cycles/128 ms).
2. When a transition is made to self-refreshing:
   a. Provide an interrupt handler to restore the refresh counter count value to the optimum value for the L version (e.g. 1024 cycles/128 ms) when a refresh counter overflow interrupt is generated.
   b. Re-set the refresh counter count value to the requested short cycle (e.g. 1024 cycles/16 ms), set refresh controller overflow interruption, and clear the refresh count register (RFCR) to 0.
   c. Set self-refresh mode.

By using this procedure, the refreshing immediately following a self-refresh will be performed in a short cycle, and when adequate refreshing ends, an interrupt is generated and the setting can be restored to the original refresh cycle.

CAS-before-RAS refreshing is performed in normal operation, in sleep mode, and in the case of a manual reset.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in the case of a manual reset.

When the bus has been released in response to a bus arbitration request, or when a transition is made to standby mode, signals generally become high-impedance, but whether the RAS and CAS signals become high-impedance or continue to be output can be controlled by the HIZCNT bit in BCR1. This enables the DRAM to be kept in the self-refreshing state.

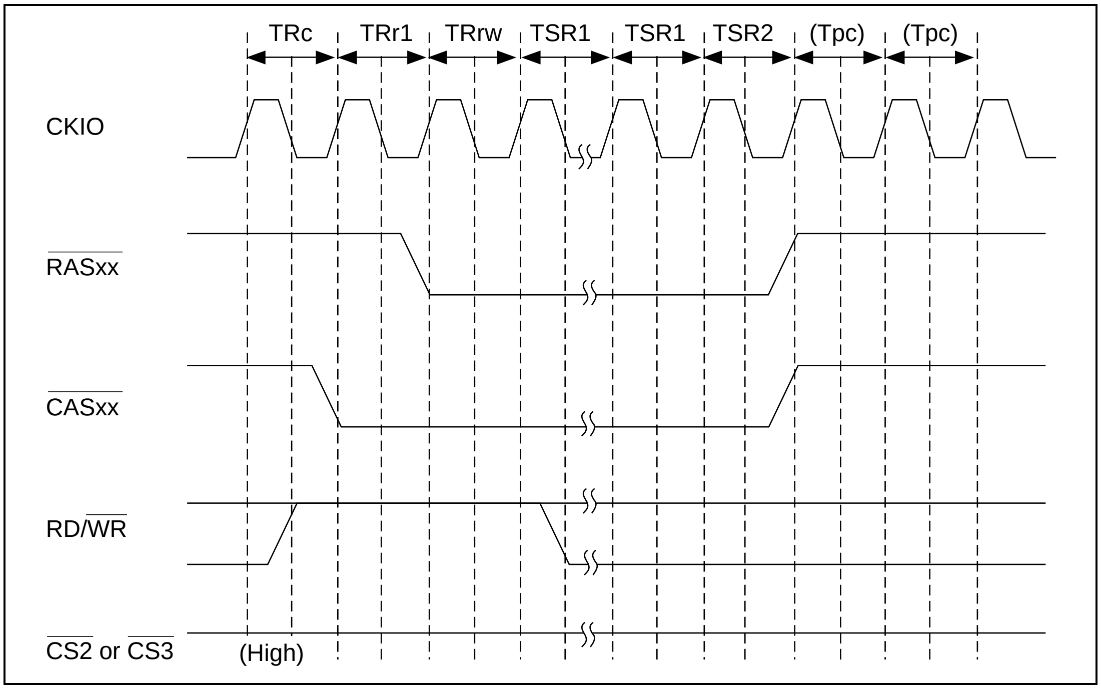


Figure 10.23 DRAM Self-Refresh Cycle Timing

Power-On Sequence: Regarding use of DRAM after powering on, it is requested that a wait time (at least 100 µs or 200 µs) during which no access can be performed be provided, followed by the prescribed number (usually 8) or more dummy CAS-before-RAS refresh cycles. As the bus state controller does not perform any special operations for a power-on reset, the necessary power-on sequence must be carried out by the initialization program executed after a power-on reset.

10.3.5 Synchronous DRAM Interface

Synchronous DRAM Direct Connection: Since synchronous DRAM can be selected by the CS signal, physical space areas 2 and 3 can be connected using RAS and other control signals in common. If the memory type bits (DRAMTP2-0) in BCR1 are set to 010, area 2 is normal memory space and area 3 is synchronous DRAM space; if set to 011, areas 2 and 3 are both synchronous DRAM space.

With the SH7709, burst length 1 burst read/single write mode is supported as the synchronous DRAM operating mode. The burst enable bit (BE) in MCR is ignored, a16-bit burst transfer is performed in a cache fill/copy-back cycle, and only one access is performed in a write-through area write or a noncacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{\text{RAS3L}}$, $\overline{\text{RAS3U}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, CS2 or CS3, DQMUU, DQMUL, DQMLU, DQMLL, and CKE. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid and fetched to the synchronous DRAM only when CS2 or CS3 is asserted. Synchronous DRAM can therefore be connected in parallel to a number of areas. CKE is negated (low) only when self-refreshing is performed, and is always asserted (high) at other times.

Commands for synchronous DRAM are specified by $\overline{\text{RAS3L}}$, $\overline{\text{RAS3U}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, and special address signals. The commands are NOP, auto-refresh (REF), self-refresh (SELF), precharge all banks (PALL), precharge specified bank (P RE), row address strobe bank active (ACTV), read (READ), read with precharge (READA), write (WRIT), write with precharge (WRITA), and mode register write (MRS).

Byte specification is performed by DQMUU, DQMUL, DQMLU, and DQMLL. A read/write is performed for the byte for which the corresponding DQM is low. In big-endian mode, DQMUU specifies an access to address 4n, and DQMLL specifies an access to address 4n + 3. In little-endian mode, DQMUU specifies an access to address 4n + 3, and DQMLL specifies an access to address 4n.

Figure 10.24 shows an example of the connection of 256k × 16-bit synchronous DRAMs.

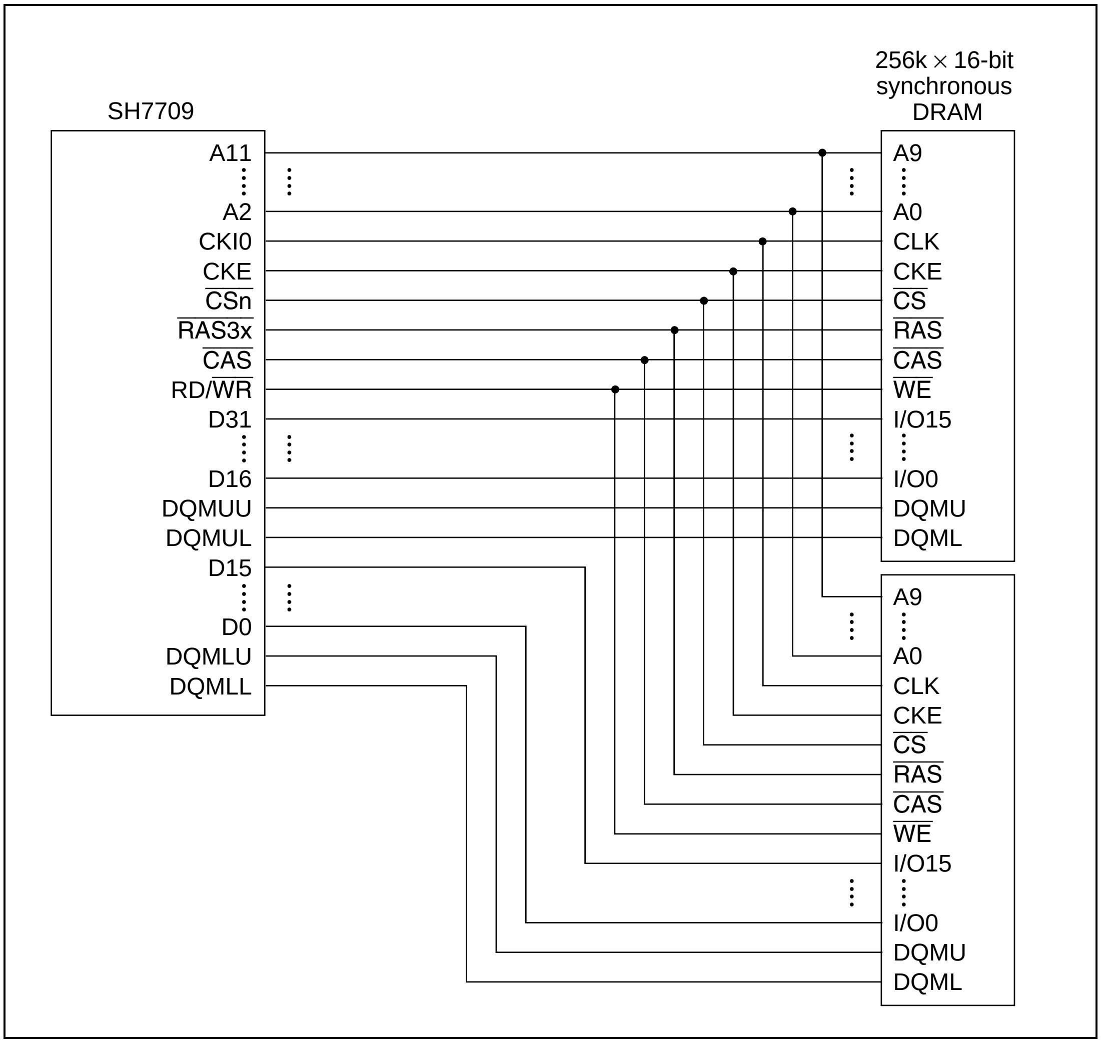


Figure 10.24 Example of Synchronous DRAM Connection

Address Multiplexing: Synchronous DRAM can be connected without external multiplexing circuitry in accordance the address multiplex specification bits AMX1-AMX0 in MCR. Table 10.14 shows the relationship between the address multiplex specification bits and the bits output at the address pins.

A25-A16 and A0 are not multiplexed; the original values are always output at these pins.

When A0, the LSB of the synchronous DRAM address, is connected to the SH7709, it performs longword address specification. Connection should therefore be made in this order: connect pin A0 of the synchronous DRAM to pin A2 of the SH7709, then connect pin A1 to pin A3.

Table 10.14 Relationship between SZ, AMX, and Address Multiplex Output

Setting			External Address Pins							
AMX1	AMX0	Output Timing	A1 to A8	A9	A10	A11	A12	A13	A14	A15
0	0	Column address	A1 to A8	A9	A10	A11	A12*1	A13*2	A14	A15
		Row address	A9 to A16	A17	A18	A19	A20	A21*2	A22	A23
0	1	Column address	A1 to A8	A9	A10	A11	L/H*1	A22*2	A14	A15
		Row address	A10 to A17	A18	A19	A20	A21	A22*2	A23	A24
1	0	Column address	A1 to A8	A9	A20	A11	L/H*1	A23*2	A14	A15
		Row address	A11 to A18	A19	A20	A21	A22	A23*2	A24	A25
1	1	Column address	A1 to A8	A9	L/H*1	A19*2	A12	A13	A14	A15
		Row address	A9 to A16	A17	A18	A19*2	A20	A21	A22	A23

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

2. Bank address specification

Table 10.15 Example of Correspondence between SH7709 and Synchronous DRAM Address Pins (AMX (1-0) = 11)

SH7709 Address Pin			Synchronous DRAM Address Pin	
	RAS Cycle	CAS Cycle		Function
A11	A19	A19	A9	BANK select bank address
A10	A18	L/H	A8	Address precharge setting
A9	A17	A9	A7	Address
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Not used	
A0	A0	A0	Not used	

Burst Read: In the following example it is assumed that four 2M × 8-bit synchronous DRAMs are connected and a 32-bit data width is used, and the burst length is 1. Following the Tr cycle in which ACTV command output is performed, a READ command is issued in the Tc1, Tc2, and Tc3 cycles, and a READA command in the Tc4 cycle, and the read data is accepted on the rising edge of the external command clock (CKIO) from cycle Td1 to cycle Td4. The Tpc cycle is used to wait for completion of auto-precharge based on the READA command inside the synchronous DRAM; no new access command can be issued to the same bank during this cycle, but access to synchronous DRAM for another area is possible. In the SH7709, the number of Tpc cycles is determined by the TPC bit specification in MCR, and commands cannot be issued for the same synchronous DRAM during this interval.

The example in figure 10.25 shows the basic timing. To connect slower synchronous DRAM, the cycle can be extended by setting WCR2 and MCR bits. The number of cycles from the ACTV command output cycle, Tr, to the READ command output cycle, Tc1, can be specified by the RCD bit in MCR, with a values of 0 to 3 specifying 1 to 4 cycles, respectively. In case of 2 or more cycles, a Trw cycle, in which an NOP command is issued for the synchronous DRAM, is inserted between the Tr cycle and the Tc cycle. The number of cycles from READ and READA command output cycles Tc1-Tc4 to the first read data latch cycle, Td1, can be specified as 1 to 3 cycles independently for areas 2 and 3 by means of A1-2W1 and A1-2W0 or A3W1 and A3W0 in WCR2. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

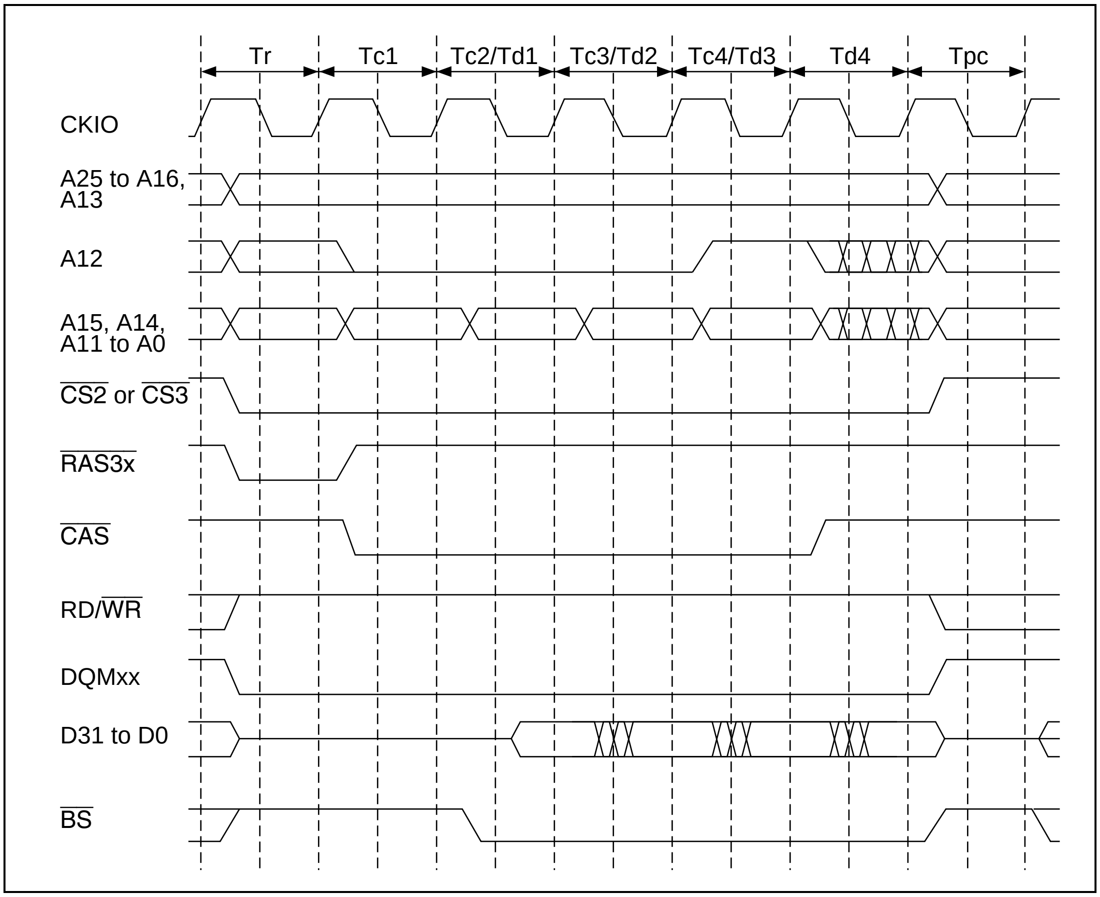


Figure 10.25 Basic Timing for Synchronous DRAM Burst Read

Figure 10.26 shows the burst read timing when RCD is set to 1, A3W1 and A3W0 are set to 10, and TPC is set to 1.

The BS cycle, which is asserted for one cycle at the start of a bus cycle for normal access space, is asserted in each of cycles Td1-Td4 in a synchronous DRAM cycle. When a burst read is performed, the address is updated each time CAS is asserted. As the unit of burst transfer is 16 bytes, address updating is performed for A3 and A2 only. The order of access is as follows: in a fill operation in the event of a cache miss, the missed data is read first, then 16-byte boundary data including the missed data is read in wraparound mode.

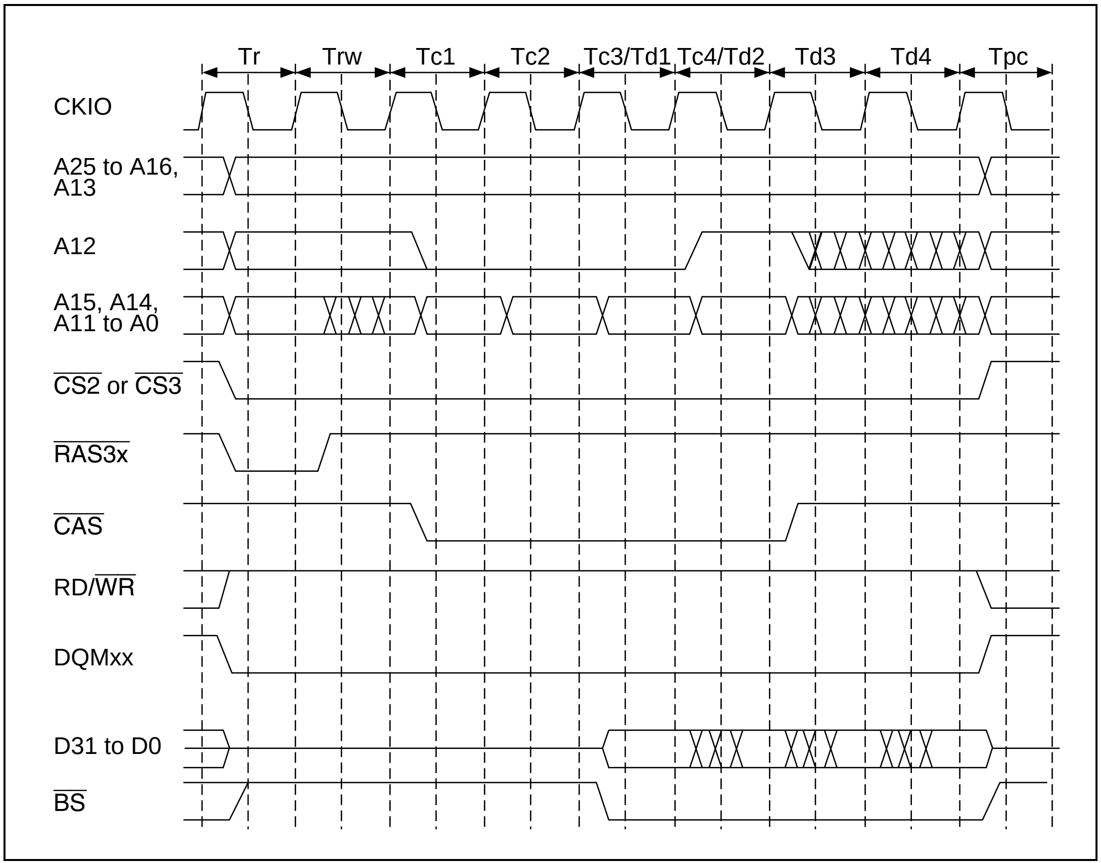


Figure 10.26 Synchronous DRAM Burst Read Wait Specification Timing

Single Read: Figure 10.27 shows the timing when a single address read is performed. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

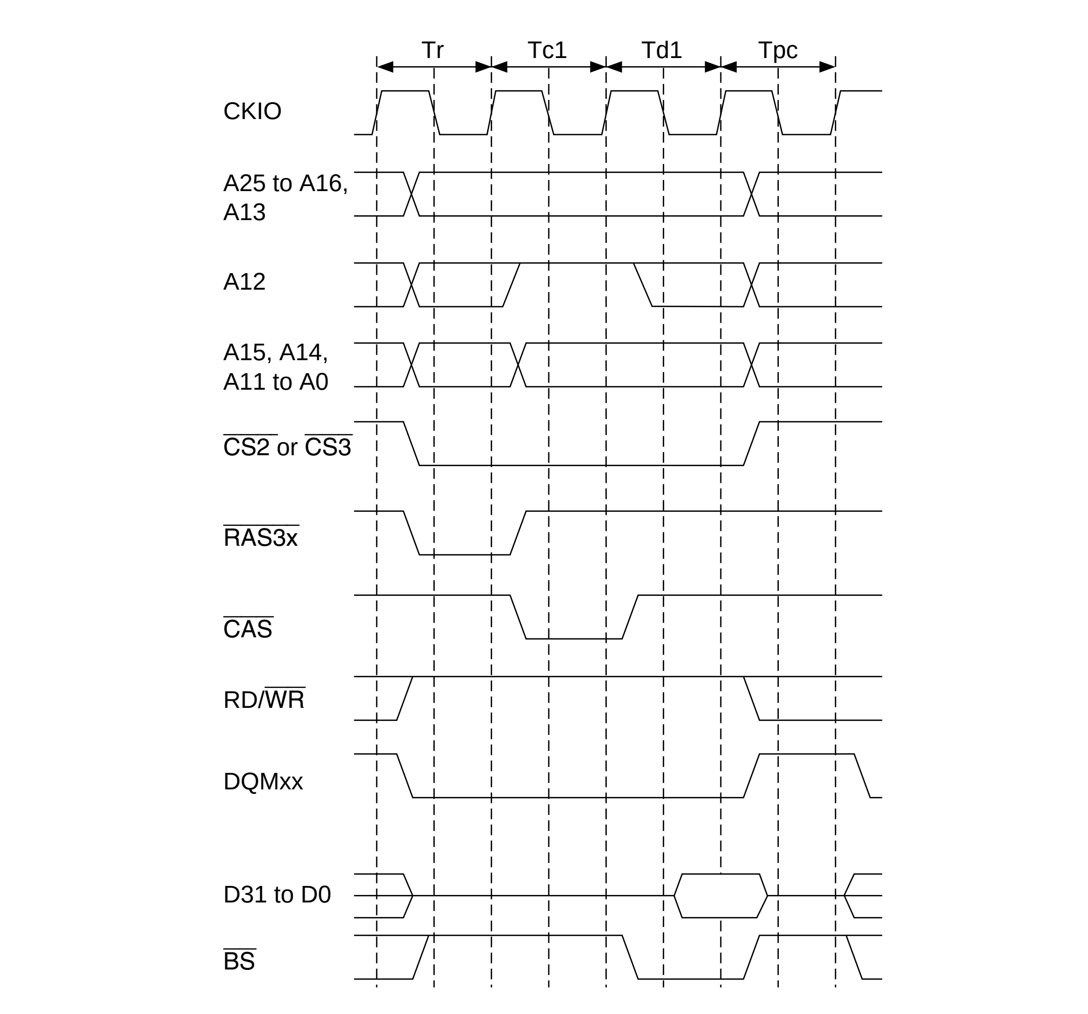


Figure 10.27 Basic Timing for Synchronous DRAM Single Read

Burst Write: The timing chart for a burst write is shown in figure 10.28. In the SH7709, a burst write occurs only in the event of cache copy-back. In a burst write operation, following the Tr cycle in which ACTV command output is performed, a WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and a WRITA command that performs auto-precharge is issued in the Tc4 cycle. In the write cycle, the write data is output at the same time as the write command. In the case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

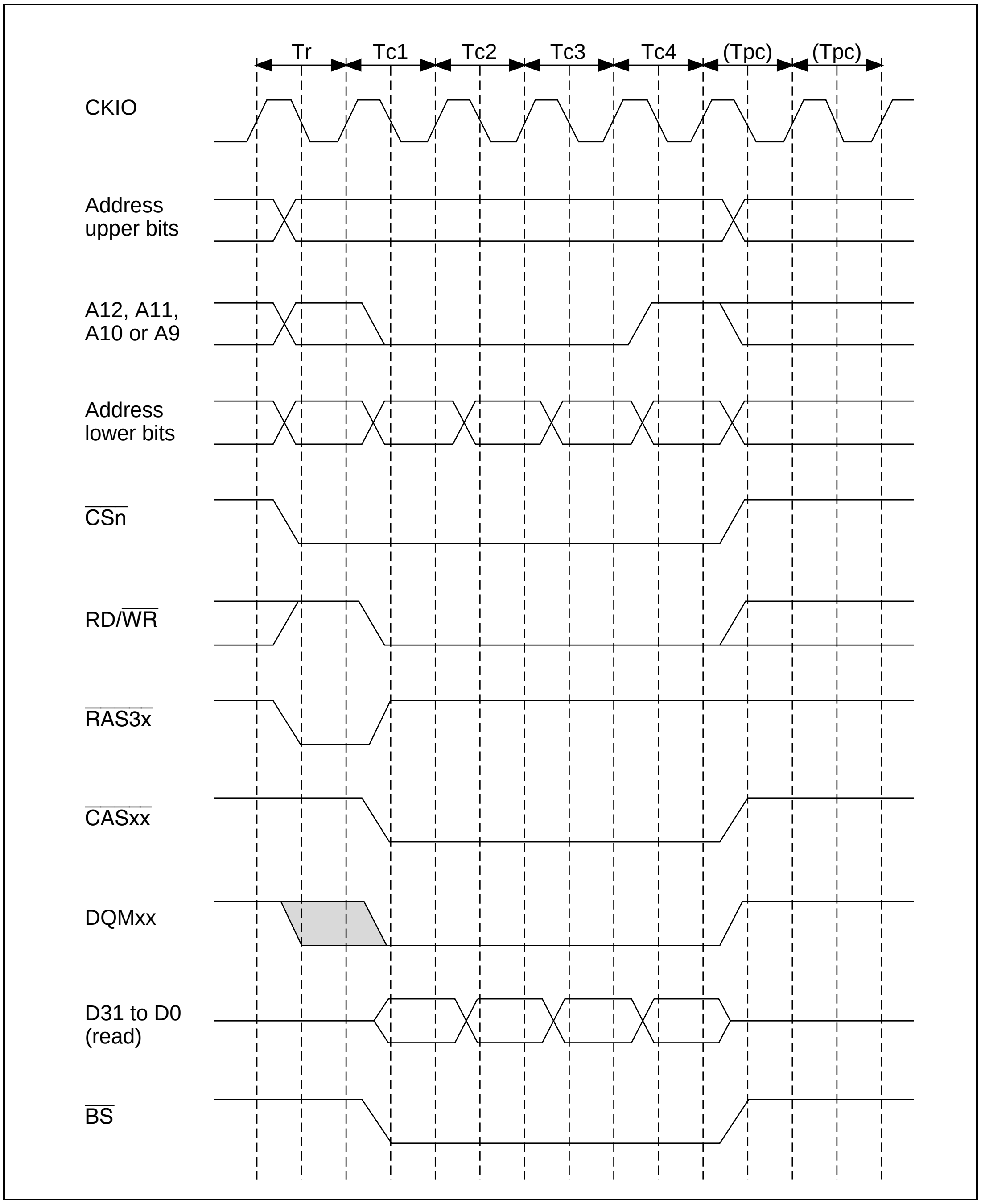


Figure 10.28 Basic Timing for Synchronous DRAM Burst Write

Single Write: The basic timing chart for write access is shown in figure 10.29. In a single write operation, following the Tr cycle in which ACTV command output is performed, a WRITA command that performs auto-precharge is issued in the Tc1 cycle. In the write cycle, the write data is output at the same time as the write command. In the case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

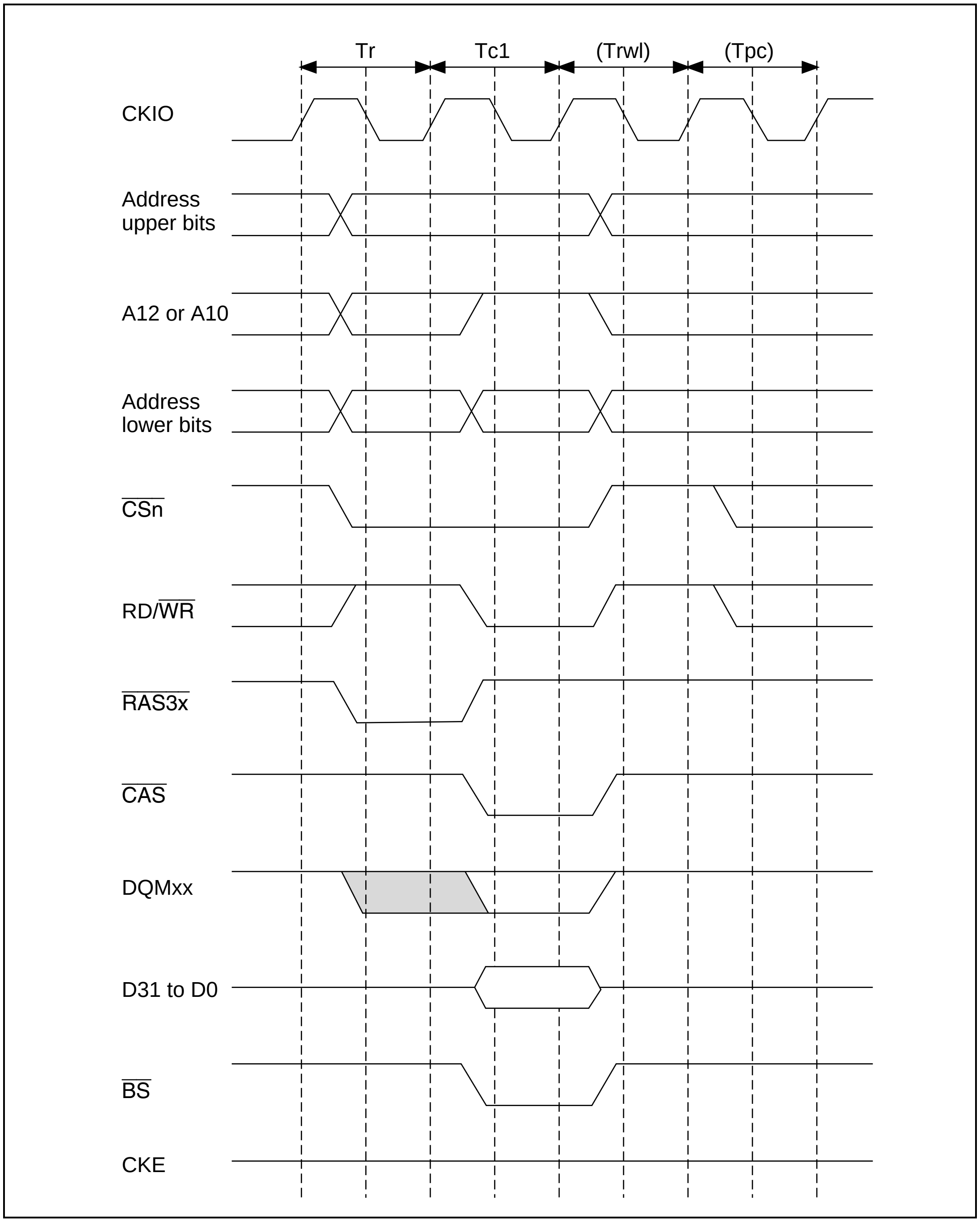


Figure 10.29 Basic Timing for Synchronous DRAM Single Write

Refreshing: The bus state controller is provided with a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-Refreshing

   Refreshing is performed at intervals determined by the input clock selected by bits CKS2-0 in RTCSR, and the value set in RTCOR. The value of bits CKS2-0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2-CKS0 setting. When the clock is selected by CKS2-CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 10.31 shows the auto-refresh cycle timing.

   First, an REF command is issued in the TRr cycle. After the TRr cycle, new command output cannot be performed for the duration of the number of cycles specified by the TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. The TRAS and SPC bits must be set so as to satisfy the synchronous DRAM refresh cycle time stipulation (active/active command delay time).

   Auto-refreshing is performed in normal operation, in sleep mode, and in the case of a manual reset.

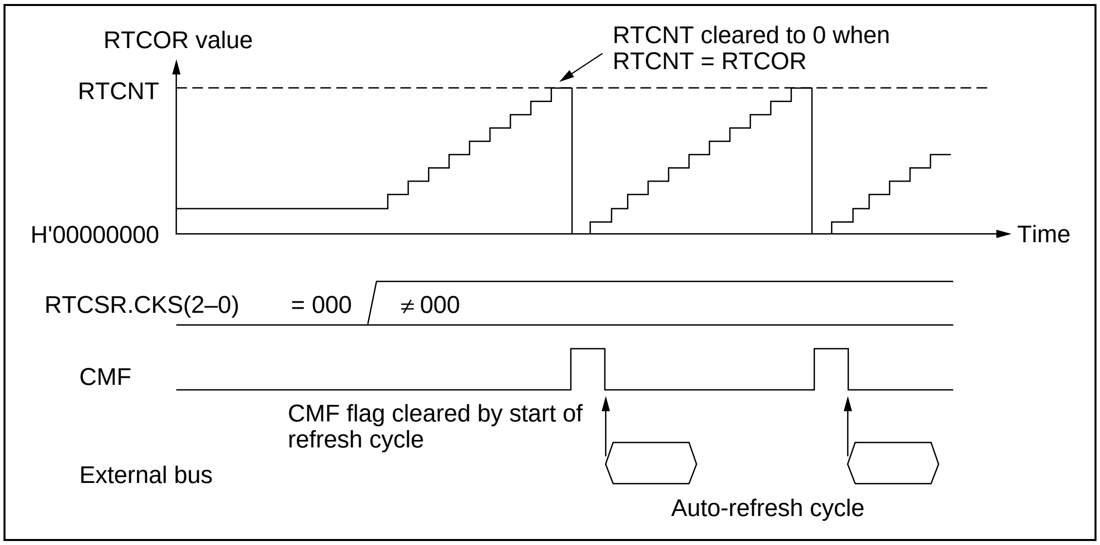


Figure 10.30 Auto-Refresh Operation

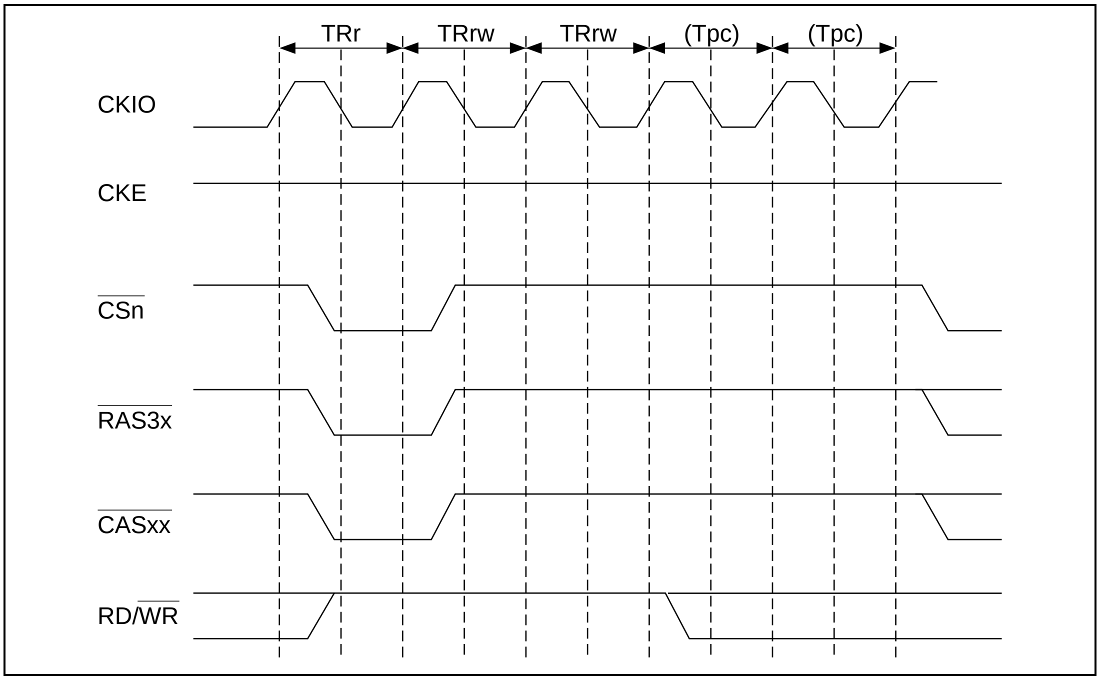


Figure 10.31 Synchronous DRAM Auto-Refresh Timing

2. Self-Refreshing

Self-refresh mode is a kind of standby mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the CKE signal is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TPC bits in MCR. Self-refresh timing is shown in figure 10.32. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the SH7709's standby function, and is maintained even after recovery from standby mode other than through a power-on reset. In the case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in the case of a manual reset.

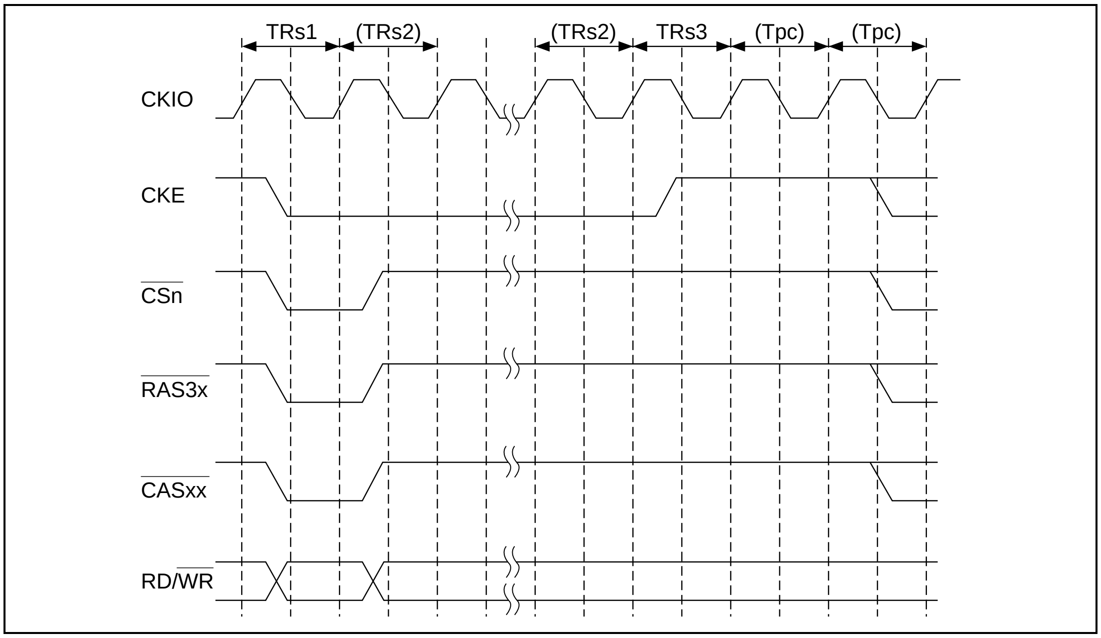


Figure 10.32 Synchronous DRAM Self-Refresh Timing

3. Relationship between Refresh Requests and Bus Cycle Requests

   If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. If a refresh request occurs when the bus has been released by the bus arbiter, refresh execution is deferred until the bus is acquired. If a match between RTCNT and RTCOR occurs while a refresh is waiting to be executed, so that a new refresh request is generated, the previous refresh request is eliminated. In order for refreshing to be performed normally, care must be taken to ensure that no bus cycle or bus mastership occurs that is longer than the refresh interval. When a refresh request is generated, the IRQOUT pin is asserted (driven low). Therefore, normal refreshing can be performed by having the IRQOUT pin monitored by a bus master other than the SH7709 requesting the bus, or the bus arbiter, and returning the bus to the SH7709. When refreshing is started, and if no other interrupt request has been generated, the IRQOUT pin is negated (driven high).

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the RAS, CAS, and RD/WR signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'FFFFD000 + X for area 2 synchronous DRAM, and to address H'FFFFE000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/write , CAS latency 1 to 3, wrap type = sequential, and burst length 1 supported by the SH7709, arbitrary data is written in a byte-size access to the following addresses.

	Area 2	Area 3
CAS latency 1	FFFFD840	FFFFE840
CAS latency 2	FFFFD880	FFFFE880
CAS latency 3	FFFFD8C0	FFFFE8C0

Mode register setting timing is shown in figure 10.33.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge all banks (PALL) command is first issued in the TRp1 cycle, then a mode register write command is issued in the TMw1 cycle.

Before mode register setting, a 100 µs idle time (depending on the memory manufacturer) must be guaranteed after powering on requested by the synchronous DRAM. If the reset signal pulse width is greater than this idle time, there is no problem in performing mode register setting immediately. The number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is usually achieved automatically while various kinds of initialization are being performed after auto-refresh setting, but a way of carrying this out more dependably is to set a short refresh request generation interval just while these dummy cycles are being executed. With simple read or write access, the address counter in the synchronous DRAM used for auto-refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.

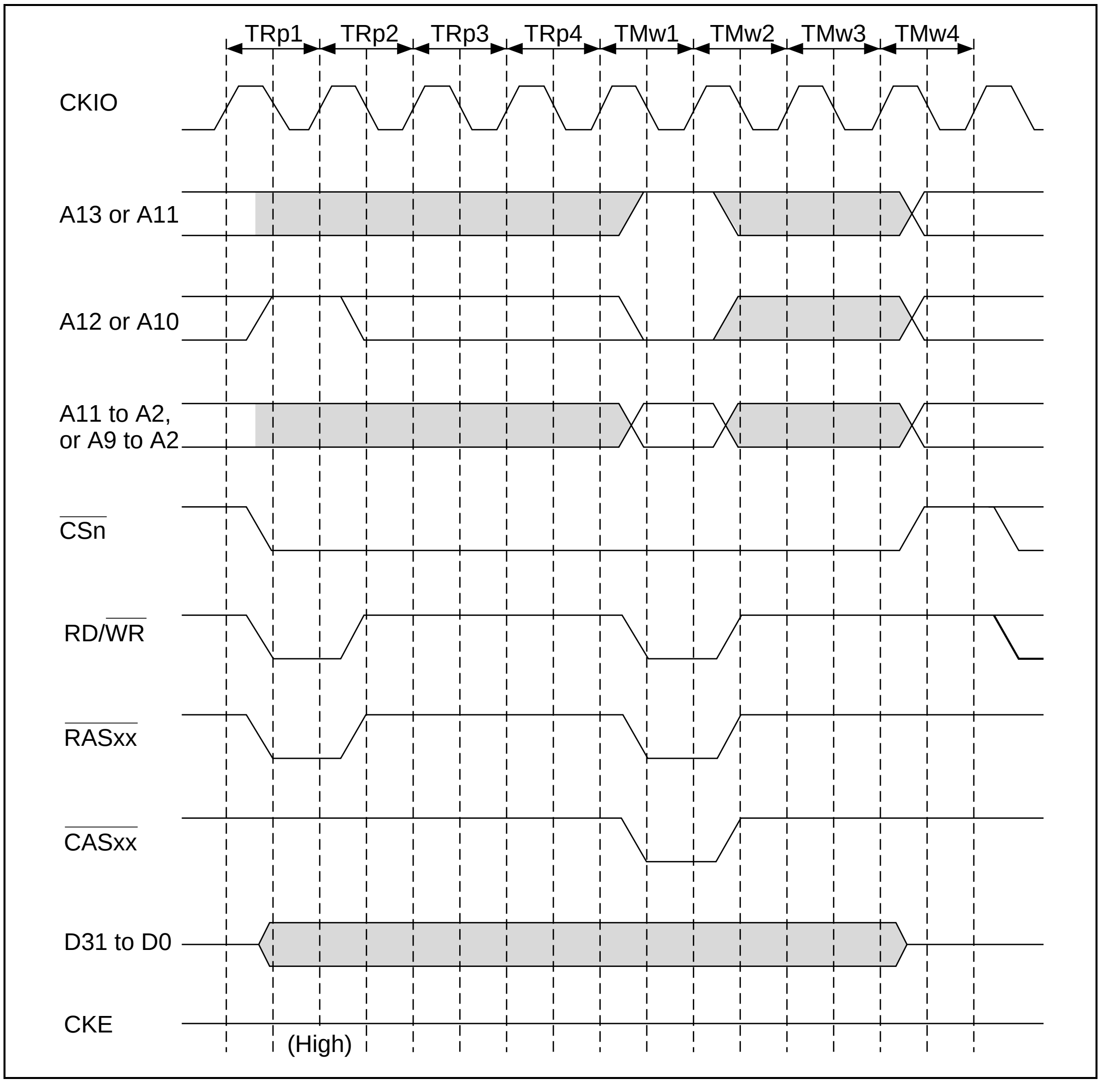


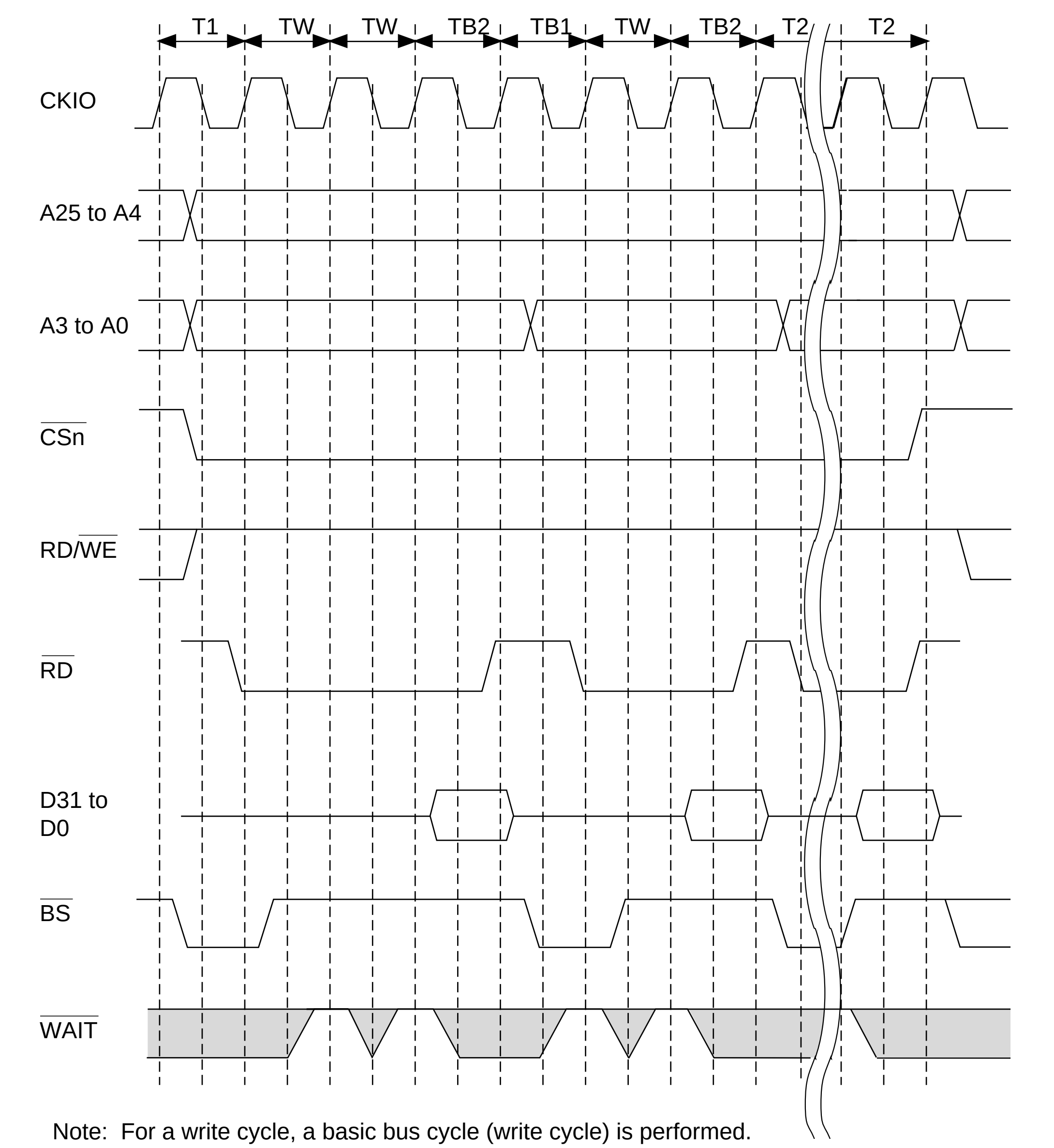
Figure 10.33 Synchronous DRAM Mode Write Timing

10.3.6 Burst ROM Interface

Setting bits A0BST (1-0), A5BST (1-0), and A6BST (1-0) in BCR1 to a non-zero value allows burst ROM to be connected to areas 0, 5, and 6. The burst ROM interface provides high-speed access to ROM that has a nibble access function. The timing for nibble access to burst ROM is shown in figure 10.34. Two wait cycles are set. Basically, access is performed in the same way as for normal space, but when the first cycle ends the CS0 signal is not negated, and only the address is changed before the next access is executed. When 8-bit ROM is connected, the number of consecutive accesses can be set as 4, 8, or 16 by bits A0BST (1-0), A5BST (1-0), or A6BST (1-0). When 16-bit ROM is connected, 4 or 8 can be set in the same way. When 32-bit ROM is connected, only 4 can be set.

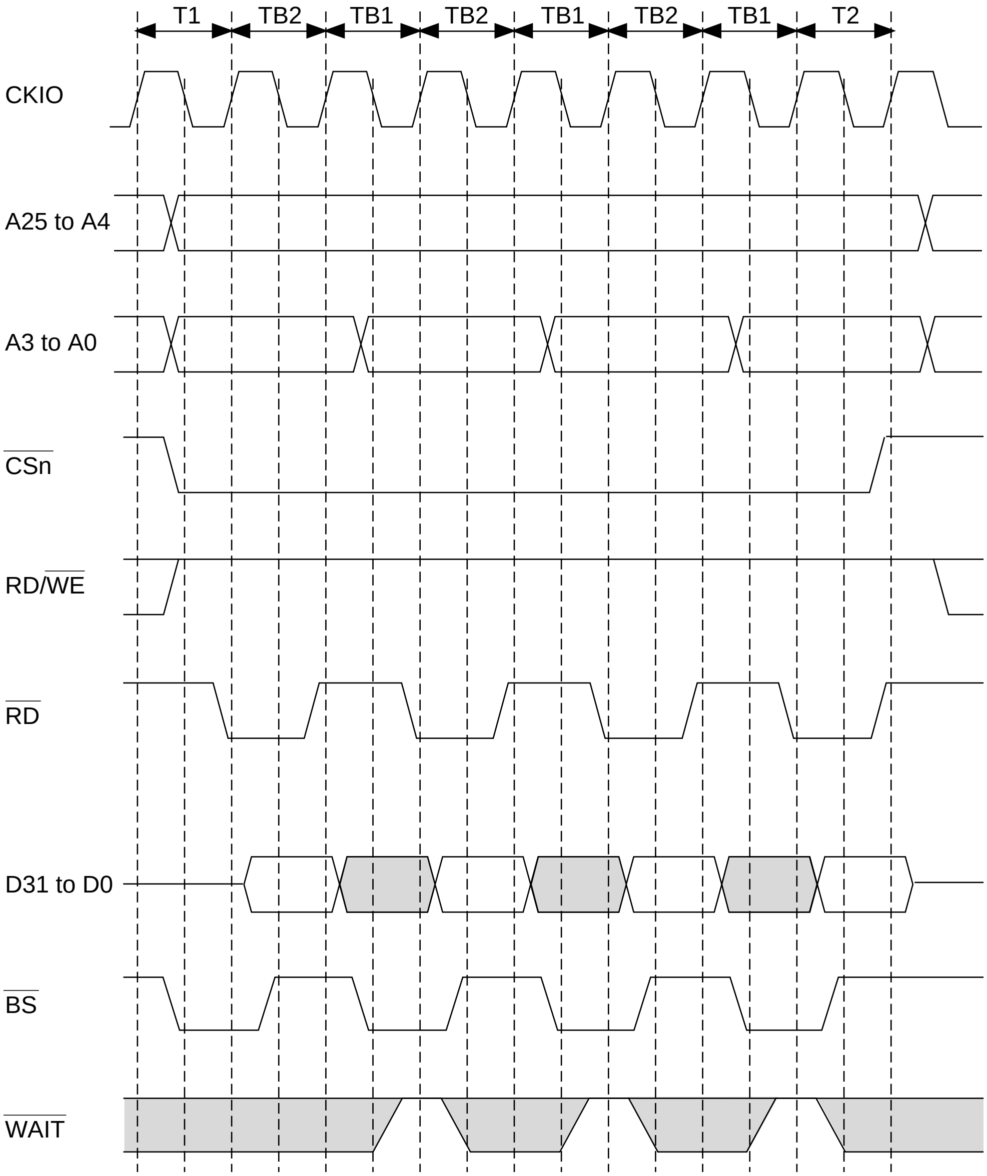
WAIT pin sampling is performed in the first access if one or more wait states are set, and is always performed in the second and subsequent accesses.

The second and subsequent access cycles also comprise two cycles when a burst ROM setting is made and the wait specification is 0. The timing in this case is shown in figure 10.35.



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 10.34 Burst ROM Wait Access Timing



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 10.35 Burst ROM Basic Access Timing

10.3.7 PCMCIA Interface

In the SH7709, setting the A5PCM bit in BCR1 to 1 makes the bus interface for physical space area 5 an IC memory card interface as stipulated in JEIDA version 4.2 (PCMCIA2.1). Setting the A6PCM bit to 1 makes the bus interface for physical space area 6 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2. When the IC memory card interface is selected, a BCR1 register setting enables page mode burst access mode to be used. This burst access mode is not stipulated in JEIDA version 4.2, but allows high-speed data access using ROM provided with a burst mode, etc.

When the PCMCIA interface is used, a bus size of 8 or 16 bits can be set by bits A5SZ1 and A5SZ0, or A6SZ1 and A6SZ0, in BCR2.

Figure 10.36 shows an example of PCMCIA card connection to the SH7709. To enable active insertion of the PCMCIA cards (i.e. insertion or removal while system power is being supplied), a 3-state buffer must be connected between the SH7709’s bus interface and the PCMCIA cards.

As operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA specifications, the PCMCIA interface for the SH7709 in big-endian mode is stipulated independently.

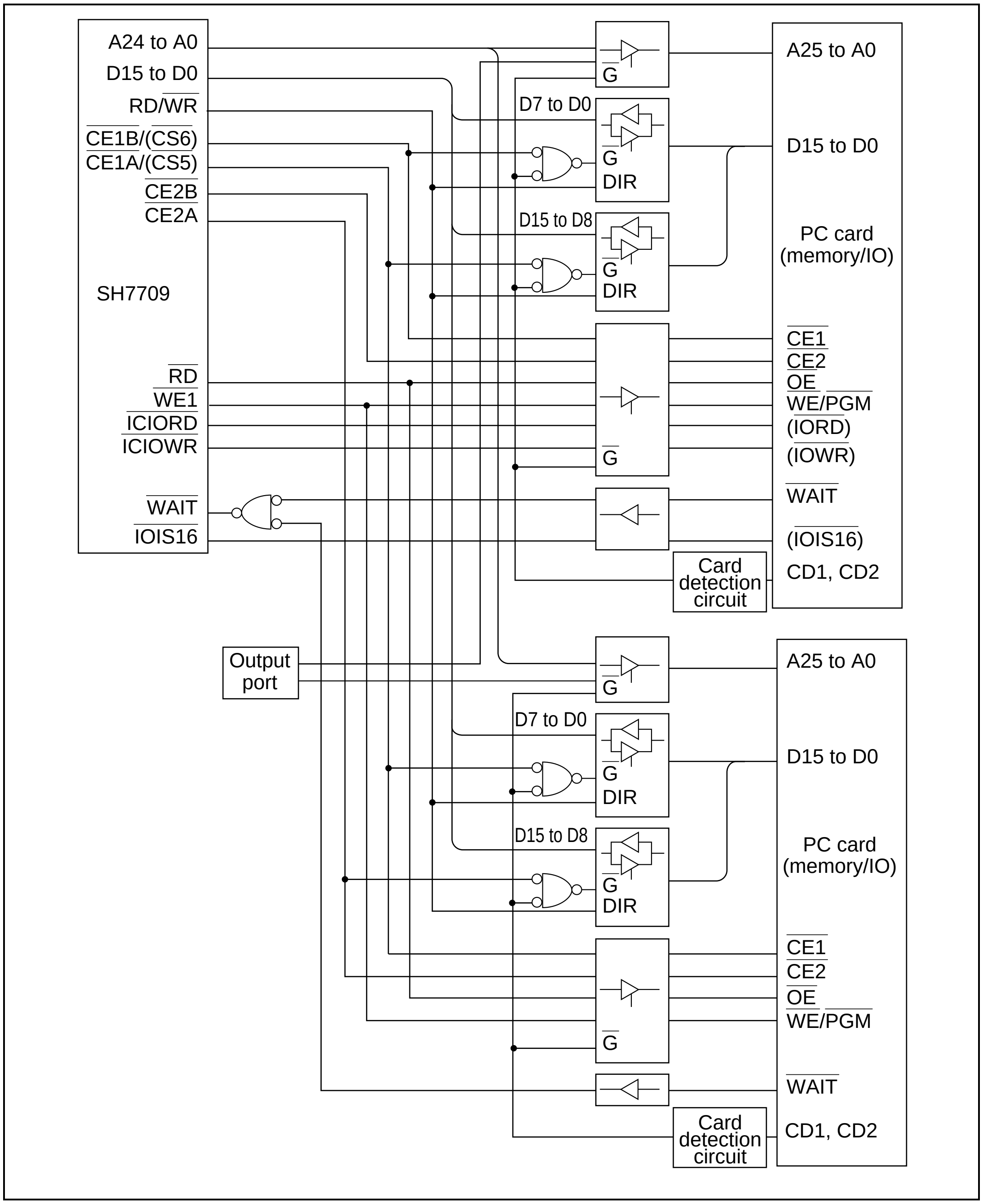


Figure 10.36 Example of PCMCIA Interface

Memory Card Interface Basic Timing: Figure 10.37 shows the basic timing for the PCMCIA IC memory card interface. When physical space areas 5 and 6 are designated as PCMCIA interface areas, bus accesses are automatically performed as IC memory card interface accesses when the lower address 32 Mbyte space of each area is accessed.

With a high external bus frequency (CKIO), the setup and hold times for the address (A24–A0), card enable ($\overline{\text{CS5}}$, $\overline{\text{CE2A}}$, $\overline{\text{CS6}}$, $\overline{\text{CE2B}}$), and write data (D15–D0) in a write cycle, become insufficient with respect to $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (the $\overline{\text{WE1}}$ pin in the SH7709). The SH7709 provides for this by enabling setup and hold times to be set for physical space areas 5 and 6 in the PCR register. Also, software waits by means of a WCR2 register setting and hardware waits by means of the $\overline{\text{WAIT}}$ pin can be inserted in the same way as for the basic interface. Figure 10.38 shows the PCMCIA memory bus wait timing.

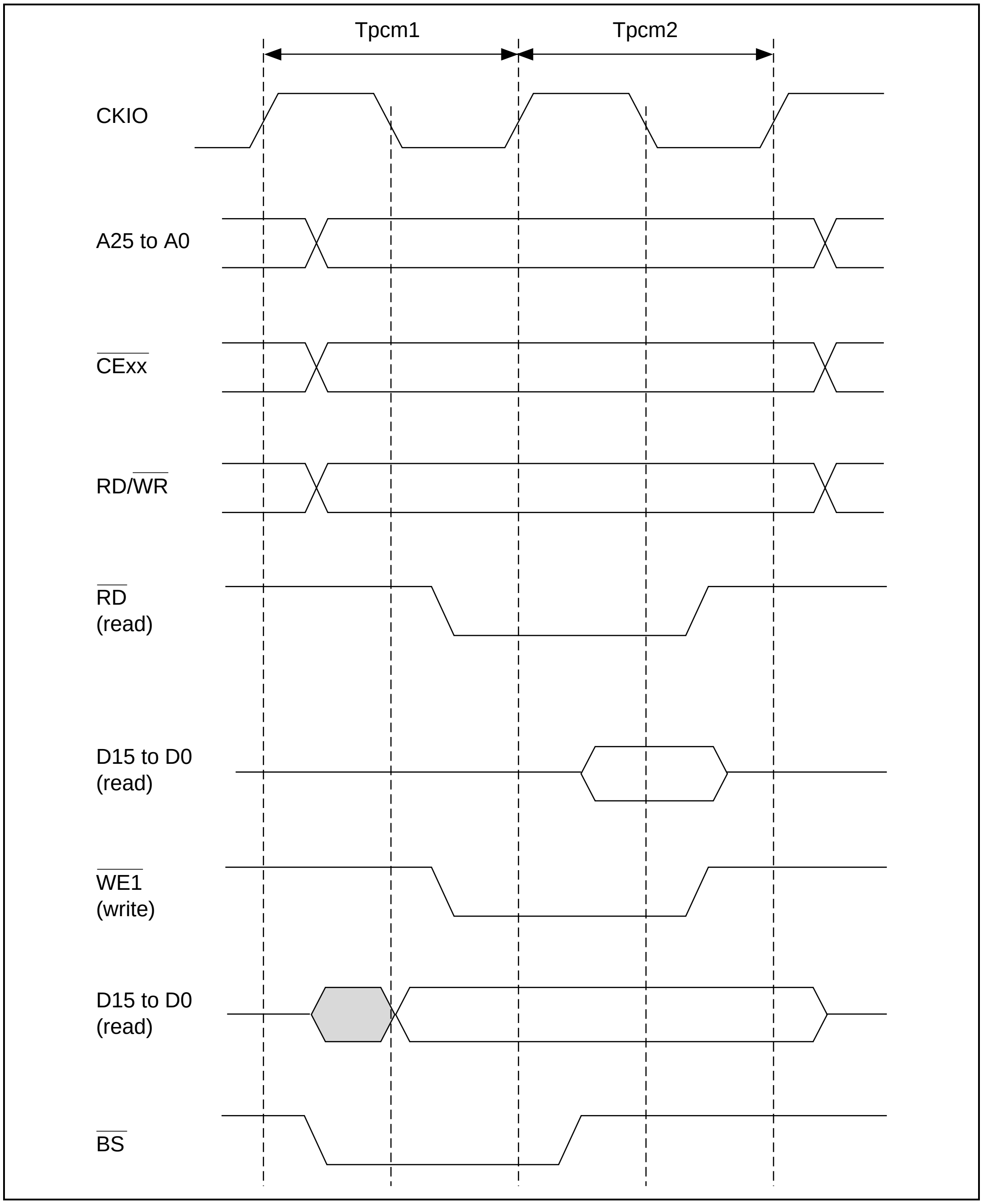


Figure 10.37 Basic Timing for PCMCIA Memory Card Interface

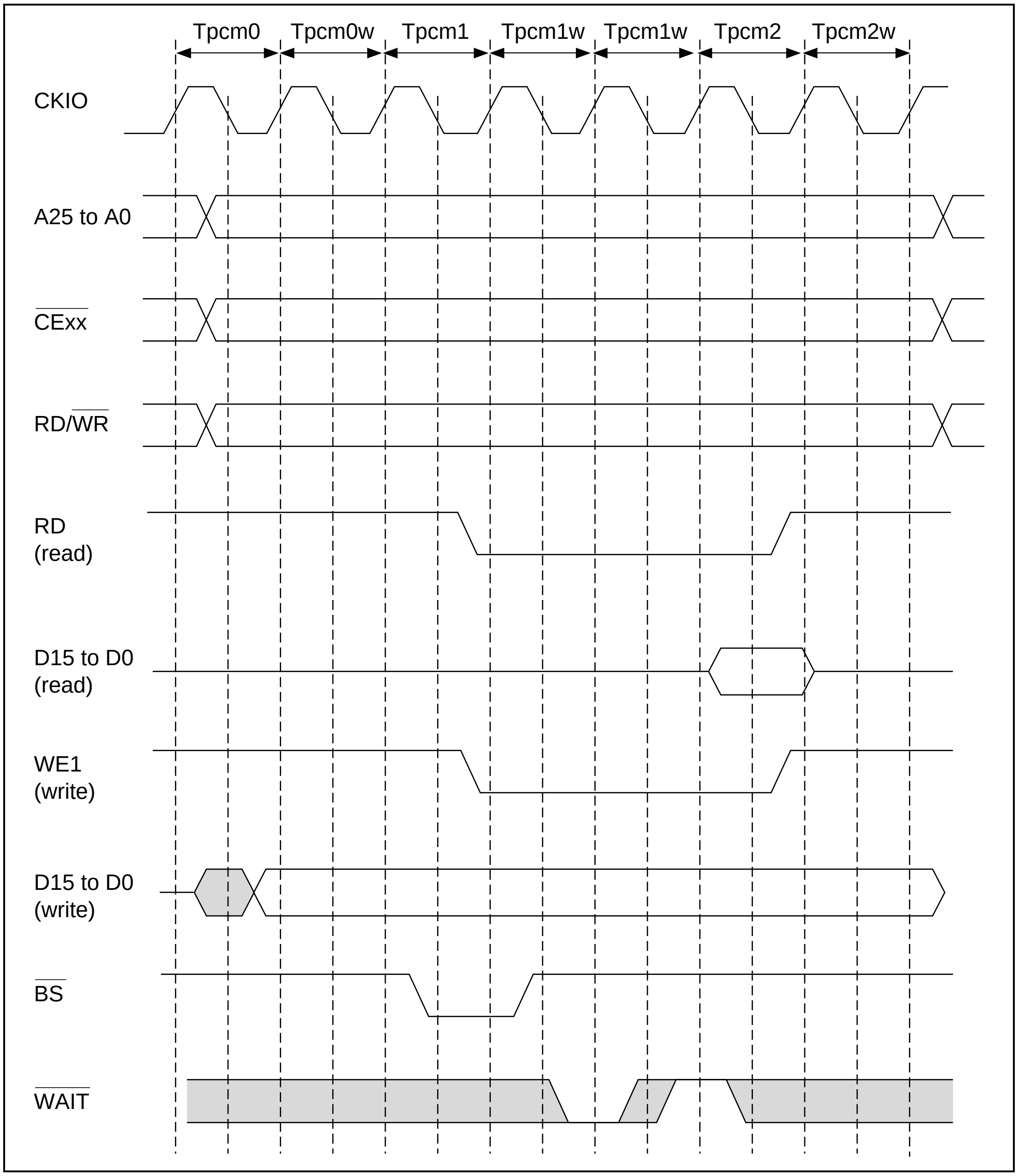


Figure 10.38 Wait Timing for PCMCIA Memory Card Interface

Memory Card Interface Burst Timing: In the SH7709, when the IC memory card interface is selected, page mode burst access mode can be used, for read access only, by setting bits A5BST1 and A5BST0 in BCR for physical space area 5, or bits A6BST1 and A6BST0 for area 6. This burst access mode is not stipulated in JEIDA version 4.2 (PCMCIA2.1), but allows high-speed data access using ROM provided with a burst mode, etc.

Burst access mode timing is shown in figures 10.39 and 10.40.

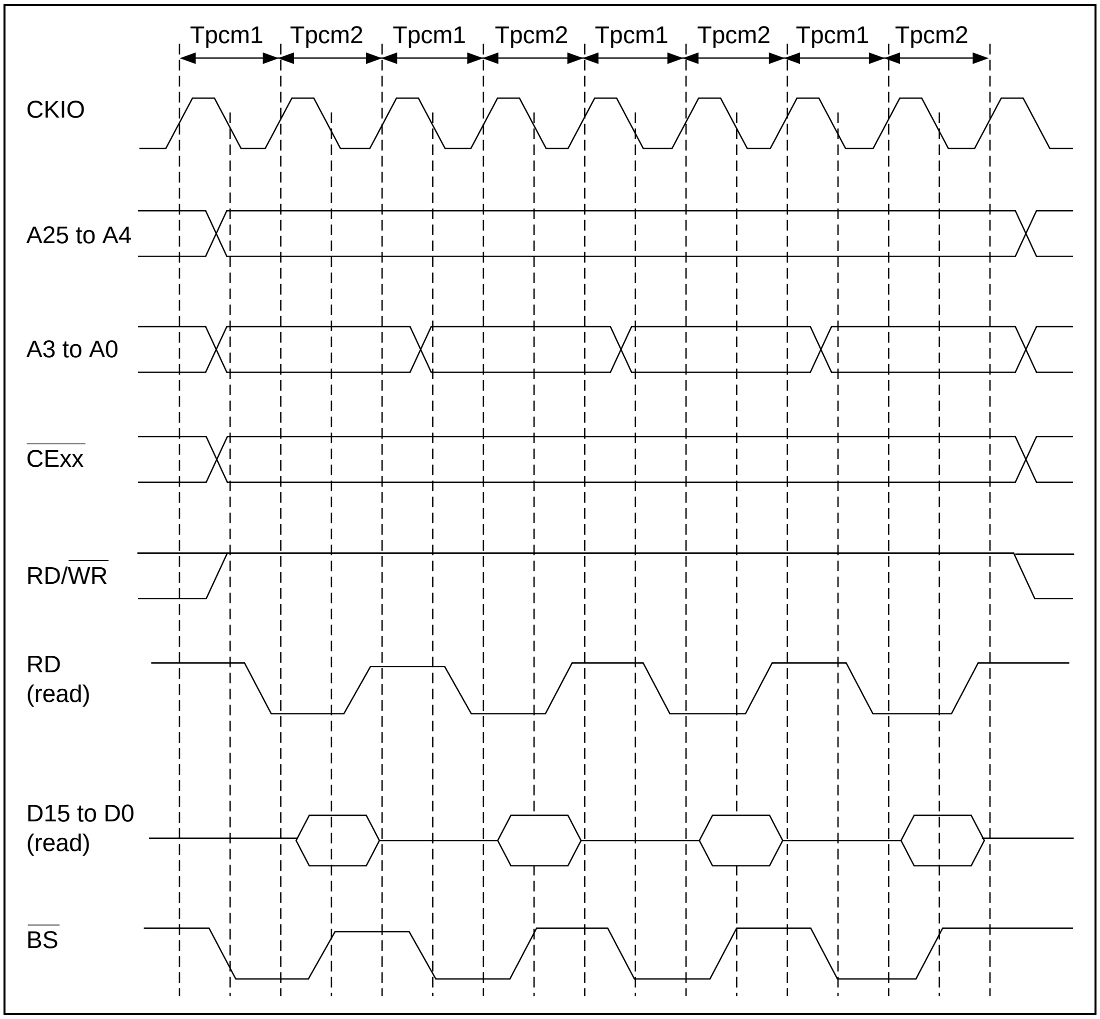


Figure 10.39 Basic Timing for PCMCIA Memory Card Interface Burst Access

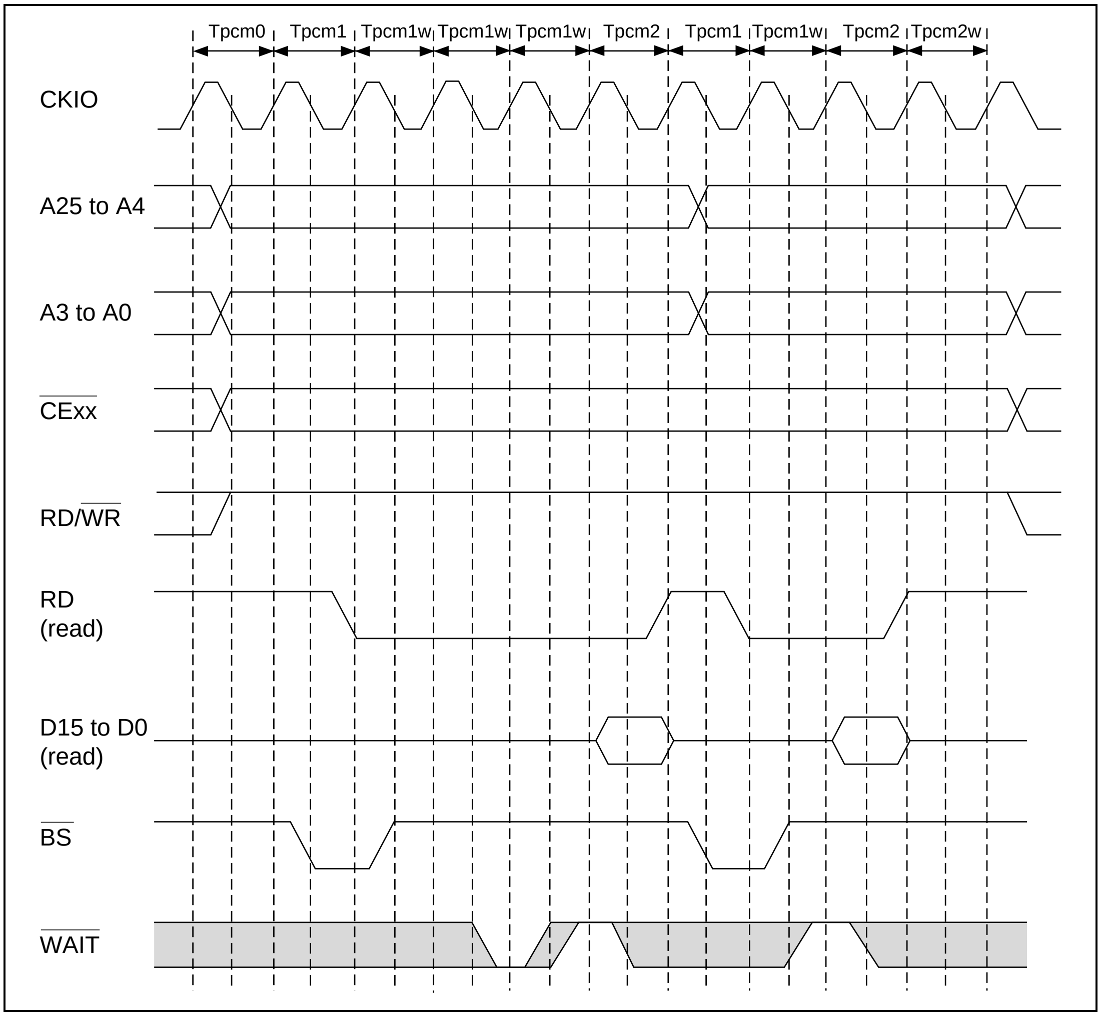


Figure 10.40 Wait Timing for PCMCIA Memory Card Interface Burst Access

When the entire 32-Mbyte memory space is used as IC memory card interface space, the common memory/attribute memory switching signal $\overline{\text{REG}}$ is generated using a port, etc. If 16-Mbytes or less of memory space is sufficient, using 16M bytes of memory space as common memory space and 16 Mbytes as attribute memory space enables the A24 pin to be used for the $\overline{\text{REG}}$ signal.

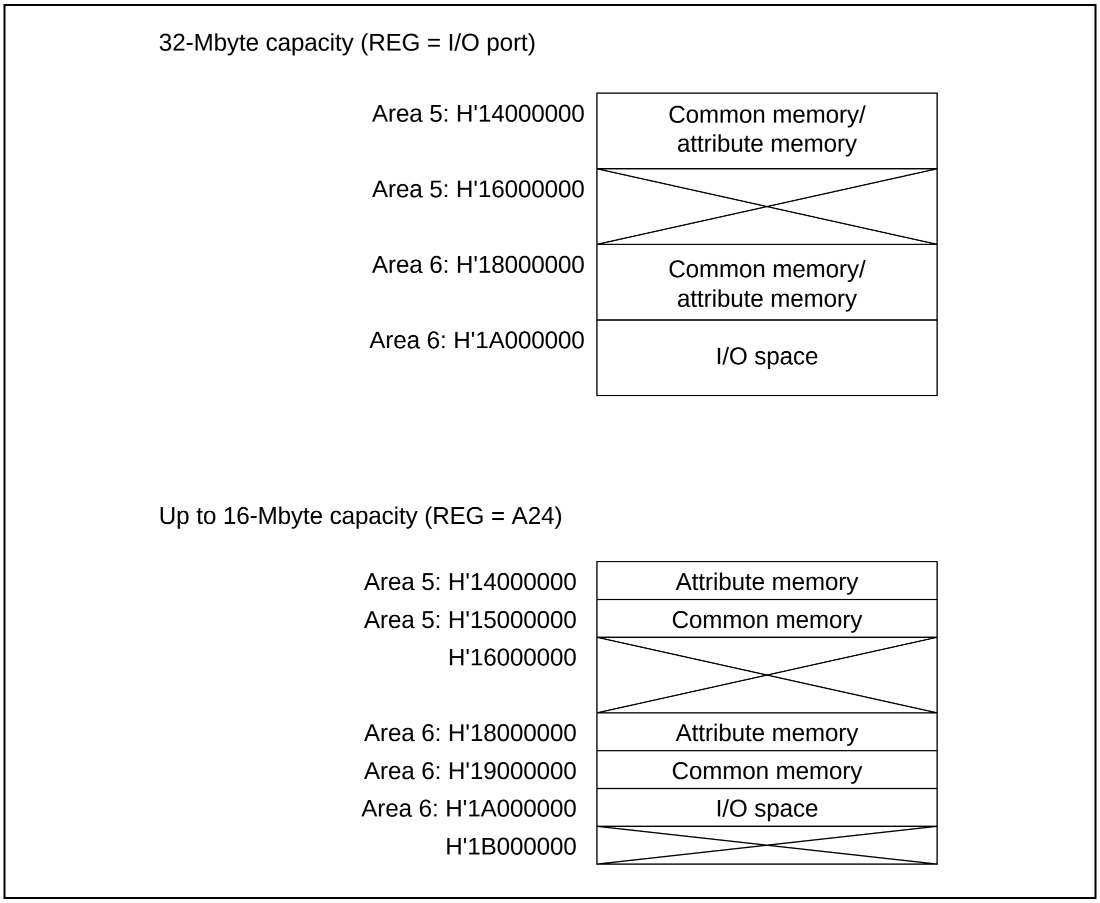


Figure 10.41 PCMCIA Space Allocation

I/O Card Interface Timing: Figures 10.42 and 10.43 show the timing for the PCMCIA I/O card interface.

The I/O card interface is supported only for physical space area 6. Switching between the I/O card interface and the IC memory card interface is performed according to the accessed address. When PCMCIA is designated for physical space area 6, the bus access is automatically performed as an I/O card interface access when a physical address from H'1A000000 to H'1BFFFFFF is accessed.

When accessing a PCMCIA I/O card, the access should be performed using a noncacheable area in virtual space (P2 or P3 space) or an area specified as noncacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, dynamic sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width is set for area 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port is recognized as being 8 bits in width. In this case, a data access for only 8 bits is performed in the I/O bus cycle being executed, followed automatically by a data access for the remaining 8 bits.

Figure 10.44 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported, and is ignored.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed low.

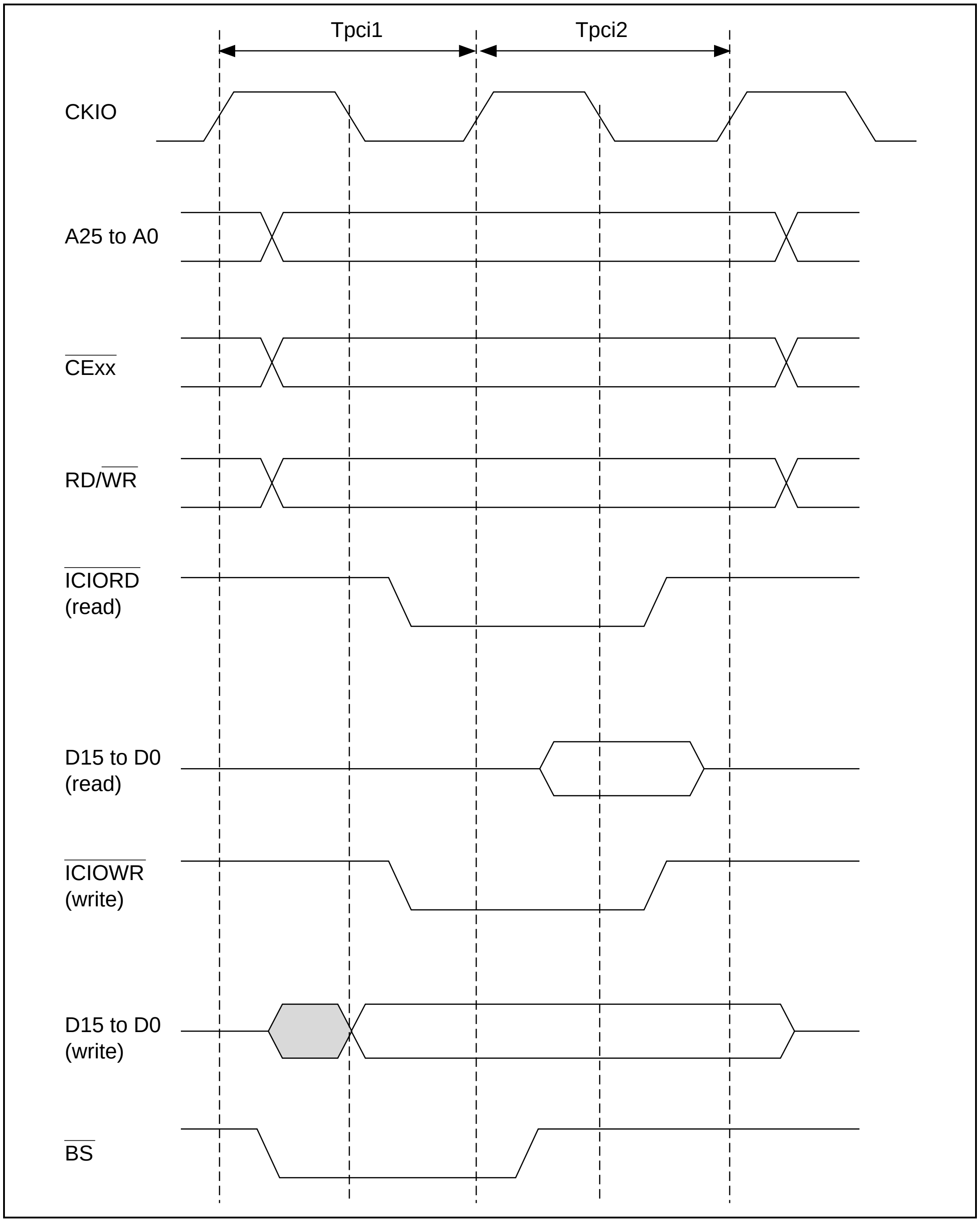


Figure 10.42 Basic Timing for PCMCIA I/O Card Interface

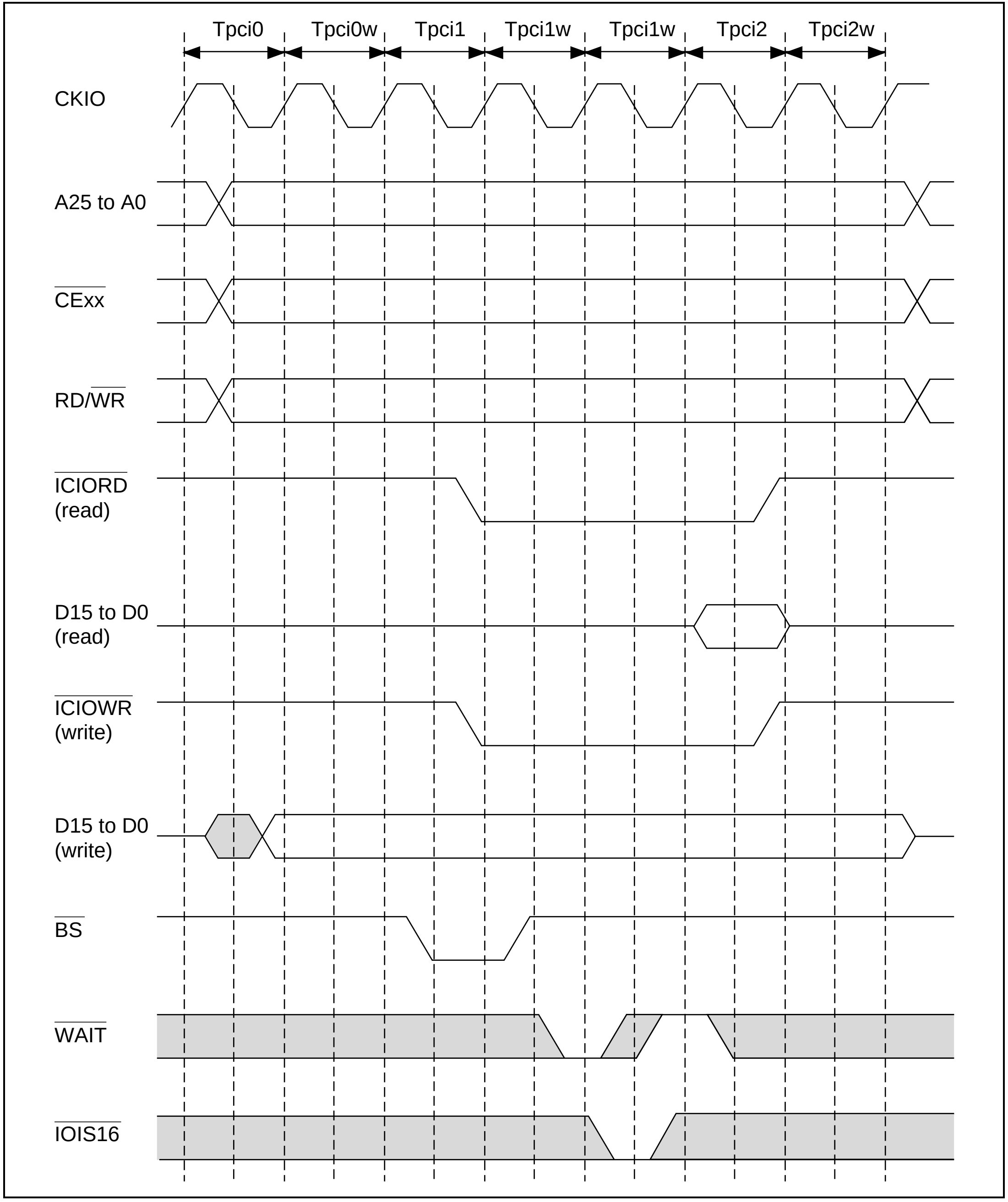


Figure 10.43 Wait Timing for PCMCIA I/O Card Interface

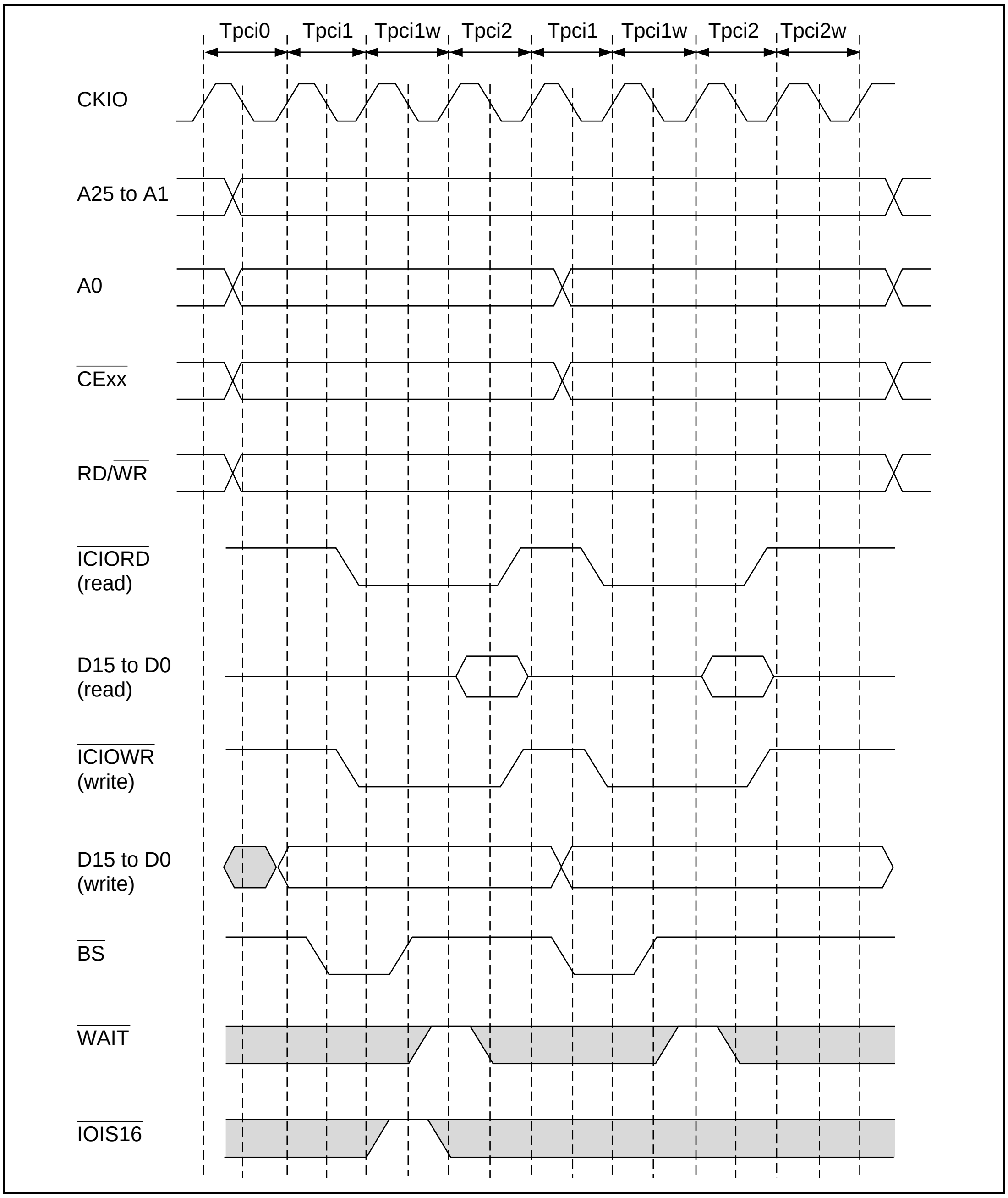


Figure 10.44 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

10.3.8 Waits between Access Cycles

A problem associated with higher external memory bus operating frequencies is that data buffer turn-off on completion of a read from a low-speed device may be too slow, causing a collision with data in the next access. This results in lower reliability or incorrect operation. To avoid this problem, a data collision prevention feature has been provided. This memorizes the preceding access area and the kind of read/write. If there is a possibility of a bus collision when the next access is started, a wait cycle is inserted before the access cycle thus preventing a data collision. There are two cases in which a wait cycle is inserted: when an access is followed by an access to a different area, and when a read access is followed by a write access from the SH7709. When the SH7709 performs consecutive write cycles, the data transfer direction is fixed (from the SH7709 to other memory) and there is no problem. With read accesses to the same area, in principle, data is output from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and AnIW0 (n = 0–6) in WCR1 specify the number of idle cycles to be inserted between access cycles when a physical space area access is followed by an access to another area, or when the SH7709 performs a write access after a read access to physical space area n. If there is originally space between accesses, the number of idle cycles inserted is the specified number of idle cycles minus the number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empty cycles are inserted for arbitration purposes.

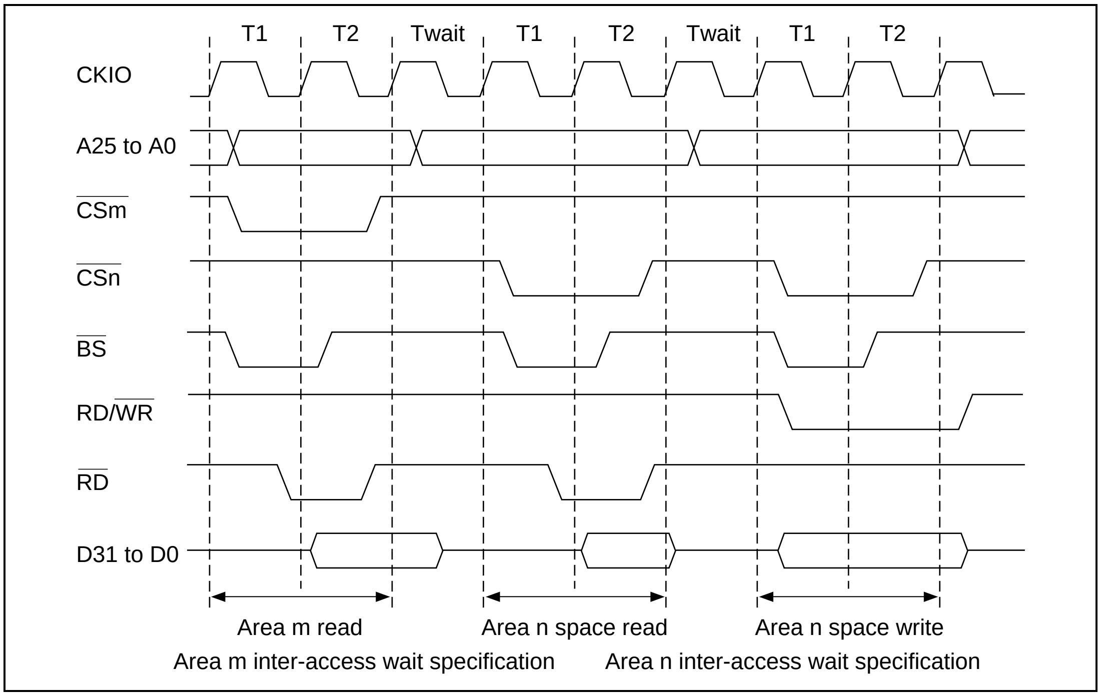


Figure 10.45 Waits between Access Cycles

10.3.9 Bus Arbitration

When a bus release request ($\overline{\mathrm{BREQ}}$) is received from an external device, buses are released after the bus cycle being executed is completed and a bus grant signal ($\overline{\mathrm{BACK}}$) is output. The bus is not released during burst transfers for cache fills. At the negation of $\overline{\mathrm{BREQ}}$, $\overline{\mathrm{BACK}}$ is negated and bus use is restarted. See Appendix B, Pin States, for the pin status when the bus is released.

The SH7709 sometimes needs to retrieve a bus it has released. For example, when memory generates a refresh request or an interrupt request internally, the SH7709 must perform the appropriate processing. The SH7709 has a bus request signal ($\overline{\mathrm{IRQOUT}}$) for this purpose. When it must retrieve the bus, it asserts the $\overline{\mathrm{IRQOUT}}$ signal. Devices asserting an external bus release request receive the assertion of the $\overline{\mathrm{IRQOUT}}$ signal and negate the $\overline{\mathrm{BREQ}}$ signal to release the bus. The SH7709 retrieves the bus and carries out the processing.

$\overline{\mathrm{IRQOUT}}$ Pin Assertion Conditions:

- When a memory refresh request has been generated but the refresh cycle has not yet begun
- When an interrupt is generated with an interrupt request level higher than the setting of the interrupt mask bits (I3–I0) in the status register (SR). (This does not depend on the SR.BL bit.)

Section 11 Direct Memory Access Controller (DMAC)

11.1 Overview

This chip includes a four-channel 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, memory-mapped external devices, and on-chip peripheral modules (IrDA, SCIF, A/D converter, D/A converter, and I/O ports). Using the DMAC reduces the burden on the CPU and increases overall operating efficiency.

11.1.1 Features

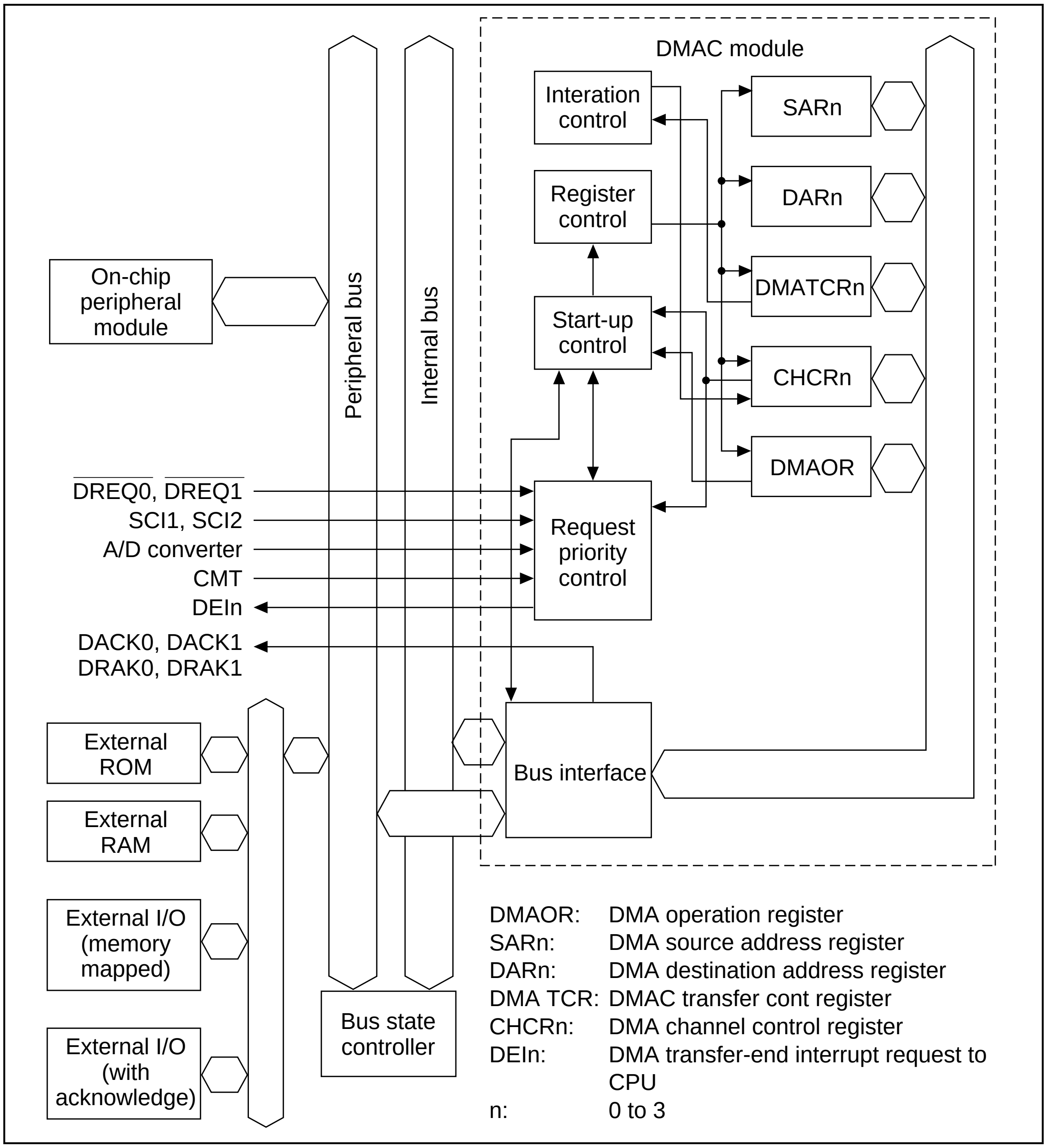
The DMAC has the following features.

- Four channels
- Physical address space
- 8-, 16-, or 32-bit data transfer unit
- 16 Mbytes (16777216 transfers)
- Address mode: Dual address mode is supported. In addition, direct address transfer mode or indirect address transfer mode can be selected.
  - Dual address mode transfer: Both the transfer source and transfer destination are accessed by address. Dual address mode has direct address transfer mode and indirect address transfer mode.
  - Direct address transfer mode: The values specified in the DMAC registers indicates the transfer source and transfer destination. 2 bus cycles are required for one data transfer. Indirect address transfer mode: Data is transferred with the address stored prior to the address specified in the transfer source address in the DMAC. Other operations are the same as those of direct address transfer mode. This function is only valid in channel 3.
- Channel functions: Transfer mode that can be specified is different in each channel.
  - Channel 0: External request can be accepted.
  - Channel 1: External request can be accepted.
  - Channel 2: This channel has a source address reload function, which reloads a source address for each 4 transfers.
  - Channel 3: In this channel, direct address mode or indirect address transfer mode can be specified.
- Reload function: The value that was specified in the source address register can be automatically reloaded for each 4 DMA transfers. This function is only valid in channel 2.

- Transfer requests
    - External request (From $\overline{\text{DREQ}}$ pins (channels 0 and 1 only). $\overline{\text{DREQ}}$ can be detected either by edge or by level)
    - On-chip module request (Requests from on-chip peripheral modules such as serial communications interface (SCI), A/D converter (A/D) and a timer (CMT). This request can be accepted in all the channels)
- Auto-request (the transfer request is generated automatically within the DMAC)
  Selectable bus modes: Cycle-steal mode or burst mode
  Selectable channel priority levels:
- Fixed mode: The channel priority is fixed.
  Round-robin mode: The priority of the channel in which the execution request was accepted is made the lowest.
- Interrupt request: An interrupt request can be generated to the CPU after transfers end by the specified counts.

11.1. 2 Block Diagram

Figure 11.1 is a block diagram of the DMAC.



DMAOR:	DMA operation register
SARn:	DMA source address register
DARn:	DMA destination address register
DMA TCR:	DMAC transfer cont register
CHCRn:	DMA channel control register
DEIn:	DMA transfer-end interrupt request to CPU
n:	0 to 3

Figure 11.1 DMAC Block Diagram

11.1.3 Pin Configuration

Table 11.1 shows the DMAC pins.

Table 11.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input from external device to channel 0
	DREQ acknowledge	DACK0	O	Strobe output to an external I/O at DMA transfer request from external device to channel 0
	DMA request acknowledge	DRAK0	O	Sampling acceptance output for DMA transfer request input from external device to channel 0
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input from external device to channel 1
	DREQ acknowledge	DACK1	O	Strobe output to an external I/O at DMA transfer request from external device to channel 1
	DMA request acknowledge	DRAK1	O	Sampling acceptance output for DMA transfer request input from external device to channel 1

11.1.4 Register Configuration

Table 11.2 summarizes the DMAC registers. DMAC has a total of 17 registers. Each channel has four control registers. One other control register is shared by all channels.

Table 11.2 DMAC Registers

Channel	Register Name	Abbr.	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'04000020	32 bits	16, 32*2
	DMA destination address register 0	DAR0	R/W	Undefined	H'04000024	32 bits	16, 32*2
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'04000028	24 bits	16, 32*3
	DMA channel control register 0	CHCR0	R/(W)*1	H'00000000	H'0400002C	32 bits	8, 16, 32*2
1	DMA source address register 1	SAR1	R/W	Undefined	H'04000030	32 bits	16, 32*2
	DMA destination address register 1	DAR1	R/W	Undefined	H'04000034	32 bits	16, 32*2
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'04000038	24 bits	16, 32*3
	DMA channel control register 1	CHCR1	R/(W)*1	H'00000000	H'0400003C	32 bits	8, 16, 32*2
2	DMA source address register 2	SAR2	R/W	Undefined	H'04000040	32 bits	16, 32*2
	DMA destination address register 2	DAR2	R/W	Undefined	H'04000044	32 bits	16, 32*2
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'04000048	24 bits	16, 32*3
	DMA channel control register 2	CHCR2	R/(W)*1	H'00000000	H'0400004C	32 bits	8, 16, 32*2
3	DMA source address register 3	SAR3	R/W	Undefined	H'04000050	32 bits	16, 32**2
	DMA destination address register 3	DAR3	R/W	Undefined	H'04000054	32 bits	16, 32*2
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'04000058	24 bits	16, 32*3
	DMA channel control register 3	CHCR3	R/(W)*1	H'00000000	H'0400005C	32 bits	8, 16, 32*2
Shared	DMA operation register	DMAOR	R/(W)*1	H'0000	H'04000060	16 bits	8, 16*2

Notes: 1. Only 0s can be written to bits 1 of CHCR0 to CHCR3, and bits 1 and 2 of DMAOR to clear flag after 1 is read.

2. If SAR0 to SAR3, DAR0 to DAR3, and CHCR0 to CHCR3 are accessed in 16 bits, the value in 16 bits that were not accessed are held.

3. DMATCR has 24 bits from DMATCR0 to DMATCR23. Therefore, even if 1s are written to upper 24 to 31 bits, it is invalid; 0s are always read if these bits are read.

4. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

11.2 Register Descriptions

11.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3)

DMA source address registers 0–3 (SAR0–SAR3) are 32-bit read/write 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 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. Specifying other addresses does not guarantee operation.

The initial value after resets or in hardware or software standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	…	0
Bit name:					…	
Initial value:	—	—	—	—	…	—
R/W:	R/W	R/W	R/W	R/W	…	R/W

11.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers include count functions, and during a DMA transfer, these registers indicate the next destination address.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address boundary. Specifying other addresses does not guarantee operation.

The initial value after resets or in hardware or software standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20		…		0
Bit name:						…		
Initial value:	—	—	—	—		…		—
R/W:	R/W	R/W	R/W	R/W		…		R/W

11.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit read/write registers that specify the DMA transfer count (bytes, words, or longwords). The number of transfers is 1 when the setting is H'000001, and 16777216 (the maximum) when H'000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

Writing to upper eight bits in DMATCR is invalid; 0s are read if these bits are read.

The initial value after resets or in hardware standby mode or software standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	...	0
Bit name:					...	
Initial value:	—	—	—	—	...	—
R/W:	R/W	R/W	R/W	R/W	...	R/W

11.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)

DMA channel control registers 0–3 (CHCR0–CHCR3) are 32-bit read/write registers that specifies operation mode, transfer method, or others in each channel. Writing to bits 31 to 24 and 7 in this register is invalid; 0 s are read if these bits are read.

Bit 20 is only used in CHCR3. it is not used in CHCR0 to CHCR2. Consequently, writing to this bit is invalid in CHCR0 to CHCR2; 0 is read if this bit is read. Bit 19 is only used in CHCR2; it is not used in CHCR0, CHCR1, and CHCR3. Consequently, writing to this bit is invalid in CHCR0, CHCR1, and CHCR3; 0 is read if this bit is read. Bits 6 and 16 to 18 are only used in CHCR0 and CHCR1; they are not used in CHCR2 and CHCR3. Consequently, writing to these bits is invalid in CHCR2 and CHCR3; 0s are read if these bits are read.

These register values are initialized to 0s after power-on resets or in software standby mode or in hardware standby mode.

Bit:	31	...	21	20	19	18	17	16
Bit name:	—	...	—	DI	RO	RL	AM	AL
Initial value:	0	...	0	0	0	0	0	0
R/W:	R	...	R	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)

Bit:	15	14	13	12	11	10	9	8
Bit name:	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
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:	7	6	5	4	3	2	1	0
Bit name:	—	DS	TM	TS1	TS0	IE	TE	DE
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

Notes: 1. Only 0 can be written to the TE bit after 1 is read.
2. DI, RO, RL, AM, AL, and DS bits are not included in some channels.

Bits 31 to 21—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 20—direct/indirect selection (DI): DI selects direct address mode or indirect address mode in channel 3.

This bit is only valid in CHCR3.   Writing to this bit is invalid in CHCR0 to CHCR2; 0 is read if this bit is read.

Bit 20: DI	Description
0	Direct address mode (initial value)
1	Indirect address mode

Bit 19—source address reload bit (RO): RO selects whether the source address initial value is reloaded in channel 2.

This bit is only valid in CHCR2. Writing to this bit is invalid in CHCR0, CHCR1, and CHCR3; 0 is read if this bit is read.

Bit 19: RO	Description
0	A source address is not reloaded (initial value)
1	A source address is reloaded

Bit 18—request check level bit (RL): RL specifies the DRAK (acknowledge of DREQ) signal output is high active or low active.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is read if this bit is read.

Bit 18: RL	Description
0	Low-active output of DRAK (initial value)
1	High-active output of DRAK

Bit 17—acknowledge mode bit (AM): AM specifies whether DACK is output in data read cycle or in data write cycle in dual address mode.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is read if this bit is read.

Bit 17: AM	Description
0	DACK output in read cycle (initial value)
1	DACK output in write cycle

Bit 16—acknowledge level (AL): AL specifies the DACK (acknowledge) signal output is high active or low active.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is read if this bit is read.

Bit 16: AL	Description
0	Low-active output of DACK (initial value)
1	High-active output of DACK

Bits 15 and 14—destination address mode bits 1, 0 (DM1 and DM0): DM1 and DM0 select whether the DMA destination address is incremented, decremented, or left fixed.

Bit 15: DM1	Bit 14: DM0	Description
0	0	Fixed destination address (initial value)
0	1	Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, and +4 in 32-bit transfer)
1	0	Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, and –4 in 32-bit transfer)
1	1	Illegal setting

Bits 13 and 12—source address mode bits 1, 0 (SM1 and SM0): SM1 and SM0 select whether the DMA source address is incremented, decremented, or left fixed.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Fixed source address (initial value)
0	1	Source address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, and +4 in 32-bit transfer)
1	0	Source address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, and –4 in 32-bit transfer)
1	1	Reserved (illegal setting)

If the transfer source is specified in indirect address, specify the address, in which the data to be transferred is stored and which is stored as data (indirect address), in source address register 3 (SAR3).

Specification of SAR3 increment or decrement in indirect address mode depends on SM1 and SM0 settings. In this case, however, the SAR3 increment or decrement value is +4, –4, or fixed to 0 regardless of the transfer data size specified in TS1 and TS0.

Bits 11–8—resource select bits 3–0 (RS3–RS0): RS3–RS0 specify which transfer requests will be sent to the DMAC.

Bit 11:	Bit 10:	Bit 9:	Bit 8:	
RS3	**RS2**	**RS1**	**RS0**	**Description**
0	0	0	0	External request*, dual address mode (initial value)
0	0	0	1	Illegal setting
0	0	1	0	Illegal setting
0	0	1	1	Illegal setting
0	1	0	0	Auto-request
0	1	0	1	Illegal setting
0	1	1	0	Illegal setting
0	1	1	1	Illegal setting
1	0	0	0	Illegal setting
1	0	0	1	Illegal setting
1	0	1	0	IrDA transmission
1	0	1	1	IrDA reception
1	1	0	0	SCIF transmission
1	1	0	1	SCIF reception
1	1	1	0	Internal A/D
1	1	1	1	CMT

Note: External request specification is valid only in channels 0 and 1. None of the request sources can be selected in channels 2 and 3.

Bit 6—$\overline{\text{DREQ}}$ select bit (DS): DS selects the sampling method of the $\overline{\text{DREQ}}$ pin that is used in external request mode is detection in low level or at the falling edge.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is read if this bit is read.

In channel 0 and 1, if an on-chip peripheral module is specified as a transfer request source or an auto-request is specified, specification of this bit is ignored and detection at the falling edge is fixed except in an auto-request.

Bit 6: DS	Description
0	$\overline{\text{DREQ}}$ detected in low level (initial value)
1	$\overline{\text{DREQ}}$ detected at falling edge

Bit 5—transmit mode (TM): TM specifies the bus mode when transferring data.

Bit 5: TM	Description
0	Cycle steal mode (initial value)
1	Burst mode

Bits 4 and 3—transmit size bits 1 and 0 (TS1, TS0): TS1 and TS0 specifies the size of data to be transferred.

Bit 4: TS1	Bit 3: TS0	Description
0	0	Byte size (8 bits) (initial value)
0	1	Word size (16 bits)
1	0	Longword size (32 bits)
1	1	Illegal setting

Bit 2—interrupt enable bit (IE): Setting this bit to 1 generates an interrupt request when data transfer end (TE = 1) by the count specified in DMATCR.

Bit 2: IE	Description
0	Interrupt request is not generated even if data transfer ends by the specified count (initial value)
1	Interrupt request is generated if data transfer ends by the specified count

Bit 1—transfer end bit (TE): TE is set to 1 when data transfer ends by the count specified in DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

Before this bit is set to 1, if data transfer ends due to an NMI interrupt or clearing the DE bit or the DME bit in DMAOR, this bit is not set to 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled.

Bit 1: TE	Description
0	Data transfer does not end by the count specified in DMATCR (initial value) Clear condition: Writing 0 after TE = 1 read, power-on reset, manual reset, standby
1	Data transfer ends by the specified count

Bit 0—DMAC enable bit (DE): DE enables channel operation.

Bit 0: DE	Description
0	Disables channel operation (initial value)
1	Enables channel operation

If an auto-request is specifies (specified in RS3 to RS0), transfer starts when this bit is set to 1. In an external request or an internal module request, transfer starts if transfer request is generated after this bit is set to 1. Clearing this bit during transfer can terminate transfer.

Even if the DE bit is set, transfer is not enabled if the TE bit is 1, the DME bit in DMAOR is 0, or the NMIF bit in DMAOR is 1.

11.2.5 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit read/write register that controls the DMAC transfer mode. Writing to bits 15 to 10 and bits 7 to 2 is invalid in this register; 0 is always read if these bits are read.

It is initialized to 0 at power-on reset, or in hardware standby mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	——	—	—	—	—	—	PR1	PR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	—	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/W

Note: Only 0 can be written to the NMIF bits after 1 is read.

Bits 9 and 8—priority mode bits 1 and 0 (PR1 and PR0): PR1 and PR0 select the priority level between channels when there are transfer requests for multiple channels simultaneously.

Bit 9: PR1	Bit 8: PR0	Description
0	0	CH0 > CH1 > CH2 > CH3 (initial value)
0	1	CH0 > CH2 > CH3 > CH1
1	0	CH2 > CH0 > CH1 > CH3
1	1	Round-robin

Bit 1—NMI Flag Bit (NMIF): NMIF indicates that an NMI interrupt occurred. This bit is set regardless of whether DMAC is in operating or halt state. The CPU cannot write 1 to this bit. Only 0 can be written to clear this bit after 1 is read.

Bit 1: NMIF	Description
0	No NMI input. DMA transfer is enabled. (initial value) To clear the NMIF bit, read 1 from it and then write 0.
1	NMI input. DMA transfer is disabled. This bit is set by occurrence of an NMI interrupt.

Bit 0—DMA master enable bit (DME): DME enables or disables DMA transfers on all channels. If the DME bit and the DE bit corresponding to each channel in CHCR are set to 1s, transfer is enabled in the corresponding channel. If this bit is cleared during transfer, transfers in all the channels can be terminated.

Even if the DME bit is set, transfer is not enabled if the TE bit is 1 or the DE bit is 0 in CHCR, or the NMIF bit is 1 in DMAOR.

Bit 0: DME	Description
0	Disable DMA transfers on all channels (initial value)
1	Enable DMA transfers on all channels

11.3 Operation

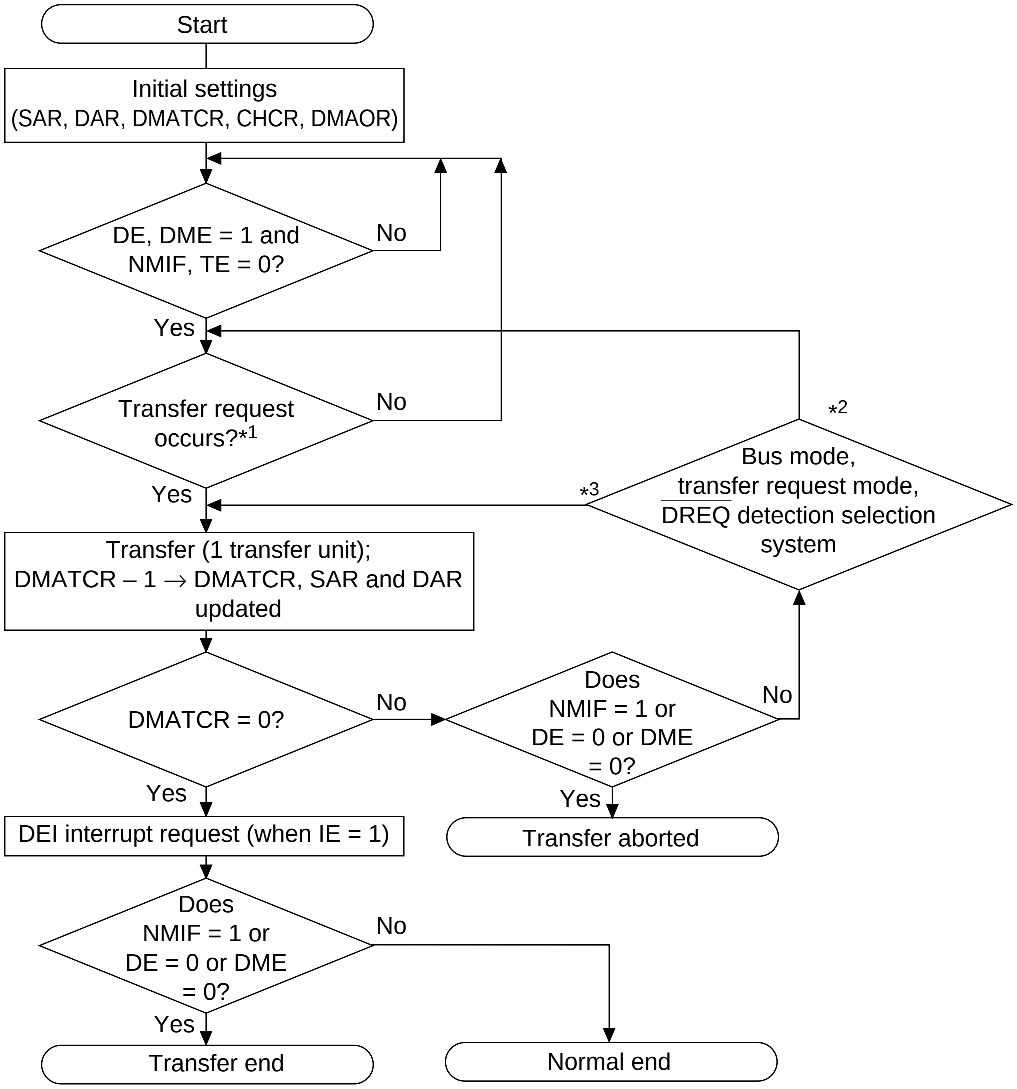
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto-request, external request, and on-chip module request. The dual address mode has direct address transfer mode and indirect address transfer mode. In the bus mode, the burst mode or the cycle steal mode can be selected.

11.3.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR) 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, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers 1 transfer unit of data (depending on the TS0 and TS1 settings). For an auto-request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfer have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an NMI interrupt is generated, the transfer is aborted. Transfers are also aborted when the DE bit of the CHCR or the DME bit of the DMAOR are changed to 0.

Figure 11.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when NMIF and TE are all 0 and the DE and DME bits are set to 1.
2. DREQ = level detection in burst mode (external request) or cycle-steal mode.
3. DREQ = edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 11.2   DMA Transfer Flowchart

11.3.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by devices and on-chip peripheral modules that are neither the source nor the destination. Transfers can be requested in three modes: auto-request, external request, and on-chip module request. The request mode is selected in the RS3–RS0 bits of the DMA channel control registers 0–3 (CHCR0–CHCR3).

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR0–CHCR3 and the DME bit of the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR0–CHCR3 and the NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{\text{DREQ}}$) of an external device. Choose one of the modes shown in table 11.3 according to the application system. When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0), a transfer is performed upon a request at the $\overline{\text{DREQ}}$ input. Choose to detect $\overline{\text{DREQ}}$ by either the falling edge or low level of the signal input with the DS bit of CHCR0–CHCR3 (DS = 0 is level detection, DS = 1 is edge detection). The source of the transfer request does not have to be the data transfer source or destination.

Table 11.3 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Any*	Any*

Note: External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, UBC, and BSC)

On-Chip Module Request: In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip module. The transfer request signals include 6 signals: the receive data full interrupts (RXI) and the transmit data empty interrupts (TXI) from two serial communication interfaces (IrDA, SCIF), the A/D conversion end interrupt (ADI) of the A/D converter and the compare match timer interrupt (CMI) of the CMT (table 11.4). When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0), a transfer is performed upon the input of a transfer request signal. The source of the transfer request does not have to be the data transfer source or destination. When RXI is set as the transfer request, however, the transfer source must be the SCI's receive data register (RDR). Likewise, when TXI is set as the transfer request, the transfer source must be the SCI's transmit data register (TDR). And if the transfer requester is the A/D converter, the data transfer source must be the A/D converter register.

Table 11.4 Selecting On-Chip Peripheral Module Request Modes with the RS Bit

RS3	RS2	RS1	RS0	DMA Transfer Request Source	DMA Transfer Request Signal	Source	Desti-nation	Bus Mode
1	0	1	0	IrDA transmitter	TXI1 (IrDA transmit data empty interrupt transfer request)	Any*	TDR1	Burst/ cycle steal
1	0	1	1	IrDA receiver	RXI1 (IrDA receive data full interrupt transfer request)	RDR1	Any*	Burst/ cycle steal
1	1	0	0	SCIF transmitter	TXI2 (SCIF transmit data empty interrupt transfer request)	Any*	TDR2	Burst/ cycle steal
1	1	0	1	SCIF receiver	RXI2 (SCIF receive data full interrupt transfer request)	RDR1	Any*	Burst/ cycle steal
1	1	1	0	A/D converter	ADI (A/D conversion end interrupt)	ADDR	Any*	Burst/ cycle steal
1	1	1	1	CMT	CMI (Compare match timer interrupt)	Any*	Any*	Burst/ cycle steal

SCI1 and SCI2: Serial communications interface channels 1 and 2

ADDR: A/D data register of A/D converter

Note: External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, BSC, UBC)

When outputting transfer requests from on-chip peripheral modules, the appropriate interrupt enable bits must be set to output the interrupt signals.

If the interrupt request signal of the on-chip peripheral module is used as a DMA transfer request signal, an interrupt is not generated to the CPU.

The DMA transfer request signals of table 11.4 are automatically withdrawn when the corresponding DMA transfer is performed. If the cycle steal mode is being employed, they are withdrawn at the first transfer; if the burst mode is being used, they are withdrawn at the last transfer.

11.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Two modes (fixed mode and round-robin mode) are selected by the priority bits PR1 and PR0 in the DMA operation register.

Fixed Mode: In these modes, the priority levels among the channels remain fixed. There are three kinds of fixed modes as follows:

CH0 > CH1 > CH2 > CH3
CH0 > CH2 > CH3 > CH1
CH2 > CH0 > CH1 > CH3

These are selected by the PR1 and the PR0 bits in the DMA operation register (DMAOR).

Round-Robin Mode: Each time one word, byte, or longword is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority order. The round-robin mode operation is shown in figure 11.3. The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after reset.

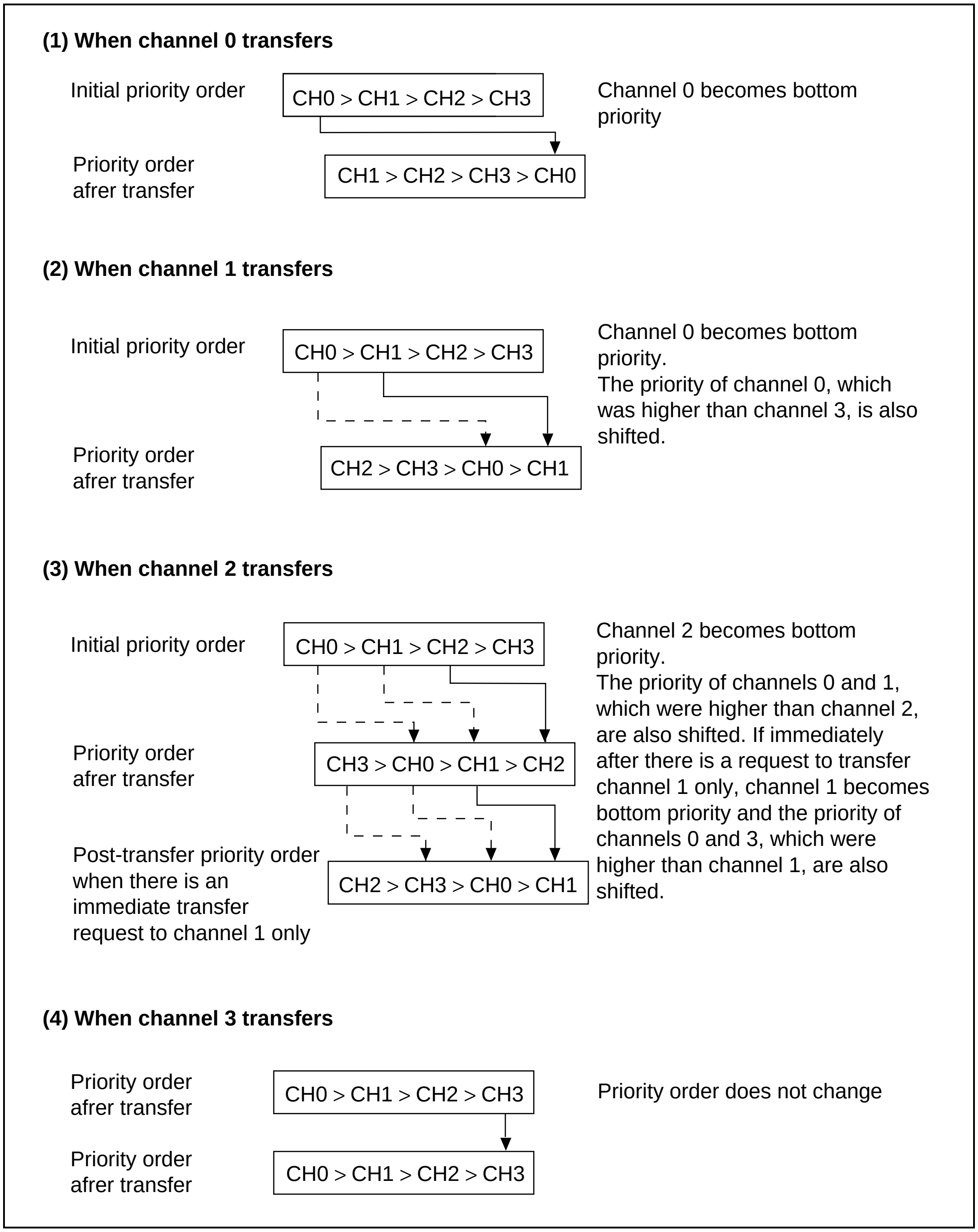

(1) When channel 0 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3
Channel 0 becomes bottom priority
Priority order afrer transfer
CH1 > CH2 > CH3 > CH0
(2) When channel 1 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3
Channel 0 becomes bottom priority.
The priority of channel 0, which was higher than channel 3, is also shifted.
Priority order afrer transfer
CH2 > CH3 > CH0 > CH1
(3) When channel 2 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3
Channel 2 becomes bottom priority.
The priority of channels 0 and 1, which were higher than channel 2, are also shifted. If immediately after there is a request to transfer channel 1 only, channel 1 becomes bottom priority and the priority of channels 0 and 3, which were higher than channel 1, are also shifted.
Priority order afrer transfer
CH3 > CH0 > CH1 > CH2
Post-transfer priority order when there is an immediate transfer request to channel 1 only
CH2 > CH3 > CH0 > CH1
(4) When channel 3 transfers
Priority order afrer transfer
CH0 > CH1 > CH2 > CH3
Priority order does not change
Priority order afrer transfer
CH0 > CH1 > CH2 > CH3

Figure 11.3 Round-Robin Mode

Figure 11.4 shows how the priority order changes when channel 0 and channel 1 transfers are requested simultaneously and a channel 3 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 1 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.

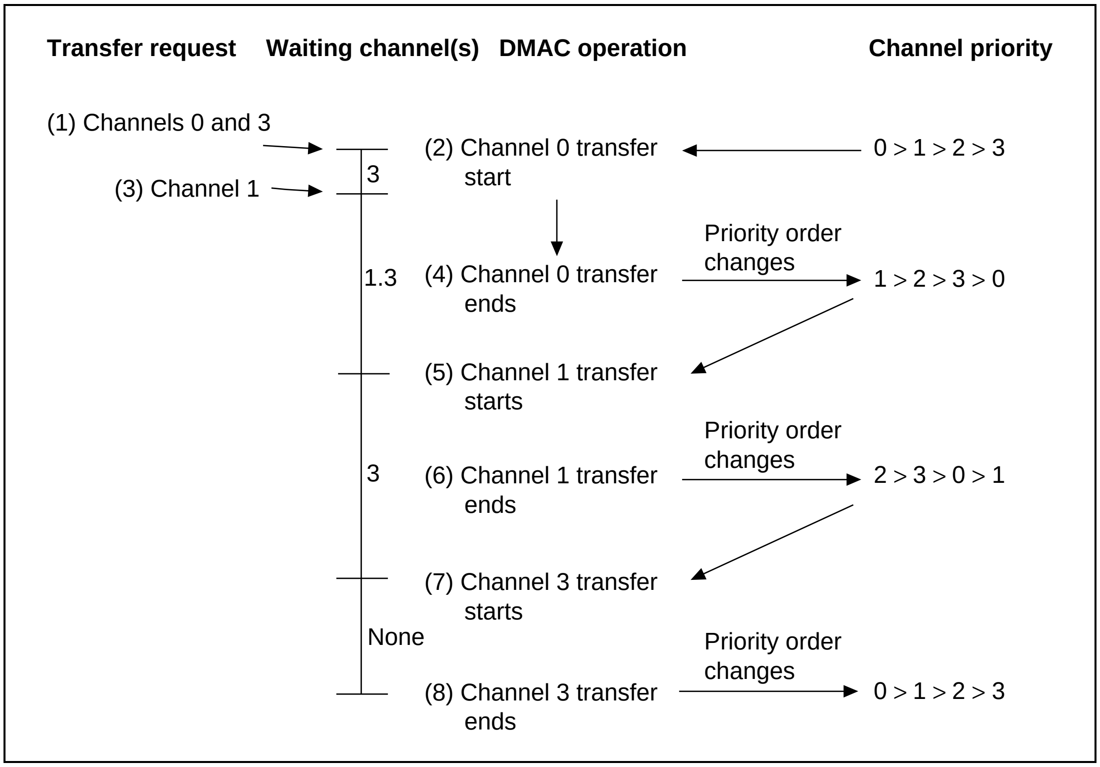


Figure 11.4 Changes in Channel Priority in Round-Robin Mode

11.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 11.5. In the dual address mode, both the transfer source address and the transfer destination address are output. The dual address mode has the direct address mode and the indirect address mode. In the direct address mode, an output address value is the data transfer target address; in the indirect address mode, the value stored in the output address, not the output address value itself, is the data transfer target address. A data transfer timing depends on the bus mode, which has cycle steal mode and burst mode.

Table 11.5 Supported DMA Transfers

	Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**
External device with DACK	Not available	Dual	Dual	Not available
External memory	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode

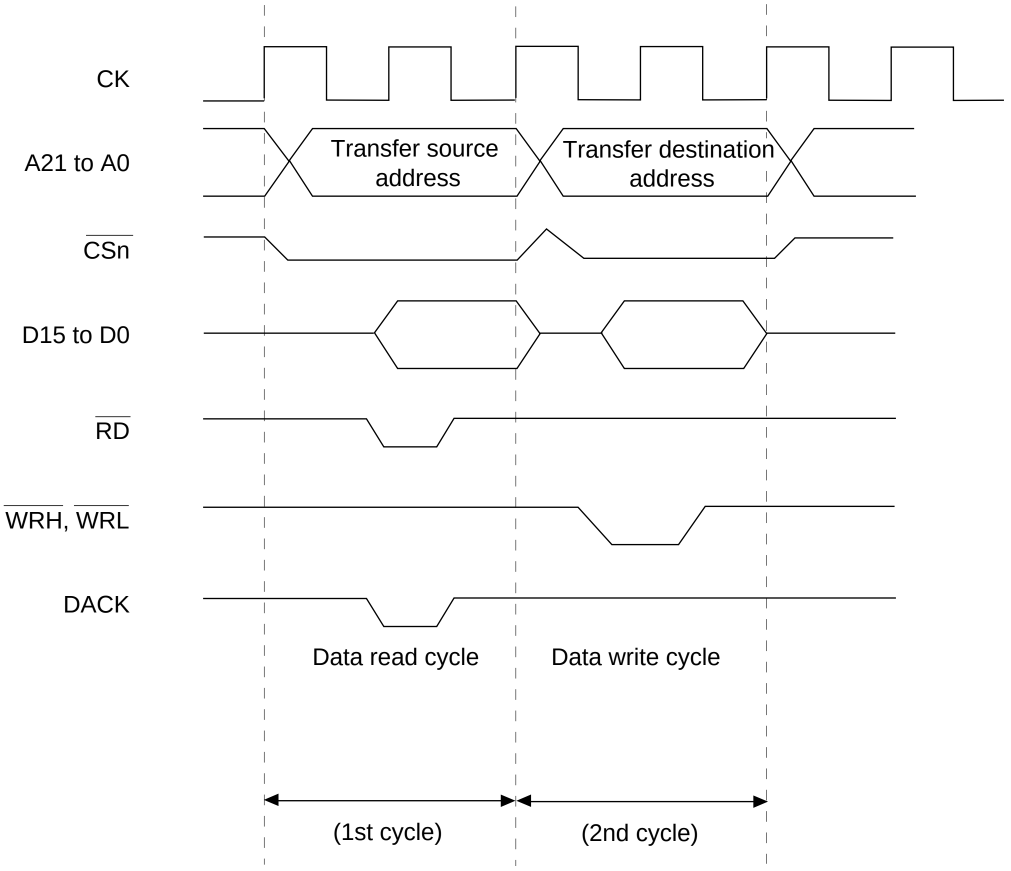
2. The dual address mode includes the direct address mode and the indirect address mode.

Address Modes:

- Dual Address Mode

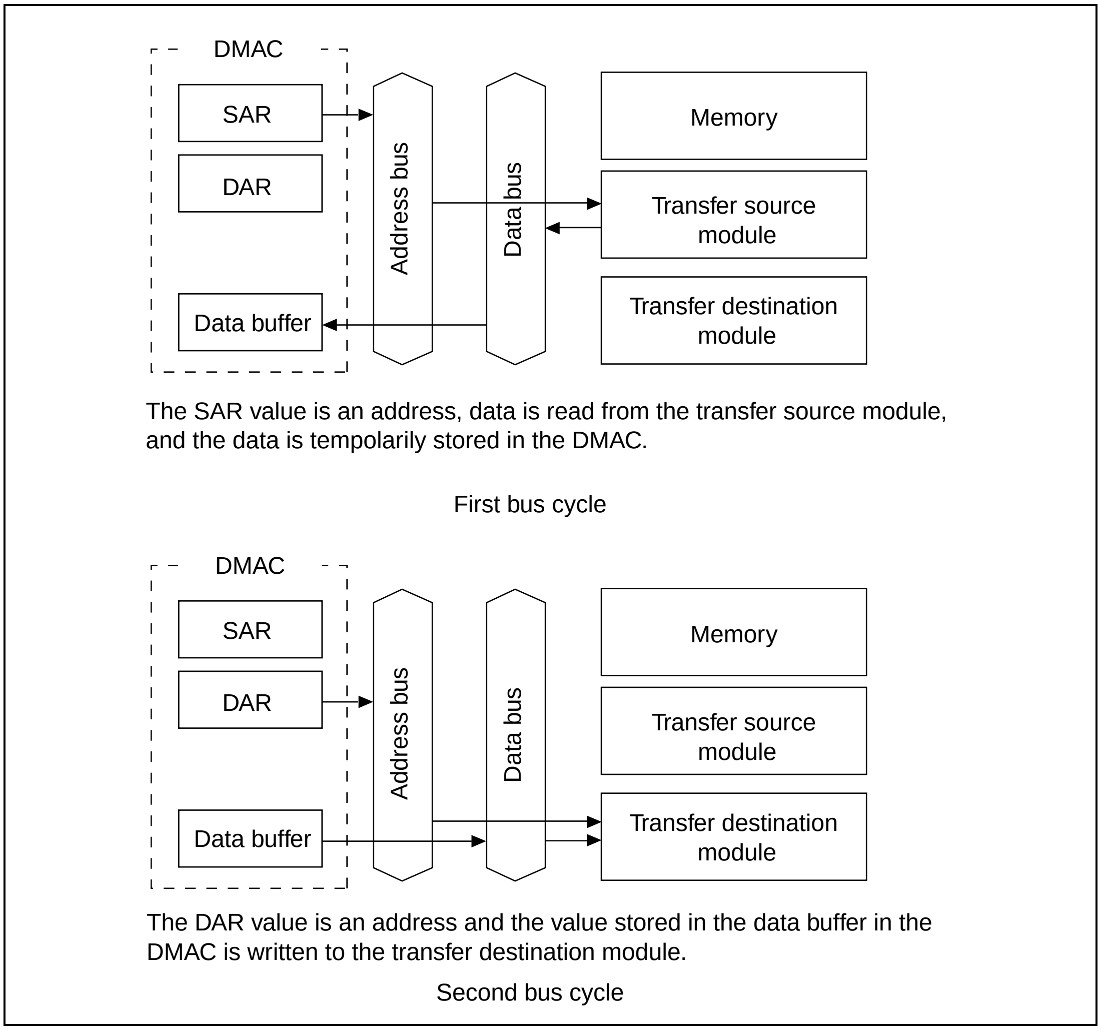
  In the dual address mode, both the transfer source and destination are accessed (selectable) by an address. The source and destination can be located externally or internally. The dual address mode has (1) direct address transfer mode and (2) indirect address transfer mode.

  (1) In the direct address transfer mode, 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 11.5 , data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a write cycle. Figure 11.6 shows an example of the timing at this time.



Note: Transfer between external memories, DACK output in a read cycle.

Figure 11.5 Example of DMA Transfer Timing in the Direct Address Mode in the Dual Mode



The SAR value is an address, data is read from the transfer source module, and the data is tempolarily 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

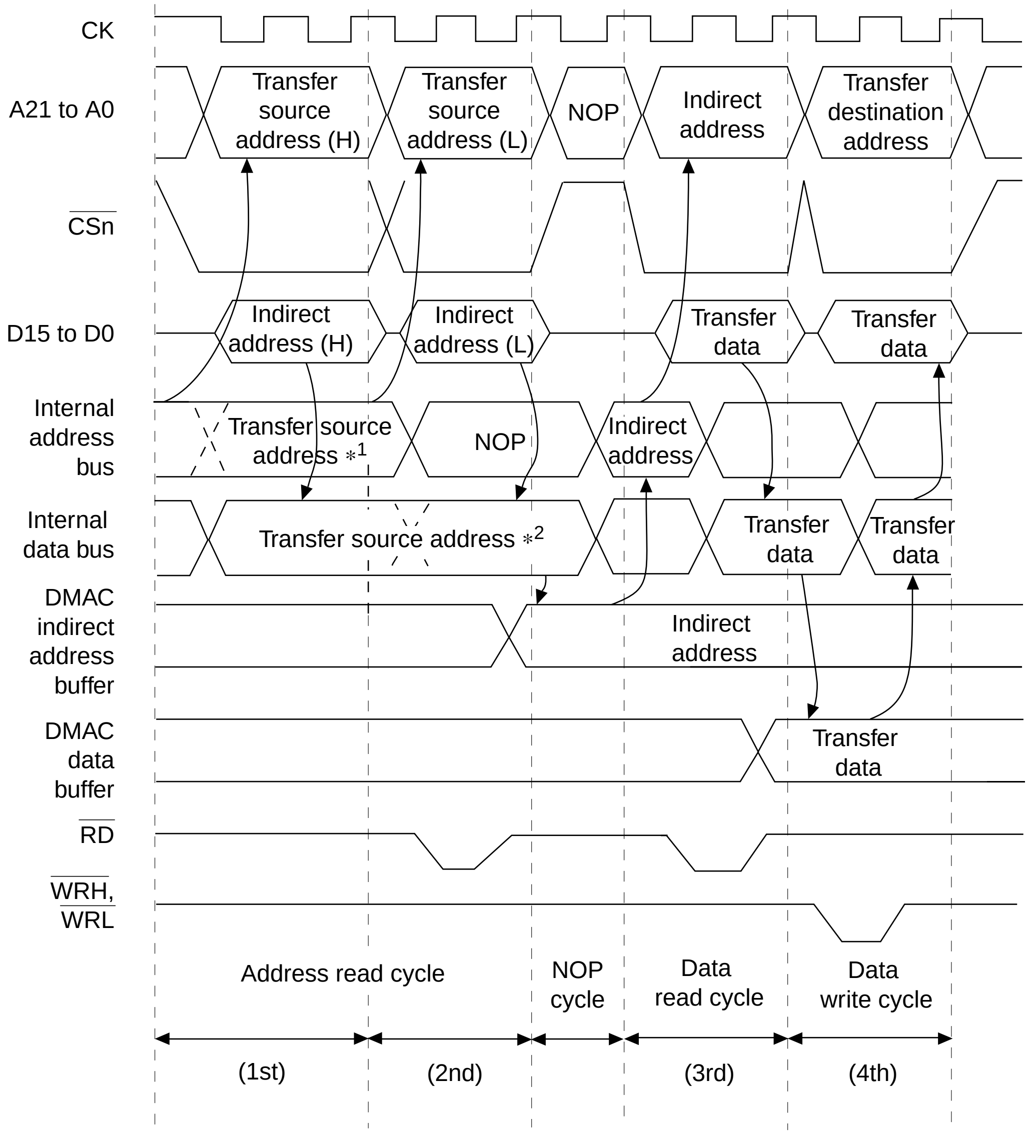
Figure 11.6 Operation of Direct Address Mode in Dual Address Mode

(2) In the indirect address transfer mode, the address of memory in which data to be transferred is stored is specified in the transfer source address register (SAR3) in the DMAC. Consequently, in this mode, the address value specified in the transfer source address register in the DMAC is read first. This value is temporarily stored in the DMAC. Next, the read value is output as an address, and the value stored in that address is stored in the DMAC again. Then, the value read afterwards is written to the address specified in the transfer destination address; this completes one DMA transfer.

Figure 11.7 shows one example. In this example, the transfer destination, the transfer source, and the storage destination of the indirect address are external memories, and transfer data is 16 or 8 bits. Figure 11.8 shows an example of the transfer timing.

In this mode, one NOP cycle (CK1 cycle shown in figure 11.8) is required to output data read as an indirect address to an address bus.

If transfer data is 32 bits, third and fourth bus cycles shown in figure 11.7 is required twice for each; a total of six bus cycles and one NOP cycle are required.



Transfer between external memories

Notes: 1. The internal address bus value does not change, and controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bus.

Figure 11.8 Example of Transfer Timing in the Indirect Address Mode in the Dual Address Mode (1)

Bus Modes: There are two bus modes: cycle steal and burst. Select the mode in the TM bits of CHCR0–CHCR3.

- Cycle-Steal Mode

  In the cycle steal mode, the bus right is given to another bus master after a one-transfer-unit (8-, 16-, or 32-bit unit) DMA transfer. When another transfer request occurs, the bus rights are obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

  In the cycle steal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination. Figure 11.8 shows an example of DMA transfer timing in the cycle steal mode. Transfer conditions shown in the figure are:

  — Dual address mode

  — $\overline{\text{DREQ}}$ level detection

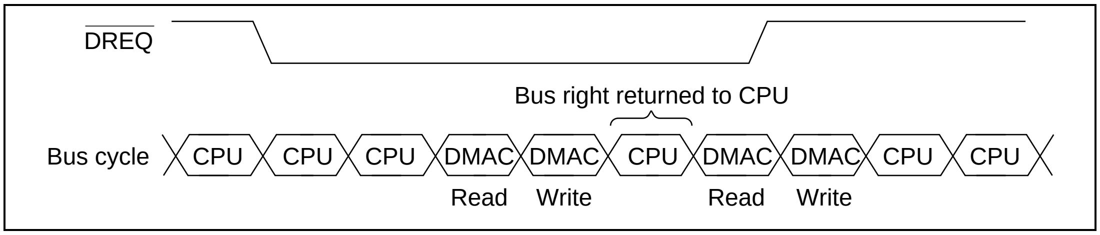
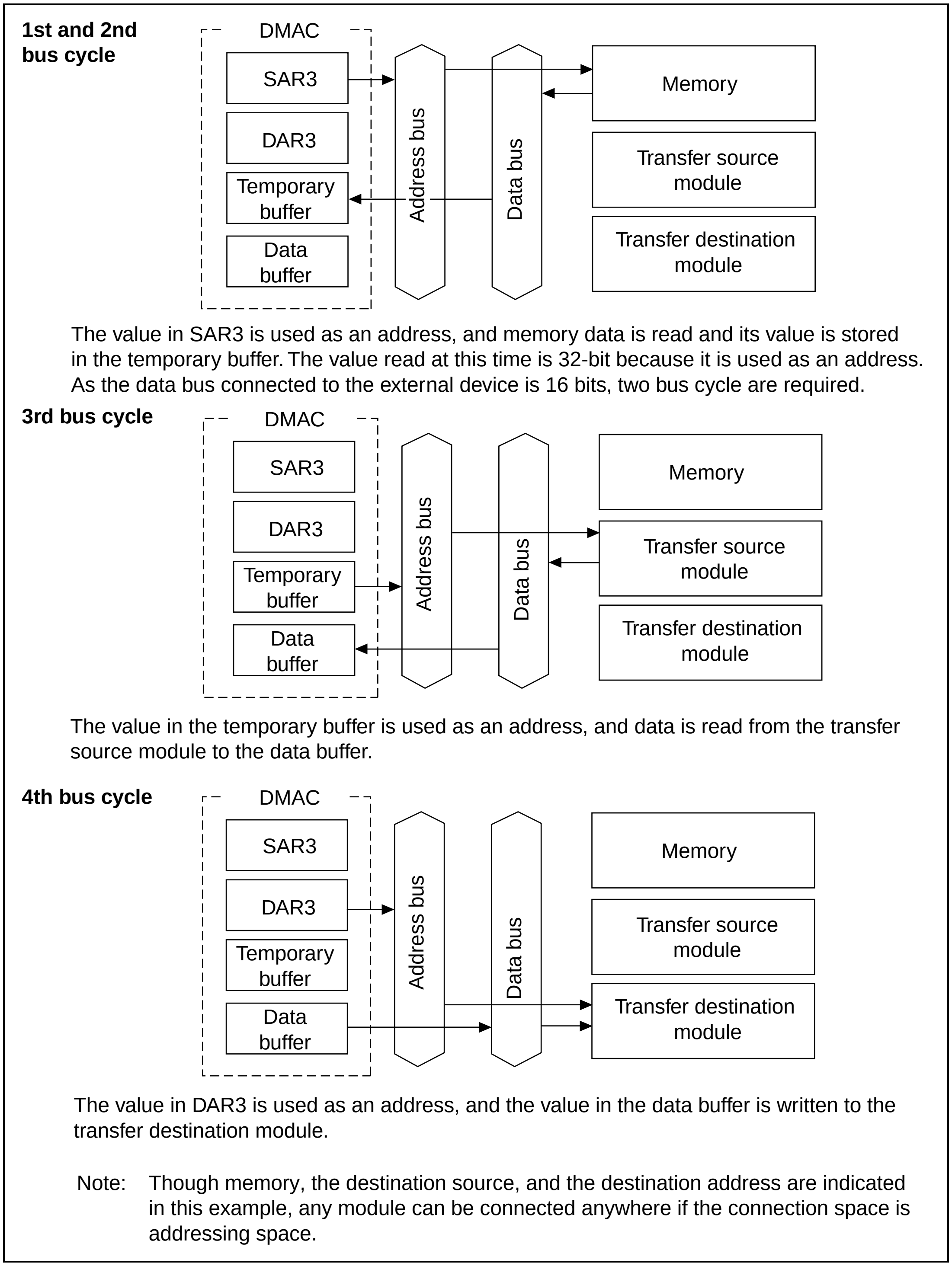


Figure 11.9 Transfer Example in the Cycle-Steal Mode

- Burst Mode

  Once the bus right is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven high, the bus passes to the other bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

  The burst mode cannot be used when the serial communications interface (SCI) is the transfer request source.



The value in SAR3 is used as an address, and memory data is read and its value is stored in the temporary buffer. The value read at this time is 32-bit because it is used as an address. As the data bus connected to the external device is 16 bits, two bus cycle are required.

The value in the temporary buffer is used as an address, and data is read from the transfer source module to the data buffer.

The value in DAR3 is used as an address, and the value in the data buffer is written to the transfer destination module.

Note: Though memory, the destination source, and the destination address are indicated in this example, any module can be connected anywhere if the connection space is addressing space.

Figure 11.10 Example of Transfer in Burst Mode ($\overline{\text{DREQ}}$ Level Detection)

Relationship between Request Modes and Bus Modes by DMA Transfer Category: Table 11.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 11.6 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32	0,1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1
	External memory and external memory	Everything*[1]	B/C	8/16/32	0–3*[5]
	External memory and memory-mapped external device	Everything*[1]	B/C	8/16/32	0–3*[5]
	Memory-mapped external device and memory-mapped external device	Everything*[1]	B/C	8/16/32	0–3*[5]
	External memory and on-chip peripheral module	Everything*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]
	Memory-mapped external device and on-chip peripheral module	Everything*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]
	On-chip peripheral module and on-chip peripheral module	Everything*[2]	B/C*[3]	8/16/32*[4]	0–3*[5]

B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests and on-chip peripheral module requests are all available. For on-chip peripheral module requests, however, SCI and A/D converter cannot be specified as the transfer request source.

2. External requests, auto requests and on-chip peripheral module requests are all available. When the SCI or A/D converter is also the transfer request source, however, the transfer destination or transfer source must be the SCI or A/D converter, respectively.

3. If the transfer request source is the SCI, cycle steal only.

4. The access size permitted when the transfer destination or source is an on-chip peripheral module register.

5. If the transfer request is an external request, channels 0 and 1 only.

Bus Mode and Channel Priority Order: When a given channel 1 is transferring in burst mode and there is a transfer request to a channel 0 with a higher priority, the transfer of the channel 0 will begin immediately.

At this time, if the priority is set in the fixed mode (CH0 > CH1), the channel 1 transfer will continue when the channel 0 transfer has completely finished, even if channel 0 is operating in the cycle steal mode or in the burst mode.

If the priority is set in the round-robin mode, channel 1 will begin operating again after channel 0 completes the transfer of one transfer unit, even if channel 0 is in the cycle steal mode or in the burst mode. The bus will then switch between the two in the order channel 1, channel 0, channel 1, channel 0.

Even if the priority is set in the fixed mode or in the round-robin mode, it will not give the bus to the CPU since channel 1 is in burst mode. This example is illustrated in figure 11.9.

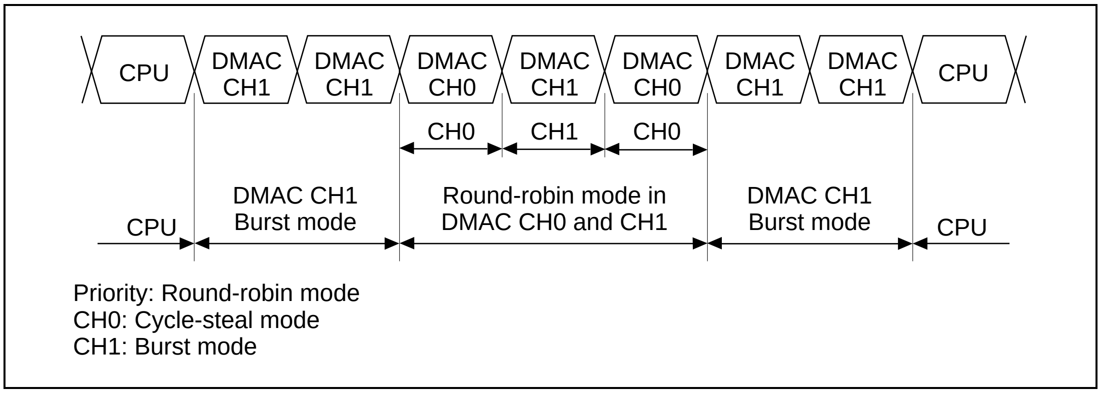


Figure 11.11 Bus Handling when Multiple Channels Are Operating

11.3.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 10, Bus State Controller.

$\overline{\text{DREQ}}$ Pin Sampling Timing: In external request mode, the $\overline{\text{DREQ}}$ pin is sampled by clock pulse (CKIO) falling edge or low level detection. When $\overline{\text{DREQ}}$ input is detected, a DMAC bus cycle is generated and DMA transfer performed, at the earliest, three states later.

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are started two cycles after the first sample.

Operation

- Cycle-Steal Mode

  In cycle-steal mode, the $\overline{\text{DREQ}}$ sampling timing is the same regardless of whether level or edge detection is used.

  For example, in figure A (cycle-steal mode, level detection), DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. If $\overline{\text{DREQ}}$ is not detected at this time, sampling is performed in each subsequent cycle.

  Thus, $\overline{\text{DREQ}}$ sampling is performed one step in advance. The third sampling operation is not performed until the idle cycle following the end of the first DMA transfer.

  The above conditions are the same whatever the number of CPU transfer cycles, as shown in figure B.

  $\overline{\text{DACK}}$ is output in a read in the example in figure A, and in a write in the example in figure B. In both cases, $\overline{\text{DACK}}$ is output for the same duration as $\overline{\text{RD}}$, $\overline{\text{WEn}}$, or $\overline{\text{CASxx}}$.

- Burst Mode, Level Detection

  In the case of burst mode with level detection, the $\overline{\text{DREQ}}$ sampling timing is the same as in cycle-steal mode.

  For example, in figure C, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. Subsequent sampling operations are performed in the idle cycle following the end of the DMA transfer cycle.

  In burst mode, also, the $\overline{\text{DACK}}$ output period is the same as in cycle-steal mode.

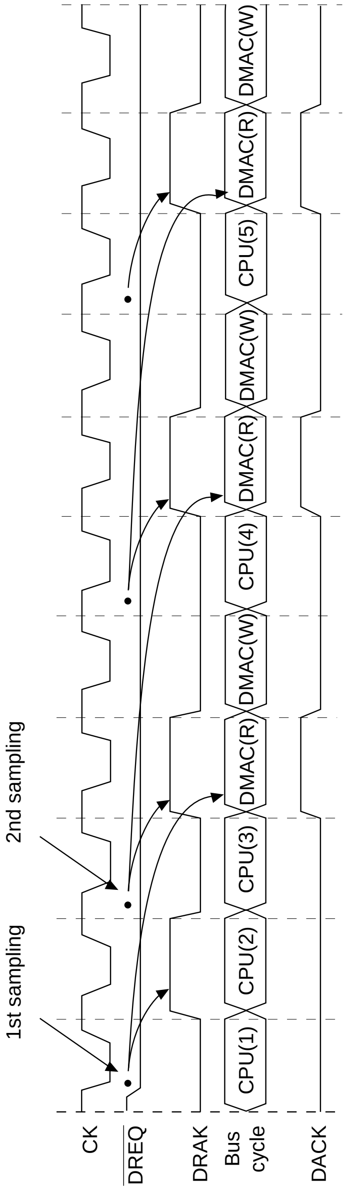
- Burst Mode, Dual Address Mode, Edge Detection

  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is only performed once.

  For example, in figure D, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. After this, DMAC transfer is executed continuously until the number of data transfers set in the DMATCR register have been completed. $\overline{\text{DREQ}}$ is not sampled during this time.

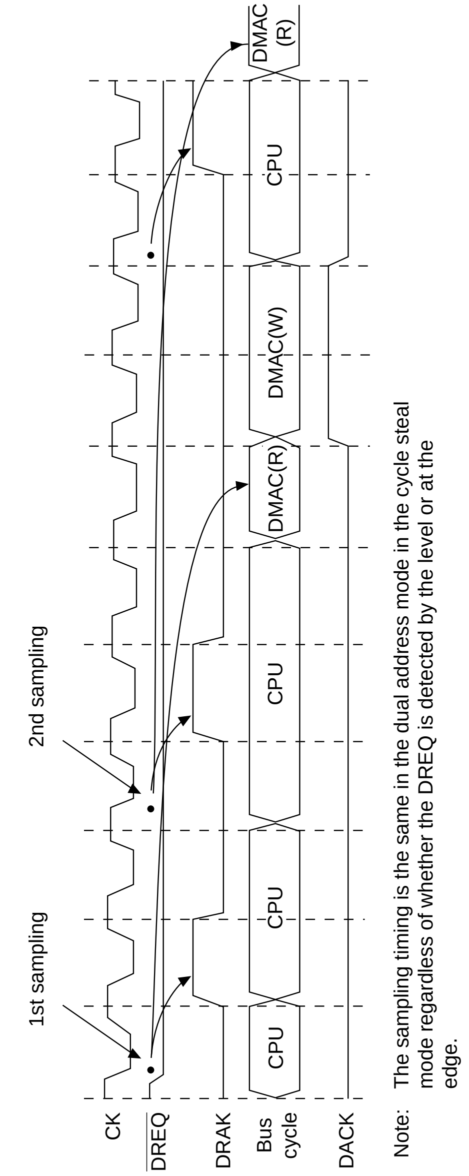
To restart DMA transfer after it has been suspended by an NMI, first clear NMIF, then input an edge request again.

In burst mode, also, the $\overline{\text{DACK}}$ output period is the same as in cycle-steal mode.



Note: The sampling timing is the same in the dual address mode in the cycle steal mode regardless of whether the DREQ is detected by the level or at the edge.

Figure 11.12 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles)



Note: The sampling timing is the same in the dual address mode in the cycle steal mode regardless of whether the DREQ is detected by the level or at the edge.

Figure 11.13 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles)

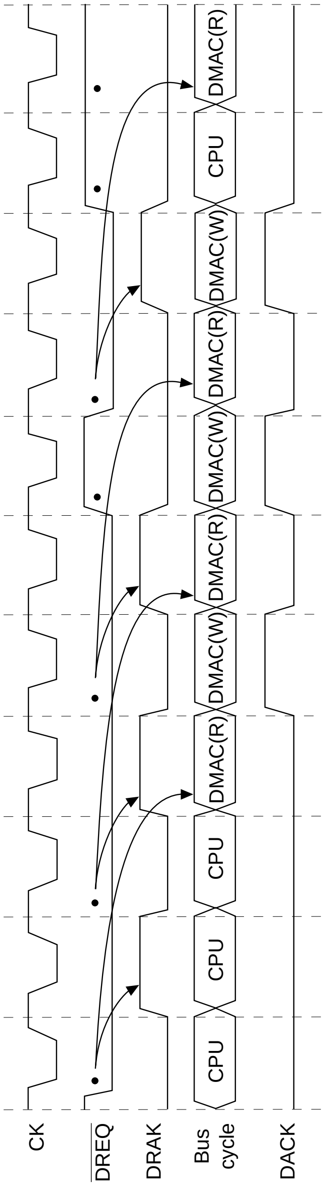


Figure 11.14 Burst Mode, Level Input

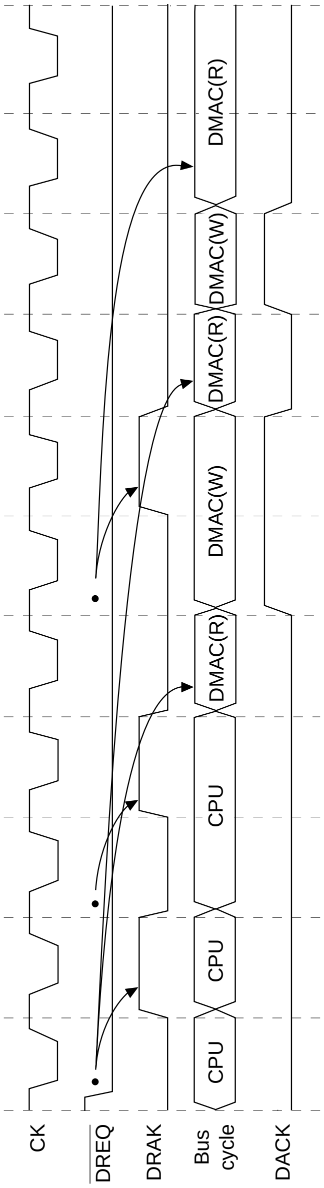


Figure 11.15 Burst Mode, Edge Input

11.3.6 Source Address Reload Function

Channel 2 includes a reload function, in which the value returns to the value set in the source address register (SAR2) for each four transfers by setting the RO bit in CHCR2. Figure 11.14 shows this operation. Figure 11.15 shows the timing chart of the source address reload function, which is under the following conditions: burst mode, auto-request, 16-bit transfer data size, SAR2 count-up, DAR2 fixed, reload function on, and usage of only channel 2.

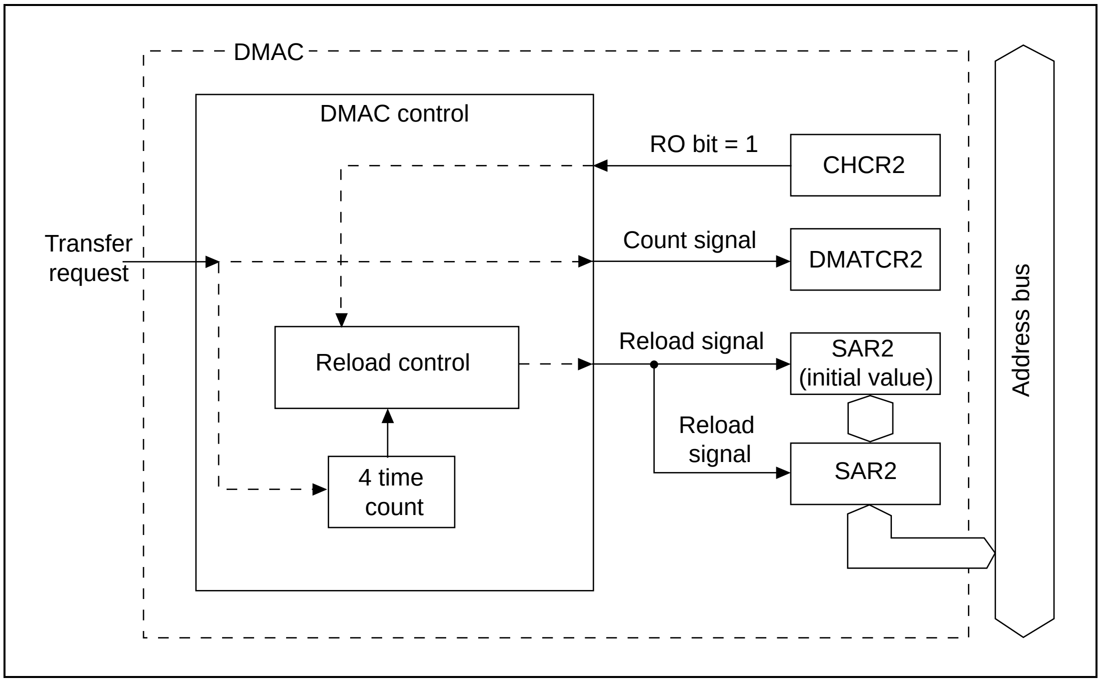


Figure 11.16 Source Address Reload Function Diagram

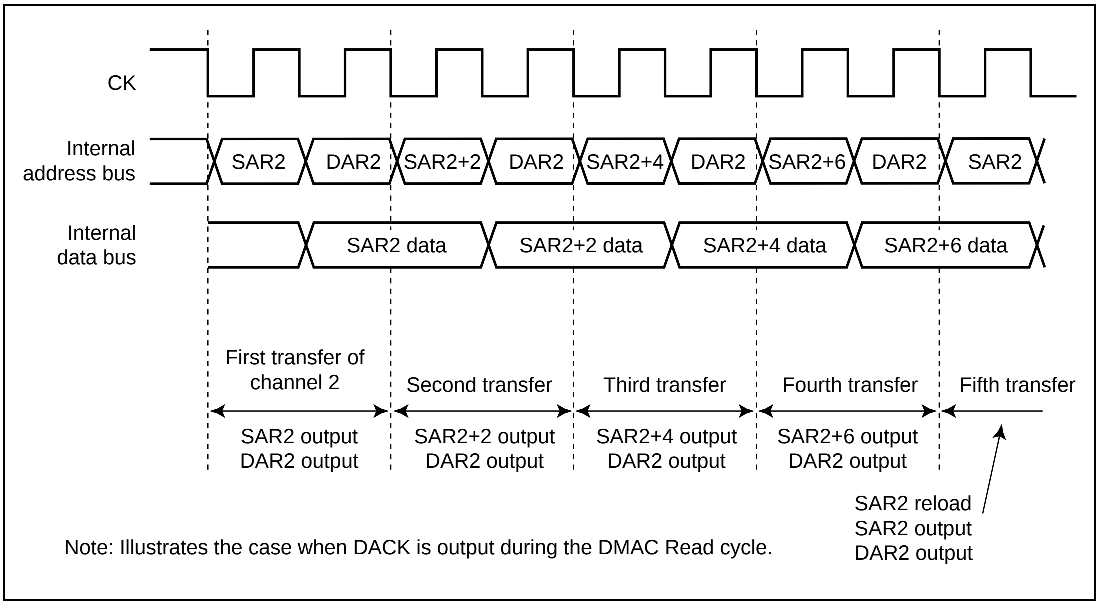


Figure 11.17 Timing Chart of Source Address Reload Function

Even if the transfer data size is 8, 16, or 32 bits, a reload function can be executed.

DMATCR, which specifies a transfer count, increments 1 each time a transfer ends regardless of whether a reload function is on or off. Consequently, be sure to specify the value multiple of four in DMATCR when the reload function is on. Specifying other values does not guarantee the operation.

Though the counters that count transfers of four times for the reload function are reset by clearing the DME bit in DMAOR or the DE bit in CHCR2, by setting the transfer end flag (TE bit in CHCR2), by inputting NMI, besides by reset or standby, the SAR2, DAR2, DMATCR2 registers are not reset. Therefore, if these sources are generated, the counters that are initialized and are not initialized exist in the DMAC; malfunction will be caused by restarting the DMAC in that state. Consequently, if these sources occur except for setting the TE bit during the usage of the reload function, set SAR2, DAR2, and DMATCR2 again.

11.3.7 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and all channels ending together. At transfer end, the following conditions are applied except the case where the value set in the DMA transfer count register (DMATCR) reaches 0.

(a) Cycle steal mode (external request, internal request, and auto-request)

When the transfer ending conditions are satisfied, DMAC transfer request acceptance is suspended. The DMAC stops operating after completing the number of transfers that it has accepted until the ending conditions are satisfied.

In the cycle steal mode, the operation is the same regardless of whether the transfer request is detected by the level or at the edge.

(b) Edge detection in the burst mode (external request, internal request, and auto-request)

The timing from the point where the ending conditions are satisfied to the point where the DMAC stops operating does not differ from that in cycle steal mode. In the edge detection in the burst mode, though only one transfer request is generated to start up the DMAC, stop request sampling is performed in the same timing as transfer request sampling in the cycle steal mode. As a result, the period when stop request is not sampled is regarded as the period when transfer request is generated, and after performing the DMA transfer for this period, the DMAC stops operating.

(c) Level detection in the burst mode (external request)

Same as described in a.

(d) Bus timing when transfers are suspended

The transfer is suspended when one transfer ends. Even if transfer ending conditions are satisfied during read in the direct address transfer in the dual address mode, the subsequent write process is executed, and after the transfer in (a) to (c) above has been executed, DMAC operation suspends.

Individual Channel Ending Conditions: There are two ending conditions. A transfer ends when the value of the channel’s DMA transfer count register (DMATCR) is 0, or when the DE bit of the channel’s CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding channel's DMA transfer ends, the transfer end flag bit (TE) is set in the CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) is requested to the CPU. This transfer ending does not apply to (a) to (d) described above.
- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel’s CHCR. The TE bit is not set when this happens. This transfer ending applies to (a) to (d) described above.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end when 1) the NMIF (NMI flag) bit is set to 1 in the DMAOR, or 2) when the DME bit in the DMAOR is cleared to 0.

- Transfers ending when the NMIF bit is set to 1 in DMAOR: When an NMI interrupt occurs, the NMIF bit is set to 1 in the DMAOR and all channels stop their transfers according to the conditions in (a) to (d) described above, and pass the bus right to other bus masters. Consequently, even if the NMI bit is set to 1 during transfer, the SAR, DAR, DMATCR are updated. The TE bit is not set. To resume the transfers after NMI interrupt exception processing, clear the NMIF bit to 0. At this time, if there are channels that should not be restarted, clear the corresponding CHCR DE bit.
- Transfers ending when DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in the DMAOR forcibly aborts the transfers on all channels. The TE bit is not set. All channels aborts their transfers according to the conditions in (a) to (d) in 11.3.7, as in NMI interrupt generation. In this case, the values in SAR, DAR, and DMATCR are also updated.

11.4 Compare Match Timer (CMT)

11.4.1 Overview

DMAC has an on-chip compare match timer (CMT) to generate DMA transfer request. The CMT has 16-bit counter.

Features

The CMT has the following features:

- Four types of counter input clock can be selected
  — One of four internal clocks (Pφ/4, Pφ/8, Pφ/16, Pφ/64) can be selected.
- Generate DMA transfer request when compare match occurs.

Block Diagram

Figure 11.16 shows a block diagram of the CMT.

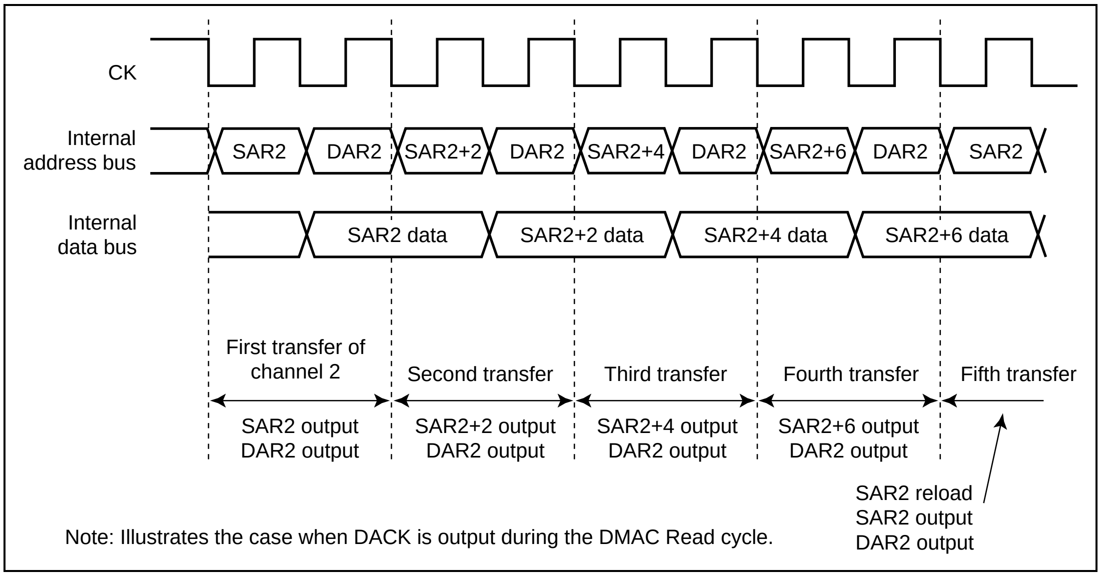


Figure 11.18 CMT Block Diagram

Register Configuration

Table 11.7 summarizes the CMT register configuration.

Table 11.7 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Access Size (Bits)
Compare match timer start register	CMSTR	R/W	H'0000	H'04000070	8, 16, 32
Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'04000072	8, 16, 32
Compare match counter 0	CMCNT0	R/W	H'0000	H'04000074	8, 16, 32
Compare match constant register 0	CMCOR0	R/W	H'FFFF	H'04000076	8, 16, 32

Note: The only value that can be written to the CMCSR0 CMF bits is a 0 to clear the flags.

11.4.2 Register Descriptions

Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT). It is initialized to H'0000 by resets. It retains its previous value in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	—	R	v	R/W	R/W

Bits 15–2—(Reserved): These bits always read as 0 and cannot be modified.

Bit 1(Reserved):

Bit 0—Count start 0 (STR0): Selects whether to operate or halt compare match timer counter 0.

Bit 0: STR0	Description
0	CMCNT0 count operation halted (initial value)
1	CMCNT0 count operation

Compare Match Timer Control/Status Register (CMCSR)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable of interrupts, and establishes the clock used for incrementation. It is initialized to H'0000 by resets. It retains its previous value in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CMF	—	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: The only value that can be written is a 0 to clear the flag.

Bits 15–8 and 5–2—(Reserved): These bits always read as 0 and cannot be modified.

Bit 7—Compare match flag (CMF): This flag indicates whether or not the CMCNT and CMCOR values have matched.

Bit 7: CMF	Description
0	CMCNT and CMCOR values have not matched (initial status) Clear condition: Write a 0 to CMF after reading a 1 from it
1	CMCNT and CMCOR values have matched

Bit 6 (Reserved)

Bits 1, 0—Clock select 1, 0 (CKS1, CKS0): These bits select the clock input to the CMCNT from among the four internal clocks obtained by dividing the system clock (ϕ). When the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	P ϕ/4 (initial status)
	1	P ϕ/8
1	0	P ϕ/16
	1	P ϕ/64

Compare Match Counter (CMCNT)

The compare match counter (CMCNT) is a 16-bit register used as an upcounter.

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with that clock. When the CMCNT value matches that of the compare match constant register (CMCOR), the CMCNT is cleared to H'0000 and the CMF flag of the CMCSR is set to 1.

The CMCNT is initialized to H'0000 by resets. It retains its previous value in standby mode.

Bit:	15	14	13	12	11	10	9	8
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:	7	6	5	4	3	2	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

Compare Match Constant Register (CMCOR)

The compare match constant register (CMCOR) is a 16-bit register that sets the compare match period with the CMCNT.

The CMCOR is initialized to H'FFFF by resets. It retains its previous value in standby mode.

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

11.4.3 Operation

Period Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the compare match constant register (CMCOR), the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. The CMCNT counter begins counting up again from H'0000.

Figure 11.19 shows the compare match counter operation.

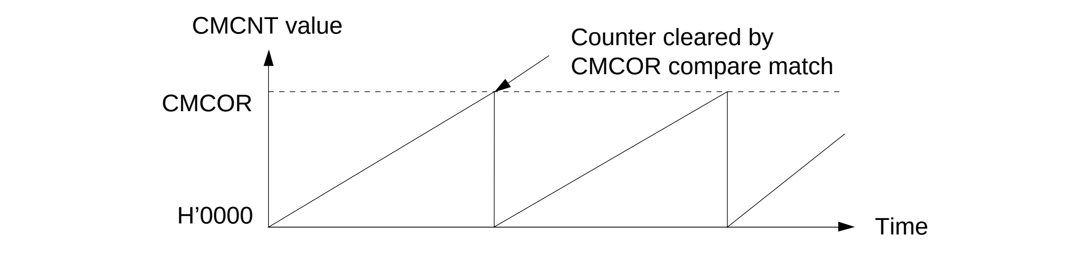


Figure 11.19 Counter Operation

CMCNT Count Timing

One of four clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) obtained by dividing the clock (Pϕ) can be selected by the CKS1, CKS0 bits of the CMCSR. Figure 11.20 shows the timing.

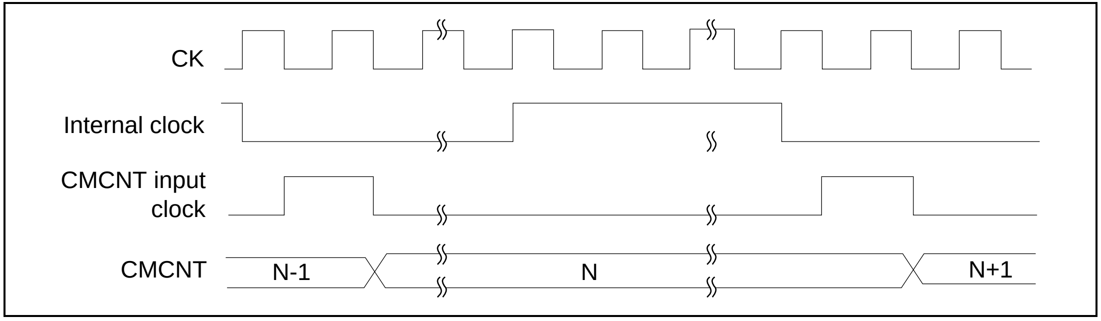


Figure 11.20 Count Timing

11.4.4 Compare Match

Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 11.21 shows the CMF bit set timing.

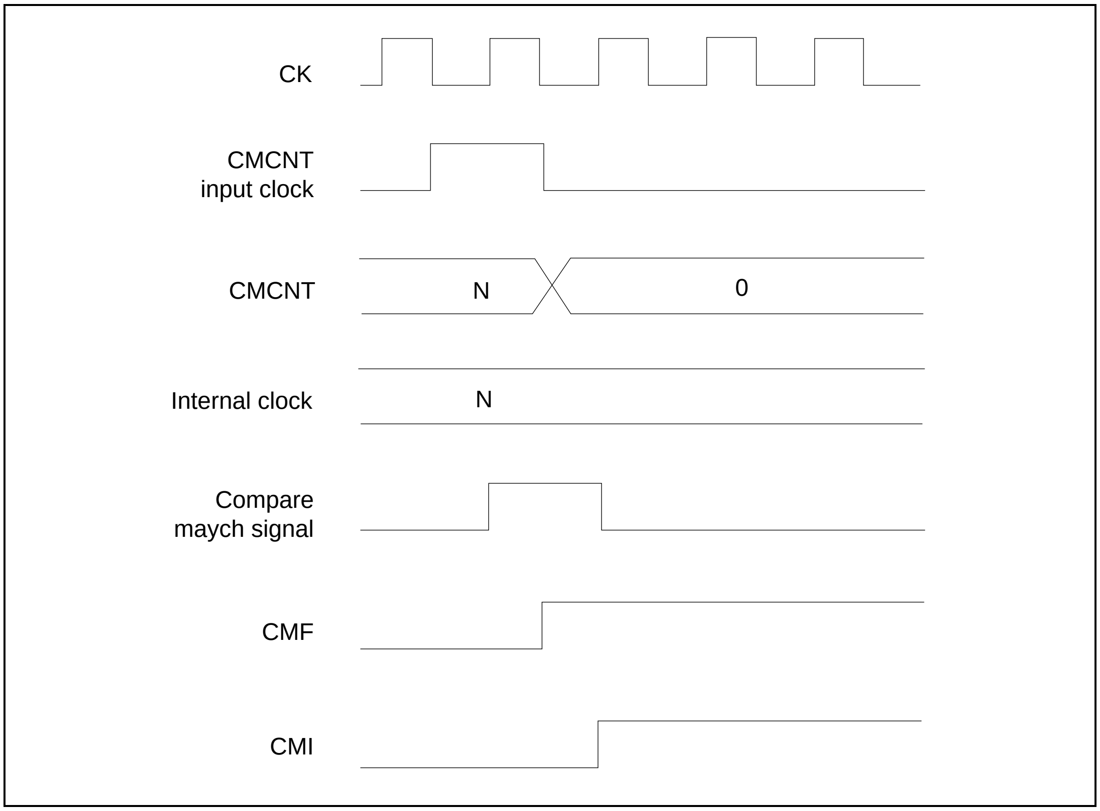


Figure 11.21 CMF Set Timing

Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared by writing a 0 to it after reading a 1. Figure 11.22 1 shows the timing when the CMF bit is cleared by the CPU.

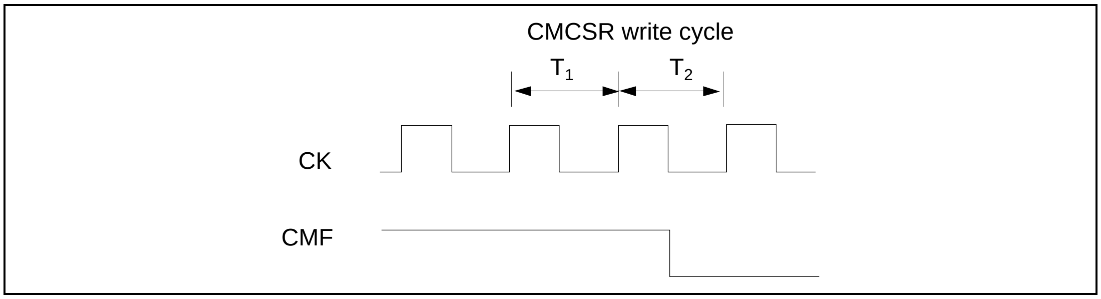


Figure 11.22 Timing of CMF Clear by the CPU

11.5 Examples of Use

11.5.1 Example of DMA Transfer between On-Chip IRDA and External Memory

In this example, receive data of on-chip IRDA is transferred to external memory using DMAC channel 3. Table 11.8 shows the transfer conditions and register settings. In addition, it is recommended that the trigger of the number of receive FIFO data in SCI1 is set to 1 (RTRG1 = RTRG0 = 0 in SCFCR).

Table 11.8 Transfer Conditions and Register Settings for Transfer between On-Chip SCI and External Memory

Transfer Conditions	Register	Setting
Transfer source: RDR1 of on-chip IRDA	SAR3	H'0400014A
Transfer destination: external memory	DAR3	H'00400000
Number of transfers: 64	DMATCR3	H'00000040
Transfer source address: fixed	CHCR3	H'00004B05
Transfer destination address: incremented		
Transfer request source: IRDA (RXI1)		
Bus mode: cycle steal		
Transfer unit: byte		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 3 > 2 > 1	DMAOR	H'0001

11.5.2 Example of DMA Transfer between AD0 and External Memory (Address Reload on)

In this example, DMA transfer is performed between channel 0 of the on-chip AD converter (transfer source) and the external memory (transfer destination) with address reload function on. Table 11.9 shows the transfer conditions and register settings.

Table 11.9 Transfer Conditions and Register Settings for Transfer between On-Chip Channel 2 of AD converter and External Memory

Transfer Conditions	Register	Setting
Transfer source: on-chip AD converter	SAR2	H'04000080
Transfer destination: external memory	DAR2	H'00400000
Number of transfers: 128 (reloading 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'00089E35
Transfer destination address: decremented		
Transfer request source: AD1		
Bus mode: burst		
Transfer unit: long word		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'0101

When the address reload function is on, the value set in SAR returns to the initially set value at each four transfers. In this example, when an interrupt request is generated from AD converter, byte data is read from the register in address H'04000080 in AD converter , and it is written to external memory address H'00400000. Since longword data has been transferred, the values in SAR and DAR are H'FFFF83E4 and H'00400004, respectively. The bus right is maintained and data transfers are successively performed because this transfer is in the burst mode.

After four transfers end, fifth and sixth transfers are performed if the address reload function is off, and the value in SAR is incremented from H'0400008C, H'04000090, H'04000094,.... If the address reload function is on, the DMA transfer stops after the fourth transfer ends, the bus request signal to the CPU is cleared. At this time, the value stored in SAR is not incremented from H'0400008C to H'04000090, but returns to the initially set value H'04000080. The value in DAR continues being incremented regardless of whether the address reload function is on or off.

As a result, the values in the DMAC are as shown in table 11.10 when the fourth transfer ends, depending on whether the address reload function is on or off.

Table 11.10 Values in the DMAC after the Fourth Transfer Ends

Items	Address reload on	Address reload off
SAR	H'04000080	H'04000090
DAR	H'003FFFFC	H'003FFFFC
DMATCR	H'0000007C	H'0000007C
Bus right	Released	Held
DMAC operation	Stops	Keeps operating
Interrupt	Not generated	Not generated
Transfer request source flag clear	Executed	Not executed

Notes: 1. An interrupt is generated regardless of whether the address reload function is on or off, if transfers are executed until the value in DMATCR reaches 0 and the IE bit in CHCR has been set to 1.

2. The transfer request source flag is cleared regardless of whether the address reload function is on or off, if transfers are executed until the value in DMATCR reaches 0.

3. Specify the burst mode to use the address reload function. This function may not be correctly executed in the cycle steal mode.

4. Set the value multiple of four in DMATCR to use the address reload function. This function may not be correctly executed if other values are specified.

11.5.3 Example of DMA Transfer between External Memory and SCIF Transmitter (SCIT2) (Indirect Address on)

In this example, DMA transfer is performed between the external memory specified with the indirect address (transfer source) and the SCIF transmitter (transfer destination) . Table 11.11 shows the transfer conditions and register settings. In addition, the trigger of the number of transmit FIFO data is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Table 11.11 Transfer Conditions and Register Settings for Transfer between External Memory and SCIF Transmitter

Transfer Conditions	Register	Setting
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'04500000	—	H'55
Transfer destination: On-chip SCFTDR2	DAR3	H'04000156
Number of transfers: 10	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR2	H'00011C01
Transfer destination address: fixed		
Transfer request source: SCIT2		
Bus mode: cycle steal		
Transfer unit: byte		
No interrupt request generated at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

If the indirect address is on, data stored in the address set in SAR is not used as transfer source data. In the indirect address, after the value stored in the address set in SAR is read, that read value is used as an address again, and the value stored in that address is read and stored in the address set in DAR.

In the example shown in table 11.10, when an SCI2 transfer request is generated, the DMAC reads the value in address H'00400000 set in SAR3. Since the value H'00450000 is stored in that address, the DMAC reads the value H'00450000. Next, the DMAC uses that read value as an address again, and reads the value H'55 stored in that address. Then, the DMAC writes the value H'55 to address H'04000156 set in DAR3; this completes one indirect address transfer.

In the indirect address, when data is read first from the address set in SAR3, the data transfer size is always longword regardless of the settings of the TS0 and the TS1 bits that specify the transfer data size. However, whether the transfer source address is fixed, incremented, or decremented is specified according to the SM0 and the SM1 bits. Therefore, in this example, though the transfer data size is specified as byte, the value in SAR3 is H'00400004 when one transfer ends. Write operation is the same as that in the normal dual address transfer.

11.6 Cautions

1. The DMA channel control registers (CHCR0–CHCR3) can be accessed in any data size. The DMA operation register (DMAOR) must be accessed in byte (eight bits) or word (16 bits).; other registers must be accessed in word (16 bits) or longword (32 bits).
2. Before rewriting the RS0–RS3 bits of CHCR0–CHCR3, first clear the DE bit to 0 (when rewriting CHCR with a byte address, be sure to set the DE bit to 0 in advance).
3. Even when the NMI interrupt is input when the DMAC is not operating, the NMIF bit of the DMAOR will be set.
4. When entering the standby mode, the DME bit in DMAOR must be cleared to 0 and the transfers accepted by the DMAC must end.
5. The on-chip peripherals which DMAC can access are IRDA, SCIF, AD converter, DA converter, PCC and I/O port. Do not access the other peripherals by DMAC.
6. When starting up the DMAC, set CHCR or DMAOR last. Specifying other registers last does not guarantee normal operation.
7. Even if the maximum number of transfers is performed in the same channel after the DMATCR count reaches 0 and the DMA transfer ends normally, write 0 to DMATCR. Otherwise, normal DMA transfer may not be performed.
8. When using the address reload function, specify the burst mode as a transfer mode. In the cycle steal mode, normal DMA transfer may not be performed.
9. When using the address reload function, set the value multiple of four in DMATCR. Specifying other values does not guarantee normal operation.
10. When detecting an external request at the falling edge, keep the external request pin high when setting the DMAC.
11. Do not access the space ranging from H'4000062 to H'400006F, which is not used in the DMAC. Accessing that space may cause malfunctions.
12. When using CH2 in reload mode with burst mode, set transfer count to four.

Section 12 Timer (TMU)

12.1 Overview

The SH7709 uses a three-channel 32-bit timer unit (TMU).

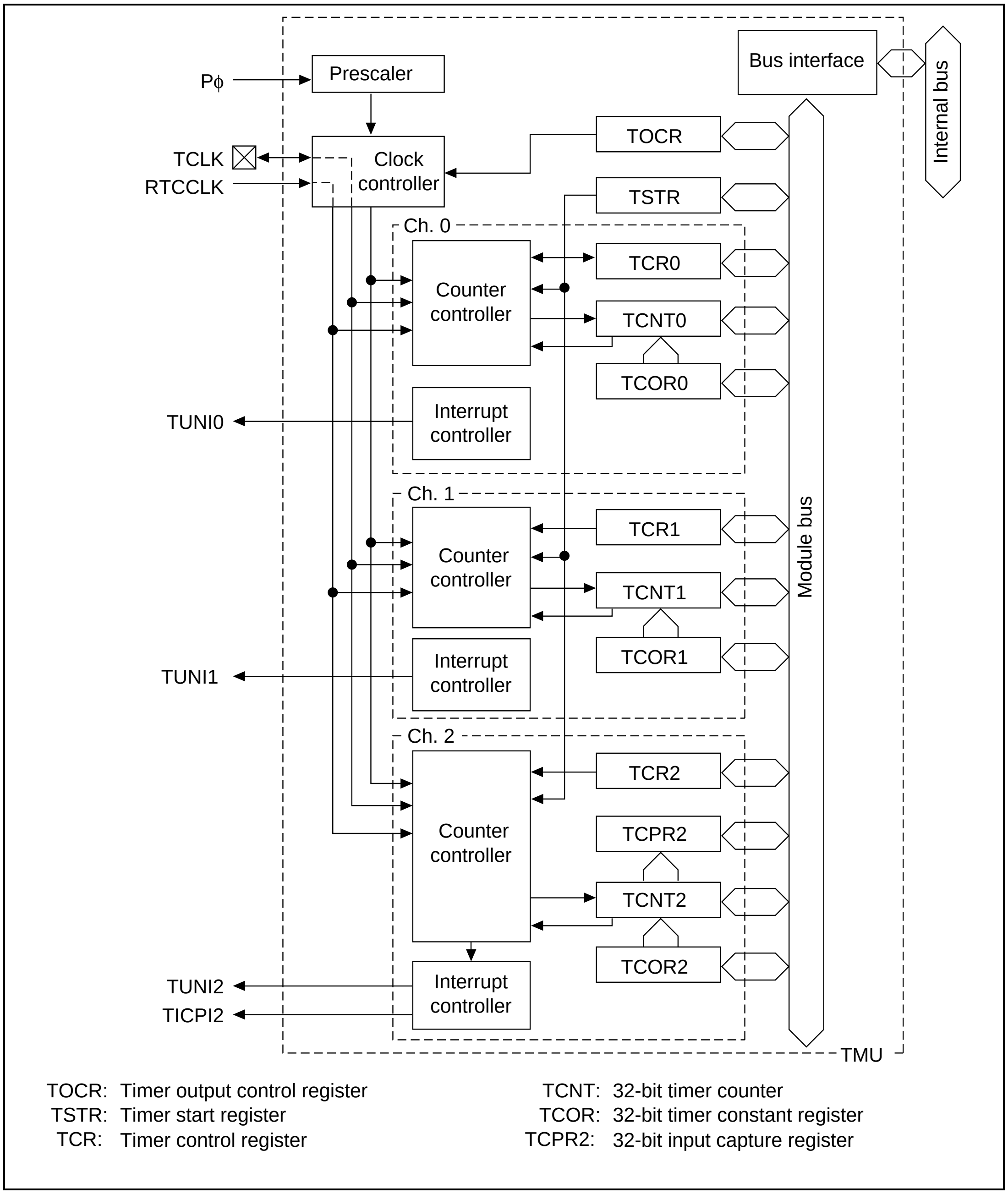
12.1.1 Features

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter
- Channel 2 is provided with an input capture function
- All channels are provided with 32-bit constant registers and 32-bit down counters that can be read or written to at any time
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection between 6 counter input clocks: External clock (TCLK), on-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256. (Pϕ is the internal clock for peripheral modules and can be selected as 1/4, 1/2, or the same frequency as that of the CPU operating clock ϕ.) See section 9, On-Chip Oscillation Circuits, for more information on the clock pulse generator.
- All channels can operate when the SH7709 is in standby mode: When the RTC output clock is being used as the counter input clock, the SH7709 is still able to count in standby mode.
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the peripheral module used has an internal bus width of 16 bits, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the counter read value caused by this time lag, a synchronization circuit is built into the TCNT so that the entire 32-bit data in the TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels: Operate the SH7709 so that the clock input to the timer counters of each channel (obtained by dividing the external clock and internal clock with the prescaler) does not exceed the maximum operating frequency.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the TMU.



TOCR: Timer output control register
TSTR: Timer start register
TCR: Timer control register
TCNT: 32-bit timer counter
TCOR: 32-bit timer constant register
TCPR2: 32-bit input capture register

Figure 12.1 TMU Block Diagram

12.1.3 Pin Configuration

Table 12.1 shows the pin configuration of the TMU.

Table 12.1 Pin Configuration

Pin Name	Symbol	I/O	Description
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/realtime clock (RTC) output pin

12.1.4 Register Configuration

Table 12.2 shows the TMU register configuration.

Table 12.2 TMU Register Configuration

Channel	Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Common	Timer output control register	TOCR	R/W	H'00	H'FFFFFE90	8
	Timer start register	TSTR	R/W	H'00	H'FFFFFE92	8
0	Timer constant register 0	TCOR0	R/W	H'FFFFFFFF	H'FFFFFE94	32
	Timer counter 0	TCNT0	R/W	H'FFFFFFFF	H'FFFFFE98	32
	Timer control register 0	TCR0	R/W	H'0000	H'FFFFFE9C	16
1	Timer constant register 1	TCOR1	R/W	H'FFFFFFFF	H'FFFFFEA0	32
	Timer counter 1	TCNT1	R/W	H'FFFFFFFF	H'FFFFFEA4	32
	Timer control register 1	TCR1	R/W	H'0000	H'FFFFFEA8	16
2	Timer constant register 2	TCOR2	R/W	H'FFFFFFFF	H'FFFFFEAC	32
	Timer counter 2	TCNT2	R/W	H'FFFFFFFF	H'FFFFFEB0	32
	Timer control register 2	TCR2	R/W	H'0000	H'FFFFFEB4	16
	Input capture register 2	TCPR2	R/W	Undefined	H'FFFFFEB8	32

12.2 TMU Registers

12.2.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit read/write register that selects whether to use the external TCLK pin as an external clock or an input capture control usage input pin, or an output pin for the on-chip RTC output clock. TOCR is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	—	—	TCOE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bits 7 to 1—Reserved: These bits always read 0. The write value should always be 0.

Bit 0—Timer Clock Pin Control (TCOE): Selects use of the timer clock pin (TCLK) as an external clock input pin or input pin for input capture control for the on-chip timer, or as an output pin for the on-chip RTC output clock.

Bit 0: TCOE	Description
0	Timer clock pin (TCLK) used as external clock input or input capture control input pin for the on-chip timer (Initial value)
1	Timer clock pin (TCLK) used as output pin for on-chip RTC output clock

12.2.2 Timer Start Register (TSTR)

TSTR is an 8-bit read/write register that selects whether to run or halt the timer counters (TCNT) for channels 0–2. TSTR is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode when the input clock selected for the channel is the on-chip RTC clock (RTCCLK). It is initialized in standby mode, changing the multiplying ratio of PLL circuit 1 or MSTP2 bit in STBCR is set to a logic one only when an external clock (TCLK) or the peripheral clock (Pϕ) is used as the input clock.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	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

Bits 7 to 3—Reserved: These bits always read 0. The write value should always be 0.

Bit 2—Counter Start 2 (STR2): Selects whether to run or halt timer counter 2 (TCNT2).

Bit 2: STR2	Description
0	Halt TCNT2 count (Initial value)
1	Start TCNT2 counting

Bit 1—Counter Start 1 (STR1): Selects whether to run or halt timer counter 1 (TCNT1).

Bit 1: STR1	Description
0	Halt TCNT1 count (Initial value)
1	Start TCNT1 counting

Bit 0—Counter Start 0 (STR0): Selects whether to run or halt timer counter 0 (TCNT0).

Bit 0: STR0	Description
0	Halt TCNT0 count (Initial value)
1	Start TCNT0 counting

12.2.3 Timer Control Register (TCR)

The timer control registers (TCR) control the timer counters (TCNT) and interrupts. The TMU has three TCR, registers one for each channel.

The TCR registers are 16-bit read/write registers that control the issuance of interrupts when the flag indicating timer counter (TCNT) underflow has been set to 1, and also carry out counter clock selection. When the external clock has been selected, they also select its edge. Additionally, TCR2 controls the channel 2 input capture function and the issuance of interrupts during input capture. The TCRs are initialized to H'0000 by a power-on reset and manual reset. They are not initialized in standby mode.

Channel 0 and 1 TCR Bit Configuration:

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	UNF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

Channel 2 TCR Bit Configuration:

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	ICPF	UNF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

Bits 15 to 10, 9 (except TCR2), 7, and 6 (except TCR2)—Reserved: These bits always read 0. The write value should always be 0.

Bit 9—Input Capture Interrupt Flag (ICPF): A function of channel 2 only: the flag is set when input capture is requested via the TCLK pin.

Bit 9: ICPF	Description
0	No input capture request has been issued. Clearing condition: When 0 is written to ICPF (Initial value)
1	Input capture has been requested via the TCLK pin. Setting condition: When an input capture is requested via the TCLK pin*

Note: Contents do not change when 1 is written to ICPF.

Bit 8—Underflow Flag (UNF): Status flag that indicates occurrence of a TCNT underflow.

Bit 8: UNF	Description
0	TCNT has not underflowed. Clearing condition: When 0 is written to UNF (Initial value)
1	TCNT has underflowed (H'00000000 → H'FFFFFFFF). Setting condition: When TCNT underflows*

Note: Contents do not change when 1 is written to UNF.

Bits 7 and 6—Input Capture Control (ICPE1, ICPE0): A function of channel 2 only: determines whether the input capture function can be used, and when used, whether or not to enable interrupts.

When using this input capture function it is necessary to set the TCLK pin to input mode with the TCOE bit in the TOCR register. Additionally, use the CKEG bit to designate use of either the rising or falling edge of the TCLK pin to set the value in TNCT2 in the input capture register (TCPR2).

Bit 7: ICPE1	Bit 6: ICPE0	Description
0	0	Input capture function is not used. (Initial value)
	1	Reserved (cannot be set)
1	0	Input capture function is used. Interrupts due to ICPF are not enabled.
	1	Input capture function is used. Interrupts due to ICPF are enabled.

Bit 5—Underflow Interrupt Control (UNIE): Controls enableing of interrupt generation when the status flag (UNF) indicating TCNT underflow has been set to 1.

Bit 5: UNIE	Description
0	Interrupts due to UNF are not enabled. (Initial value)
1	Interrupts due to UNF are enabled.

Bits 4 and 3—Clock Edge 1, 0 (CKEG1, CKEG0): These bits select the external clock edge when the external clock is selected, or when the input capture function is used.

Bit 4: CKEG1	Bit 3: CKEG0	Description
0	0	Count/capture register set on rising edge (Initial value)
	1	Count/capture register set on falling edge
1	—	Count/capture register set on both rising and falling edge

Bits 2 to 0—Timer Prescalers 2–0 (TPSC2–TPSC0): These bits select the TCNT count clock.

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count on Pφ/4 (Initial value)
		1	Internal clock: count on Pφ/16
	1	0	Internal clock: count on Pφ/64
		1	Internal clock: count on Pφ/256
1	0	0	Internal clock: count on clock output of on-chip RTC (RTCCLK)
		1	External clock: count on TCLK pin input
	1	0	Reserved
		1	Reserved

12.2.4 Timer Constant Register (TCOR)

The timer constant registers are 32-bit registers. The TMU has three TCOR registers, one for each of the three channels.

TCOR is a 32-bit read/write register. When a TCNT count-down results in an underflow, the TCOR value is set in TCNT and the count-down continues from that value. TCOR is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode, and retains its contents.

TCOR:

Bit:	31	30	29	28	27	26	25	24
Bit name:								
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:	23	22	21	20	19	18	17	16
Bit name:								
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:	15	14	13	12	11	10	9	8
Bit name:								
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:	7	6	5	4	3	2	1	0
Bit name:								
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

12.2.5 Timer Counters (TCNT)

The timer counters are 32-bit read/write registers. The TMU has three timer counters, one for each channel.

TCNT counts down upon input of a clock. The clock input is selected using the TPSC2–TPSC0 bits in the timer control register (TCR).

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the count-down continues from that value.

Because the internal bus for the SH7709 on-chip supporting modules is 16 bits wide, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. Since TCNT counts sequentially, this time lag can create discrepancies between the data in the upper and lower halves. To correct the discrepancy, a buffer register is connected to TCNT so that upper and lower halves are not read separately. The entire 32-bit data in TCNT can thus be read at once.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode, and retains its contents.

TCNT:

Bit:	31	30	29	28	27	26	25	24
Bit name:								
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:	23	22	21	20	19	18	17	16
Bit name:								
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:	15	14	13	12	11	10	9	8
Bit name:								
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:	7	6	5	4	3	2	1	0
Bit name:								
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

12.2.6 Input Capture Register (TCPR2)

The input capture register (TCPR2) is a read-only 32-bit register built only into timer 2. Control of TCPR2 setting conditions due to the TCLK pin is affected by the input capture function bits (ICPE1/ICPE2 and CKEG1/CKEG0)) in TCR2. When a TCPR2 setting indication due to the TCLK pin occurs, the value of TCNT2 is copied into TCPR2.

TCNT2 is not initialized by a power-on reset or manual reset, or in standby mode.

TCPR2:

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

12.3 TMU Operation

12.3.1 Overview

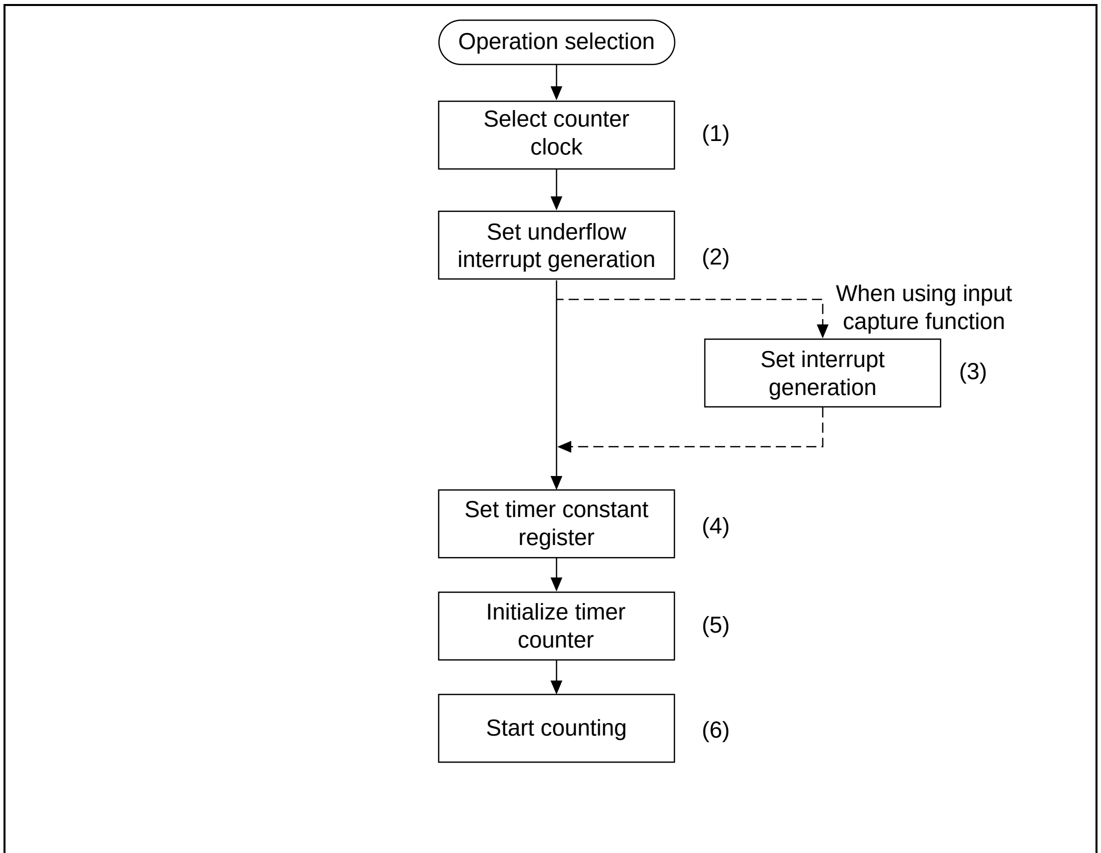
Each of the three channels has a 32-bit timer counter (TCNT) and a 32-bit timer constant register. The TCNT counts down. The auto-reload function enables synchronized counting and counting by external events. Channel 2 has an input capture function.

12.3.2 Basic Functions

Counter Operation: When the STR0–STR2 bits in the timer start register (TSTR) are set, the corresponding timer counter (TCNT) starts counting. When a TCNT underflows (H'00000000 → H'FFFFFFFF), the UNF flag of the corresponding timer control register (TCR) is set. At this time, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the down-count operation is continued.

The count operation is set as follows (figure 12.2):

1. Select the counter clock with the TPSC2–TPSC0 bits in the timer control register (TCR). If the external clock is selected, set the TCLK pin to input mode with the TOCE bit in TOCR, and select its edge with the CKEG1 and CKEG0 bits in TCR.
2. Use the UNIE bit in TCR to set whether to generate an interrupt when TCNT underflows.
3. When using the input capture function, set the ICPE bits in TCR, including the choice of whether or not to use the interrupt function (channel 2 only).
4. Set a value in the timer constant register (TCOR) (the cycle is the set value plus 1).
5. Set the initial value in the timer counter (TCNT).
6. Set the STR bit in the timer start register (TSTR) to 1 to start operation.



Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 12.2 Setting the Count Operation

Auto-Reload Count Operation: Figure 12.3 shows the TCNT auto-reload operation.

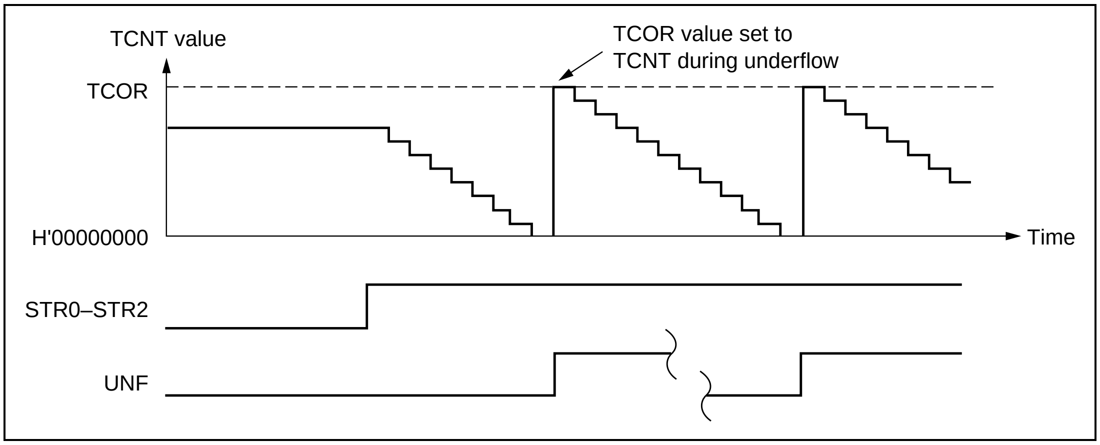


Figure 12.3 Auto-Reload Count Operation

TCNT Count Timing:

- Internal Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select whether peripheral module clock Pϕ or one of the four internal clocks created by dividing it is used (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256). Figure 12.4 shows the timing.

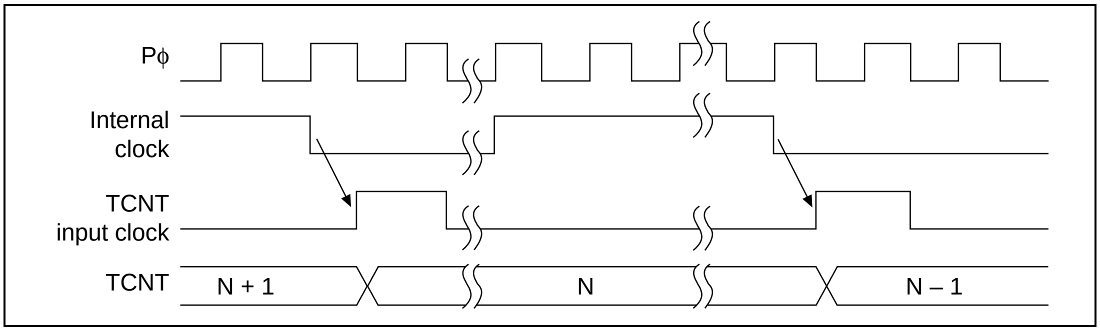


Figure 12.4 Count Timing when Internal Clock Is Operating

- External Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the external clock (TCLK) as the timer clock. Use the CKEG1 and CKEG0 bits in TCR to select the detection edge. Rise, fall or both may be selected. The pulse width of the external clock must be at least 1.5 peripheral module clock cycles for single edges or 2.5 peripheral module clock cycles for both edges. A shorter pulse width will result in accurate operation. Figure 12.5 shows the timing for both-edge detection.

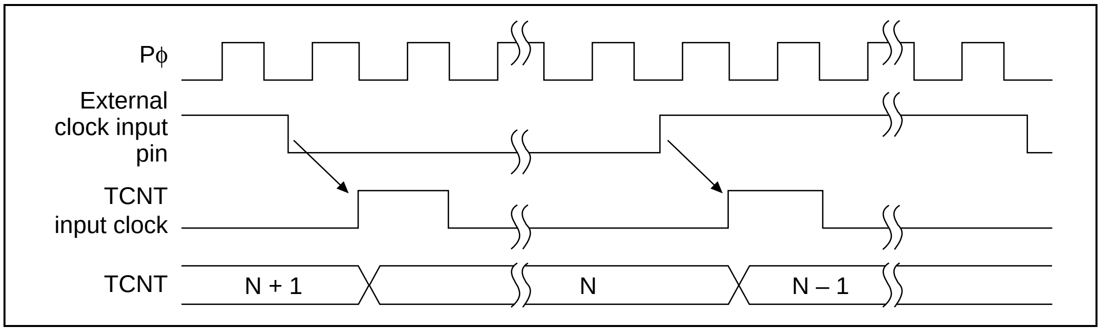


Figure 12.5 Count Timing when External Clock Is Operating (Both Edges Detected)

- On-Chip RTC Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the on-chip RTC clock as the timer clock. Figure 12.6 shows the timing.

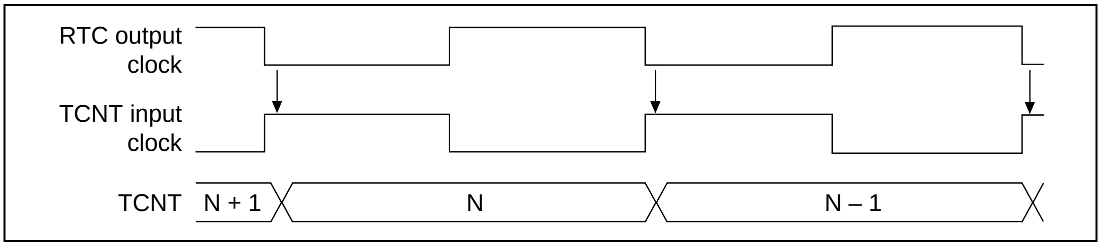


Figure 12.6 Count Timing when On-Chip RTC Clock Is Operating

Input Capture Function: Channel 2 has an input capture function (figure 12.7). When using the input capture function, set the TCLK pin to input mode with the TCOE bit in the timer output control register (TOCR) and set the timer operation clock to internal clock or on-chip RTC clock with the TPCS2–TPCS0 bits in the timer control register (TCR2). Also, designate use of the input capture function and whether to generate interrupts on using it with the IPCE1–IPCE0 bits in TCR2, and designate the use of either the rising or falling edge of the TCLK pin to set the timer counter (TNCT2) value into the input capture register (TCPR2) with the CKEG1–CKEG0 bits in TCR2.

The input capture function cannot be used in standby mode.

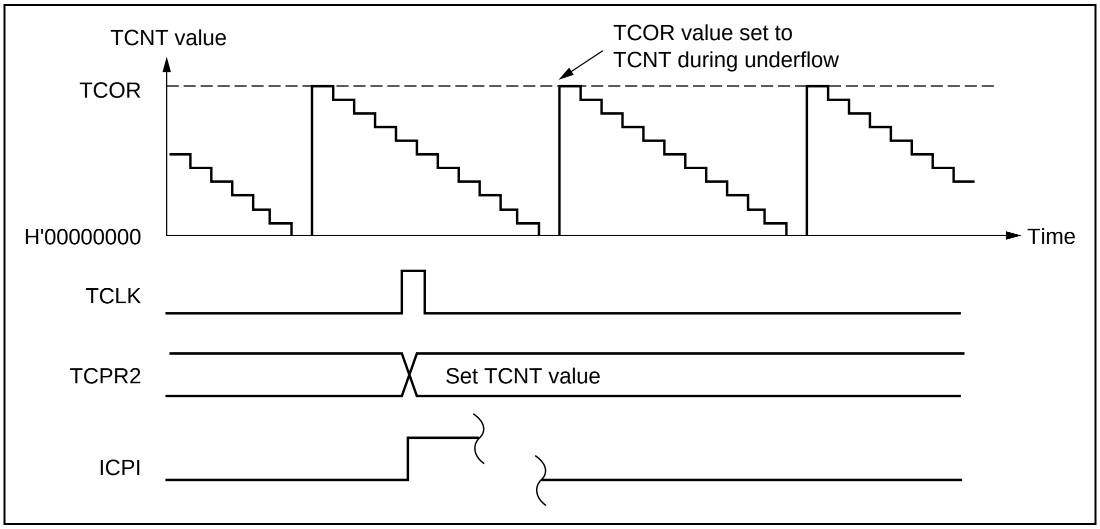


Figure 12.7 Operation Timing when Using the Input Capture Function (Using TCLK Rising Edge)

12.4 Interrupts

There are two sources of TMU interrupts: underflow interrupts and interrupts when using the input capture function.

12.4.1 Status Flag Set Timing

UNF is set to 1 when the TCNT underflows (H'00000000 → H'FFFFFFFF). Figure 12.8 shows the timing.

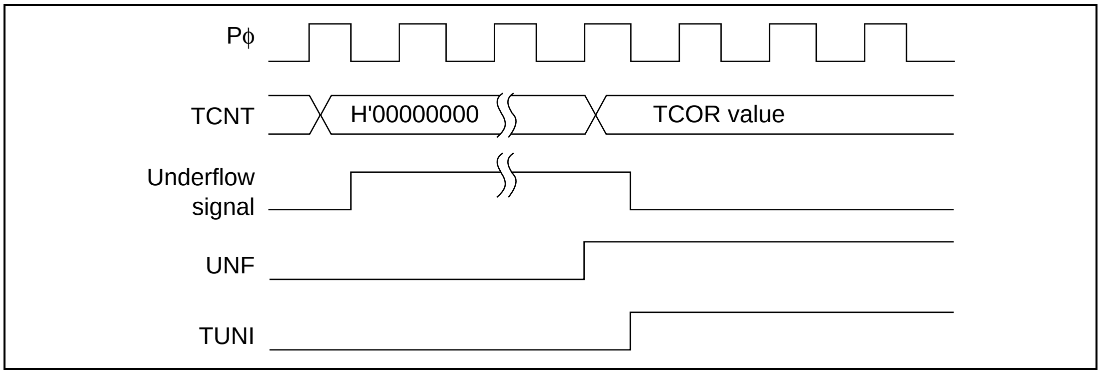


Figure 12.8 UNF Set Timing

12.4.2 Status Flag Clear Timing

The status flag can be cleared by writing a 0 from the CPU. Figure 12.9 shows the timing.

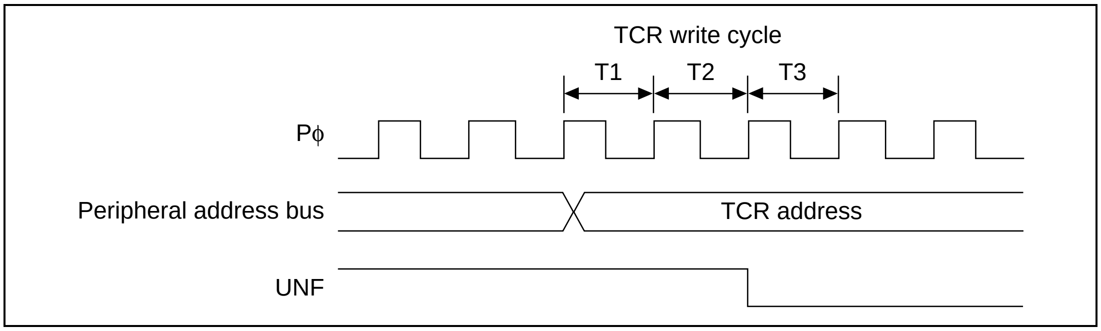


Figure 12.9 Status Flag Clear Timing

12.4.3 Interrupt Sources and Priorities

The TMU produces underflow interrupts for each channel. When the interrupt request flag and interrupt enable bit are both set to 1, the interrupt is requested. Codes are set in the exception source register (INTEVT, INTEVT2) for these interrupts and interrupt handling occurs according to the codes.

The relative priorities of channels can be changed using the interrupt controller (see section 6, Interrupt Controller). Table 12.3 lists TMU interrupt sources.

Table 12.3 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↓
2	TUNI2	Underflow interrupt 2	
2	TICPI2	Input capture interrupt 2	Low

12.5 Usage Notes

12.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits for the channel (STR2–STR0) in the timer start register (TSTR) to halt timer counting.

12.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When timer counting and register read processing are performed simultaneously, the register value before TCNT counting down (with synchronization processing) is read.

Section 13 Realtime Clock (RTC)

13.1 Overview

The SH7709 has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

13.1.1 Features

- Clock and calendar functions (BCD display): seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary display)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: frame comparison of seconds, minutes, hours, date, day of the week, and month can be used as conditions for the alarm interrupt
- Cyclic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year correction

13.1.2 Block Diagram

The following abbreviations are used in the block diagram of the RTC (figure 13.1):

R64CNT:	64-Hz counter	RSECAR:	Second alarm register
RSECCNT:	Second counter	RMINAR:	Minute alarm register
RMINCNT:	Minute counter	RHRAR:	Hour alarm register
RHRCNT:	Hour counter	RWKAR:	Day of the week alarm register
RWKCNT:	Day of the week counter	RDAYAR:	Date alarm register
RDAYCNT:	Date counter	RMONAR:	Month alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2

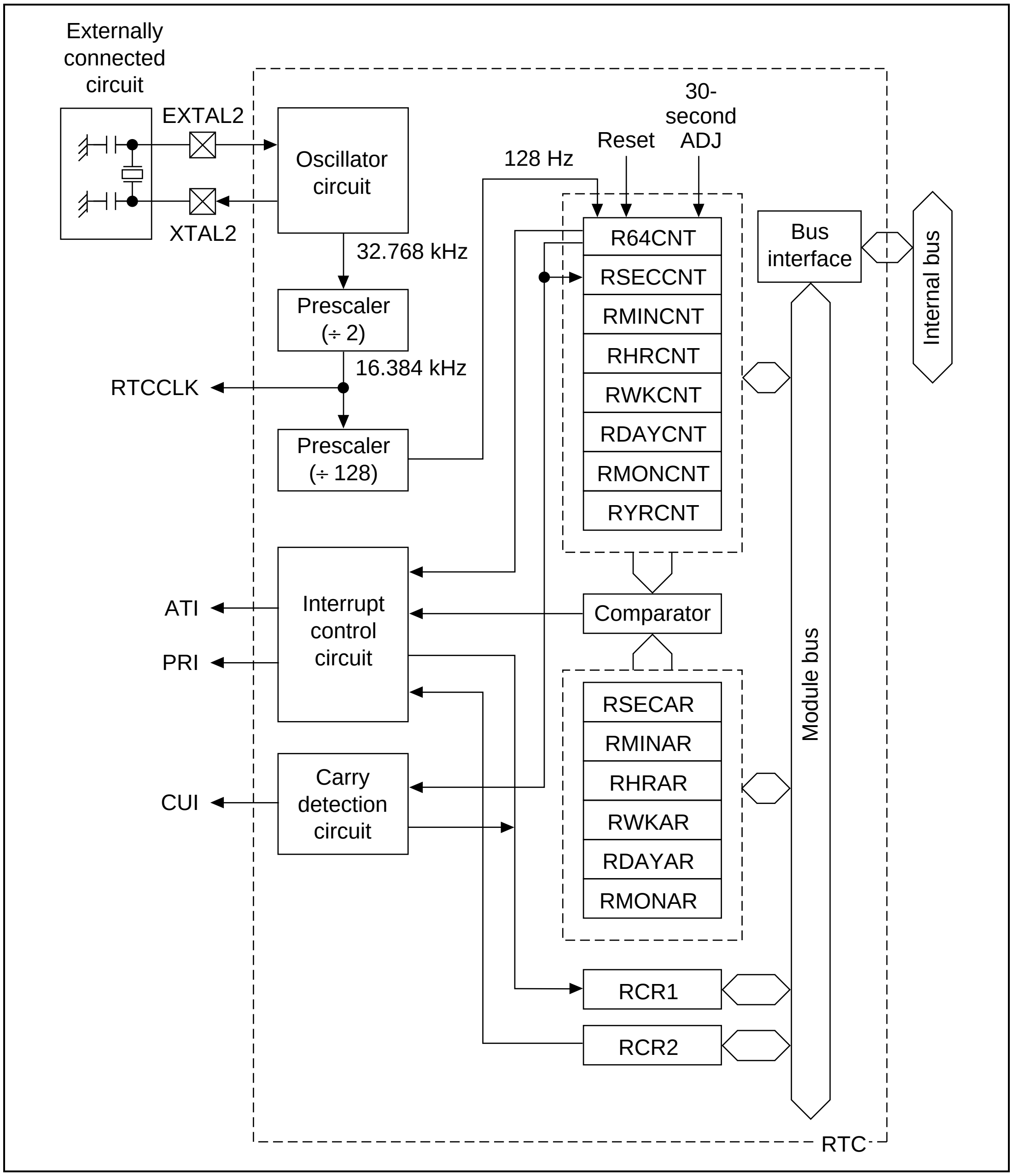


Figure 13.1 RTC Block Diagram

13.1.3 Pin Configuration

Table 13.1 shows the RTC pin configuration.

Table 13.1 RTC Pin Configuration

Pin Name	Symbol	I/O	Description
RTC oscillator crystal pin	EXTAL2	I	Connects crystal to RTC oscillator
RTC oscillator crystal pin	XTAL2	O	Connects crystal to RTC oscillator
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/realtime clock (RTC) output pin (shared by TMU)
Dedicated power-supply pin for RTC	VCC(RTC)	—	Dedicated power-supply pin for RTC*
Dedicated GND pin for RTC	GND(RTC)	—	Dedicated GND pin for RTC*

Note: Power must be supplied to the RTC power supply pins even when the RTC is not used. Even if only the RTC is used, power must be supplied to all power supply pins, including these pins.

In standby mode, also, power must be supplied to all power supply pins, including these pins.

13.1.4 RTC Register Configuration

Table 13.2 shows the RTC register configuration.

Table 13.2 RTC Registers

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	Undefined	H'FFFFFEC0	8
Second counter	RSECCNT	R/W	Undefined	H'FFFFFEC2	8
Minute counter	RMINCNT	R/W	Undefined	H'FFFFFEC4	8
Hour counter	RHRCNT	R/W	Undefined	H'FFFFFEC6	8
Day of week counter	RWKCNT	R/W	Undefined	H'FFFFFEC8	8
Date counter	RDAYCNT	R/W	Undefined	H'FFFFFECA	8
Month counter	RMONCNT	R/W	Undefined	H'FFFFFECC	8
Year counter	RYRCNT	R/W	Undefined	H'FFFFFECE	8
Second alarm register	RSECAR	R/W	Undefined*	H'FFFFFED0	8
Minute alarm register	RMINAR	R/W	Undefined*	H'FFFFFED2	8
Hour alarm register	RHRAR	R/W	Undefined*	H'FFFFFED4	8
Day of week alarm register	RWKAR	R/W	Undefined*	H'FFFFFED6	8
Date alarm register	RDAYAR	R/W	Undefined*	H'FFFFFED8	8
Month alarm register	RMONAR	R/W	Undefined*	H'FFFFFEDA	8
RTC control register 1	RCR1	R/W	H'00	H'FFFFFEDC	8
RTC control register 2	RCR2	R/W	H'09	H'FFFFFEDE	8

Note: Only the ENB bits of each register are initialized.

13.2 RTC Registers

13.2.1 64-Hz Counter (R64CNT)

The 64-Hz counter (R64CNT) is an 8-bit read-only register that indicates the status of the RTC divider circuit between 64 Hz and 1 Hz.

R64CNT is reset to H'00 by setting the RESET bit in RTC control register 2 (RCR2) or the ADJ bit in RCR2 to 1.

R64CNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit 7 always reads 0.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

13.2.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded second section of the RTC. The count operation is performed by a carry for each second of the 64 Hz counter.

The range that can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	10 seconds			1 second			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded minute section of the RTC. The count operation is performed by a carry for each minute of the second counter.

The range that can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	10 minutes			1 minute			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded hour section of the RTC. The count operation is performed by a carry for each 1 hour of the minute counter.

The range that can be set is 00–23 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	10 hours		1 hour			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

13.2.5 Day of the Week Counter (RWKCNT)

The day of the week counter (RWKCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded day of week section of the RTC. The count operation is performed by a carry for each day of the date counter.

The range that can be set is 0–6 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	Day of week		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Days of the week are coded as shown in table 13.3.

Table 13.3 Day-of-Week Codes (RWKCNT)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

13.2.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded date section of the RTC. The count operation is performed by a carry for each day of the hour counter.

The range that can be set is 01–31 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode.

The RDAYCNT range that can be set changes with each month and in leap years. Please confirm the correct setting.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	10 days		1 day			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

13.2.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded month section of the RTC. The count operation is performed by a carry for each month of the date counter.

The range that can be set is 00–12 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

13.2.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded year section of the RTC. The least significant 2 digits of the western calendar year are displayed. The count operation is performed by a carry for each year of the month counter.

The range that can be set is 00–99 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2.

RYRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fractional result of 0.

Bit:	7	6	5	4	3	2	1	0
Bit name:	10 years				1 year			
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded second section counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSECAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	10 seconds			1 second			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded minute section counter RMINCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	10 minutes			1 minute			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13.2.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded hour section counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–23 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	—	10 hours		1 hour			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

13.2.12 Day of the Week Alarm Register (RWKAR)

The day of the week alarm register (RWKAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded day of week section counter RWKCNT of the RTC. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 0–6 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	—	—	—	—	Day of week		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Days of the week are coded as shown in table 13.4.

Table 13.4 Day-of-Week Codes (RWKAR)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

13.2.13 Date Alarm Register (RDAYAR)

The date alarm register (RDAYAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded date section counter RDAYCNT of the RTC. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 01–31 (decimal) + ENB bit. Errant operation will result if any other value is set. The RDAYCNT range that can be set changes with some months and in leap years. Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	—	10 days		1 day			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

13.2.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded month section counter RMONCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 01–12 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

13.2.15 RTC Control Register 1 (RCR1)

The RTC control register 1 (RCR1) is an 8-bit read/write register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag. Because flags are sometimes set after an operand read, do not use this register in read-modify-write processing.

RCR1 is initialized to H'00 by a power-on reset. In a manual reset, all bits are initialized to 0 except for the CF flag, which is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	CF	—	—	CIE	AIE	—	—	AF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit 7—Carry Flag (CF): Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to R64CNT or RSECCNT occurs. A count register value read at this time cannot be guaranteed; another read is required.

Bit 7: CF	Description
0	No count up of R64CNT or RSECCNT. Clearing condition: When 0 is written to CF (Initial value)
1	Count up of R64CNT or RSECCNT. Setting condition: When 1 is written to CF

Bits 6, 5, 2, and 1—Reserved: These bits always read 0. The write value should always be 0.

Bit 4—Carry Interrupt Enable Flag (CIE): When the carry flag (CF) is set to 1, the CIE bit enables interrupts.

Bit 4: CIE	Description
0	A carry interrupt is not generated when the CF flag is set to 1 (Initial value)
1	A carry interrupt is generated when the CF flag is set to 1

Bit 3—Alarm Interrupt Enable Flag (AIE): When the alarm flag (AF) is set to 1, the AIE bit allows interrupts.

Bit 3: AIE	Description
0	An alarm interrupt is not generated when the AF flag is set to 1 (Initial value)
1	An alarm interrupt is generated when the AF flag is set to 1

Bit 0—Alarm Flag (AF): The AF flag is set to 1 when the alarm time set in an alarm register (only registers with ENB bit set to 1) matches the clock and calendar time.

Bit 0: AF	**Description**
0	Clock/counter and alarm register have not matched since last reset to 0. Clearing condition: When 0 is written to AF (Initial value)
1	Setting condition: Clock/counter and alarm register have matched (only registers with ENB set)*

Note: Contents do not change when 1 is written to AF.

13.2.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit read/write register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop control. It is initialized to H'09 by a power-on reset.It is initialized except for RTCEN and START by a manual reset. It is not initialized in standby mode, and retains its contents.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START
Initial value:	0	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 7—Periodic Interrupt Flag (PEF): Indicates interrupt generation with the period designated by the PES bits. When set to 1, PEF generates periodic interrupts.

Bit 7: PEF	**Description**
0	Interrupts not generated with the period designated by the PES bits. Clearing condition: When 0 is written to PEF (Initial value)
1	Interrupts generated with the period designated by the PES bits. Setting condition: When 1 is written to PEF

Bits 6–4—Periodic Interrupt Flags (PES2–PES0): These bits specify the periodic interrupt.

Bit 6: PES2	Bit 5: PES1	Bit 4: PES0	Description
0	0	0	No periodic interrupts generated (Initial value)
		1	Periodic interrupt generated every 1/256 second
	1	0	Periodic interrupt generated every 1/64 second
		1	Periodic interrupt generated every 1/16 second
1	0	0	Periodic interrupt generated every 1/4 second
		1	Periodic interrupt generated every 1/2 second
	1	0	Periodic interrupt generated every 1 second
		1	Periodic interrupt generated every 2 seconds

Bit 3—RTCEN: Controls the operation of the crystal oscillator for the RTC.

Bit 3: RTCEN	Description
0	Halts the crystal oscillator for the RTC.
1	Runs the crystal oscillator for the RTC. (Initial value)

Bit 2—30-Second Adjustment (ADJ): When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit will be simultaneously reset. This bit always reads 0.

Bit 2: ADJ	Description
0	Runs normally. (Initial value)
1 (write)	30-second adjustment.

Bit 1—Reset (RESET): When 1 is written, initializes the divider circuit. This bit always reads 0.

Bit 1: RESET	Description
0	Runs normally. (Initial value)
1 (Write)	Divider circuit is reset.

Bit 0—Start Bit (START): Halts and restarts the counter (clock).

Bit 0: START	Description
0	Second/minute/hour/day/week/month/year counter halts.
1	Second/minute/hour/day/week/month/year counter runs normally. (Initial value)

Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

13.3 RTC Operation

13.3.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

13.3.2 Setting the Time

Part (a) in figure 13.2 shows how to set the time when the clock is stopped. This works when the entire calendar or clock is to be set.

Part (b) in figure 13.2 describes how to set the clock when the clock is running. This works when only part of the calendar or clock needs to be reset (e.g., changing only the seconds or only the hour). The write status is checked using the carry flags. When there is a carry during the writing of new data, the new data is automatically updated. Since this causes errors in the data, the data must be rewritten if the carry flag is set to 1.

The interrupt function can be used to determine the status of the carry flag.

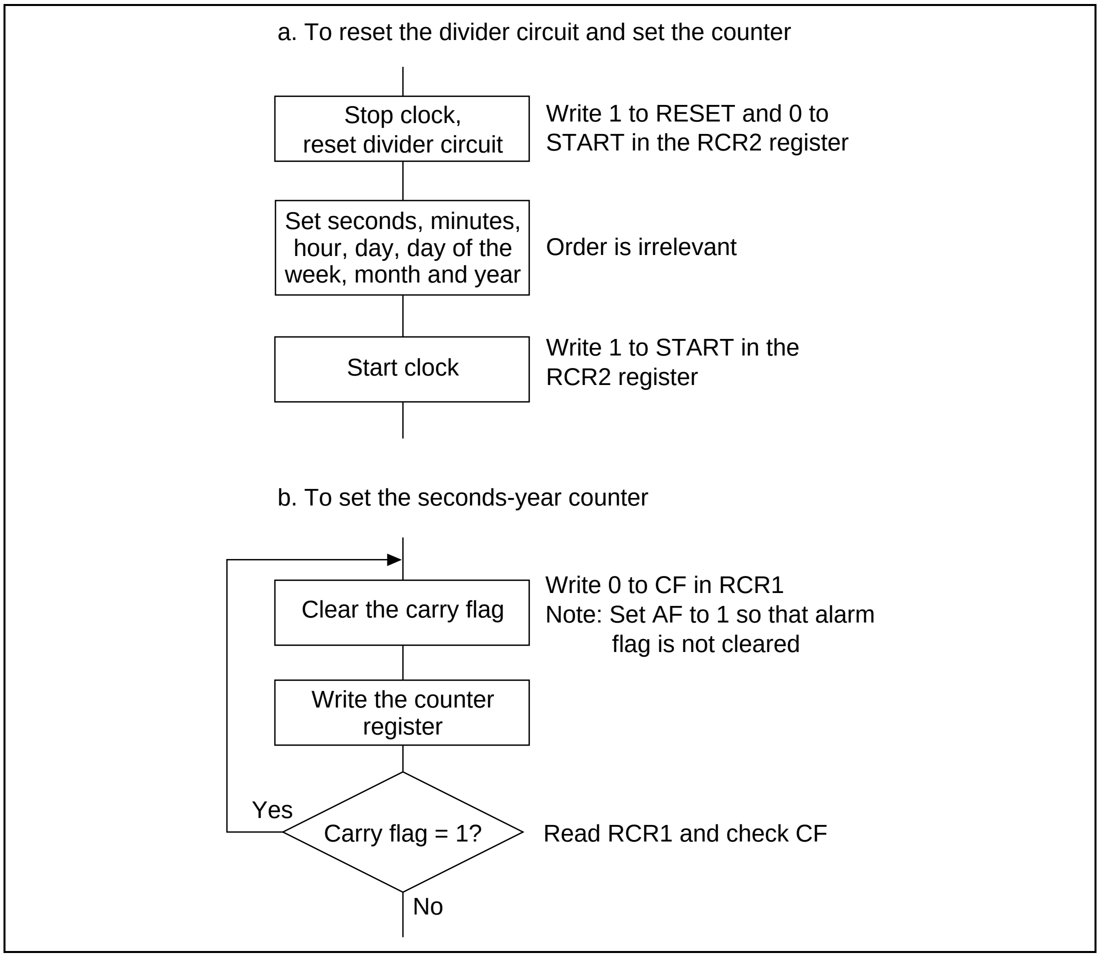

a. To reset the divider circuit and set the counter
Stop clock, reset divider circuit
Write 1 to RESET and 0 to START in the RCR2 register
Set seconds, minutes, hour, day, day of the week, month and year
Order is irrelevant
Start clock
Write 1 to START in the RCR2 register
b. To set the seconds-year counter
Clear the carry flag
Write 0 to CF in RCR1
Note: Set AF to 1 so that alarm flag is not cleared
Write the counter register
Carry flag = 1?
Yes
No
Read RCR1 and check CF

Figure 13.2 Setting the Time

13.3.3 Reading the Time

Figure 13.3 shows how to read the time. If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 13.3 shows the method of reading the time without using interrupts; part (b) in figure 13.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

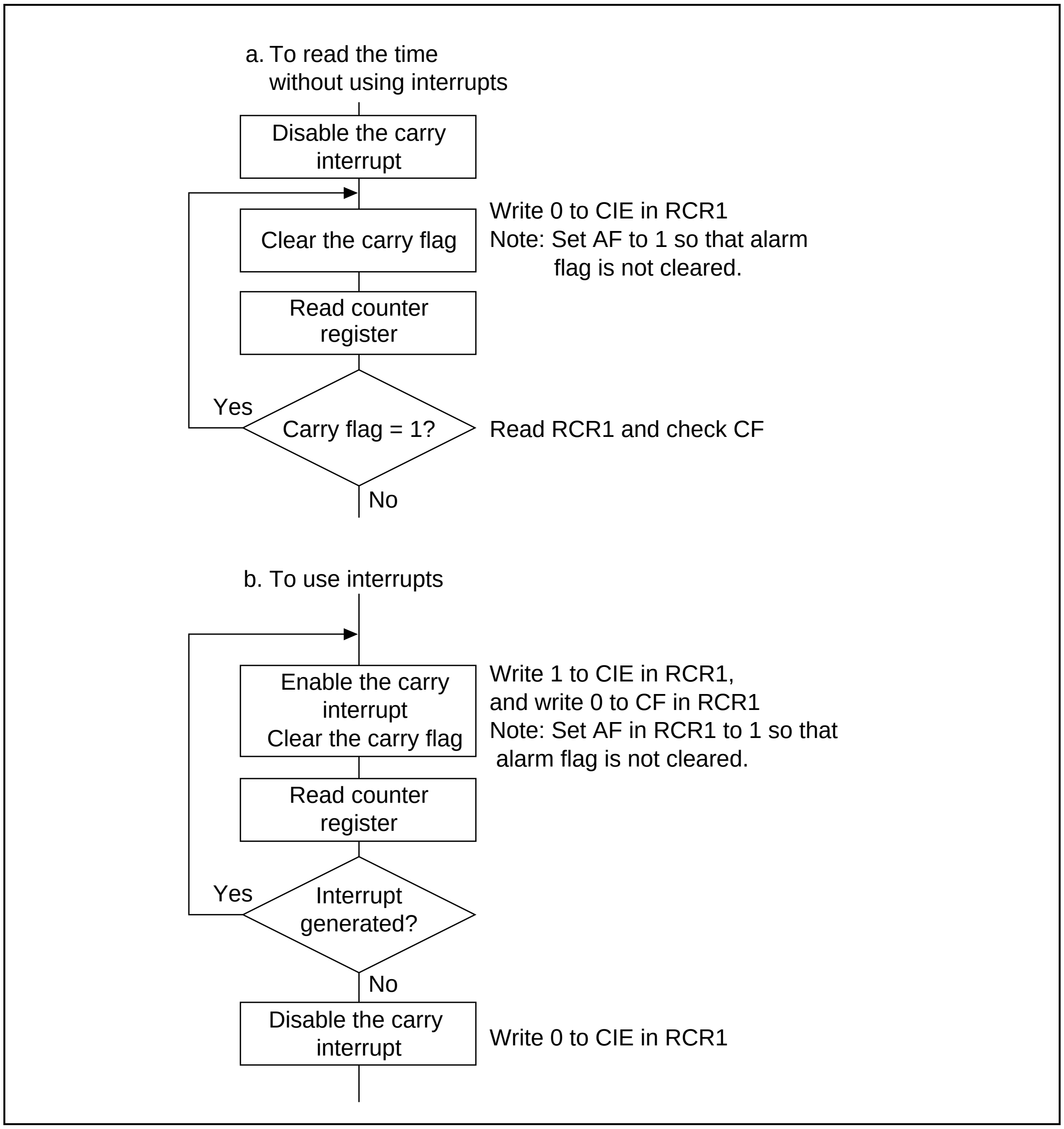


Figure 13.3 Reading the Time

13.3.4 Alarm Function

Figure 13.4 shows how to use the alarm function.

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, or any combination of these. Set the ENB bit (bit 7) in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is placed to 0.

When the clock and alarm times match, a 1 is set in the AF bit (bit 0) in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the AIE bit (bit 3) in RCR1, an interrupt is generated when an alarm occurs.

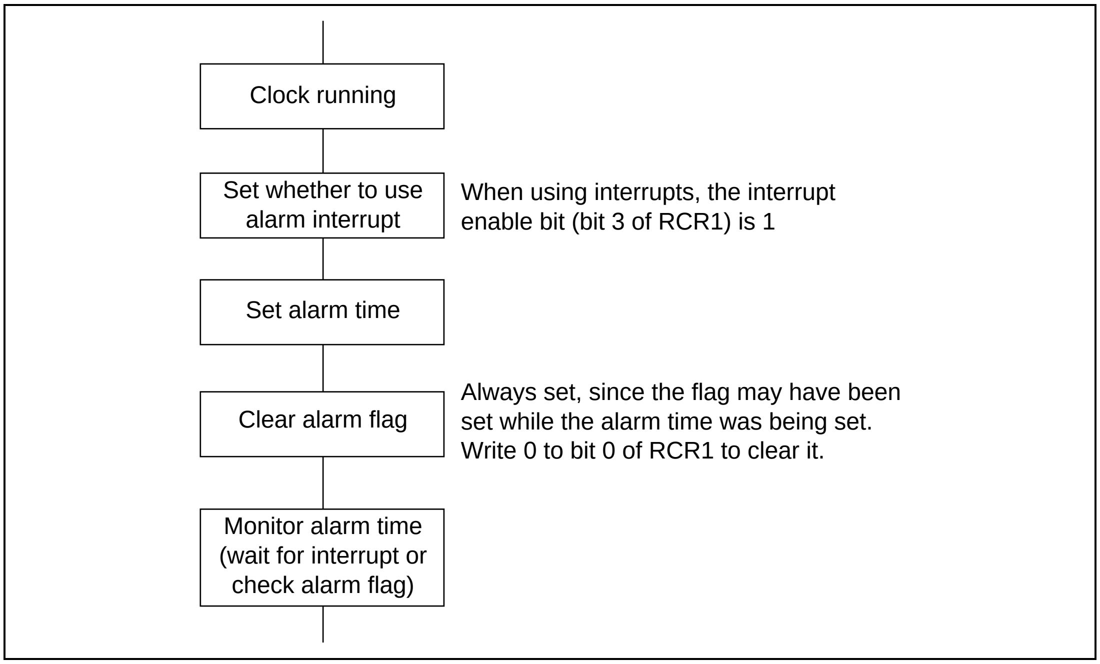

Clock running
Set whether to use alarm interrupt
When using interrupts, the interrupt enable bit (bit 3 of RCR1) is 1
Set alarm time
Clear alarm flag
Always set, since the flag may have been set while the alarm time was being set. Write 0 to bit 0 of RCR1 to clear it.
Monitor alarm time (wait for interrupt or check alarm flag)

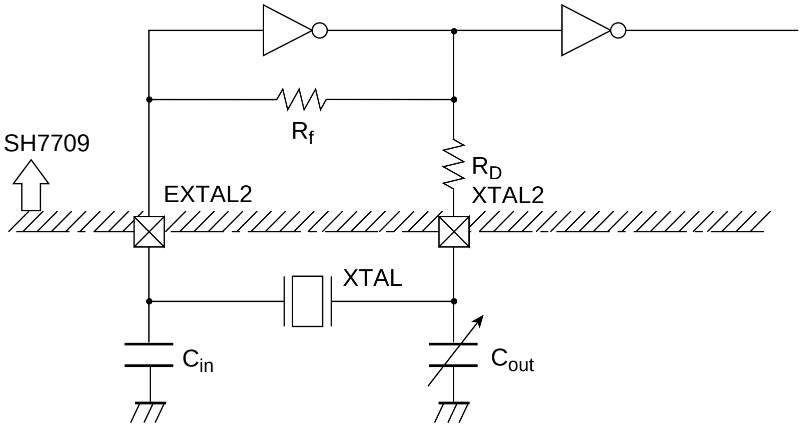
Figure 13.4 Using the Alarm Function

13.3.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 13.5, and the RTC crystal oscillator circuit in figure 13.5.

Table 13.5 Recommended Oscillator Circuit Constants (Recommended Values)

fosc	Cin	Cout
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes: 1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.

2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ

3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.

4. The crystal oscillation settling 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, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 13.5 Example of Crystal Oscillator Circuit Connection

Section 14 Serial Communication Interface (SCI)

14.1 Overview

The SH-3 has an on-chip serial communication interface (SCI) that supports both asynchronous and clock synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors. The SCI supports a smart card interface, which is a serial communications feature for IC card interfaces that conforms to the ISO/IEC standard 7816-3 for identification cards. See section 15, Smart Card Interface, for more information.

14.1.1 Features

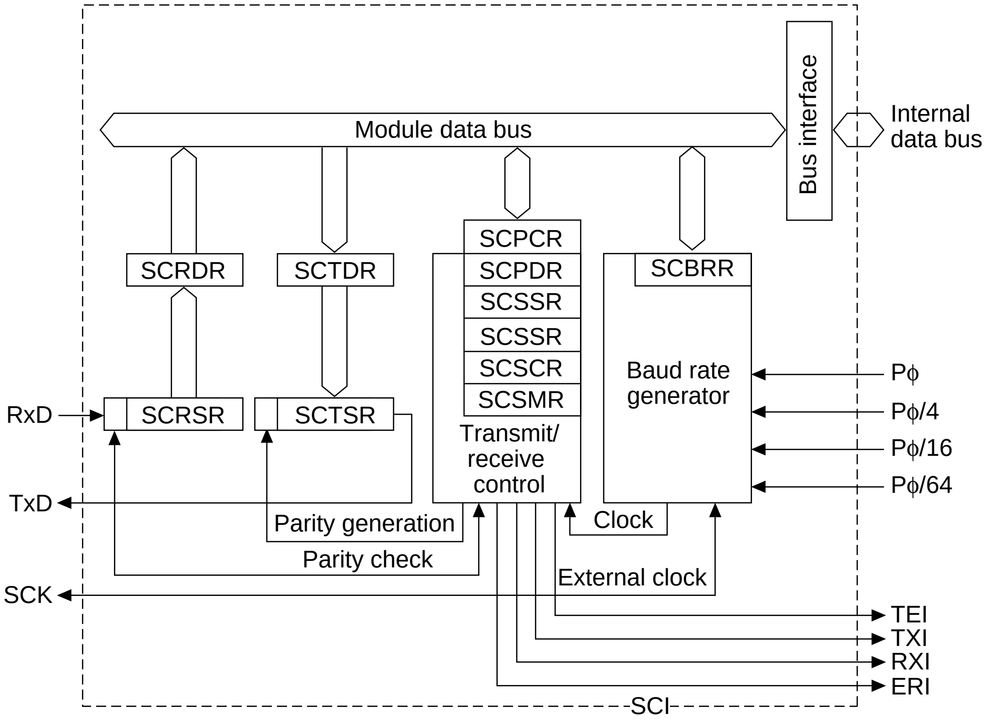
Select asynchronous or clock synchronous as the serial communications mode.

- Asynchronous mode:
  - Serial data communications are synched by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Multiprocessor bit: 1 or 0
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: By reading the RxD level directly from the port SC data register (SCPDR) when a framing error occurs
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: Eight bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)

- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.
- When the SCI is not in use, it can be stopped by halting the clock supplied to it, saving power.

14.1.2 Block Diagram

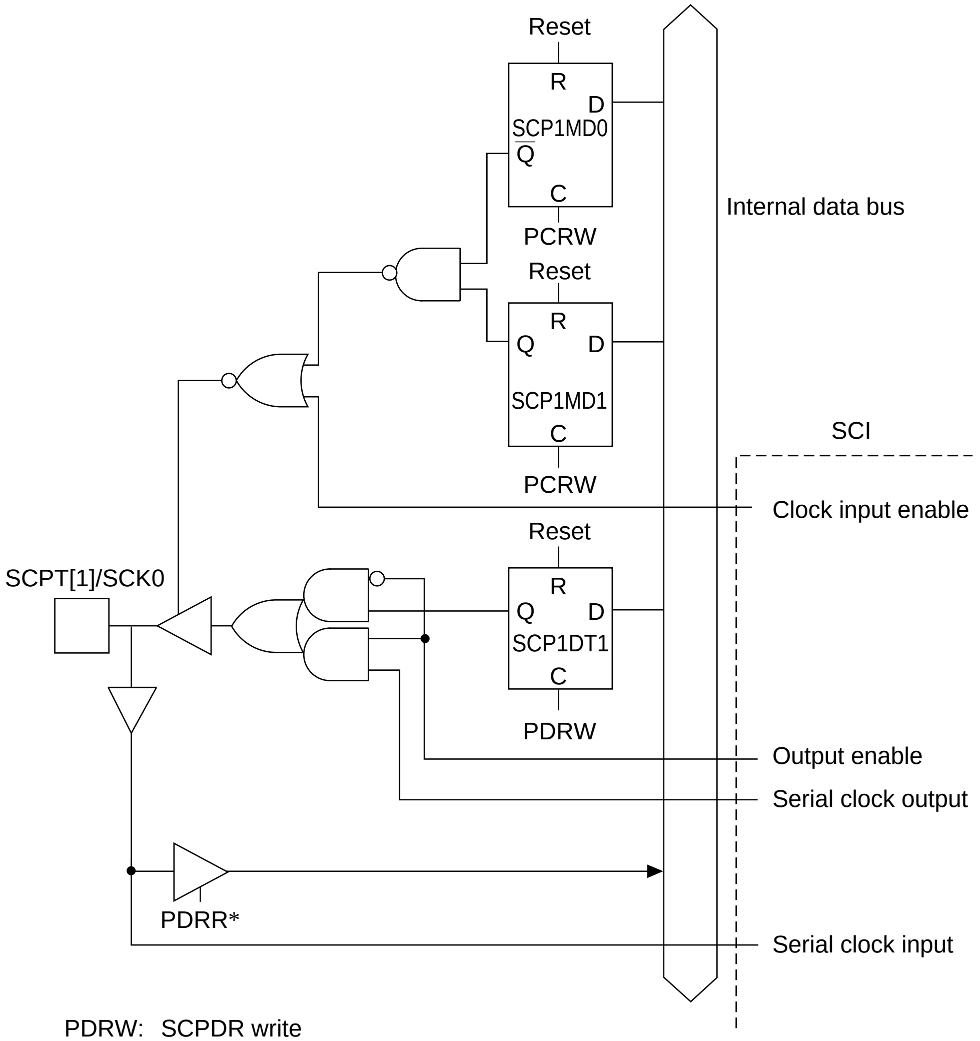
Figure 14.1 shows a block diagram of the SCI.



SCRSR:	Receive shift register	SCSCR:	Serial control register
SCRDR:	Receive data register	SCSSR:	Serial status register
SCTSR:	Transmit shift register	SCBRR:	Bit rate register
SCTDR:	Transmit data register	SCPDR:	Port SC data register
SCSMR:	Serial mode register	SCPCR:	Port SC control register

Figure 14.1 SCI Block Diagram

Figures 14.2, 14.3, and 14.4 show the block diagrams of the SCI I/O port.



PDRW: SCPDR write
PDRR: SCPDR read
PCRW: SCPCR write

Note: * When reading the SCK0 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP1MD1 bit in SCPCR to 1 (see section 14.2.8).

Figure 14.2 SCPT[1]/SCK0 Pin

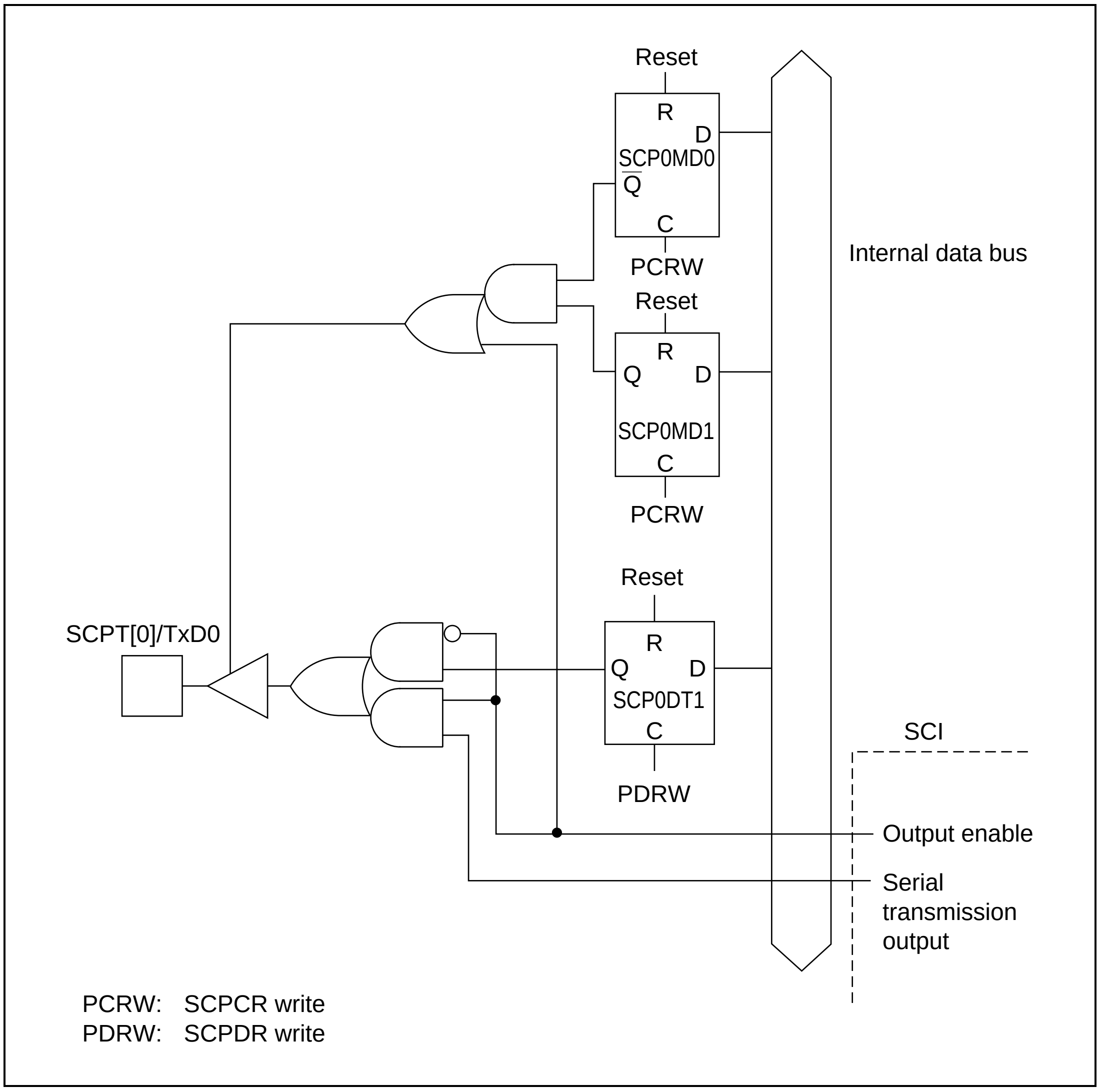
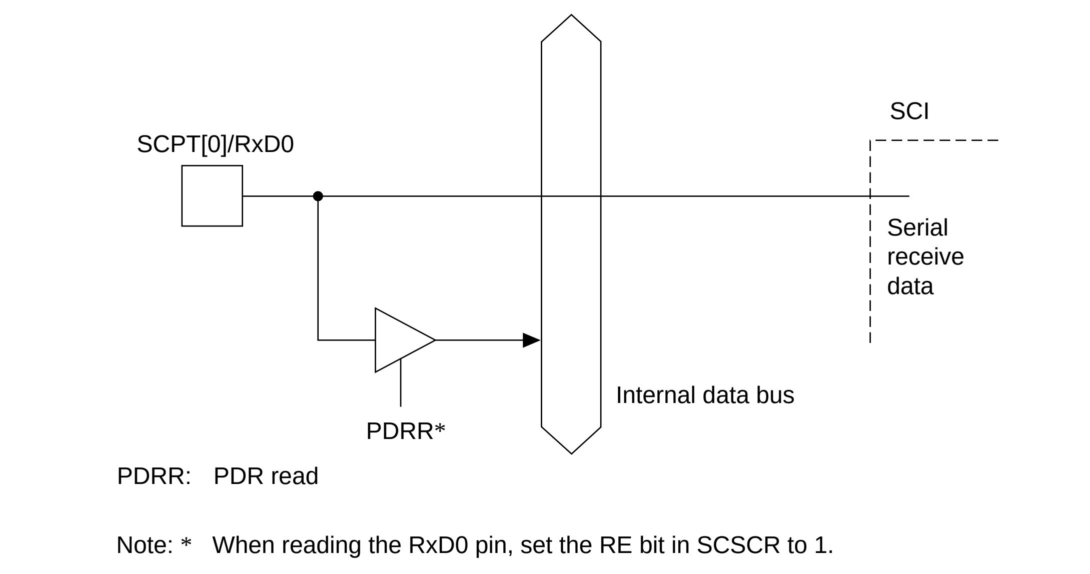


Figure 14.3 SCPT[0]/TxD0 Pin



PDRR: PDR read

Note: * When reading the RxD0 pin, set the RE bit in SCSCR to 1.

Figure 14.4 SCPT[0]/RxD0 Pin

14.1.3 Pin Configuration

The SCI has the serial pins summarized in table 14.1.

Table 14.1 SCI Pins

Pin Name	Symbol	Input/Output	Function
Serial clock pin	SCK0	Input/output	Clock input/output
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

Note: They are made to function as serial pins by performing SCI operation settings with the TE, RE, CKEI, and CKEO bits in SCSCR and the C/$\overline{A}$ bit in SCSMR. Break status transmission and detection can be performed by means of the SCIsSCSPTR register.

14.1.4 Register Configuration

Table 14.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 15.2 Registers

Register Name	Abbr.	R/W	Initial Value	Address	Access size
Serial mode register	SCSMR	R/W	H'00	H'FFFFFE80	8
Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*[1]	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
Port SC data register	SCPDR	R/W	H'00	H'04000136	8
Port SC control register	SCPCR	R/W	H'A888	H'04000116	16

Notes: 1. The only value that can be written is a 0 to clear the flags.

2. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

14.2 Register Descriptions

14.2.1 Receive Shift Register

The receive shift register (SCRSR) receives serial data. Data input at the RxD pin is loaded into the SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCRDR. The CPU cannot read or write the SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	—	—	—	—	—	—	—	—

14.2.2 Receive Data Register

The receive data register (SCRDR) stores serial receive data. The SCI completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into the SCRDR for storage. The SCRSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write the SCRDR. The SCRDR is initialized to H'00 by a reset or in standby or module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

14.2.3 Transmit Shift Register

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR) into the SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the SCTDR into the SCTSR and starts transmitting again. If the TDRE bit of the SCSSR is 1, however, the SCI does not load the SCTDR contents into the SCTSR. The CPU cannot read or write the SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	—	—	—	—	—	—	—	—

14.2.4 Transmit Data Register

The transmit data register (SCTDR) is an eight-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCTDR into the SCTSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in the SCTDR during serial transmission from the SCTSR.

The CPU can always read and write the SCTDR. The SCTDR is initialized to H'FF by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

14.2.5 Serial Mode Register

The serial mode register (SCSMR) is an eight-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR. The SCSMR is initialized to H'00 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0
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 7—Communication Mode (C/$\overline{\text{A}}$): Selects whether the SCI operates in the asynchronous or clock synchronous mode.

Bit 7: C/A	Description
0	Asynchronous mode (initial value)
1	Clock synchronous mode

Bit 6—Character Length (CHR): Selects seven-bit or eight-bit data in the asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	Description
0	Eight-bit data (initial value)
1	Seven-bit data. (When seven-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted.)

Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data, in the asynchronous mode. In the clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.

Bit 5: PE	Description
0	Parity bit not added or checked (initial value)
1	Parity bit added and checked. When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.

Bit 4—Parity Mode ($O/\overline{E}$): Selects even or odd parity when parity bits are added and checked. The $O/\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The $O/\overline{E}$ setting is ignored in the clock synchronous mode, or in the asynchronous mode when parity addition and check is disabled.

Bit 4: O/E	Description
0	Even parity (initial value). If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
1	Odd parity. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length in the asynchronous mode. This setting is used only in the asynchronous mode. It is ignored in the clock synchronous mode because no stop bits are added.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description
0	One stop bit (initial value). In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two bits of 1 are added at the end of each transmitted character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode ($O/\overline{E}$) bits are ignored. The MP bit setting is used only in the asynchronous mode; it is ignored in the clock synchronous mode. For the multiprocessor communication function, see section 14.3.3, Multiprocessor Communication.

Bit 2: MP	Description
0	Multiprocessor function disabled (initial value)
1	Multiprocessor format selected

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 14.2.9, Bit Rate Register.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ (initial value)
	1	Pϕ/4
1	0	Pϕ/16
	1	Pϕ/64

Note: Pϕ: Peripheral clock

14.2.6 Serial Control Register

The serial control register (SCSCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR. The SCSCR is initialized to H'00 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
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 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the transmit data register empty bit (TDRE) in the serial status register (SCSSR) is set to 1 due to transfer of serial transmit data from the SCTDR to the SCTSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TXI) is disabled (initial value). The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.
1	Transmit-data-empty interrupt request (TXI) is enabled

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the receive data register full bit (RDRF) in the serial status register (SCSSR) is set to 1 due to transfer of serial receive data from the SCRSR to the SCRDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled (initial value). RXI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled.

Bit 5—Transmit Enable (TE): Enables or disables the SCI serial transmitter.

Bit 5: TE	Description
0	Transmitter disabled (initial value). The transmit data register empty bit (TDRE) in the serial status register (SCSSR) is locked at 1.
1	Transmitter enabled. Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SCSSR) is cleared to 0 after writing of transmit data into the SCTDR. Select the transmit format in the SCSMR before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled (initial value). Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
1	Receiver enabled. Serial reception starts when a start bit is detected in the asynchronous mode, or synchronous clock input is detected in the clock synchronous mode. Select the receive format in the SCSMR before setting RE to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in the asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SCSMR) is set to 1 during reception. The MPIE setting is ignored in the clock synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (initial value). MPE is cleared to 0 when MPIE is cleared to 0, or the multiprocessor bit (MPB) is set to 1 in receive data.
1	Multiprocessor interrupts are enabled. Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SCSSR) are disabled until data with a multiprocessor bit of 1 is received. The SCI does not transfer receive data from the SCRSR to the SCRDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SCSSR). When it receives data that includes MPB = 1, the SCSSR's MPB flag is set to 1, and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (if the TIE and RIE bits in the SCSCR are set to 1), and allows the FER and ORER bits to be set.

Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if SCTDR does not contain new transmit data when the MSB is transmitted.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) requests are disabled* (initial value)
1	Transmit-end interrupt (TEI) requests are enabled.*

Note: The TEI request can be cleared by reading the TDRE bit in the serial status register (SCSSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0, or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only in the asynchronous mode, and only when the SCI is internally clock (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or when an external clock source is selected (CKE1 = 1). Before selecting the SCI operating mode in the serial mode register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clock source, see table 14.9 in section 14.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored) (Initial value)
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output (Initial value)
	1	Asynchronous mode	Internal clock, SCK pin used for clock output*1
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*2
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input
	1	Asynchronous mode	External clock, SCK pin used for clock input*2
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input

Notes: 1. The output clock frequency is the same as the bit rate.

2. The input clock frequency is 16 times the bit rate.

14.2.7 Serial Status Register

The serial status register (SCSSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate SCI operating status.

The CPU can always read and write the SCSSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written. The SCSSR is initialized to H'84 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: The only value that can be written is a 0 to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from the SCTDR into the SCTSR and new serial transmit data can be written in the SCTDR.

Bit 7: TDRE	Description
0	SCTDR contains valid transmit data. TDRE is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE or data is written in SCTDR.
1	SCTDR does not contain valid transmit data (initial value) TDRE is set to 1 when the chip is reset or enters standby mode, the TE bit in the serial control register (SCSCR) is cleared to 0, or SCTDR contents are loaded into SCTSR, so new data can be written in SCTDR.

Bit 6—Receive Data Register Full (RDRF): Indicates that SCRDR contains received data.

Bit 6: RDRF	Description
0	SCRDR does not contain valid received data (initial value). RDRF is cleared to 0 when the chip is reset or enters standby mode, software reads RDRF after it has been set to 1, then writes 0 in RDRF, or data is read from SCRDR.
1	SCRDR contains valid received data. RDRF is set to 1 when serial data is received normally and transferred from SCRSR to SCRDR.

Note: The SCRDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the received data is lost.

Bit 5—Overrun Error (ORER): Indicates that data reception aborted due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally (initial value).*[1] ORER is cleared to 0 when the chip is reset or enters standby mode or software reads ORER after it has been set to 1, then writes 0 in ORER.
1	A receive overrun error occurred.*[2] ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1.

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value.

2. SCRDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception aborted due to a framing error in the asynchronous mode.

Bit 4: FER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value. FER is cleared to 0 when the chip is reset or enters standby mode or software reads FER after it has been set to 1, then writes 0 in FER.
1	A receive framing error occurred. When the stop bit length is two bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0.

Bit 3—Parity Error (PER): Indicates that data reception (with parity) aborted due to a parity error in the asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally (initial value). Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value. PER is cleared to 0 when the chip is reset or enters standby mode or software reads PER after it has been set to 1, then writes 0 in PER.
1	A receive parity error occurred. When a parity error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In the clock synchronous mode, serial transmitting is also disabled. PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/E) in the serial mode register (SCSMR).

Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the SCTDR did not contain valid data, so transmission has ended. TEND is a read-only bit and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE, or data is written in SCTDR.
1	End of transmission (initial value). TEND is set to 1 when the chip is reset or enters standby mode, TE is cleared to 0 in the serial control register (SCSCR), or TDRE is 1 when the last bit of a one-byte serial character is transmitted.

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in the asynchronous mode. The MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0 (initial value). If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value.
1	Multiprocessor bit value in receive data is 1.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in the asynchronous mode. The MPBT setting is ignored in the clock synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (initial value)
1	Multiprocessor bit value in transmit data is 1

14.2.8 Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR)

The port SC control register (SCPCR) and port SC data register (SCPDR) control input/output and data for the port multiplexed with the serial communication interface (SCI) pins.

SCPCR settings are used to perform input/output control, to enable data written in SCPDR to be output to the TxD pin, and input data to be read from the RxD pin, and to control serial transmission/reception breaks.

It is also possible to read data on the SCK pin, and write output data.

SCPCR

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	SCP7 MD1	SCP7 MD0	SCP6 MD1	SCP6 MD0	SCP5 MD1	SCP5 MD0	SCP4 MD1	SCP4 MD0	SCP3 MD1	SCP3 MD0	SCP2 MD1	SCP2 MD0	SCP1 MD1	SCP1 MD0	SCP0 MD1	SCP0 MD0
Initial value:	1	0	1	0	1	0	0	0	1	0	0	0	1	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

SCPDR

Bit:	7	6	5	4	3	2	1	0
Bit name:	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT
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

Note: * = Undefined

SCI pin input/output and data control are performed by bits 3—0 of SCPCR and bits 1 and 0 of SCPDR.

Serial Port Clock Port Input/Output (SCP1MD1, SCP1MD0): These bits specify serial port SCK pin input/output. When the SCK pin is actually used as a port input/output pin, clear the C/$\overline{\text{A}}$ bit of SCSMR and bits CKE1 and CKE0 of SCSCR to 0.

Bit 3	Bit 2	
SCP1MD1	**SCP1MD0**	**Description**
0	0	SCP1DT bit value is not output to SCK pin
0	1	SCP1DT bit value is output to SCK pin
1	0	SCK pin value is read from SCP1DT bit
1	1	

Serial Port Clock Port Data (SCP1DT): Specifies the serial port SCK pin input/output data. Input or output is specified by the SCP1MD0 and SCP1MD1 bits. In output mode, the value of the SCP1DT bit is output to the SCK pin. In input mode, the SCK pin value is read from the SCP1DT bit.

Bit 1	
SCP1DT	**Description**
0	Input/output data is low
1	Input/output data is high

Serial Port Break Input/Output (SCP0MD1, SCP0MD0): These bits specify the serial port TxD pin output condition. When the TxD pin is actually used as a port output pin and outputs the value set with the SCP0DT bit, clear the TE bit of SCSCR to 0.

Bit 1	Bit 0	
SCP0MD1	**SCP0MD0**	**Description**
0	0	SCP0DT bit value is not output to TxD pin
0	1	SCP0DT bit value is output to TxD pin

Serial Port Break Data (SCP0DT): Specifies the serial port RxD pin input data and TxD pin output data. The TxD pin output condition is specified by the SCP0MD0 and SCP0MD1 bits. When the TxD pin is set to output mode, the value of the SCP0DT bit is output to the TxD pin. The RxD pin value is read from the SCP0DT bit regardless of the values of the SCP0MD0 and SCP0MD1 bits, if SCSCR.RE is set to 1. The initial value of this bit after a power-on reset is undefined.

Bit 0	
SCP0DT	**Description**
0	Input/output data is low
1	Input/output data is high

Block diagrams of the SCI I/O ports are shown in figures 14.2, 14.3, and 14.4.

14.2.9 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR. The SCBRR is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in the two channels.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

The SCBRR setting is calculated as follows:

Asynchronous mode: $N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

Clock synchronous mode: $N = [P\phi/(8 \times 2^{2n-1} \times B)] \times 10^6 - 1$

B: Bit rate (bit/s)

N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)

Pϕ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 14.3.)

Table 14.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error for the asynchronous mode by the following formula:
$\text{Error (\%)} = \{P(\phi \times 10^6)/[(N + 1) \times B \times 64 \times 2^{2n-1}] - 1\} \times 100$

Table 14.4 lists examples of SCBRR settings in the asynchronous mode; table 14.5 lists examples of SCBRR settings in the clock synchronous mode.

Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode

	P ϕ (MHz)								
	2			**2.097152**			**2.4576**		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26
150	1	103	0.16	1	108	0.21	1	127	0.00
300	0	207	0.16	0	217	0.21	0	255	0.00
600	0	103	0.16	0	108	0.21	0	127	0.00
1200	0	51	0.16	0	54	–0.70	0	63	0.00
2400	0	25	0.16	0	26	1.14	0	31	0.00
4800	0	12	0.16	0	13	–2.48	0	15	0.00
9600	0	6	–6.99	0	6	–2.48	0	7	0.00
19200	0	2	8.51	0	2	13.78	0	3	0.00
31250	0	1	0.00	0	1	4.86	0	1	22.88
38400	0	1	–18.62	0	0	–14.67	0	1	0.00

	P ϕ (MHz)								
	3			**3.6864**			**4**		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	212	0.03	2	64	0.70	2	70	0.03
150	1	155	0.16	1	191	0.00	1	207	0.16
300	1	77	0.16	1	95	0.00	1	103	0.16
600	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	4	–2.34	0	5	0.00	0	6	–6.99
31250	0	2	0.00	—	—	—	0	3	0.00
38400	—	—	—	0	2	0.00	0	2	8.51

Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	P φ (MHz)								
	4.9152			5			6		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	86	0.31	2	88	–0.25	2	106	–0.44
150	1	255	0.00	2	64	0.16	2	77	0.16
300	1	127	0.00	1	129	0.16	1	155	0.16
600	0	255	0.00	1	64	0.16	1	77	0.16
1200	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	7	0.00	0	7	1.73	0	9	–2.34
31250	0	4	–1.70	0	4	0.00	0	5	0.00
38400	0	3	0.00	0	3	1.73	0	4	–2.34

	P φ (MHz)								
	6.144			7.3728			8		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	79	0.00	2	95	0.00	2	103	0.16
300	1	159	0.00	1	191	0.00	1	207	0.16
600	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	0.00	0	5	0.00	0	6	–6.99

Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	P ϕ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	174	–0.26	2	177	–0.25	1	212	0.03	2	217	0.08
150	1	127	0.00	2	129	0.16	1	155	0.16	2	159	0.00
300	0	255	0.00	2	64	0.16	1	77	0.16	2	79	0.00
600	0	127	0.00	1	129	0.16	0	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	0	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	38	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	19	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	9	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	4	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	2	0.00	0	11	2.40
38400	0	1	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	P ϕ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	2	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	1	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73

Table 14.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	P ϕ (MHz)											
	24			24.576			28.7			30		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73

Table 14.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode

	P ϕ (MHz)									
Bit Rate	**4**		**8**		**16**		**28.7**		**30**	
(bits/s)	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**
110	—	—	—	—	—	—	—	—	—	—
250	1	249	3	124	3	249	—	—	—	—
500	1	124	2	249	3	124	3	223	3	
1k	1	249	2	124	2	249	3	111	3	
2.5k	1	99	1	199	2	99	2	178	2	
5k	0	199	1	99	1	199	2	89	2	
10k	0	99	0	199	1	99	1	178	1	
25k	0	39	0	79	0	159	1	71	1	
50k	0	19	0	39	0	79	0	143	0	
100k	0	9	0	19	0	39	0	71	0	
250k	0	3	0	7	0	15	0	28	0	
500k	0	1	0	3	0	7	0	13	0	
1M	0	0*	0	1	0	3	0	6	0	

Note: Settings with an error of 1% or less are recommended.
Blank: No setting possible
— : Setting possible, but error occurs
* : Continuous transmit/receive not possible

Table 14.6 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used. Tables 14.7 and 14.8 list the maximum rates for external clock input.

Table 14.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 14.7 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

Table 14.8 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0

14.3 Operation

14.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses. Asynchronous/clock synchronous mode and the transmission format are selected in the serial mode register (SCSMR), as listed in table 14.9. The SCI clock source is selected by the combination of the C/$\overline{\text{A}}$ bit in the serial mode register (SCSMR) and the CKE1 and CKE0 bits in the serial control register (SCSCR), as listed in table 14.10.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER) and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The transmission/reception format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 14.9 Serial Mode Register Settings and SCI Communication Formats

	SCSMR Settings					SCI Communication Format			
Mode	**Bit 7 C/A**	**Bit 6 CHR**	**Bit 5 PE**	**Bit 2 MP**	**Bit 3 STOP**	**Data Length**	**Parity Bit**	**Multiprocessor Bit**	**Stop Bit Length**
Asynchronous	0	0	0	0	0	8-bit	Not set	Not set	1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
		1	0		0	7-bit	Not set		1 bit
					1				2 bits
			1		0		Set		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Clock synchronous	1	*	*	*	*	8-bit		Not set	None

Note: Asterisks (*) indicate don't-care bits.

Table 14.10 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSMR	SCSCR Settings		SCI Transmit/Receive Clock	
Mode	**Bit 7 C/A**	**Bit 1 CKE1**	**Bit 0 CKE0**	**Clock Source**	**SCK Pin Function**
Asynchronous mode	0	0	0	Internal	SCI does not use the SCK pin
			1		Outputs a clock with frequency matching the bit rate
		1	0	External	Inputs a clock with frequency 16 times the bit rate
			1		
Clock synchronous mode	1	0	0	Internal	Outputs the synchronous clock
			1		
		1	0	External	Inputs the synchronous clock
			1		

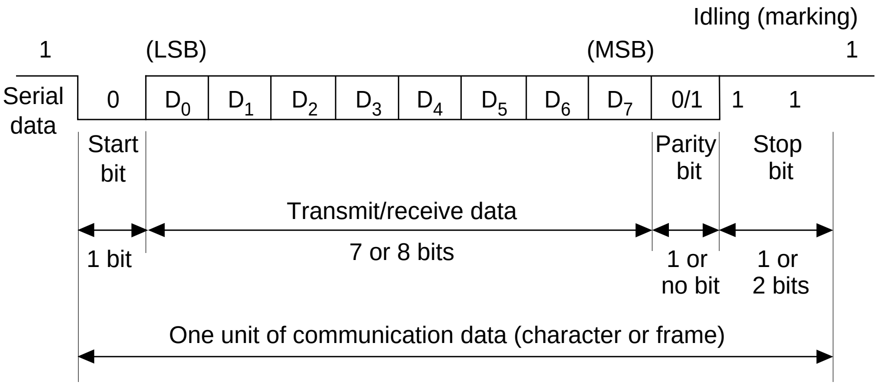
14.3.2 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 14.5 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.



Example: 8-bit data with parity and two stop bits

Figure 14.5 Data Format in Asynchronous Communication

Transmit/Receive Formats: Table 14.11 lists the 12 communication formats that can be selected in the asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 14.11 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-Bit data								STOP		
0	0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	0	START	8-Bit data								P	STOP	
0	1	0	1	START	8-Bit data								P	STOP	STOP
1	0	0	0	START	7-Bit data							STOP			
1	0	0	1	START	7-Bit data							STOP	STOP		
1	1	0	0	START	7-Bit data							P	STOP		
1	1	0	1	START	7-Bit data							P	STOP	STOP	
0	—	1	0	START	8-Bit data								MPB	STOP	

Table 14.11 Serial Communication Formats (Asynchronous Mode) (cont)

SCSMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	—	1	1	START	8-Bit data								MPB	STOP	STOP
1	—	1	0	START	7-Bit data							MPB	STOP		
1	—	1	1	START	7-Bit data							MPB	STOP	STOP	

Notes: — : Don’t care bits
START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 14.10).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 14.6 so that the rising edge of the clock occurs at the center of each transmit data bit.

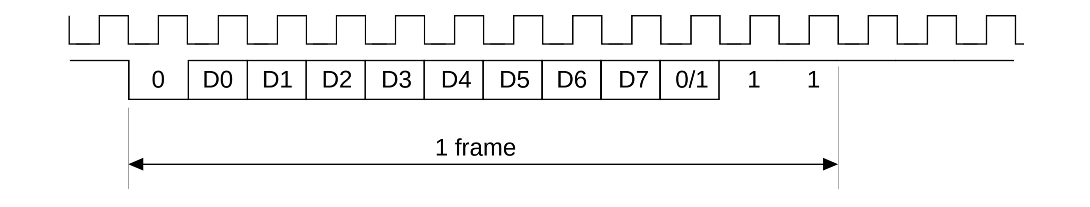


Figure 14.6 Output Clock and Serial Data Timing (Asynchronous Mode)

Transmitting and Receiving Data (SCI Initialization (Asynchronous Mode)): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

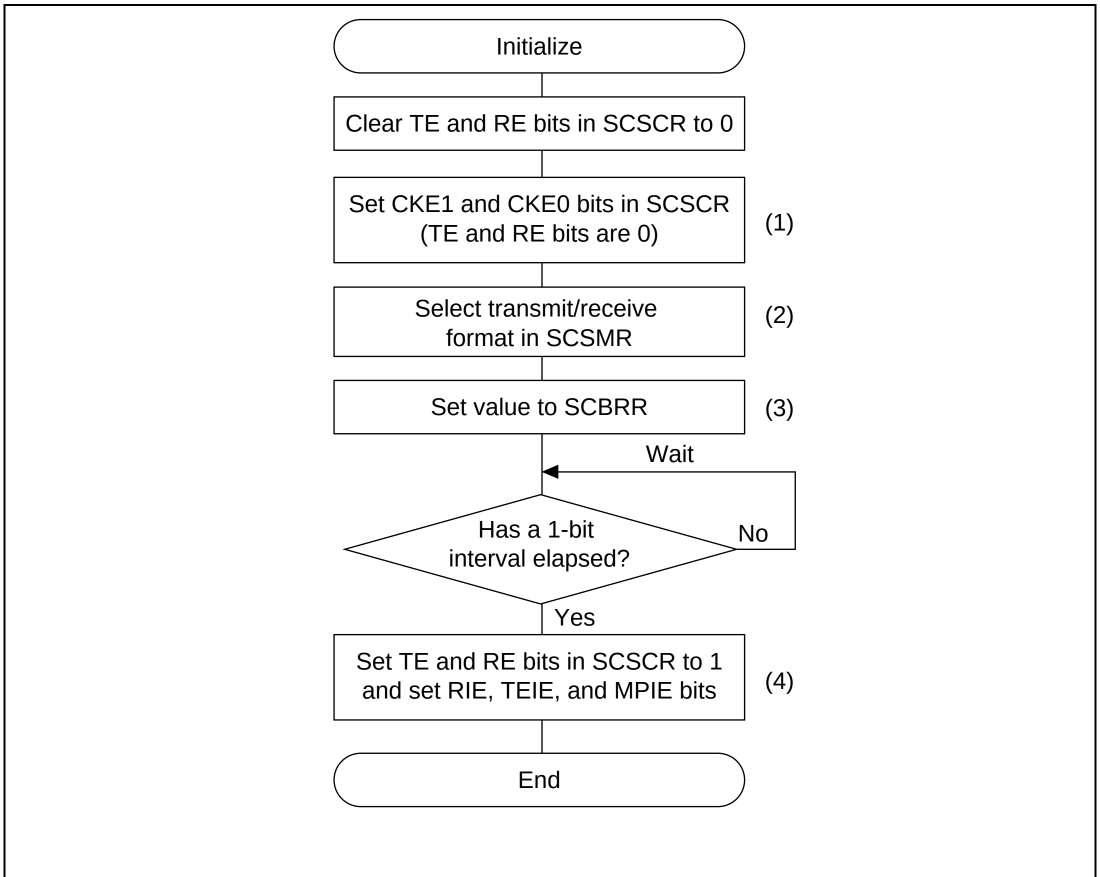
When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 14.7 is a sample flowchart for initializing the SCI. The procedure for initializing the SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCSCR.
2. Select the communication format in the serial mode register (SCSMR).
3. Write the value corresponding to the bit rate in the bit rate register (SCBRR) unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE, and MPIE as necessary.

Setting TE or RE enables the SCI to use the TxD or RxD pin. The initial states are the mark transmit state, and the idle receive state (waiting for a start bit).

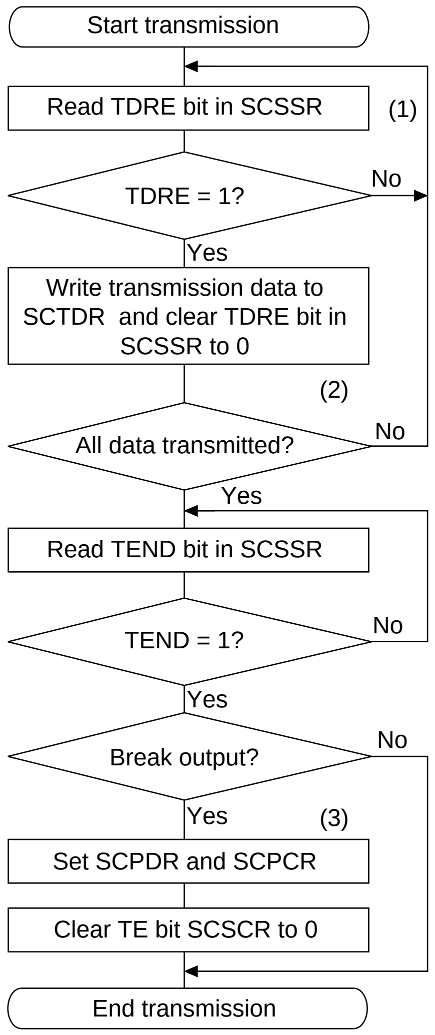

Initialize
Clear TE and RE bits in SCSCR to 0
Set CKE1 and CKE0 bits in SCSCR (TE and RE bits are 0)
(1)
Select transmit/receive format in SCSMR
(2)
Set value to SCBRR
(3)
Wait
Has a 1-bit interval elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TEIE, and MPIE bits
(4)
End

Note: Circled numbers refer to the preceding procedure.

Figure 14.7 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 14.8 shows a sample flowchart for transmitting serial data. The procedure for transmitting serial data is:

1. SCI status check and transmit data write: read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial transmission: Set the port SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE bit to 0 in the serial control register (SCSCR). For SCPCR and SCPDR settings, see section 14.2.8.



Note: Circled numbers refer to the preceding procedure.

Figure 14.8 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from the SCTDR into the transmit shift register (SCTSR).
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCSCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD pin:
    a. Start bit: One 0 bit is output.
    b. Transmit data: Seven or eight bits of data are output, LSB first.
    c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
    d. Stop bit: One or two 1 bits (stop bits) are output.
    e. Marking: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SCSSR, outputs the stop bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 14.9 shows an example of SCI transmit operation in the asynchronous mode.

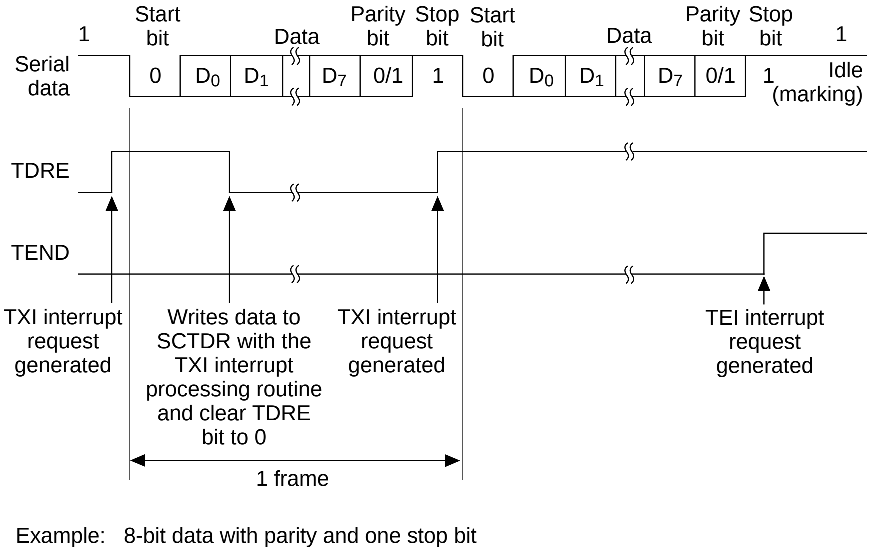
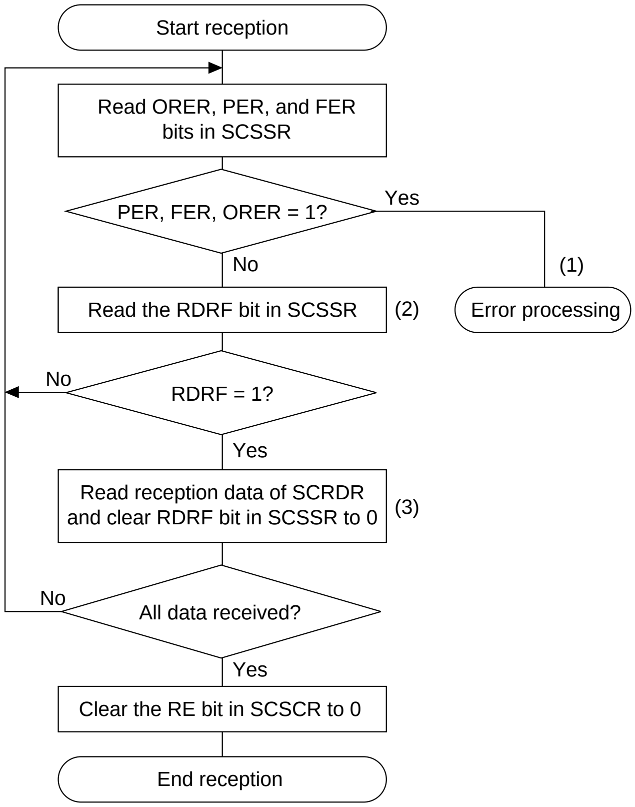

1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
Serial data
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle (marking)
TDRE
TEND
TXI interrupt request generated
Writes data to SCTDR with the TXI interrupt processing routine and clear TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame
Example: 8-bit data with parity and one stop bit

Figure 14.9 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figure 14.10 shows a sample flowchart for receiving serial data. The procedure for receiving serial data after enabling the SCI for reception is:

1. Receive error handling and break detection: If a receive error occurs, read the ORER, PER and FER bits of the SCSSR to identify the error. After executing the necessary error handling, clear ORER, PER and FER all to 0. Receiving cannot resume if ORER, PER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
2. SCI status check and receive-data read: Read the serial status register (SCSSR), check that RDRF is set to 1, then read receive data from the receive data register (SCRDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read the RDRF and SCRDR bits and clear RDRF to 0 before the stop bit of the current frame is received.


Start reception
Read ORER, PER, and FER bits in SCSSR
PER, FER, ORER = 1?
Yes
No
(1)
Read the RDRF bit in SCSSR
(2)
Error processing
No
RDRF = 1?
Yes
Read reception data of SCRDR and clear RDRF bit in SCSSR to 0
(3)
No
All data received?
Yes
Clear the RE bit in SCSCR to 0
End reception

Note: Circled numbers refer to the preceding procedure.

Figure 14.10 Sample Flowchart for Receiving Serial Data

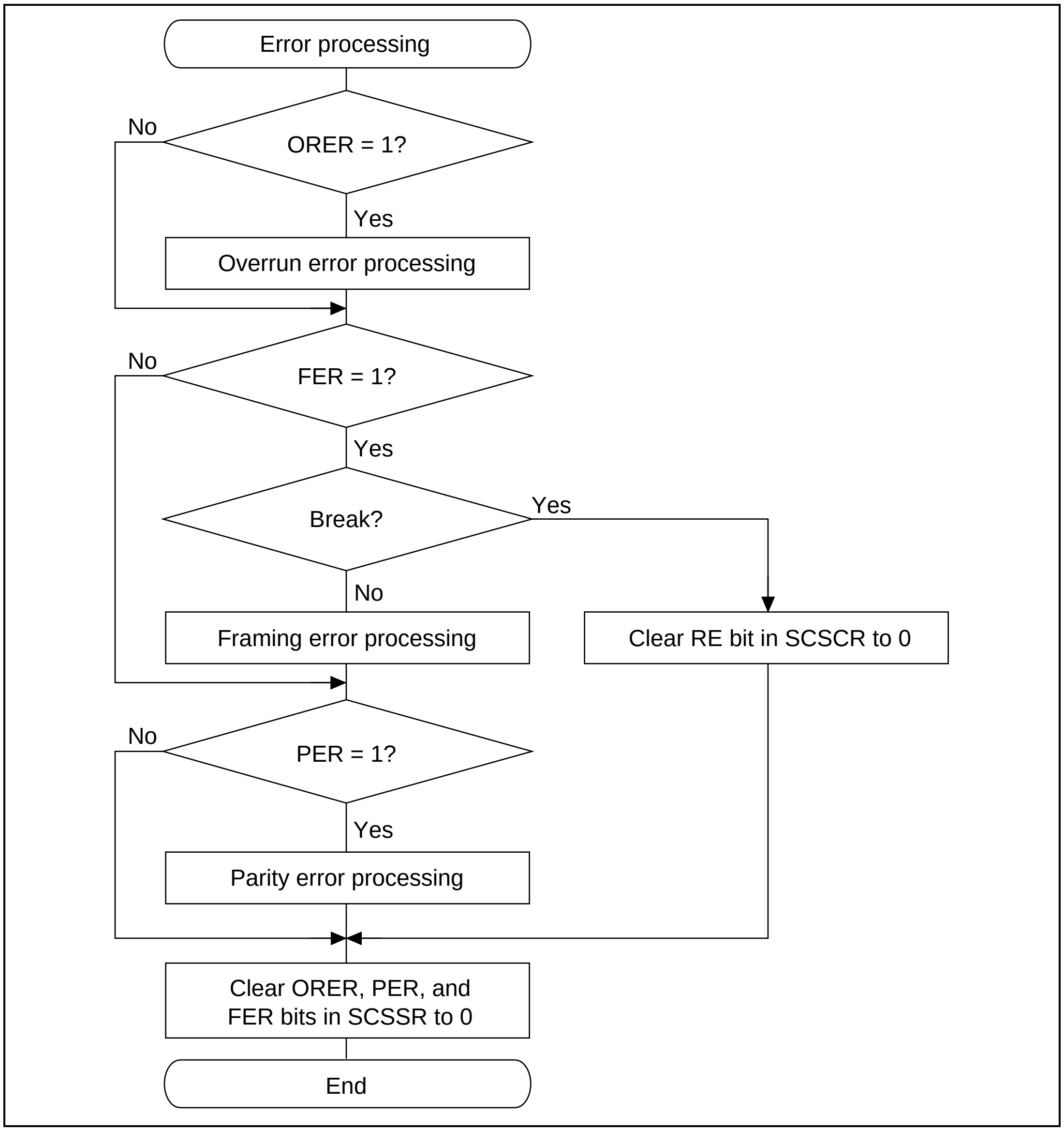


Figure 14.10 Sample Flowchart for Receiving Serial Data (cont)

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.
2. Receive data is shifted into the SCRSR in order from the LSB to the MSB.
3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:
    a. Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{\text{E}}$ bit in the SCSMR.
    b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
    c. Status check: RDRF must be 0 so that receive data can be loaded from the SCRSR into the SCRDR.

    If these checks all pass, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If one of the checks fails (receive error), the SCI operates as indicated in table 14.12.

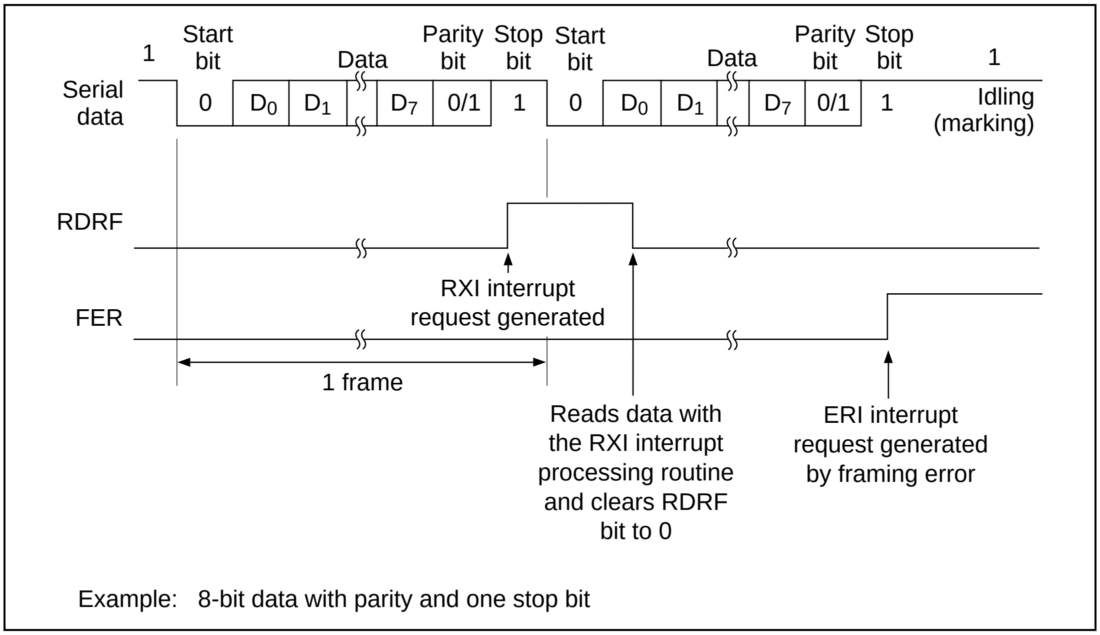
Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is not set to 1. Be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 14.12 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SCSSR	Receive data not loaded from SCRSR into SCRDR
Framing error	FER	Stop bit is 0	Receive data loaded from SCRSR into SCRDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SCSMR	Receive data loaded from SCRSR into SCRDR

Figure 14.11 shows an example of SCI receive operation in the asynchronous mode.



1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
Serial data
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idling (marking)
RDRF
RXI interrupt request generated
FER
1 frame
Reads data with the RXI interrupt processing routine and clears RDRF bit to 0
ERI interrupt request generated by framing error
Example: 8-bit data with parity and one stop bit

Figure 14.11 SCI Receive Operation

14.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 14.12 shows an example of communication among processors using the multiprocessor format.

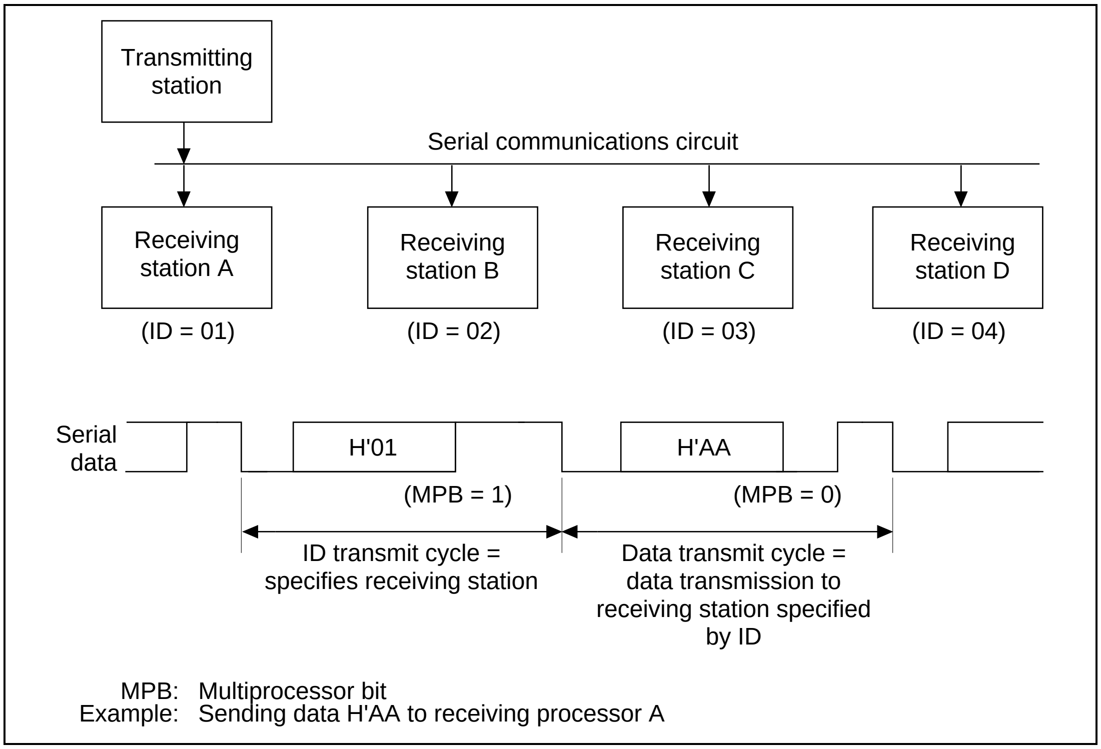

Transmitting station
Serial communications circuit
Receiving station A
Receiving station B
Receiving station C
Receiving station D
(ID = 01)
(ID = 02)
(ID = 03)
(ID = 04)
Serial data
H'01
H'AA
(MPB = 1)
(MPB = 0)
ID transmit cycle = specifies receiving station
Data transmit cycle = data transmission to receiving station specified by ID
MPB: Multiprocessor bit
Example: Sending data H'AA to receiving processor A

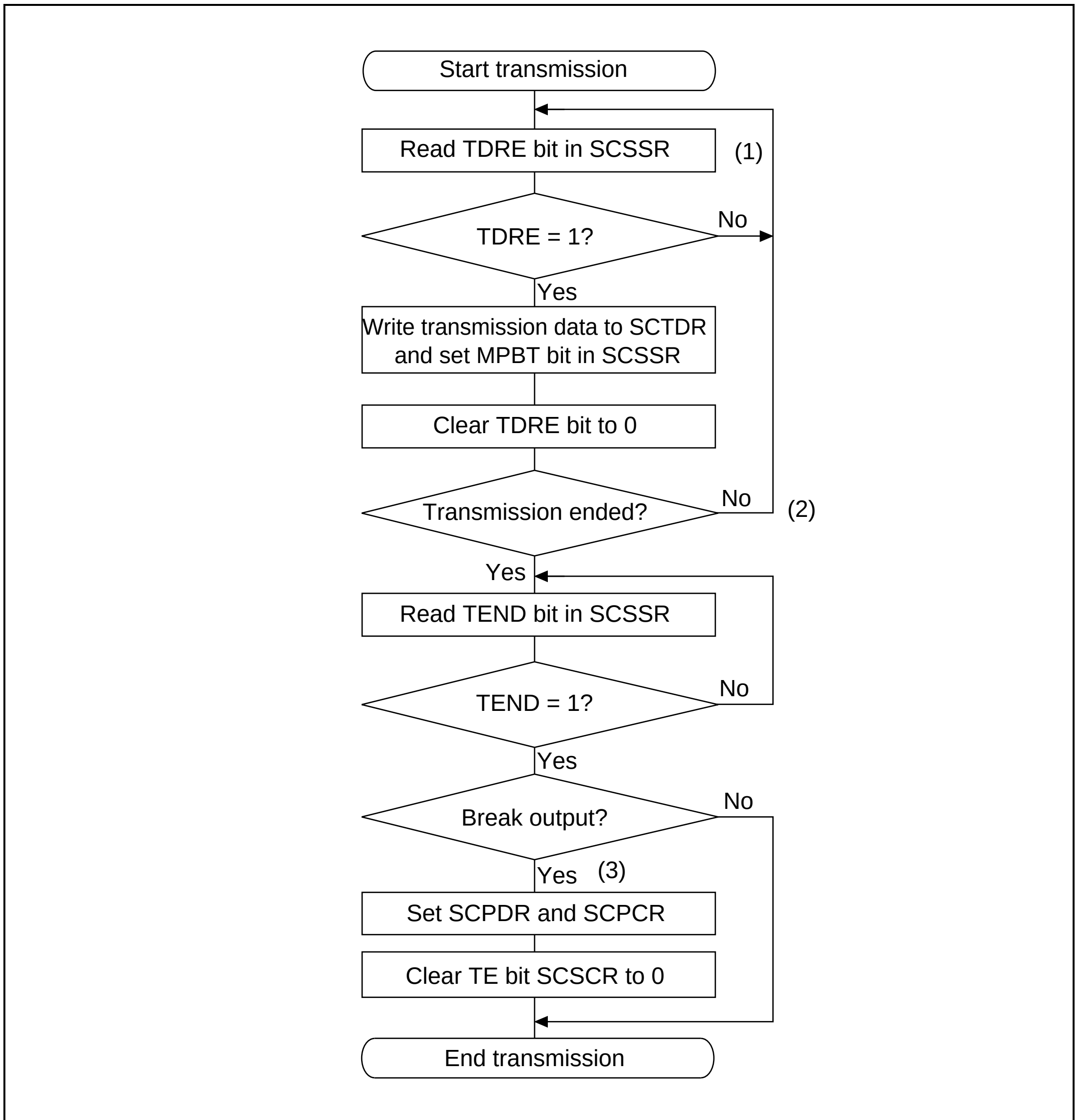
Figure 14.12 Communication Among Processors Using Multiprocessor Format

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 14.11.

Clock: See the description in the asynchronous mode section.

Transmitting Multiprocessor Serial Data: Figure 14.13 shows a sample flowchart for transmitting multiprocessor serial data. The procedure for transmitting multiprocessor serial data is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR). Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SCSSR. Finally, clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.
3. To output a break at the end of serial transmission: Set the port SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE bit to 0 in the serial control register (SCSCR). For SCPCR and SCPDR settings, see section 14.2.8.



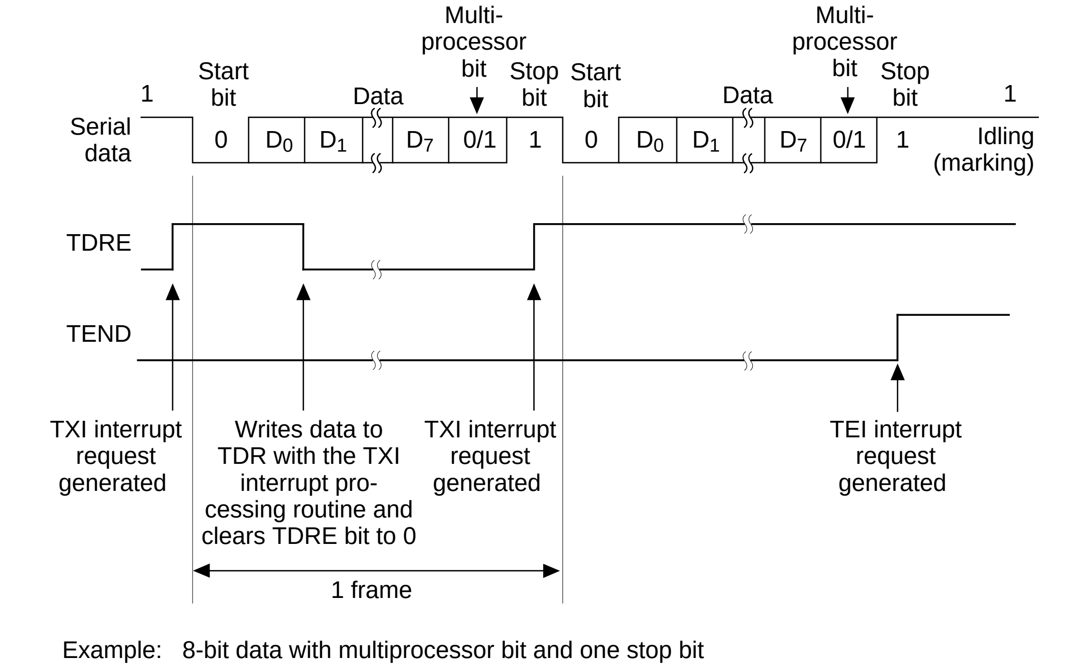
Note: Circled numbers refer to the preceding procedure.

Figure 14.13 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from the SCTDR into the transmit shift register (SCTSR).
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD pin:
   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits are output, LSB first.
   c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, outputs the stop bit, then continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 14.14 shows SCI transmission in the multiprocessor format.



Serial data
1
Start bit
0
D0
D1
Data
D7
Multi-processor bit
0/1
Stop bit
1
Start bit
0
D0
D1
Data
D7
Multi-processor bit
0/1
Stop bit
1
1
Idling (marking)
TDRE
TEND
TXI interrupt request generated
Writes data to TDR with the TXI interrupt pro-cessing routine and clears TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame

Example: 8-bit data with multiprocessor bit and one stop bit

Figure 14.14 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 14.15 shows a sample flowchart for receiving multiprocessor serial data. The procedure for receiving multiprocessor serial data is:

1. ID receive cycle: Set the MPIE bit in the serial control register (SCSCR) to 1.
2. SCI status check and compare to ID reception: Read the serial status register (SCSSR), check that RDRF is set to 1, then read data from the receive data register (SCRDR) and compare with the processor’s own ID. If the ID does not match the receive data, set MPIE to 1 again and clear RDRF to 0. If the ID matches the receive data, clear RDRF to 0.
3. SCI status check and data receiving: Read SCSSR, check that RDRF is set to 1, then read data from the receive data register (SCRDR).
4. Receive error handling and break detection: If a receive error occurs, read the ORER and FER bits in SCSSR to identify the error. After executing the necessary error processing, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.

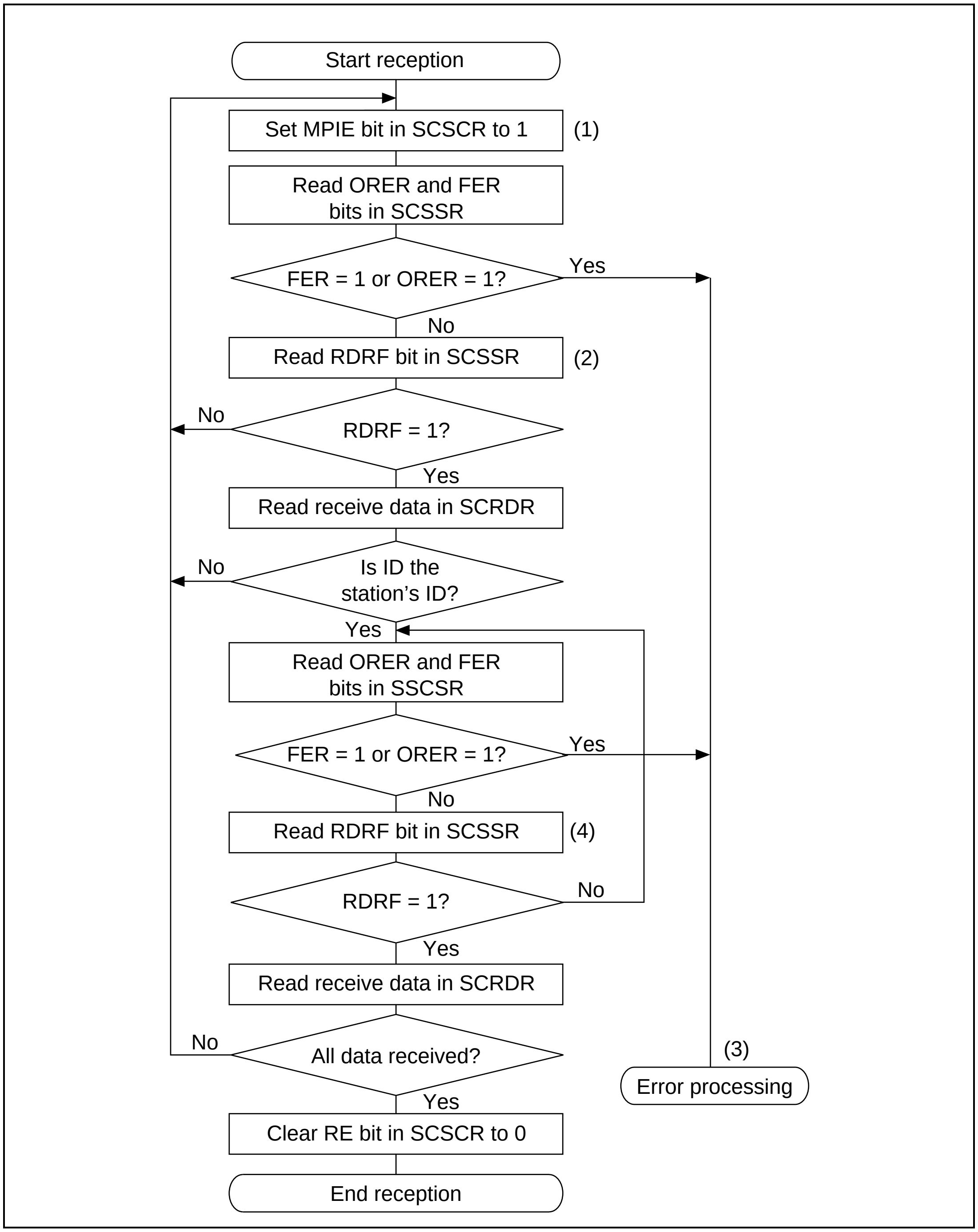


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data

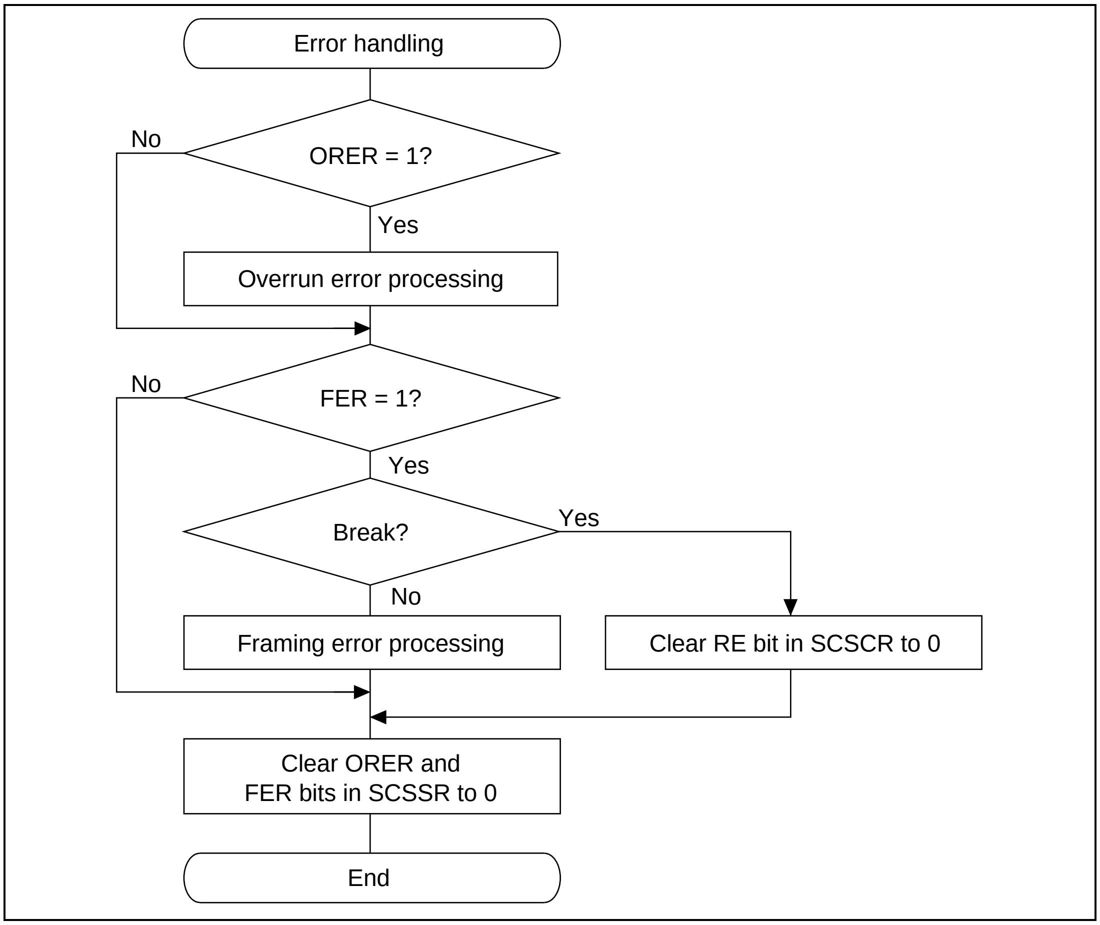


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figure 14.16 shows an example of SCI receive operation using a multiprocessor format.

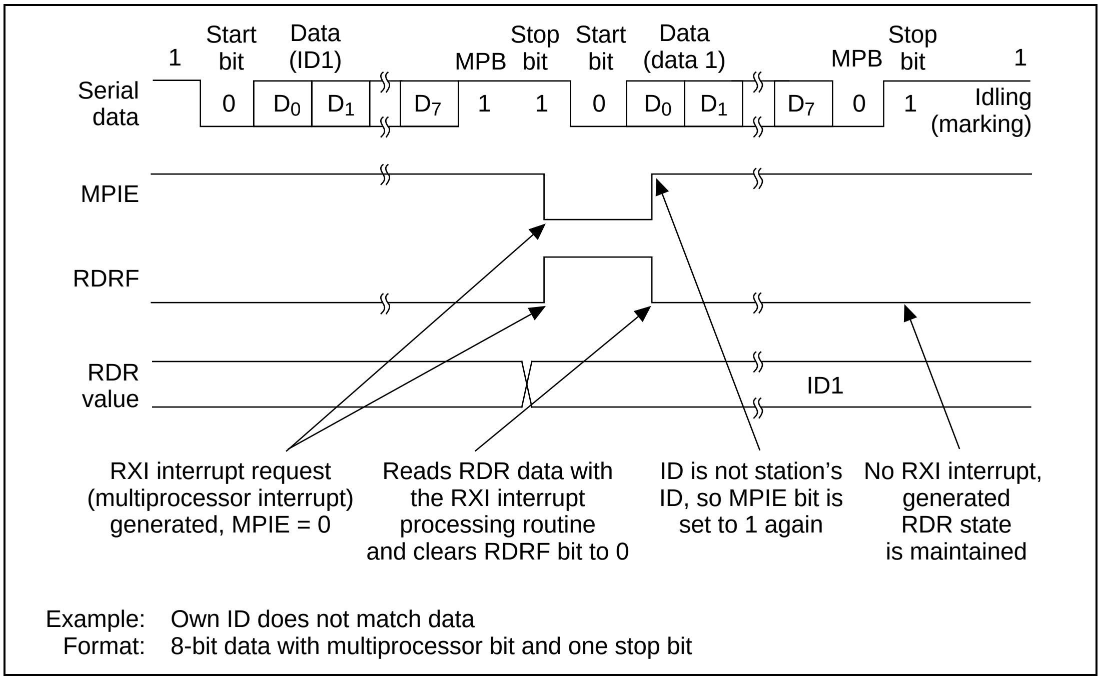

1
Start bit
Data (ID1)
MPB
Stop bit
Start bit
Data (data 1)
MPB
Stop bit
1
Serial data
0
D_0
D_1
D_7
1
1
0
D_0
D_1
D_7
0
1
Idling (marking)
MPIE
RDRF
RDR value
ID1
RXI interrupt request (multiprocessor interrupt) generated, MPIE = 0
Reads RDR data with the RXI interrupt processing routine and clears RDRF bit to 0
ID is not station's ID, so MPIE bit is set to 1 again
No RXI interrupt, generated RDR state is maintained
Example: Own ID does not match data
Format: 8-bit data with multiprocessor bit and one stop bit

Figure 14.16 Example of SCI Receive Operation

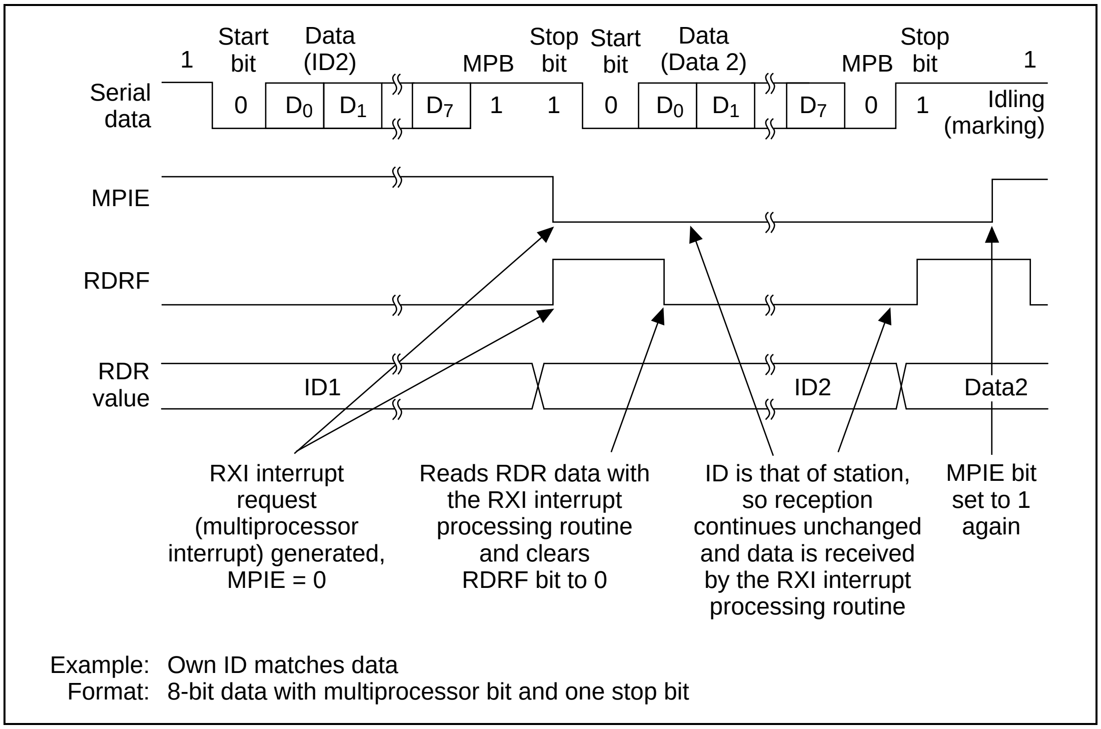

1
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data 2)
MPB
Stop bit
1
Serial data
0
D_0
D_1
D_7
1
1
0
D_0
D_1
D_7
0
1
Idling (marking)
MPIE
RDRF
RDR value
ID1
ID2
Data2
RXI interrupt request (multiprocessor interrupt) generated, MPIE = 0
Reads RDR data with the RXI interrupt processing routine and clears RDRF bit to 0
ID is that of station, so reception continues unchanged and data is received by the RXI interrupt processing routine
MPIE bit set to 1 again
Example: Own ID matches data
Format: 8-bit data with multiprocessor bit and one stop bit

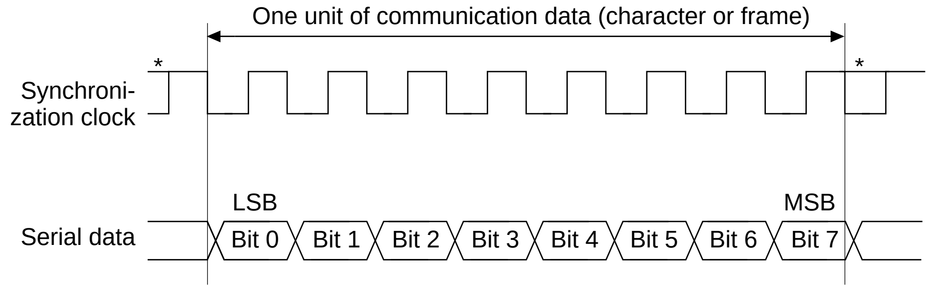
Figure 14.16 Example of SCI Receive Operation (cont)

14.3.4 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14.17 shows the general format in clock synchronous serial communication.



Note: High except in continuous transmitting or receiving

Figure 14.17 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In the clock synchronous mode, the SCI transmits or receives data by synchronizing with the falling edge of the serial clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the $C/\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR). See table 14.10.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the SCI receives in 2-character units, so a 16 pulse synchronization clock is output. To receive in 1-character units, select an external clock source.

Transmitting and Receiving Data: SCI Initialization (clock synchronous mode). Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 14.18 is a sample flowchart for initializing the SCI. The procedure for initializing the SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE and RE cleared to 0.
2. Select the communication format in the serial mode register (SCSMR).
3. Write the value corresponding to the bit rate in the bit rate register (SCBRR) unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE and MPIE. Setting TE and RE allows use of the TxD and RxD pins.

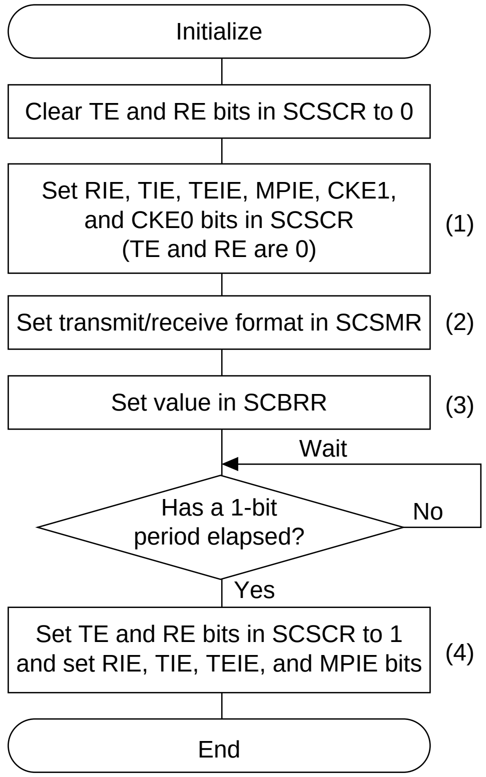

Initialize
Clear TE and RE bits in SCSCR to 0
Set RIE, TIE, TEIE, MPIE, CKE1, and CKE0 bits in SCSCR (TE and RE are 0)
(1)
Set transmit/receive format in SCSMR
(2)
Set value in SCBRR
(3)
Wait
Has a 1-bit period elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TIE, TEIE, and MPIE bits
(4)
End

Figure 14.18 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Clock Synchronous Mode): Figure 14.19 shows a sample flowchart for transmitting serial data. The procedure for transmitting serial data is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0.
2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

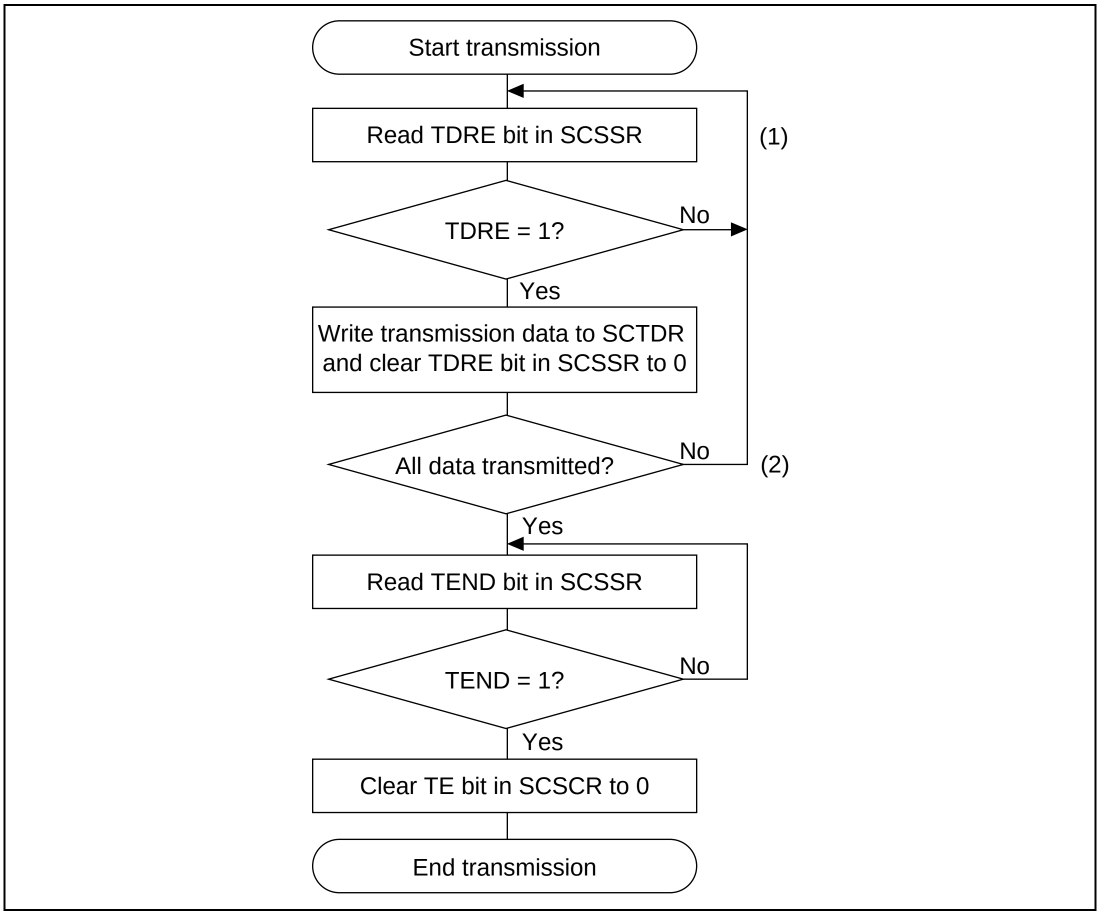


Figure 14.19 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data and loads this data from the SCTDR into the transmit shift register (SCTSR).
2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, transmits the MSB, then holds the transmit data pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14.20 shows an example of SCI transmit operation.

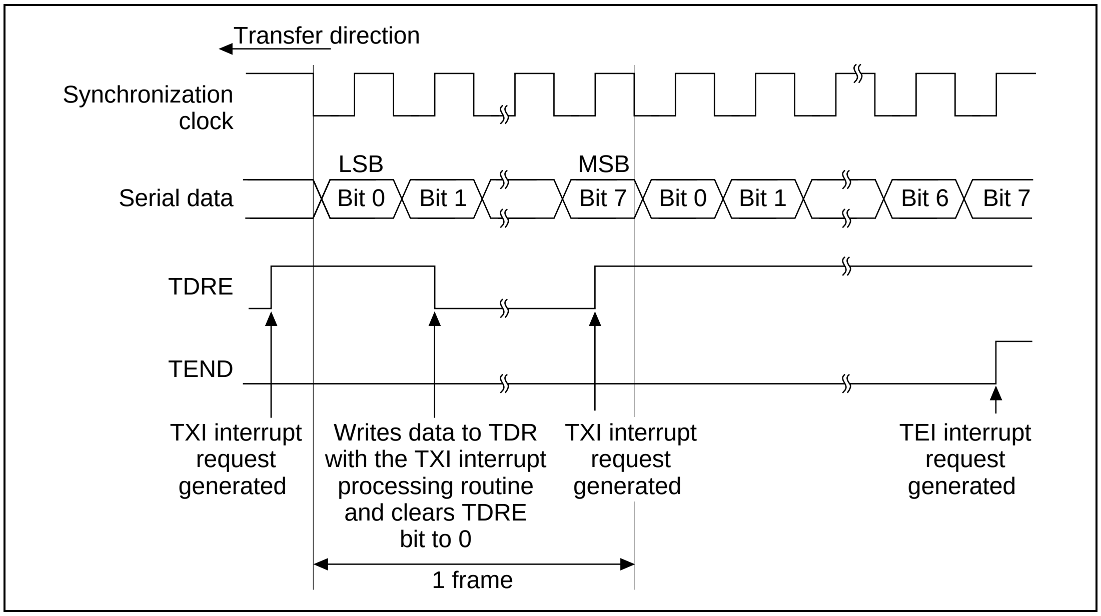

Transfer direction
Synchronization clock
LSB
MSB
Serial data
Bit 0
Bit 1
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
TDRE
TEND
TXI interrupt request generated
Writes data to TDR with the TXI interrupt processing routine and clears TDRE bit to 0
TXI interrupt request generated
TEI interrupt request generated
1 frame

Figure 14.20 Example of SCI Transmit Operation

Receiving Serial Data (Clock Synchronous Mode): Figure 14.21 shows a sample flowchart for receiving serial data. When switching from the asynchronous mode to the clock synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

The procedure for receiving serial data is:

1. Receive error handling and break detection: If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error handling, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
2. SCI status check and receive data read: Read the serial status register (SCSSR), check that RDRF is set to 1, then read receive data from the receive data register (SCRDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received.

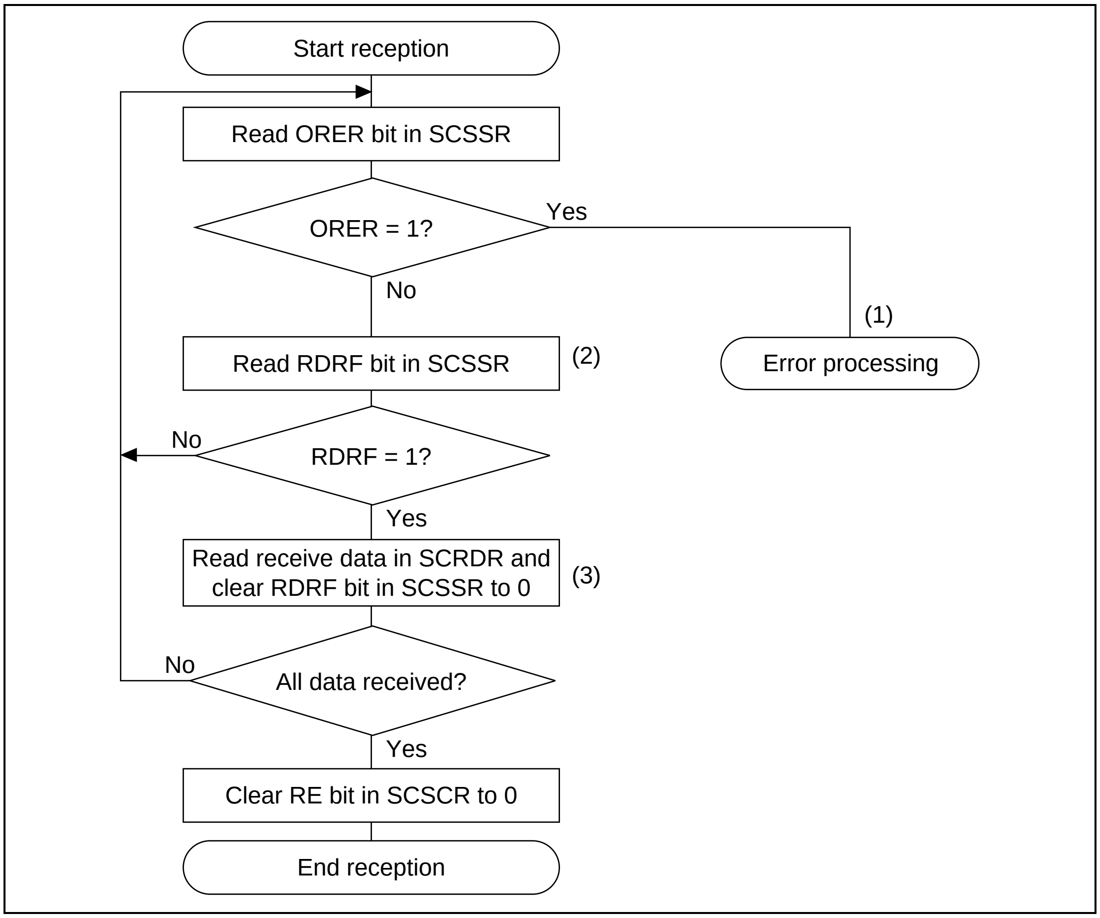


Figure 14.21 Sample Flowchart for Serial Receiving

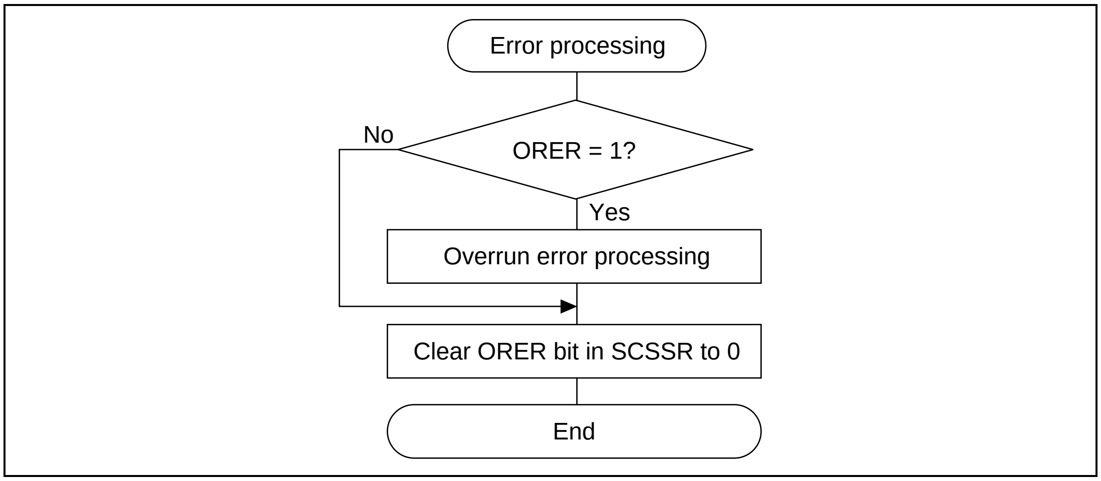


Figure 14.21 Sample Flowchart for Serial Receiving (cont)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into the SCRSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from the SCRSR into the SCRDR. If this check is passed, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If the check is not passed (receive error), the SCI operates as indicated in table 14.12. This state prevents further transmission or reception. While receiving, the RDRF bit is not set to 1. Be sure to clear the error flag.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 14.22 shows an example of the SCI receive operation.

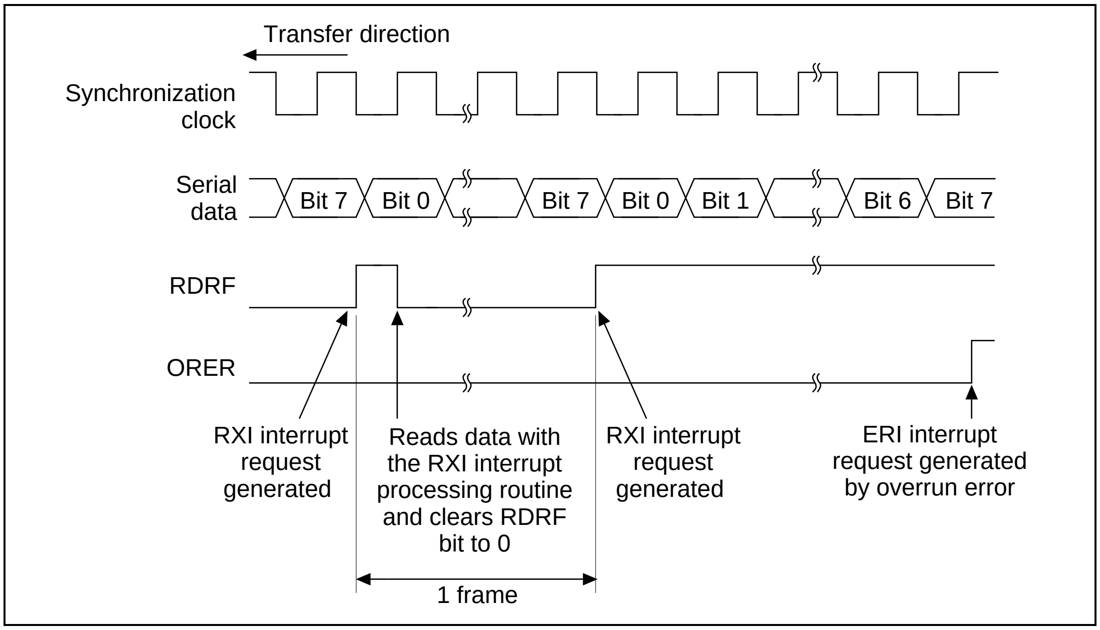
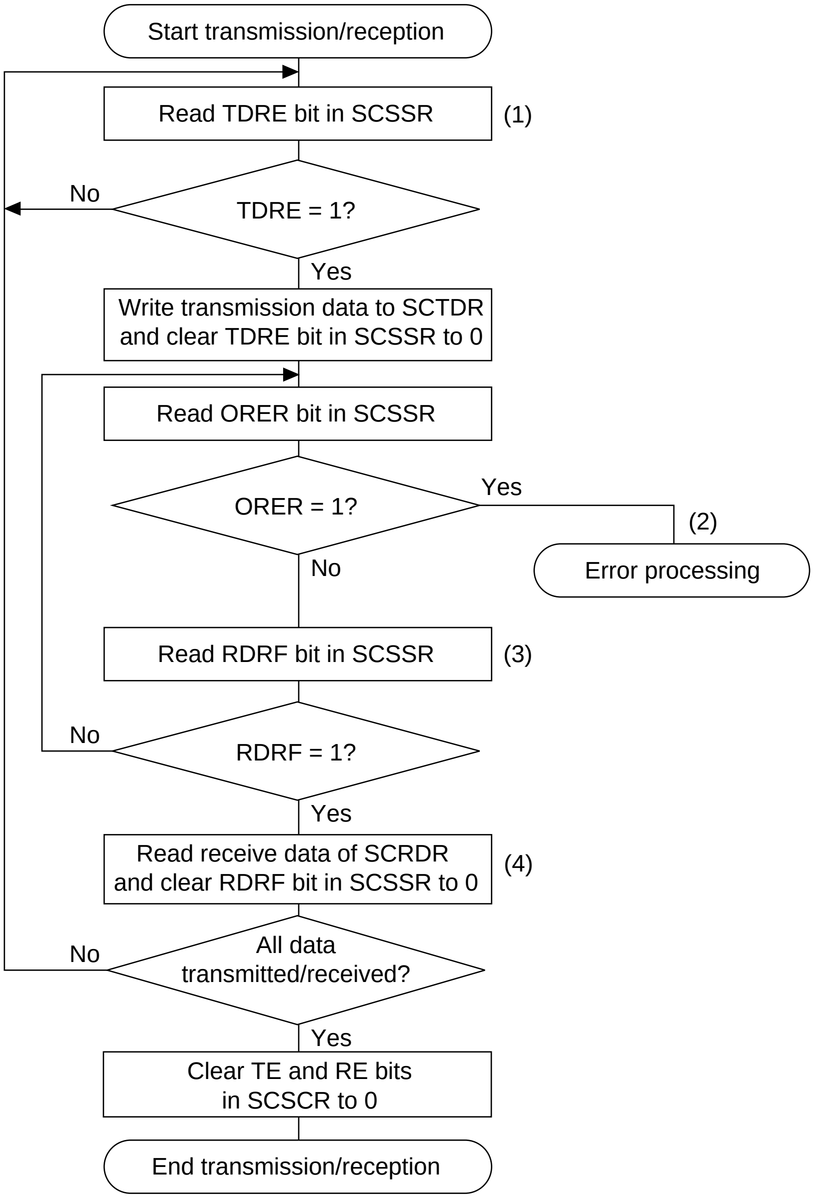

Transfer direction
Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
RDRF
ORER
RXI interrupt request generated
Reads data with the RXI interrupt processing routine and clears RDRF bit to 0
RXI interrupt request generated
ERI interrupt request generated by overrun error
1 frame

Figure 14.22 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode): Figure 14.23 shows a sample flowchart for transmitting and receiving serial data simultaneously. The procedure for setting the SCI to transmit and receive serial data simultaneously is:

1. SCI status check and transmit data write: Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0. The TXI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.
2. Receive error handling: If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the serial status register (SCSSR), check that RDRF is set to 1, then read receive data from the receive data register (SCRDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. To continue transmitting and receiving serial data: Read the RDRF bit and SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted.


Start transmission/reception
Read TDRE bit in SCSSR
(1)
TDRE = 1?
No
Yes
Write transmission data to SCTDR and clear TDRE bit in SCSSR to 0
Read ORER bit in SCSSR
ORER = 1?
Yes
(2)
Error processing
No
Read RDRF bit in SCSSR
(3)
RDRF = 1?
No
Yes
Read receive data of SCRDR and clear RDRF bit in SCSSR to 0
(4)
All data transmitted/received?
No
Yes
Clear TE and RE bits in SCSCR to 0
End transmission/reception

Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, simultaneously clear TE and RE to 0, then simultaneously set TE and RE to 1.

Figure 14.23 Sample Flowchart for Serial Transmitting

14.4 SCI Interrupt Sources

The SCI has four interrupt sources in each channel: Transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 14.13 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in the serial control register (SCSCR). Each interrupt request is sent separately to the interrupt controller.

TXI is requested when the TDRE bit in the SCSSR is set to 1. TDRE is automatically cleared to 0 when data is written in the transmit data register (SCTDR).

RXI is requested when the RDRF bit in the SCSSR is set to 1. RDRF is automatically cleared to 0 when the receive data register (SCRDR) is read.

ERI is requested when the ORER, PER, or FER bit in the SCSSR is set to 1.

TEI is requested when the TEND bit in the SCSSR is set to 1. Where the TXI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 14.13 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is Cleared
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	
TXI	Transmit data empty (TDRE)	↓
TEI	Transmit end (TEND)	Low

See section 4, Exception Processing, for information on the priority order and relationship to non-SCI interrupts.

14.5 Notes on Use

Note the following points when using the SCI.

SCTDR Write and TDRE Flags: The TDRE bit in the serial status register (SCSSR) is a status flag indicating loading of transmit data from the SCTDR into the SCTSR. The SCI sets TDRE to 1 when it transfers data from the SCTDR to the SCTSR. Data can be written to the SCTDR regardless of the TDRE bit status. If new data is written in the SCTDR when TDRE is 0, however, the old data stored in the SCTDR will be lost because the data has not yet been transferred to the SCTSR. Before writing transmit data to the SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 14.14 indicates the state of the SCSSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the SCRSR contents cannot be transferred to the SCRDR, so receive data is lost.

Table 14.14 SCSSR Status Flags and Transfer of Receive Data

	SCSSR Status Flags				**Receive Data Transfer**
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**SCRSR → SCRDR**
Overrun error	1	1	0	0	X*[1]
Framing error	0	0	1	0	O*[2]
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Notes: 1. Receive data is not transferred from SCRSR to SCRDR.

2. Receive data is transferred from SCRSR to SCRDR.

Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state, the input from the RxD pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

Sending a Break Signal: The TxD pin input/output condition and level can be determined by means of the SCP0DT bit of the port SC data register (SCPDR) and bits SCP0MD0 and SCP0MD1 of the port SC control register (SCPCR). These bits can be used to send breaks. To send a break during serial transmission, clear the SCP0DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission status, and 0 is output from the TxD pin.

TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.

Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode: In the asynchronous mode, the SCI operates on a base clock of 16 times the transfer rate frequency. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 14.24).

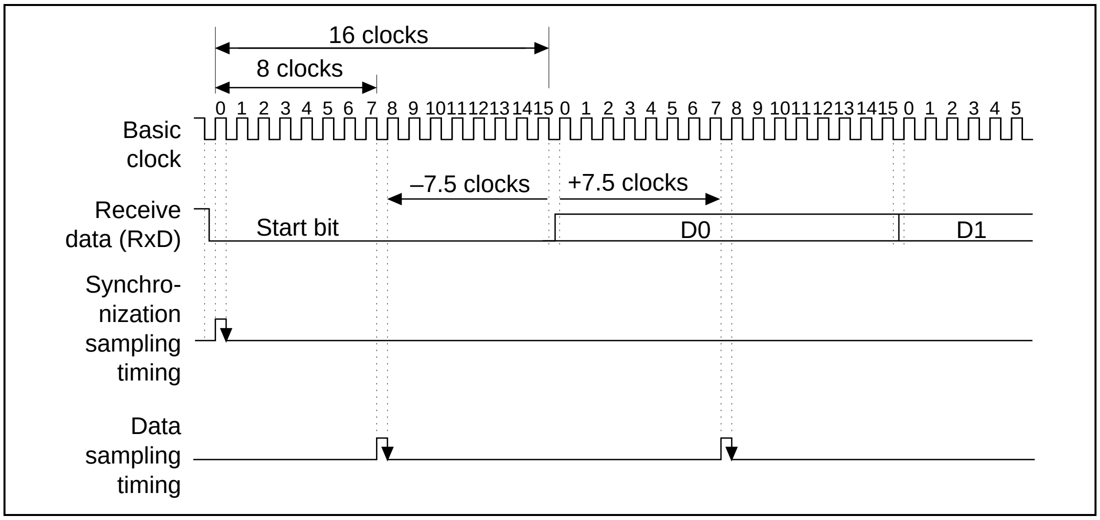


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the asynchronous mode can therefore be expressed as in equation 1.

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where:

M = Receive margin (%)
N = Ratio of clock frequency to bit rate (N = 16)
D = Clock duty cycle (D = 0–1.0)
L = Frame length (L = 9–12)
F = Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as in equation 2.

Equation 2:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

Cautions for Clock Synchronous External Clock Mode:

- Set TE = RE = 1 only when the external clock SCK is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising edge of the RxD D7 bit SCK input, but it cannot be copied to SCRDR.

Caution for Clock Synchronous Internal Clock Mode: When receiving, RDRF is 1 when RE is set to zero 1.5 clocks after the rising edge of the RxD D7 bit SCK output, but it cannot be copied to SCRDR.

Section 15 Smart Card Interface

15.1 Overview

As an added serial communications interface function, the SCI supports an IC card (smart card) interface that conforms to the ISO/IEC standard 7816-3 for identification of cards. Register settings are used to switch between the ordinary serial communication interface and the smart card interface.

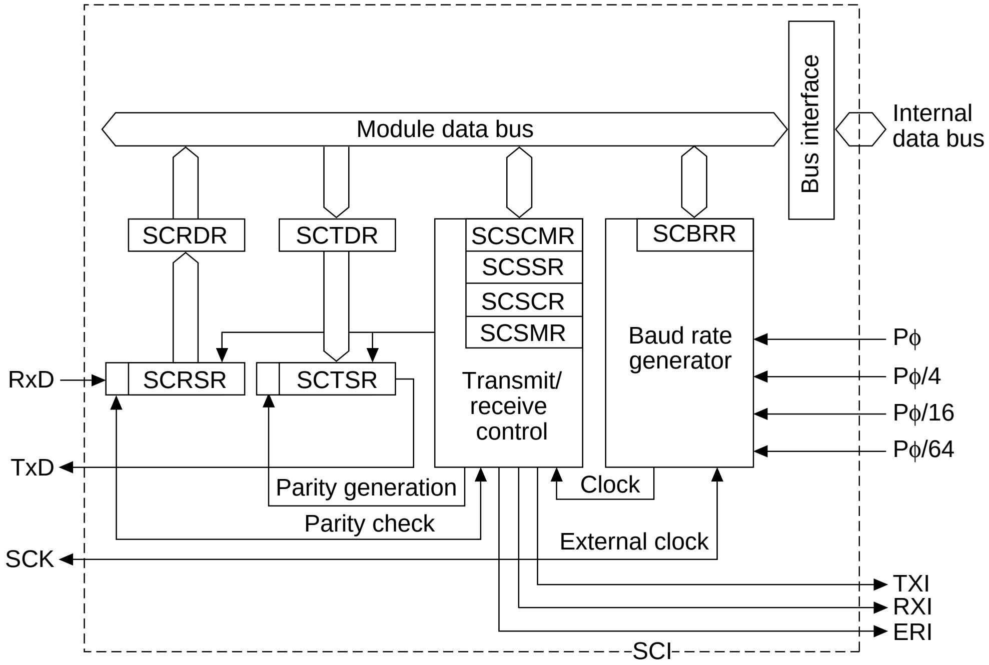
15.1.1 Features

The smart card interface has the following features:

- Asynchronous mode
  - Data length: Eight bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communication-error interrupts are requested independently.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the smart card interface.



SCSCMR:	Smart card mode register
SCRSR:	Receive data register
SCRDR:	Receive data register
SCTSR:	Transmit shift register
SCTDR:	Transmit data register
SCSMR:	Serial mode register
SCSCR:	Serial control register
SCSSR:	Serial status register
SCBRR:	Bit rate register

Figure 15.1 Smart Card Interface Block Diagram

15.1.3 Pin Configuration

Table 15.1 summarizes the smart card interface pins.

Table 15.1 SCI Pins

Pin Name	Symbol	Input/Output	Function
Serial clock pin	SCK0	Output	Clock output
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

15.1.4 Register Configuration

Table 15.2 summarizes the registers used by the smart card interface. The SCSMR, SCBRR, SCSCR, SCTDR, and SCRDR registers are the same as in the ordinary SCI function. They are described in section 14, Serial Communication Interface.

Table 15.2 Registers

Register Name	Abbr.	R/W	Initial Value*[3]	Address	Access Size
Serial mode register	SCSMR	R/W	H'00	H'FFFFFE80	8
Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*[1]	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
Smart card mode register	SCSCMR	R/W	H'00*[2]	H'FFFFFE8C	8

Notes: 1. Only 0 can be written, to clear the flags.

2. Bits 0, 2, and 3 are cleared. The value of the other bits is undefined.

3. Initialized by a power-on or manual reset.

15.2 Register Descriptions

This section describes the registers added for the smart card interface and the bits whose functions are changed.

15.2.1 Smart Card Mode Register (SCSCMR)

The smart card mode register (SCSCMR) is an 8-bit read/write register that selects smart card interface functions. SCSMR bits 0, 2, and 3 are initialized to 0 by a reset and in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	SDIR	SINV	—	SMIF
Initial value:	—	—	—	—	0	0	—	0
R/W:	R	R	R	R	R/W	R/W	R	R/W

Bits 7 to 4 and 1—Reserved: An undefined value will be returned if these bits are read.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3: SDIR	Description
0	Contents of SCTDR are transferred LSB first, receive data is stored in SCRDR LSB first. (Initial value)
1	Contents of SCTDR are transferred MSB first, receive data is stored in SCRDR MSB first.

Bit 2—Smart Card Data Inversion (SINV): Specifies whether to invert the logic level of the data. This function is used in combination with bit 3 for transmitting and receiving with an inverse convention card. SINV does not affect the logic level of the parity bit. See section 16.3.4, Register Settings, for information on how parity is set.

Bit 2: SINV	Description
0	Contents of SCTDR are transferred unchanged, receive data is stored in SCRDR unchanged. (Initial value)
1	Contents of SCTDR are inverted before transfer, receive data is inverted before storage in SCRDR.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables the smart card interface function.

Bit 0 : SMIF	Description
0	Smart card interface function disabled (Initial value)
1	Smart card interface function enabled

15.2.2 Serial Status Register (SCSSR)

In the smart card interface mode, the function of SCSSR bit 4 is changed. The setting conditions for bit 2, the TEND bit, are also changed.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: Only 0 can be written, to clear the flag.

Bits 7 to 5: These bits have the same function as in the ordinary SCI. See section 13, Serial Communication Interface, for more information.

Bit 4—Error Signal Status (ERS): In the smart card interface mode, bit 4 indicates the status of the error signal returned from the receiving side during transmission. The smart card interface cannot detect framing errors.

Bit 4: ERS	Description
0	Receiving ended normally with no error signal. (Initial value) ERS is cleared to 0 when the chip is reset or enters standby mode, or when software reads ERS after it has been set to 1, then writes 0 in ERS.
1	An error signal indicating a parity error was transmitted from the receiving side. ERS is set to 1 if the error signal sampled is low.

Note: The ERS flag maintains its status even when the TE bit in SCSCR is cleared to 0.

Bits 3 to 0: These bits have the same function as in the ordinary SCI. See section 13, Serial Communication Interface, for more information. The setting conditions for bit 2, the transmit end bit (TEND), are changed as follows.

Bit 2: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE, or when data is written in SCTDR.
1	End of transmission. (Initial value) TEND is set to 1 when: • the chip is reset or enters standby mode, • the TE bit in SCSCR is 0 and the FER/ERS bit is also 0, • the $C/\overline{A}$ bit in SCSMR is 0, and TDRE = 1 and FER/ERS = 0 (normal transmission) 2.5 etu after a one-byte serial character is transmitted, or • the $C/\overline{A}$ bit in SCSMR is 1, and TDRE = 1 and FER/ERS = 0 (normal transmission) 1.0 etu after a one-byte serial character is transmitted.

Note: etu is an abbreviation of elementary time unit, which is the period for the transfer of 1 bit.

15.3 Operation

15.3.1 Overview

The primary functions of the smart card interface are described below.

1. Each frame consists of 8 data bits and 1 parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (elementary time units: the period for 1 bit to transfer) from the end of the parity bit to the start of the next frame.
3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu has elapsed from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at least 2 etu from the time the error signal is sampled.
5. Only start-stop type asynchronous communication functions are supported; no synchronous communication functions are available.

15.3.2 Pin Connections

Figure 15.2 shows the pin connection diagram for the smart card interface. During communication with an IC card, transmission and reception are both carried out over the same data transfer line, so connect the TxD and RxD pins on the chip. Pull up the data transfer line to the power supply V_{CC} side with a resistor.

When using the clock generated by the smart card interface on an IC card, input the SCK pin output to the IC card's CLK pin. This connection is not necessary when the internal clock is used on the IC card.

Use the chip's port output as the reset signal. Apart from these pins, the power and ground pin connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed communication is possible and auto-diagnosis can be performed.

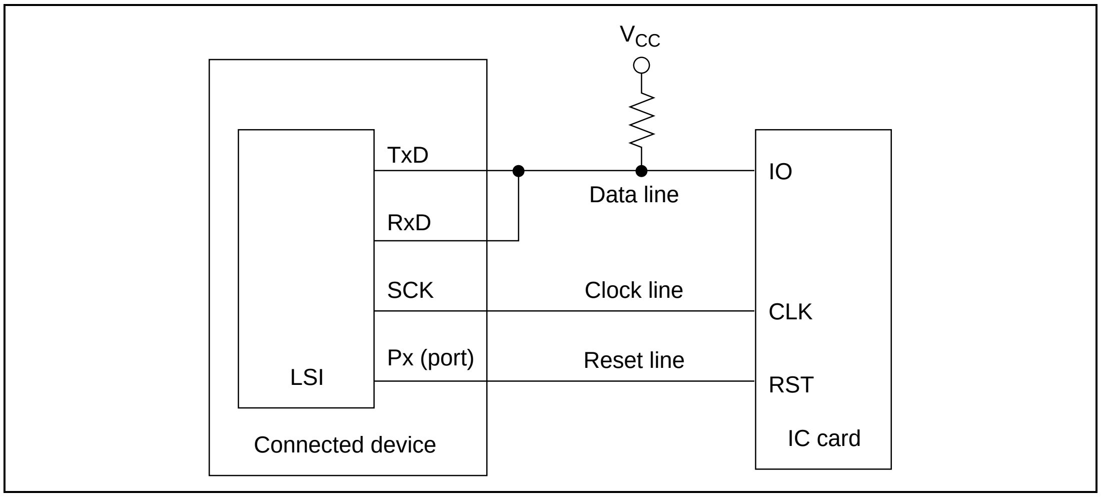


Figure 15.2 Pin Connection Diagram for the Smart Card Interface

15.3.3 Data Format

Figure 15.3 shows the data format for the smart card interface. In this mode, parity is checked every frame while receiving and error signals sent to the transmitting side whenever an error is detected so that data can be re-transmitted. During transmission, error signals are sampled and data re-transmitted whenever an error signal is detected.

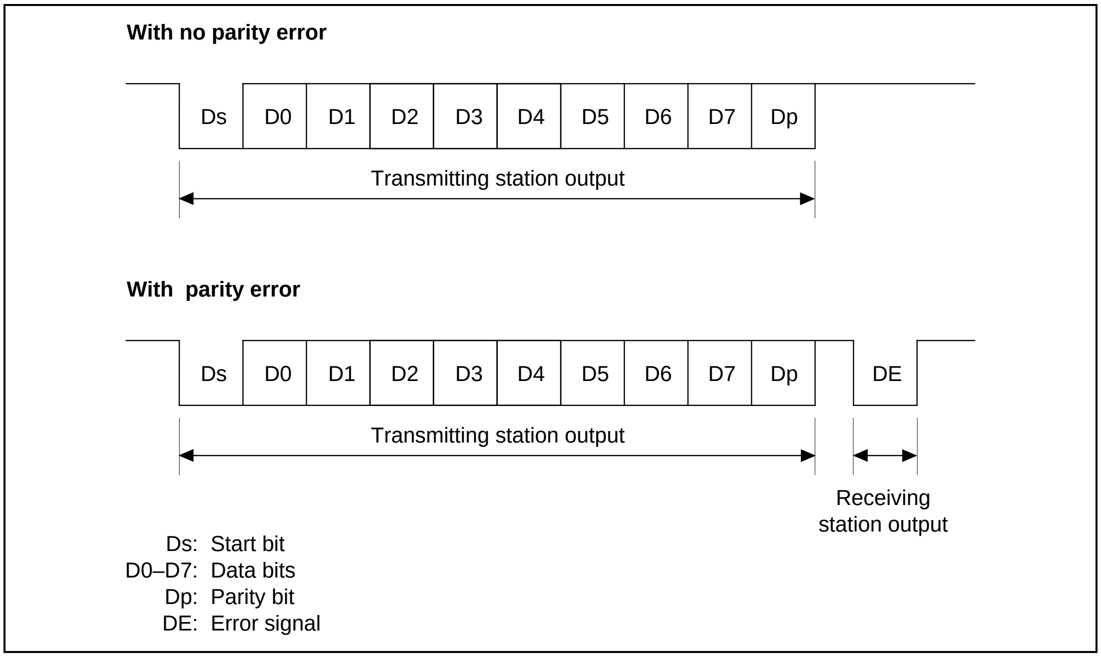


Figure 15.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is high impedance when not in use and is fixed high with a pull-up resistor.
2. The transmitting side starts one frame of data transmission. The data frame starts with a start bit (Ds, low level). The start bit is followed by eight data bits (D0–D7) and a parity bit (Dp).
3. On the smart card interface, the data line returns to high impedance after this. The data line is pulled high with a pull-up resistor.
4. The receiving side checks parity. When the data is received normally with no parity errors, the receiving side then waits to receive the next data. When a parity error occurs, the receiving side outputs an error signal (DE, low level) and requests re-transfer of data. The receiving station returns the signal line to high impedance after outputting the error signal for a specified period. The signal line is pulled high with a pull-up resistor.
5. The transmitting side transmits the next frame of data unless it receives an error signal. If it does receive an error signal, it returns to step 2 to re-transmit the erroneous data.

15.3.4 Register Settings

Table 15.3 shows the bit map of the registers that the smart card interface uses. Bits shown as 1 or 0 must be set to the indicated value. The settings for the other bits are described below.

Table 15.3 Register Settings for the Smart Card Interface

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCSMR	H'FFFFFE80	C/$\overline{A}$	0	1	O/$\overline{E}$	1	0	CKS1	CKS0
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE1	CKE0
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	0	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—	SMIF

Note: Dashes indicate unused bits.

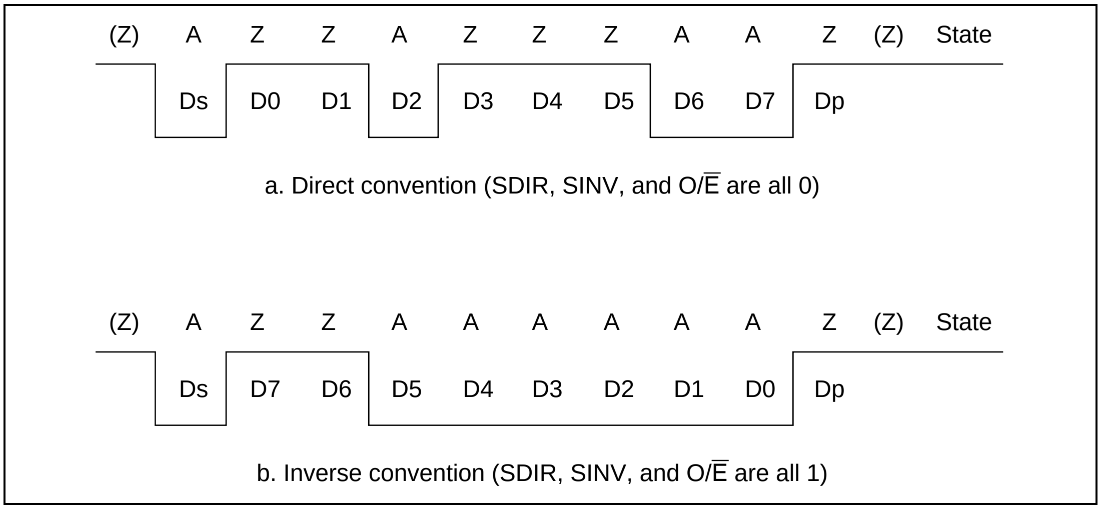
1. Setting the serial mode register (SCSMR): Set the O/$\overline{E}$ bit to 0 when the IC card uses the direct convention or to 1 when it uses the inverse convention. Select the on-chip baud rate generator clock source with the CKS1 and CKS0 bits (see section 15.3.5, Clock).
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 15.3.5, Clock, to see how to calculate the set value.
3. Setting the serial control register (SCSCR): The TIE, RIE, TE and RE bits function as they do for the ordinary SCI. See section 14, Serial Communication Interface, for more information. The CKE0 bit specifies the clock output. When no clock is output, set 0; when a clock is output, set 1.
4. Setting the smart card mode register (SCSCMR): The SDIR and SINV bits are both set to 0 for IC cards that use the direct convention and both to 1 when the inverse convention is used. The SMIF bit is set to 1 for the smart card interface.

Figure 15.4 shows sample waveforms for register settings of the two types of IC cards (direct convention and inverse convention) and their start characters.

In the direct convention type, the logical 1 level is state Z, the logical 0 level is state A, and communication is LSB first. The start character data is H'3B. The parity bit is even (from the smart card standards), and thus a 1.

In the inverse convention type, the logical 1 level is state A, the logical 0 level is state Z, and communication is MSB first. The start character data is H'3F. The parity bit is even (from the smart card standards), and thus a 0, which corresponds to state Z.

Only data bits D7–D0 are inverted by the SINV bit. To invert the parity bit, set the O/$\overline{E}$ bit in SCSMR to odd parity mode. This applies to both transmission and reception.


(Z) A Z Z A Z Z Z A A Z (Z) State
Ds D0 D1 D2 D3 D4 D5 D6 D7 Dp
(Z) A Z Z A A A A A A Z (Z) State
Ds D7 D6 D5 D4 D3 D2 D1 D0 Dp

a. Direct convention (SDIR, SINV, and O/$\overline{E}$ are all 0)

b. Inverse convention (SDIR, SINV, and O/$\overline{E}$ are all 1)

Figure 15.4 Waveform of Start Character

15.3.5 Clock

Only the internal clock generated by the on-chip baud rate generator can be used as the communication clock in the smart card interface. The bit rate for the clock is set by the bit rate register (SCBRR) and the CKS1 and CKS0 bits in the serial mode register (SCSMR), and is calculated using the equation below. Table 15.5 shows sample bit rates. If clock output is then selected by setting CKE0 to 1, a clock with a frequency 372 times the bit rate is output from the SCK0 pin.

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where:

N = Value set in SCBRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
Pϕ = Peripheral module operating frequency (MHz)*
n = 0–3 (table 15.4)

Table 15.4 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 15.5 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)						
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**	**18.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: The bit rate is rounded to two decimal places.

Calculate the value to be set in the bit rate register (SCBRR) from the operating frequency and the bit rate. N is an integer in the range 0 ≤ N ≤ 255, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 15.6 Examples of SCBRR Settings for Bit Rate B (Bit/s) (n = 0)

φ (MHz) (9600 Bits/s)													
7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
0	0.00	1	30.00	1	25.00	1	8.99	1	0.00	1	12.01	2	15.99

Table 15.7 Maximum Bit Rates for Frequencies (Smart Card Interface Mode)

Pφ (MHz)	Maximum Bit Rate (Bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error}(\%) = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

Table 15.8 shows the relationship between transmit/receive clock register set values and output states on the smart card interface.

Table 15.8 Register Set Values and SCK Pin

	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{\text{A}}$**	**CKE1**	**CKE0**	**Output**	**State**
1[*1]	1	0	0	0	Port	Determined by setting of port register SPB1IO and SPB1DT bits
	1	0	0	1	⎍⎍⎍	SCK (serial clock) output state
2[*2]	1	1	0	0	Low output	Low output state
	1	1	0	1	⎍⎍⎍	SCK (serial clock) output state
3[*2]	1	1	1	0	High output	High output state
	1	1	1	1	⎍⎍⎍	SCK (serial clock) output state

Notes: 1. The SCK output state changes as soon as the CKE0 bit is modified. The CKE1 bit should be cleared to 0.

2. The clock duty remains constant despite stopping and starting of the clock by modification of the CKE0 bit.

15.3.6 Data Transmission and Reception

Initialization: Initialize the SCI using the following procedure before sending or receiving data. Initialization is also required for switching from transmit mode to receive mode or from receive mode to transmit mode. Figure 15.5 shows a flowchart of the initialization process.

1. Clear TE and RE in the serial control register (SCSCR) to 0.
2. Clear error flags FER/ERS, PER, and ORER to 0 in the serial status register (SCSSR).
3. Set the C/$\overline{\text{A}}$ bit, parity bit (O/$\overline{\text{E}}$ bit), and baud rate generator select bits (CKS1 and CKS0 bits) in the serial mode register (SCSMR). At this time also clear the CHR and MP bits to 0 and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR). When the SMIF bit is set to 1, the TxD and RxD pins both switch from ports to SCI pins and become high impedance.
5. Set the value corresponding to the bit rate in the bit rate register (SCBRR).
6. Set the clock source select bits (CKE1 and CKE0 bits) in the serial control register (SCSCR). Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0. When the CKE0 bit is set to 1, a clock is output from the SCK0 pin.
7. After waiting at least 1 bit, set the TIE, RIE, TE, and RE bits in SCSCR. Do not set the TE and RE bits simultaneously unless performing auto-diagnosis.

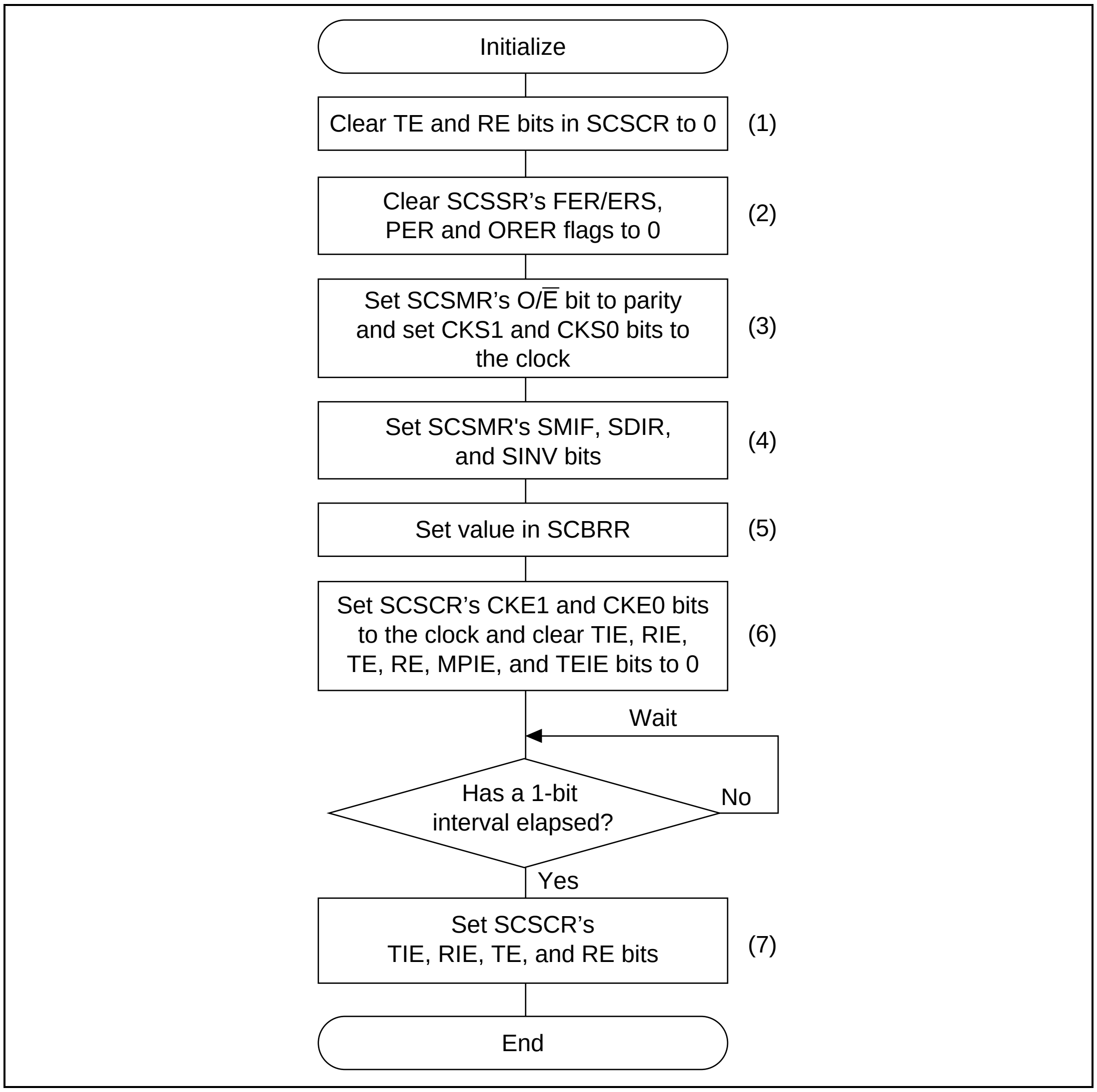

Initialize
Clear TE and RE bits in SCSCR to 0
(1)
Clear SCSSR's FER/ERS, PER and ORER flags to 0
(2)
Set SCSMR's O/$\overline{E}$ bit to parity and set CKS1 and CKS0 bits to the clock
(3)
Set SCSMR's SMIF, SDIR, and SINV bits
(4)
Set value in SCBRR
(5)
Set SCSCR's CKE1 and CKE0 bits to the clock and clear TIE, RIE, TE, RE, MPIE, and TEIE bits to 0
(6)
Wait
Has a 1-bit interval elapsed?
No
Yes
Set SCSCR's TIE, RIE, TE, and RE bits
(7)
End

Figure 15.5 Initialization Flowchart (Example)

Serial Data Transmission: The handling procedures in the smart card mode differ from ordinary SCI processing because data is retransmitted when an error signal is sampled during a data transmission. This results in the transmission processing flowchart shown in figure 15.6.

1. Initialize the smart card interface mode as described in initialization above.
2. Check that the FER/ERS bit in SCSSR is cleared to 0.
3. Repeat steps 2 and 3 until the TEND flag in SCSSR is set to 1.
4. Write the transmit data into SCTDR, clear the TDRE flag to 0 and start transmitting. The TEND flag will be cleared to 0.
5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

This processing can be interrupted. When the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested when the TEND flag is set to 1 at the end of the transmission. When the RIE bit is set to 1 and interrupt requests are enabled, a communication error interrupt (ERI) will be requested when the ERS flag is set to 1 when an error occurs in transmission. See Interrupt Operation below for more information.

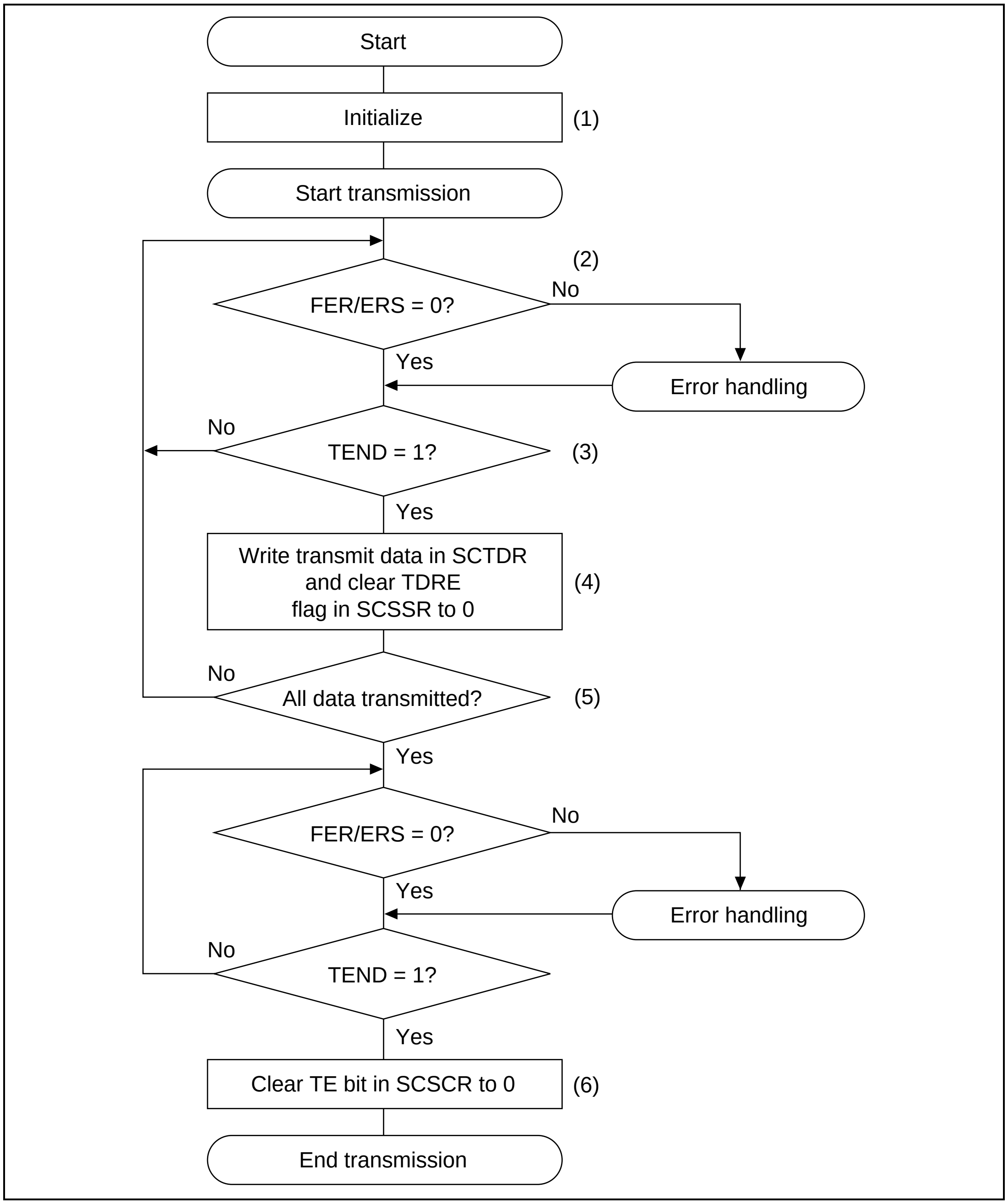


Figure 15.6 Transmission Flowchart

Serial Data Reception: The handling procedures in the smart card mode are the same as in ordinary SCI processing. The reception processing flowchart is shown in figure 15.7.

1. Initialize the smart card interface mode as described above in Initialization and in figure 15.5.
2. Check that the ORER and PER flags in SCSSR are cleared to 0. If either flag is set, clear both to 0 after performing the appropriate error handling procedures.
3. Repeat steps 2 and 3 until the RDRF flag is set to 1.
4. Read the receive data from SCRDR.
5. To receive more data, clear the RDRF flag to 0 and return to step 2.
6. To end reception, clear the RE bit to 0.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at the end of the reception. When an error occurs during reception and either the ORER or PER flag is set to 1, a communication error interrupt (ERI) will be requested. See Interrupt Operation, below, for more information.

The received data will be transferred to SCRDR even when a parity error occurs during reception and PER is set to 1, so this data can still be read.

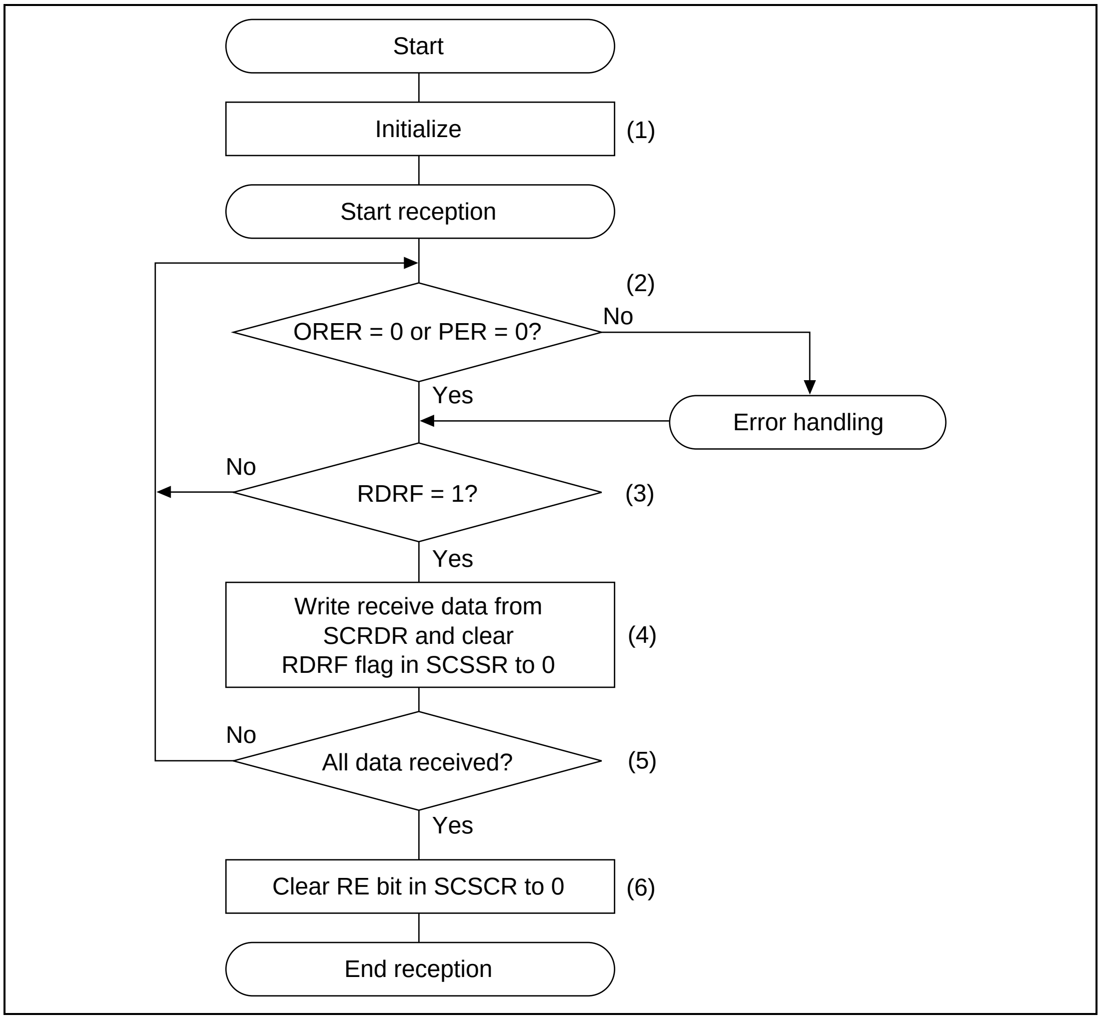


Figure 15.7 Reception Flowchart (Example)

Switching Modes: When switching from receive mode to transmit mode, check that the receive operation is completed before starting initialization and setting RE to 0 and TE to 1. The RDRF, PER, and ORER flags can be used to check if reception is completed. When switching from transmit mode to receive mode, check that the transmit operation is completed before starting initialization and setting TE to 0 and RE to 1. The TEND flag can be used to check if transmission is completed.

Interrupt Operation: In the smart card interface mode, there are three types of interrupts: transmit-data-empty (TXI), communication error (ERI) and receive-data-full (RXI). In this mode, the transmit-end interrupt (TEI) cannot be requested.

Set the TEND flag in SCSSR to 1 to request a TXI interrupt. Set the RDRF flag in SCSSR to 1 to request an RXI interrupt. Set the ORER, PER, or FER/ERS flag in SCSSR to 1 to request an ERI interrupt (table 15.9).

Table 15.9 Smart Card Mode Operating Status and Interrupt Sources

Mode	Status	Flag	Mask Bit	Interrupt Source
Transmit mode	Normal	TEND	TIE	TXI
	Error	FER/ERS	RIE	ERI
Receive mode	Normal	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

15.4 Usage Notes

When the SCI is used as a smart card interface, be sure that all criteria in sections 15.4.1 and 15.4.2 are applied.

15.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI runs on a basic clock with a frequency of 372 times the transfer rate. During reception, the SCI samples the fall of the start bit using the base clock to achieve internal synchronization. Receive data is latched internally on the rising edge of the 186th basic clock cycle (figure 15.8).

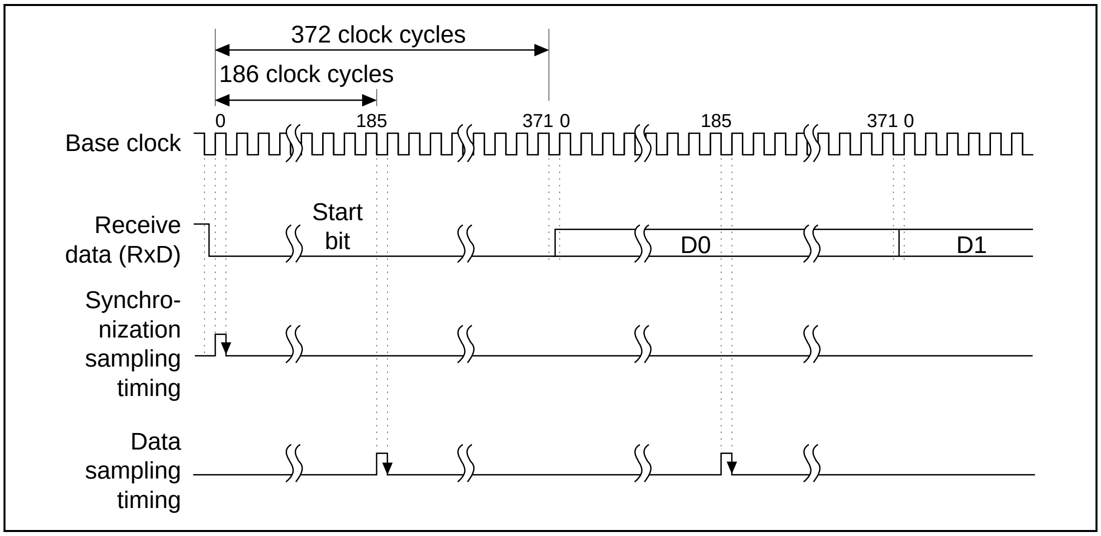


Figure 15.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Where:

M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

15.4.2 Retransmission (Receive and Transmit Modes)

Retransmission by the SCI in Receive Mode: Figure 15.9 shows the retransmission operation in the SCI receive mode.

1. When the received parity bit is checked and an error is found, the PER bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the PER bit before the next parity bit is sampled.
2. The RDRF bit in SCSSR is not set in the frame that caused the error.
3. When the received parity bit is checked and no error is found, the PER bit in SCSSR is not set.
4. When the received parity bit is checked and no error is found, reception is considered to have been completed normally and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state status when it transmits the error signal.

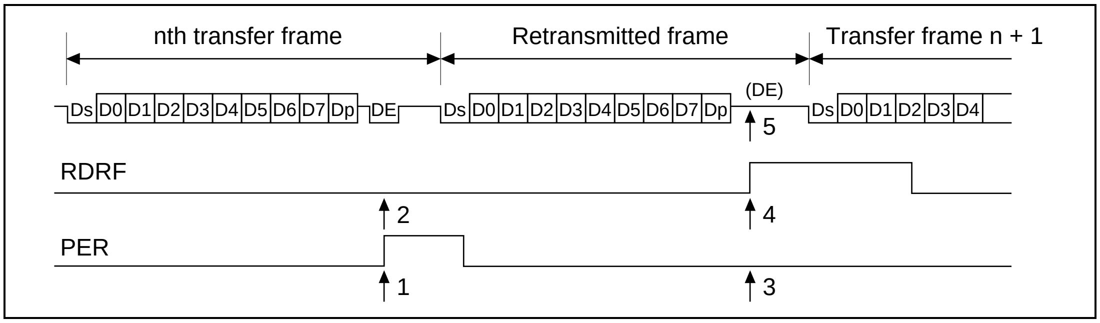


Figure 15.9 Retransmission in SCI Receive Mode

Retransmission by the SCI in Transmit Mode: Figure 15.10 shows the retransmission operation in the SCI transmit mode.

1. After transmission of one frame is completed, the FER/ERS bit in SCSSR is set to 1 when a error signal is returned from the receiving side. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the FER/ERS bit before the next parity bit is sampled.
2. The TEND bit in SCSSR is not set in the frame that received the error signal that indicated the error.
3. The FER/ERS bit in SCSR is not set when no error signal is returned from the receiving side.
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is set to 1 when the transmission of the frame that includes the retransmission is considered completed. If the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.

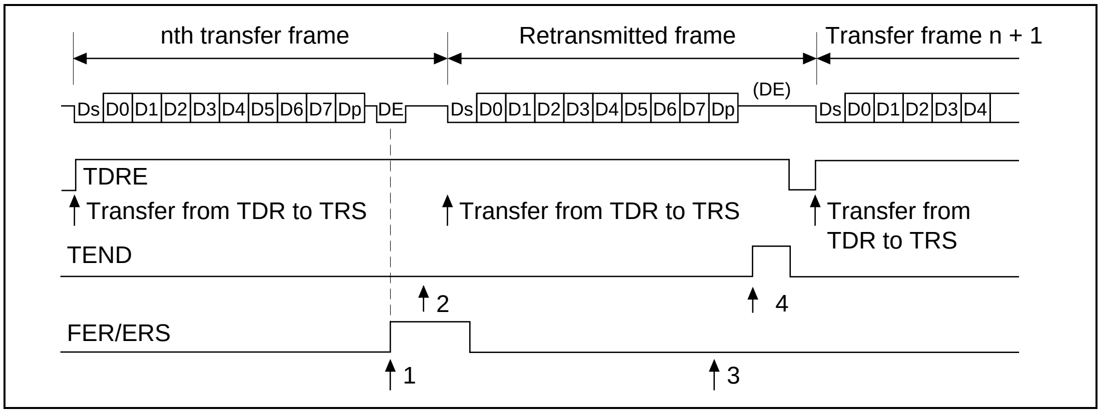


Figure 15.10 Retransmission in SCI Transmit Mode

Section 16 Serial Communication Interface with FIFO (SCIF)

16.1 Overview

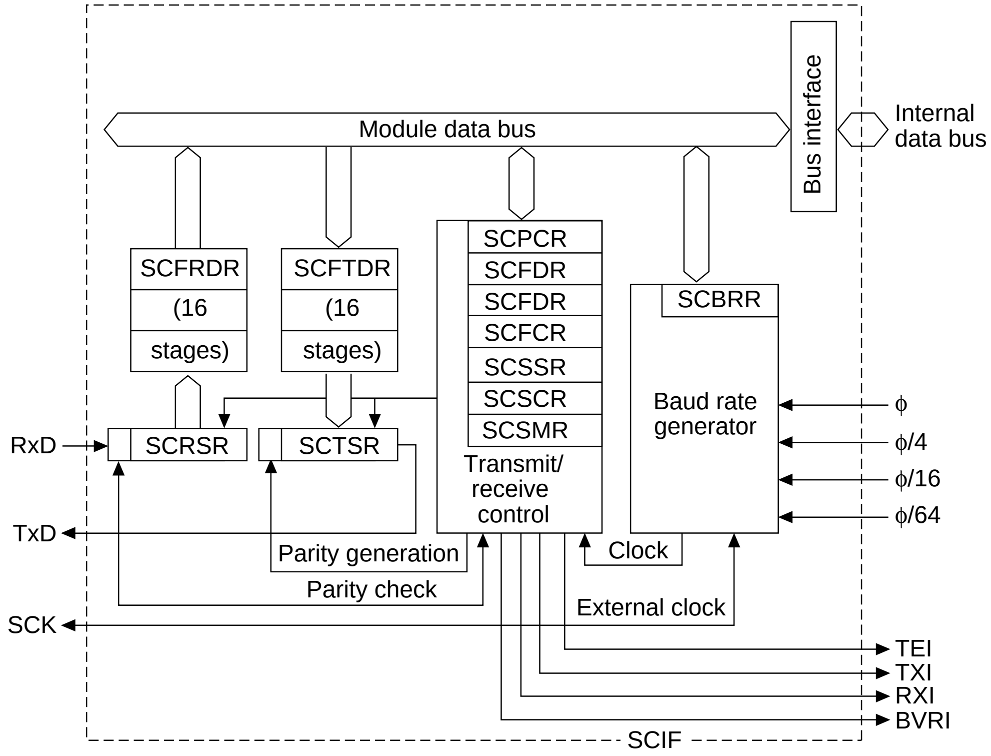
This LSI has two-channel serial communication interface with FIFO (SCIF) that supports asynchronous serial communication. It also has 16-stage FIFO registers for both transfer and receive that enables this LSI efficient high-speed continuous communication.

16.1.1 Features

- Asynchronous serial communication:
  - — Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - — Data length: Seven or eight bits
  - — Stop bit length: One or two bits
  - — Parity: Even, odd, or none
  - — Receive error detection: Parity and framing errors
  - — Break detection: Break is detected when the receive data next the generated framing error is the space ϕ level and has the framing error. It is also detected by reading the RxD level directly from the port SC data register (SCPDR) when a framing error occurs
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interrupts are requested independently. The direct memory access controller (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- On-chip modem control functions (RTS and CTS)
- The quantity of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be known.
- The time-out error (DR) can be detected in receiving.

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the SCI.

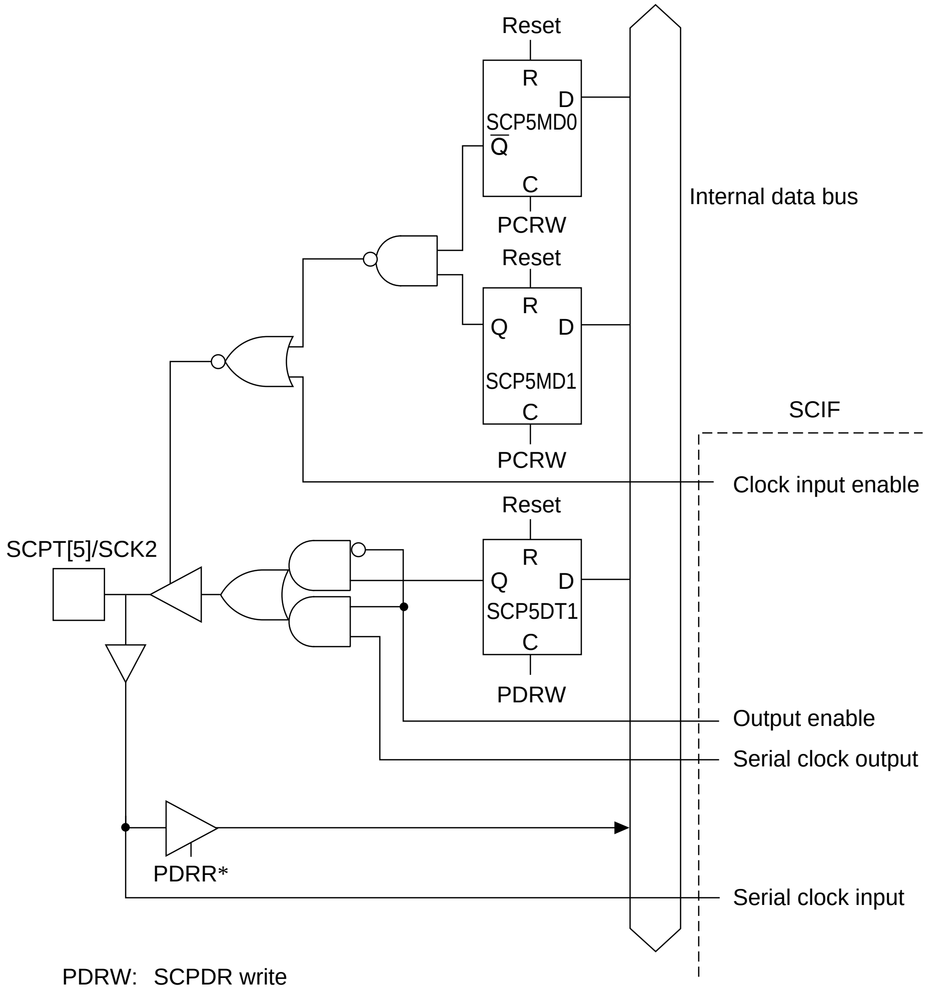


SCRSR:	Receive shift register	SCSSR:	Serial status register
SCFRDR:	Receive FIFO data register	SCBRR:	Bit rate register
SCTSR:	Transmit shift register	SCFCR:	FIFO control register
SCFTDR:	Transmit FIFO data register	SCFDR:	Number of FIFO data register
SCSMR:	Serial mode register	SCPDR:	Port SC data register
SCSCR:	Serial control register	SCPCR:	Port SC control register

Figure 16.1 SCI Block Diagram

Figures 16.2 to 16.4 show the SCIF I/O ports.

SCIF pin input/output and data control is performed by bits 11—8 of SCPCR and bits 5 and 4 of SCPDR. For details, see section 14.2.8.



PDRW:	SCPDR write
PDRR:	SCPDR read
PCRW:	SCPCR write

Note: * When reading the SCK2 pin, clear the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP5MD1 bit in SCSPR to 1 (see section 14.2.8).

Figure 16.2 SCPT[5]/SCK2 Pin

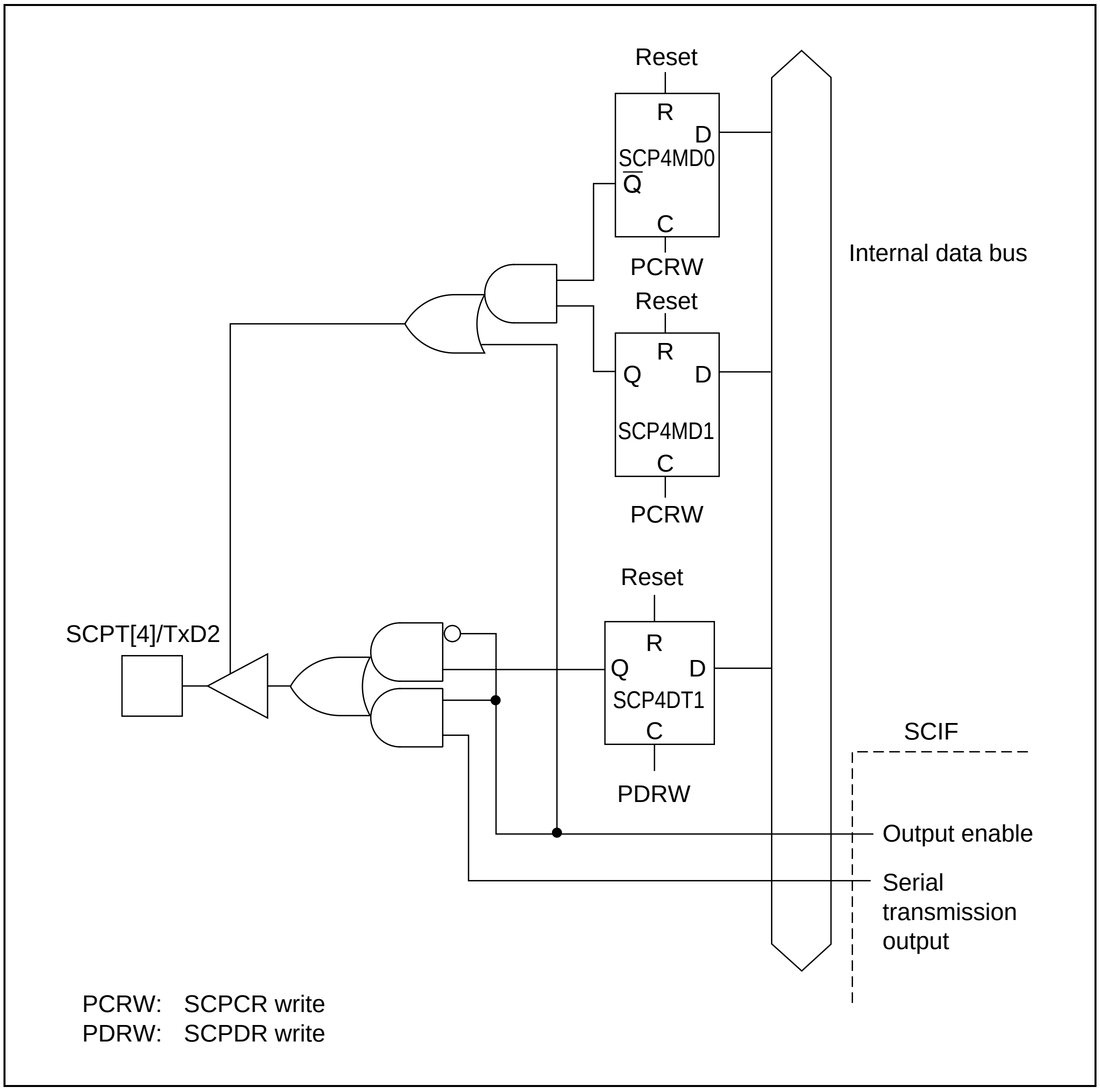
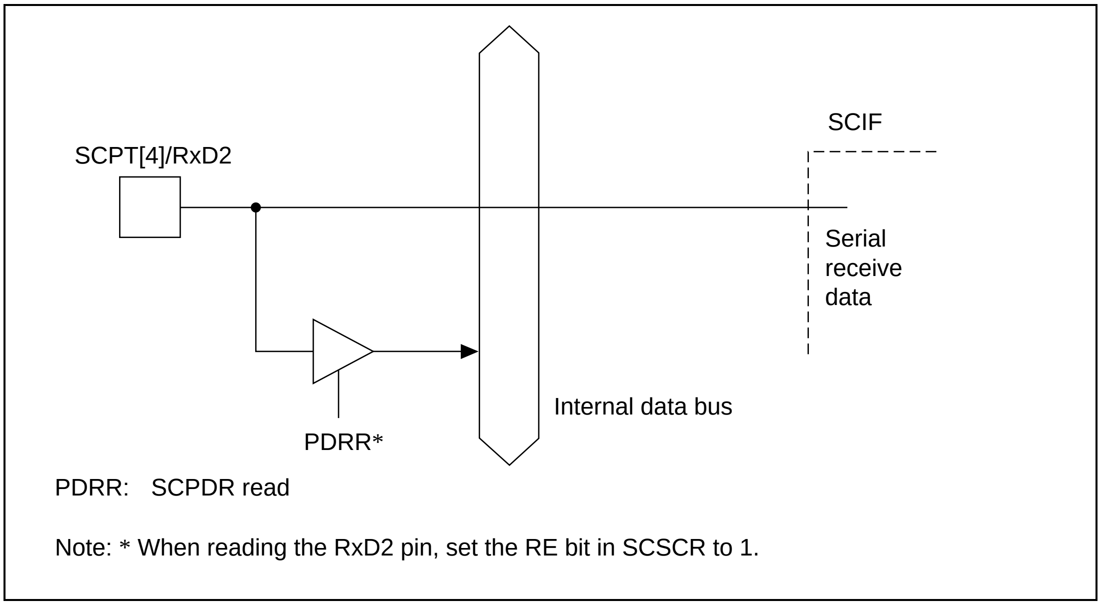


Figure 16.3 SCPT[4]/TxD2 Pin



PDRR: SCPDR read

Note: * When reading the RxD2 pin, set the RE bit in SCSCR to 1.

Figure 16.4 SCPT[4]/RxD2 Pin

16.1.3 Pin Configuration

The SCIF has the serial pins summarized in table 16.1.

Table 16.1 SCIF Pins

Pin Name	Symbol	Input/Output	Function
Serial clock pin	SCK2	Input/output	Clock input/output
Receive data pin	RxD2	Input	Receive data input
Transmit data pin	TxD2	Output	Transmit data output
Request to send pin	$\overline{\text{RTS2}}$	Output	Request to send
Clear to send pin	$\overline{\text{CTS2}}$	Input	Clear to send

16.1.4 Register Configuration

Table 16.2 summarizes the SCIF internal registers. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 16.2 Registers

Register Name	Abbr.	R/W	Initial Value	Address	Access size
Serial mode register 2	SCSMR2	R/W	H'00	H'04000150	8 bits
Bit rate register 2	SCBRR2	R/W	H'FF	H'04000152	8 bits
Serial control register 2	SCSCR2	R/W	H'00	H'04000154	8 bits
Transmit FIFO data register 2	SCFTDR2	W	—	H'04000156	8 bits
Serial status register 2	SCSSR2	R/(W)*1	H'0060	H'04000158	16 bits
Receive data FIFO register 2	SCFRDR2	R	Undefined	H'0400015A	8 bits
FIFO control register 2	SCFCR2	R/W	H'00	H'0400015C	8 bits
FIFO data count set register 2	SCFDR2	R	H'0000	H'0400015E	16 bits

Notes: 1. Only 0 can be written to clear the flag.
2. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

16.2 Register Descriptions

16.2.1 Receive Shift Register

The receive shift register (SCRSR) receives serial data. Data input at the RxD pin is loaded into the SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCFRDR, which is a receive FIFO register. The CPU cannot read or write the SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	—	—	—	—	—	—	—	—

16.2.2 Receive FIFO Data Register

The 16-byte receive FIFO data register (SCFRDR) stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into the SCFRDR for storage. Continuous receive is enabled until 16 bytes are stored.

The CPU can read but not write the SCFRDR. When data is read without received data in the receive FIFO data register, the value is undefined. When the received data in this register becomes full, the subsequent serial data is lost.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	R	R	R	R	R	R	R	R

16.2.3 Transmit Shift Register

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data from the transmit FIFO data register (SCFTDR) into the SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the SCFTDR into the SCTSR and starts transmitting again. The CPU cannot read or write the SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	—	—	—	—	—	—	—	—

16.2.4 Transmit FIFO Data Register

The transmit FIFO data register (SCFTDR) is a 16-byte 8-bit-length FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into the SCTSR and starts serial transmission. Continuous serial transmission is performed until the transmit data in the SCFTDR becomes empty. The CPU can always write to the SCFTDR.

When the transmit data in the SCFTDR is full (16 bytes), next data cannot be written. If attempted to write, the data is ignored.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
R/W:	W	W	W	W	W	W	W	W

16.2.5 Serial Mode Register

The serial mode register (SCSMR) is an eight-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR. The SCSMR is initialized to H'00 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	0	CHR	PE	O/E	STOP	0	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit 6—Character Length (CHR): Selects seven-bit or eight-bit data in the asynchronous mode.

Bit 6: CHR	Description
0	Eight-bit data (initial value)
1	Seven-bit data. (When seven-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.)

Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data.

Bit 5: PE	Description
0	Parity bit not added or checked (initial value)
1	Parity bit added and checked. When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/E) setting. Receive data parity is checked according to the even/odd (O/E) mode setting.

Bit 4—Parity Mode (O/E): Selects even or odd parity when parity bits are added and checked. The O/E setting is used only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The O/E setting is ignored when parity addition and check is disabled.

Bit 4: O/E	Description
0	Even parity (initial value). If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
1	Odd parity. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description
0	One stop bit (initial value). In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two bits of 1 are added at the end of each transmitted character.

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 13.2.9, Bit Rate Register.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ (initial value)
	1	Pϕ/4
1	0	Pϕ/16
	1	Pϕ/64

Note: Pϕ: Peripheral clock

16.2.6 Serial Control Register

The serial control register (SCSCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR. The SCSCR is initialized to H'00 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TIE	RIE	TE	RE	0	0	CKE1	CKE0
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 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag of the serial FIFO status register (SCFSR) is set to1.

Bit 7: TIE	Description
0	Transmit-FIFO-data-empty interrupt request (TXI) is disabled (initial value). The TXI interrupt request can be cleared by writing the greater quantity of transmit data than the specified number of transmission triggers to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.
1	Transmit-FIFO-data-empty interrupt request (TXI) is enabled

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full (RXI) and receive-error (ERI) interrupts requested when the serial receive data is transferred from the receive shift register (SCRSR) to receive FIFO data register (SCFRDR), when the quantity of data in the receive FIFO register becomes more than the specified number of receive triggers, and when the RDRF flag of SCSSR is set to1.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and receive break interrupt (BRI) requests are disabled (initial value). RXI and ERI interrupt requests can be cleared by reading the DR, ER, or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. At RDF, read 1 from the RDF flag and clear it to 0, after reading the received data from SCRDR until the quantity of received data becomes less than the specified number of the receive triggers.
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled.

Bit 5—Transmit Enable (TE): Enables or disables the SCIF serial transmitter.

Bit 5: TE	Description
0	Transmitter disabled (initial value).
1	Transmitter enabled. Serial transmission starts after writing of transmit data into the SCFTDR. Select the transmit format in the SCSMR and SCFCR and reset the TFIFO before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCIF serial receiver.

Bit 4: RE	Description
0	Receiver disabled (initial value). Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, FER, PER, and ORER). These flags retain their previous values.
1	Receiver enabled. Serial reception starts when a start bit is detected. Select the receive format in the SCSMR before setting RE to 1.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only when the SCI is operating with the internal clock (CKE1 = 0). The CKE0 setting is ignored when an external clock source is selected (CKE1 = 1). Before selecting the SCIF operating mode in the serial mode register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clock source, see table 16.8 in section 16.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description
0	0	Internal clock, SCK pin used for input pin (input signal is ignored) (Initial value)
	1	Internal clock, SCK pin used for clock output*[1]
1	0	External clock, SCK pin used for clock input*[2]
	1	External clock, SCK pin used for clock input*[2]

Notes: 1. The output clock frequency is 16 times the bit rate.
2. The input clock frequency is 16 times the bit rate.

16.2.7 Serial Status Register

The serial status register (SCSSR) is a 16-bit register. The upper 8 bits indicate the number of receive errors in the data of the receive FIFO register, and the lower 8 bits indicate SCI operating status.

The CPU can always read and write the SCSSR, but cannot write 1 in the status flags (ER, TEND, TDFE, BRK, OPER, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written. The SCSSR is initialized to H'0060 by a reset or in standby and module standby modes.

Lower 8 bits:	7	6	5	4	3	2	1	0
Bit name:	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	1	1	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: The only value that can be written is a 0 to clear the flag.

Bit 7—Receive Error (ER): Indicates that a parity error has occurred when received data includes a framing error or a parity. *[1]

Bit 7: ER	Description
0	Receive is in progress, or receive is normally completed. ER is cleared to 0 when the chip is reset or enters standby mode, or when 0 is written after 1 is read from ER.
1	A framing error or a parity error has occurred. ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one-data receive*[2], or when the total number of 1's in the received data and in the parity bit does not match the even/odd parity specification specified by the O/E bit of the SCSMR.

Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the FR bit, which retains its previous value. Even if a receive error occurs, the received data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCRDR includes a receive error can be detected by the FER and PER bits of SCSSR.

2. In 2-stop mode, only the first stop bit is checked; the second stop bit is not checked.

Bit 6—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the SCFTDR did not contain valid data, so transmission has ended.

Bit 6: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when data is written in SCTDR.
1	End of transmission (initial value). TEND is set to 1 when the chip is reset or enters standby mode, TE is cleared to 0 in the serial control register (SCSCR), or when SCFTDR does not contain received data when the last bit of a one-byte serial character is transmitted.

Bit 5—Transmit FIFO Data Empty (TDFE): Indicates that data is transferred from transmit FIFO data register (SCFTDR) to transmit shift register (SCTSR), the quantity of data in SCFTDR becomes less than the number of transmission triggers specified by the TTRG1 and TTRG0 bits in FIFO control register (SCFCR), and writing the transmit data to SCFTDR is enabled.

Bit 5: TDFE	Description
0	The quantity of transmit data written to SCFTDR is greater than the specified number of transmission triggers. TDFE is cleared to 0 when the data exceeding the specified number of transmission triggers is written to SCFTDR, software reads TDFE after it has been set to 1, then writes 0 in TDFE.
1	The quantity of transmit data in SCFTDR is less than the specified number of transmission triggers. TDFE is set to 1 at reset or at standby mode, or when the quantity of transmission data in SCFTDR becomes less than the specified number of transmission triggers as a result of transmission. *[1]

Note: Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data which can be written when TDFE is 1 is "16 minus the specified number of transmission triggers". If attempted to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFTDR.

Bit 4—Break Detection (BRK): Indicates that a break signal is detected in received data.

Bit 4: BRK	Description
0	No break signal is being received (initial value). BRK is cleared to 0 when the chip is reset or enters standby mode, or software reads BRK after it has been set to 1, then writes 0 in BRK.
1	The break signal is received.*1 BRK is set to 1 when data including a framing error is received and a framing error occurs with space 0 in the subsequent received data.

Note: When a break is detected, transfer of the received data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of the received data resumes. The received data of a frame in which a break signal is detected is transferred to SCFRDR. After this, however, no received data is transferred until a break ends with the received signal being mark 1 and the next data is received.

Bit 3—Framing Error (FER): Indicates a framing error in the data read from the receive FIFO data register (SCFRDR).

Bit 3: FER	Description
0	No receive framing error occurred in the data read from SCFRDR (initial value). FER is cleared to 0 when the chip is power on reset or enters standby mode, or when no framing error is present in the data read from SCFRDR.
1	A receive framing error occurred in the data read from SCFRDR. FER is set to 1 when a framing error is present in the data read from SCFRDR.

Bit 2—Parity Error (PER): Indicates a parity error in the data read from the receive FIFO data register (SCFRDR).

Bit 2: PER	Description
0	No receive parity error occurred in the data read from SCFRDR (initial value). PER is cleared to 0 when the chip is power on reset or enters standby mode, or when no parity error is present in the data read from SCFRDR.
1	A receive framing error occurred in the data read from SCFRDR. PER is set to 1 when a parity error is present in the data read from SCFRDR.

Bit 1—Receive FIFO Data Full (RDF): Indicates that received data is transferred to the receive FIFO data register (SCFRDR), the quantity of data in SCFRDR becomes more than the number of receive triggers specified by the RTRG1 and RTRG0 bits in FIFO control register (SCFCR).

Bit 1: RDF	Description
0	The quantity of transmit data written to SCFRDR is less than the specified number of receive triggers (initial value). RDF is cleared to 0 at power onreset or at standby mode, or cleared when the SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified number of receive triggers, when 1 is read from RDF, and 0 is then written.
1	The quantity of receive data in SCFRDR is more than the specified number of receive triggers. RDF is set to 1 when the quantity of receive data which is greater than the specified number of receive triggers is stored in SCFRDR. *[1]

Note: Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data which can be read when RDF is 1 is the specified number of receive triggers. If attempted to read after all data in the SCFRDR have been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFTDR.

Bit 0—Receive Data Ready (DR): Indicates that the receive FIFO data register (SCFRDR) stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 ETU has elapsed from the last stop bit.

Bit 0: DR	Description
0	Receive is in progress, or no received data remains in SCFRDR after completing receive normally (initial value). DR is cleared to 0 when the chip is power on reset or enters standby mode, or software reads DR after it has been set to 1, then writes 0 in DR.
1	Next receive data is not received. DR is set to 1 when SCFRDR stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 ETU has elapsed from the last stop bit. *[1]

Note: This is equivalent to 1.5 frames with the 8-bit 1-stop-bit format. (ETU: Element Time Unit)

Upper 8 bits:	15	14	13	12	11	10	9	8
Bit name:	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 15–12—Number of Parity Errors (PER): Indicates the quantity of data including a parity error in the received data stored in the receive FIFO data register (SCFRDR). The value indicated by the bits 15 to 12 represents the number of parity errors in SCFRDR.

Bits 11–8—Number of Framing Errors (FER): Indicates the quantity of data including a framing error in the received data stored in SCFRDR. The value indicated by the bits 11 to 8 represents the number of framing errors in SCFRDR.

16.2.8 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR. The SCBRR is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in the two channels.

Bit:	7	6	5	4	3	2	1	0
Bit name:								
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

The SCBRR setting is calculated as follows:

Asynchronous mode:

$N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

B: Bit rate (bit/s)

N: SCBRR setting for baud rate generator ($0 \le N \le 255$)

Pϕ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 16.3.)

Table 16.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error by the following formula:

$\text{Error } (\%) = \{P(\phi \times 10^6)/[(N + 1) \times B \times 64 \times 2^{2n-1}] - 1 \} \times 100$

Table 16.4 lists examples of SCBRR settings.

Table 16.4 Bit Rates and SCBRR Settings

	P ϕ (MHz)								
	2			2.097152			2.4576		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26
150	1	103	0.16	1	108	0.21	1	127	0.00
300	0	207	0.16	0	217	0.21	0	255	0.00
600	0	103	0.16	0	108	0.21	0	127	0.00
1200	0	51	0.16	0	54	–0.70	0	63	0.00
2400	0	25	0.16	0	26	1.14	0	31	0.00
4800	0	12	0.16	0	13	–2.48	0	15	0.00
9600	0	6	–6.99	0	6	–2.48	0	7	0.00
19200	0	2	8.51	0	2	13.78	0	3	0.00
31250	0	1	0.00	0	1	4.86	0	1	22.88
38400	0	1	–18.62	0	0	–14.67	0	1	0.00

Table 16.4 Bit Rates and SCBRR Settings (cont)

	P ϕ (MHz)								
	3			3.6864			4		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	212	0.03	2	64	0.70	2	70	0.03
150	1	155	0.16	1	191	0.00	1	207	0.16
300	1	77	0.16	1	95	0.00	1	103	0.16
600	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	4	–2.34	0	5	0.00	0	6	–6.99
31250	0	2	0.00	0	3	-7.84	0	3	0.00
38400	—	—	—	0	2	0.00	0	2	8.51

	P ϕ (MHz)								
	4.9152			5			6		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	0.31	2	88	–0.25	2	106	–0.44
150	1	255	0.00	2	64	0.16	2	77	0.16
300	1	127	0.00	1	129	0.16	1	155	0.16
600	0	255	0.00	1	64	0.16	1	77	0.16
1200	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	7	0.00	0	7	1.73	0	9	–2.34
31250	0	4	–1.70	0	4	0.00	0	5	0.00
38400	0	3	0.00	0	3	1.73	0	4	–2.34

Table 16.4 Bit Rates and SCBRR Settings (cont)

	P φ (MHz)								
	6.144			7.3728			8		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	79	0.00	2	95	0.00	2	103	0.16
300	1	159	0.00	1	191	0.00	1	207	0.16
600	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	0.00	0	5	0.00	0	6	–6.99

	P φ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	174	–0.26	2	177	–0.25	1	212	0.03	2	217	0.08
150	1	127	0.00	2	129	0.16	1	155	0.16	2	159	0.00
300	0	255	0.00	2	64	0.16	1	77	0.16	2	79	0.00
600	0	127	0.00	1	129	0.16	0	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	0	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	38	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	19	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	9	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	4	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	2	0.00	0	11	2.40
38400	0	1	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 16.4 Bit Rates and SCBRR Settings (cont)

	P ϕ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	2	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	1	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	-1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
115200	0	3	0.00	0	3	8.51	0	4	6.67	0	4	8.51
500000	0	0	-7.84	0	0	0.00	0	0	22.9	0	0	25.0

Table 16.4 Bit Rates and SCBRR Settings (cont)

	P φ (MHz)											
	24			**24.576**			**28.7**			**30**		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73
115200	0	6	–6.99	0	6	–4.76	0	7	–2.68	0	7	1.73
500000	0	1	–25.0	0	1	–23.2	0	1	–10.3	0	1	–6.25

Table 16.5 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used. Table 16.6 list the maximum rates for external clock input.

Table 16.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
P ϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 16.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

P ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

16.2.9 FIFO Control Register (SCFCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
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

The FIFO control register (SCFCR) resets the number of data in the transmit and receive FIFO registers, sets the number of trigger data, and contains the permit bit for the loop back test. The SCFCR is always read and written by the CPU. It is initialized to H'00 by the reset, the module standby function, or in the standby mode.

Bits 7 and 6—Trigger of the Number of Receive FIFO Data (RTRG1 and RTRG0): Set the number of receive data which sets the receive data full (RDF) flag in the serial status register (SCSSR). These bits set the RDF flag when the number of receive data stored in the receive FIFO register (SCFRDR) is increased more than the number of setting triggers listed below.

Bit 7: RTRG1	Bit 6: RTRG0	Number of Received Triggers
0	0	1*1
0	1	4
1	0	8
1	1	14

Note: 1. Initial state.

Bits 5 and 4—Trigger of the Number of Transmit FIFO Data (TTRG1 and TTRG0): Set the number of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCSSR). These bits set the TDFE flag when the number of transmit data in the transmit FIFO data register (SCFTDR) is decreased less than the number of setting triggers listed below.

Bit 5: TTRG1	Bit 4: TTRG0	Number of Transmitted Triggers
0	0	8 (8)*1
0	1	4 (12)
1	0	2 (14)
1	1	1 (15)

Note: 1. Initial state. Values in brackets mean the number of empty SCFTDR when a flag occurs.

Bit 3—Modem Control Enable (MCE): Enables the modem control signals CTS and RTS.

Bit 3: MCE	Description
0	Disables the modem signal *1 (Initial state)
1	Enables the modem signal

Note: The CTS is fixed to active 0 regardless of the input value, and the RTS is also fixed to 0.

Bit 2—Transmit FIFO Data Register Reset (TFRST): Disables the transmit data in the transmit FIFO data register and resets the data to the empty state.

Bit 2: TFRST	Description
0	Disables reset operation *1 (Initial state)
1	Enables reset operation

Note: Reset is operated in resets or the standby mode.

Bit 1—Receive FIFO Data Register Reset (RFRST): Disables the receive data in the receive FIFO data register and resets the data to the empty state.

Bit 1: RFRST	Description
0	Disables reset operation *1 (Initial state)
1	Enables reset operation

Note: Reset is operated in resets or the standby mode.

Bit 0—Loop Back Test (LOOP): Internally connects the transmit output pin (TXD) and receive input pin (RXD) and enables the loop back test.

Bit 0: LOOP	Description
0	Disables the loop back test (Initial state)
1	Enables the loop back test

16.2.10 Register of the Number of FIFO Data (SCFDR)

The SCFDR is a 16-bit register which indicates the number of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR). It indicates the number of transmit data in the SCFTDR with the upper eight bits, and the number of receive data in the SCFRDR with the lower eight bits. The SCFDR is always read from the CPU.

Lower 8 Bits:	7	6	5	4	3	2	1	0
Bit name:	0	0	0	R4	R3	R2	R1	R0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The SCFDR indicates the number of receive data stored in the SCFRDR. The H'00 means no receive data, and the H'10 means that the full of receive data are stored in the SCFRDR.

Upper 8 Bits:	15	14	13	12	11	10	9	8
Bit name:	0	0	0	T4	T3	T2	T1	T0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The SCFDR indicates the number of non-transmitted data stored in the SCFTDR. The H'00 means no transmit data, and the H'10 means that the full of transmit data are stored in the SCFTDR.

16.3 Operation

16.3.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually. Refer to section 14.3.2, SCI, for asynchronous mode operation. The SCIF has the 16-byte FIFO buffer for both transmit and receive, reduces an overhead of the CPU, and enables continuous high-speed communication. Moreover, it has the $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals as the modem control signals. The transmission format is selected in the serial mode register (SCSMR), as listed in table 16.6. The SCI clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as listed in table 16.7.

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), receive FIFO data full, receive data ready, and breaks.
- In transmitting, it is possible to detect transmit FIFO data empty.
- The number of stored data for both the transmit and receive FIFO registers is displayed.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Table 16.6 Serial Mode Register Settings and SCI Communication Formats

	SCSMR Settings			SCI Communication Format			
Mode	**Bit 6 CHR**	**Bit 5 PE**	**Bit 3 STOP**	**Data Length**	**Parity Bit**	**Multiprocessor Bit**	**Stop Bit Length**
Asynchronous	0	0	0	8-bit	Not set	Not set	1 bit
			1				2 bits
		1	0		Set		1 bit
			1				2 bits
	1	0	0	7-bit	Not set		1 bit
			1				2 bits
		1	0		Set		1 bit
			1				2 bits

Table 16.7 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSCR Settings		SCI Transmit/Receive Clock	
	Bit 1	Bit 0		
Mode	CKE1	CKE0	Clock Source	SCK Pin Function
Asynchronous mode	0	0	Internal	SCI does not use the SCK pin
		1		Outputs a clock with a frequency 16 times the bit rate
	1	0	External	Inputs a clock with frequency 16 times the bit rate
		1		

16.3.2 Serial Operation

Transmit/Receive Formats: Table 16.8 lists the eight communication formats that can be selected. The format is selected by settings in the serial mode register (SCSMR).

Table 16.8 Serial Communication Formats

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-Bit data								STOP		
0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	START	8-Bit data								P	STOP	
0	1	1	START	8-Bit data								P	STOP	STOP
1	0	0	START	7-Bit data							STOP			
1	0	1	START	7-Bit data							STOP	STOP		
1	1	0	START	7-Bit data							P	STOP		
1	1	1	START	7-Bit data							P	STOP	STOP	

Notes: START: Start bit
STOP: Stop bit
P: Parity bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/A bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 16.7).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

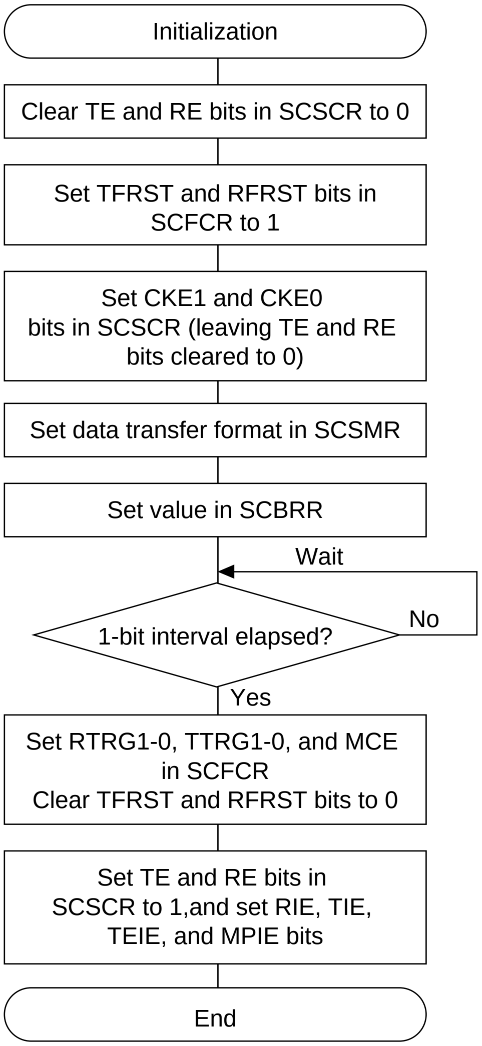
When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is 16 times the bit rate.

Transmitting and Receiving Data (SCIF Initialization): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCSSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data are transmitted and the TEND flag in the SCSSR is set. The transmitting data enters the high impedance state after clearing to 0 although the bit can be cleared to 0 in transmitting. Set the TFRST bit in the SCFCR to 1 and reset the SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 16.5 is a sample flowchart for initializing the SCIF. The procedure for initializing the SCIF is:


Initialization
Clear TE and RE bits in SCSCR to 0
Set TFRST and RFRST bits in SCFCR to 1
Set CKE1 and CKE0 bits in SCSCR (leaving TE and RE bits cleared to 0)
Set data transfer format in SCSMR
Set value in SCBRR
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1-0, TTRG1-0, and MCE in SCFCR Clear TFRST and RFRST bits to 0
Set TE and RE bits in SCSCR to 1,and set RIE, TIE, TEIE, and MPIE bits
End

1. Set the clock selection in SCSCR.
   Be sure to clear bits RIE TIE, TE, and RE to 0.
   When clock output is selected, it is output immediately after SCSCR settings are made.

2. Set the data transfer format in SCSMR.

3. Write a value corresponding to the bit rate into the bit rate register (SCBRR).
   (Not necessary if an external clock is used.)

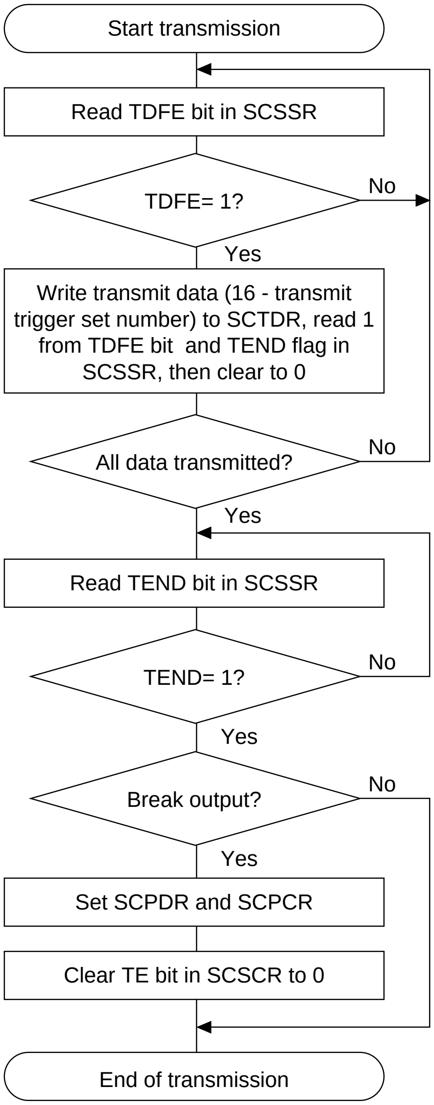
4. Wait at least one bit interval, then set the TE bit or RE bit in SCSR to 1. Also set the RIE and TIE bits.
   Setting the TE and RE bits enables the TxD and RxD pins to be used. When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 16.5 Sample SCIF Initialization Flowchart

- Serial data transmission

Figure 16.6 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



1. SCIF status check and transmit data write:

   Read serial status register (SCSSR) and check that the TDFE flag is set to 1, then write transmit data to the transmit FIFO data register (SCFTDR), read 1 from the TDFE and TEND flags, then clear these flags to 0.

   The number of transmit data bytes that can be written is 16 - (transmit trigger set number).

2. Serial transmission continuation procedure:

   To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

3. Break output at the end of serial transmission: To output a break in serial transmission, set the port SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE bit to 0 in the serial control register (SCSCR). For information on SCPDR and SCPCR, see section 14.2.8.

In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of the FIFO data register (SCFDR).

Figure 16.6 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCSSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 - transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit: One 0-bit is output.

   b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   c. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)

   d. Stop bit(s): One or two 1-bits (stop bits) are output.

   e. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.

Figure 16.7 shows an example of the operation for transmission.

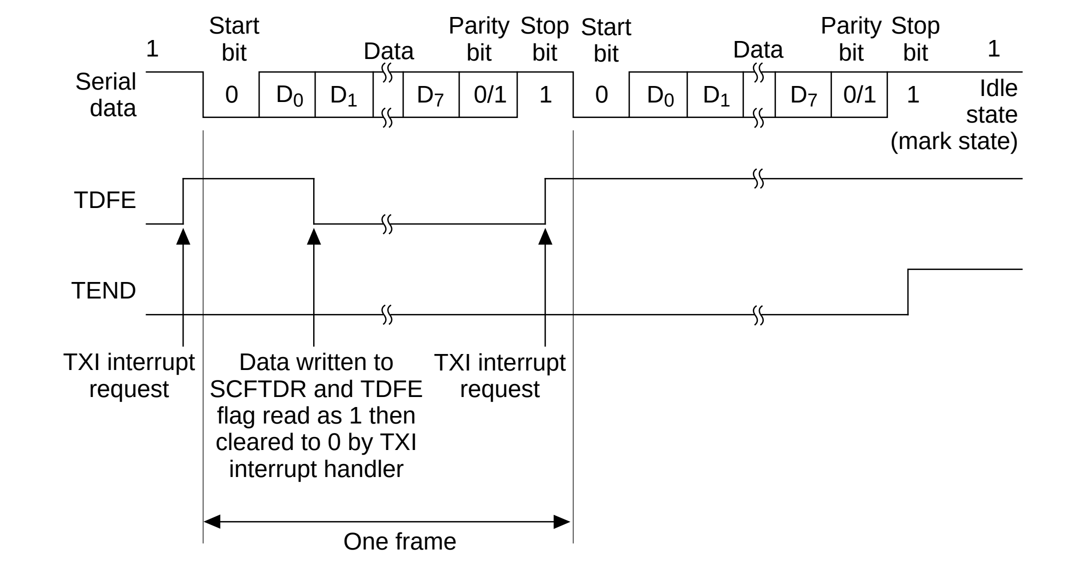


Figure 16.7 Example of Transmit Operation
(Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 16.8 shows an example of the operation when modem control is used.

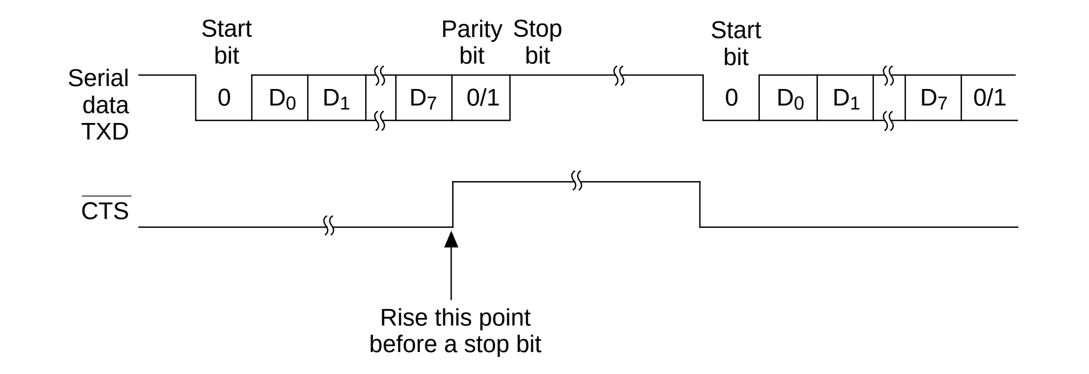
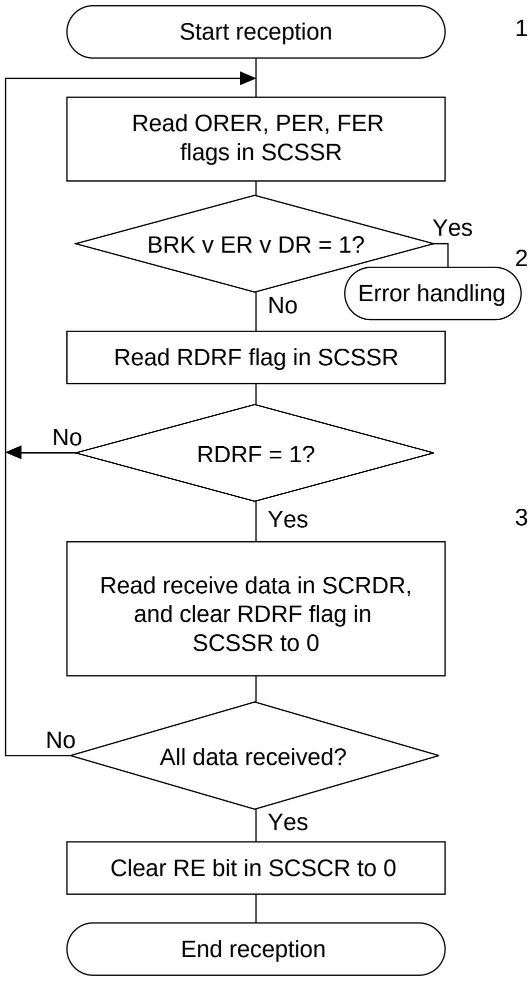


Figure 16.8 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

- Serial data reception

Figure 16.9 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

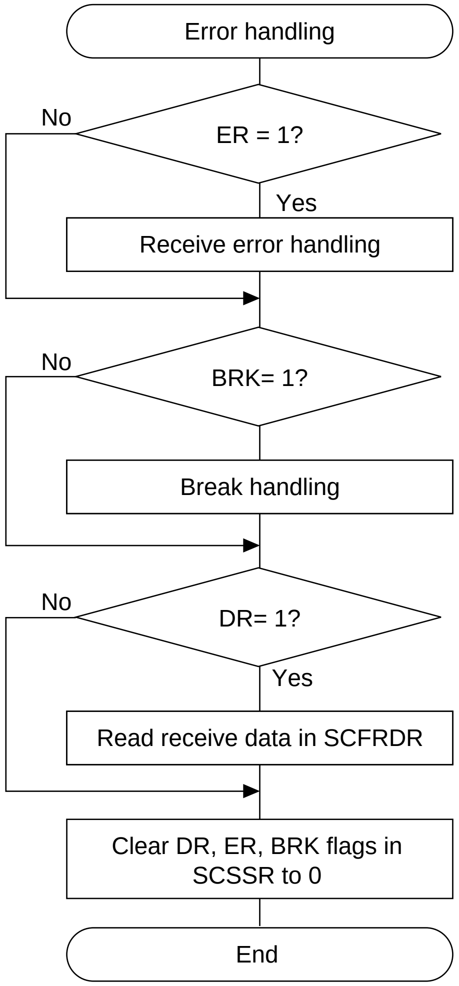


1. Receive error handling and break detection: Read the DR, ER, and BRK flags in SCSSR to identify any error, perform the appropriate error handling, then clear the DR, ER, and BRK flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

2. SCIF status check and receive data read : Read the serial status register (SCSSR) and check that RDF = 1, then read the receive data in the receive FIFO data register (SCFRDR), read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

3. Serial reception continuation procedure: To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDR flag, then clear the RDR flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading the lower bits of SCFDR.

Figure 16.9 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR can be ascertained from the FER and PER bits in SCSSR.

2. When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00 and the break data in which a framing error occurred is stored.

Figure 16.10 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 the receive shift register (SCRSR) to SCFRDR.
   c. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: Reception is not suspended when a receive error occurs.

4. If the RIE bit in SCSR 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 in SCSR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.
   If the RIE bit in SCSR is set to 1 when the BRK flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 16.11 shows an example of the operation for reception.

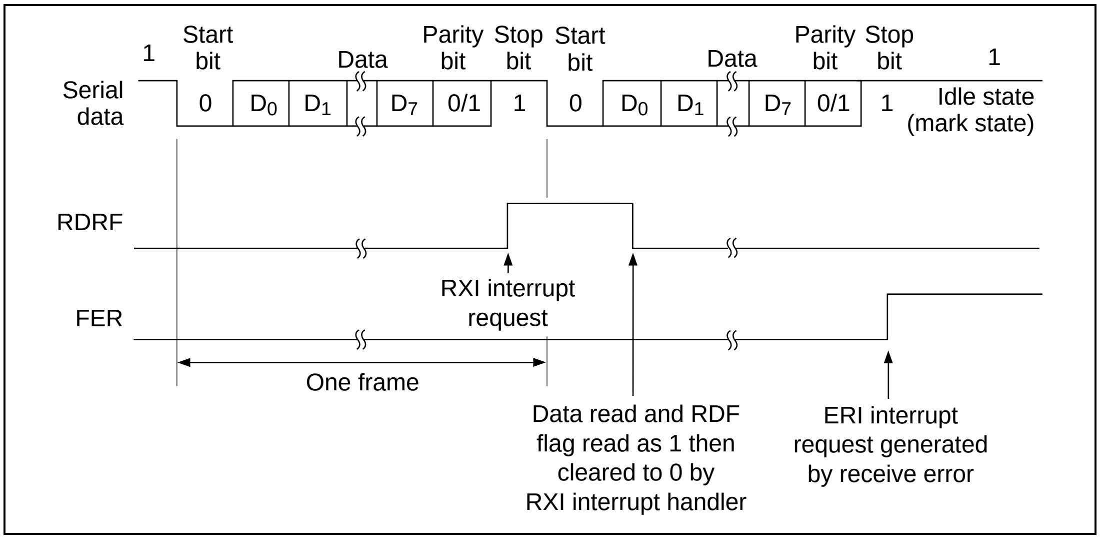


Figure 16.11 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR is full and reception is not possible.

Figure 16.12 shows an example of the operation when modem control is used.

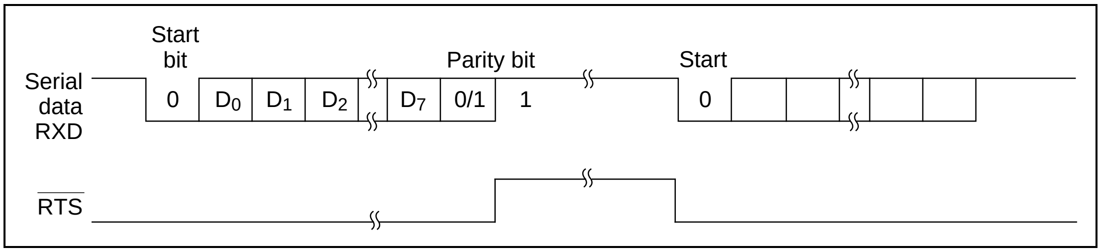


Figure 16.12 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

16.4 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 16.9 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE and RIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When the TDFE flag in the serial status register (SCSSR) is set to 1, a TXI interrupt request is generated. The DMAC can be activated and data transfer performed when this interrupt is generated. The TDFE flag is automatically cleared to 0 when data is written to the transmit data register (SCFTDR) by the DMAC.

When the RDF flag in SCSSR is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed when the RDF flag in SCSSR is set to 1. The RDF flag is automatically cleared to 0 when data is read from the receive data register (SCFRDR) by the DMAC.

When the ER flag in SCSSR is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR is set to 1, a BRI interrupt request is generated.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 16.9 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 data FIFO full flag (RDF) or data ready flag (DR)	Possible (RDF only)	↓
BRI	Interrupt initiated by break flag (BRK)	Not possible	
TxI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

See section 4, Exception Handling, for priorities and the relationship with non-SCIF interrupts.

16.5 Notes on Use

Note the following when using the SCIF.

1. SCFTDR Writing and the TDFE Flag: The TDFE flag in the serial status register (SCSSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.
   However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.
   The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).
2. SCFRDR Reading and the RDF Flag: The RDF flag in the serial status register (SCSSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.
   However, if the number of data bytes in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after all the receive data has been read.
   The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).
3. Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate, so if the BRK flag is cleared to 0 it will be set to 1 again.

4. Sending a Break Signal: The input/output condition and level of the TxD pin are determined by the SCP4DT bit in the port SC data register (SCPDR) and bits SCP4MD0 and SCP4MD1 in the port SC control register (SCPCR). This feature can be used to send a break signal.
   To send a break signal during serial transmission, clear the CP4DT bit to 0 (designating low level), then set the SCP4MD0 and SCP4MD1 bits to 0 and 1, respectively, and finally 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 status, and 0 is output from the TxD pin.
5. TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.
6. Receive Data Sampling Timing and Receive Margin: The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 16.13.

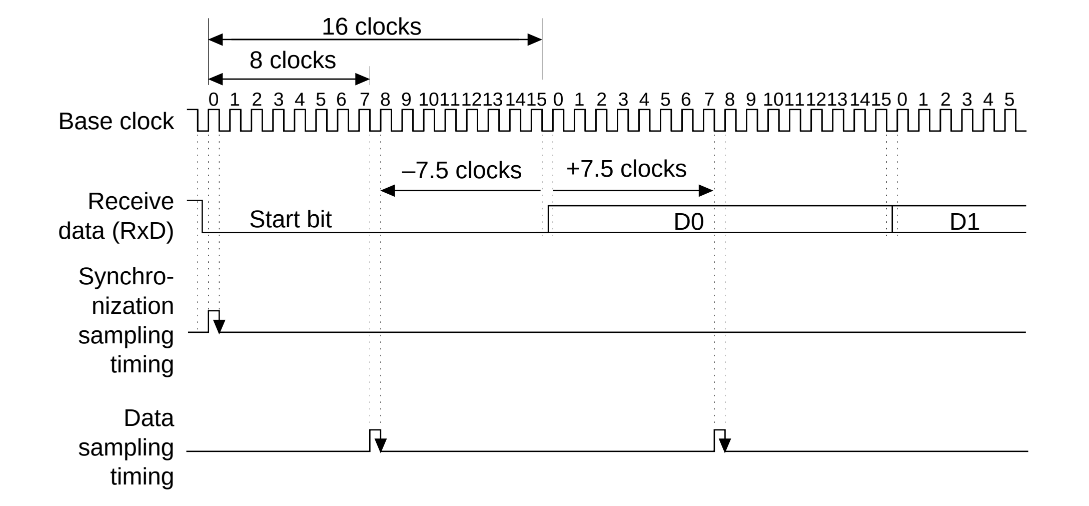


Figure 16.13 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \cdots\cdots (1)$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

When D = 0.5 and F = 0:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \cdots\cdots (2) \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 17 IrDA

17.1 Overview

This LSI has an on-chip infrared data association (IrDA) interface which is based on the IrDA 1.0 system and can perform infrared communication. It also can be used as the SCIF by setting registers.

17.1.1 Features

- Based on the IrDA 1.0 system
- Asynchronous serial communication
  - Data length: Eight bits
  - Stop bit length: One bit
  - Parity bit: None
- On-chip 16-stage FIFO buffers for both transmit and receive
- On-chip baud rate generator with selectable bit rates
- Guard functions not to affect the receiver in transmitting
- Clock supply halted to reduce power consumption in using no IrDA

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the IrDA.

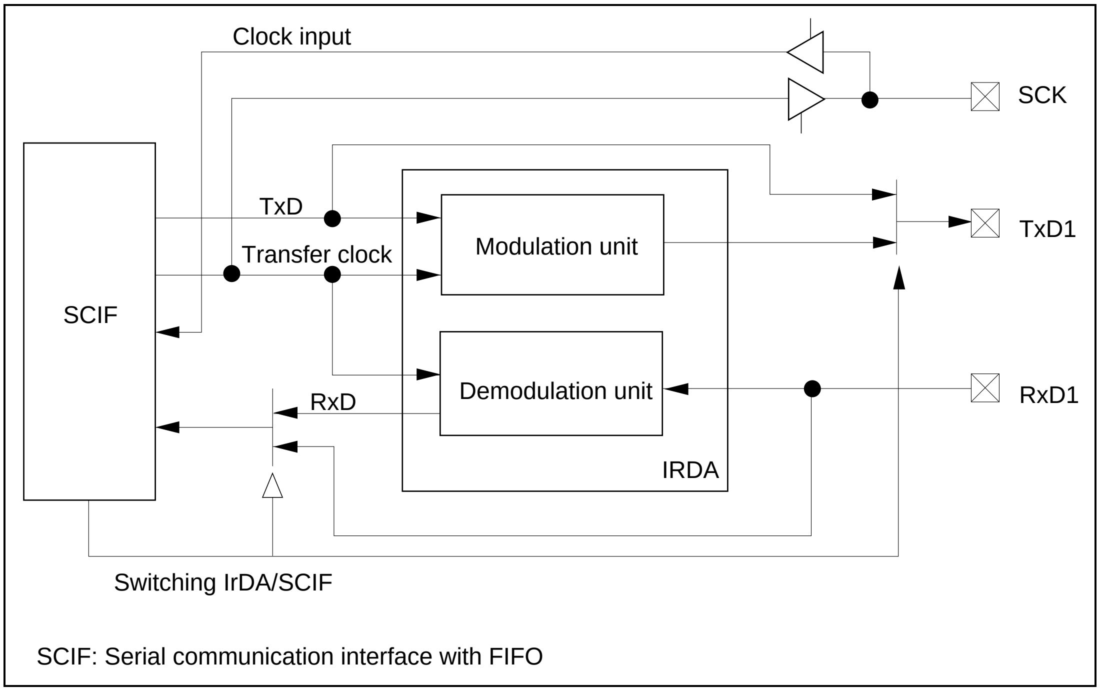
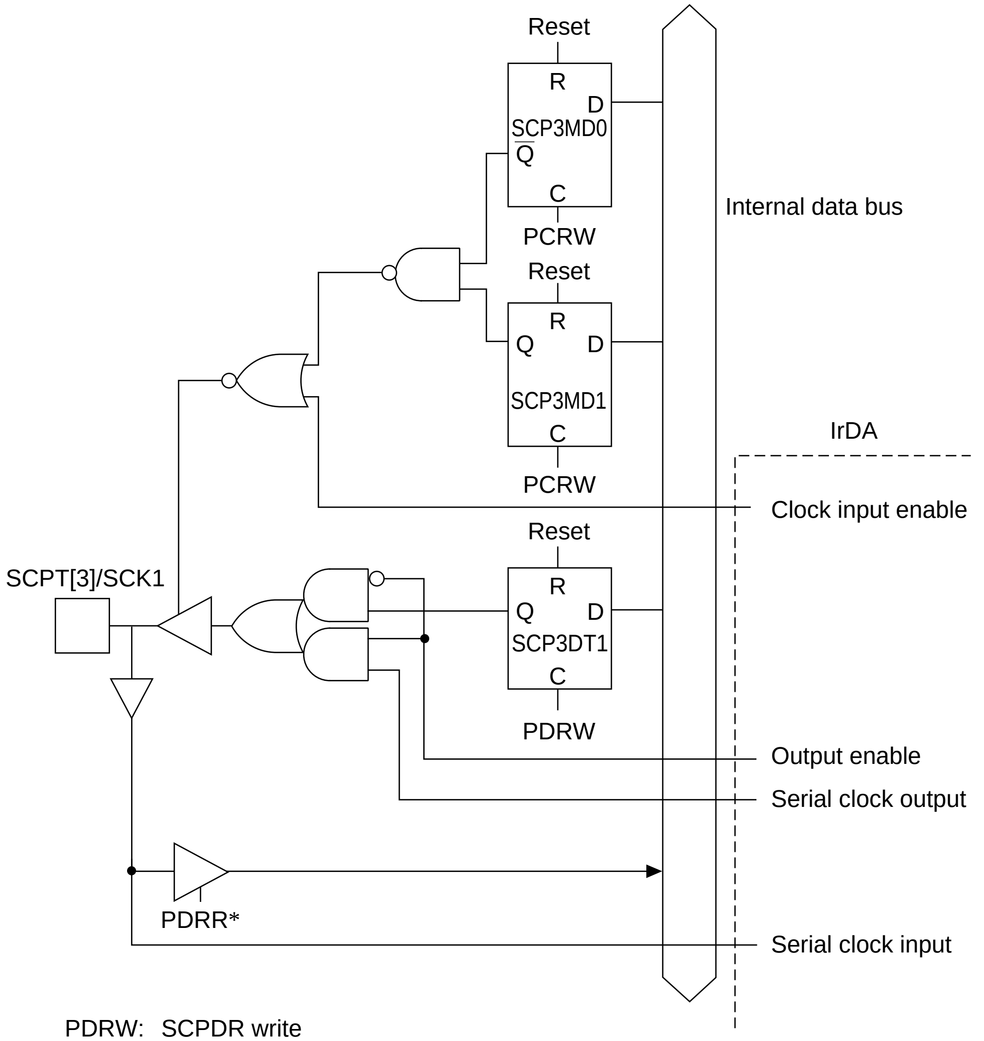


Figure 17.1 IrDA Block Diagram

Figures 17.2 to 17.4 show the IrDA I/O port pins.

SCIF pin input/output and data control is performed by bits 7—4 of SCPCR and bits 3 and 2 of SCPDR. For details, see section 14.2.8.



PDRW: SCPDR write
PDRR: SCPDR read
PCRW: SCPCR write

Note: * When reading the SCK1 pin, clear the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP3MD1 bit in SCSPR to 1 (see section 14.2.8).

Figure 17.2 SCPT[3]/SCK1 Pin

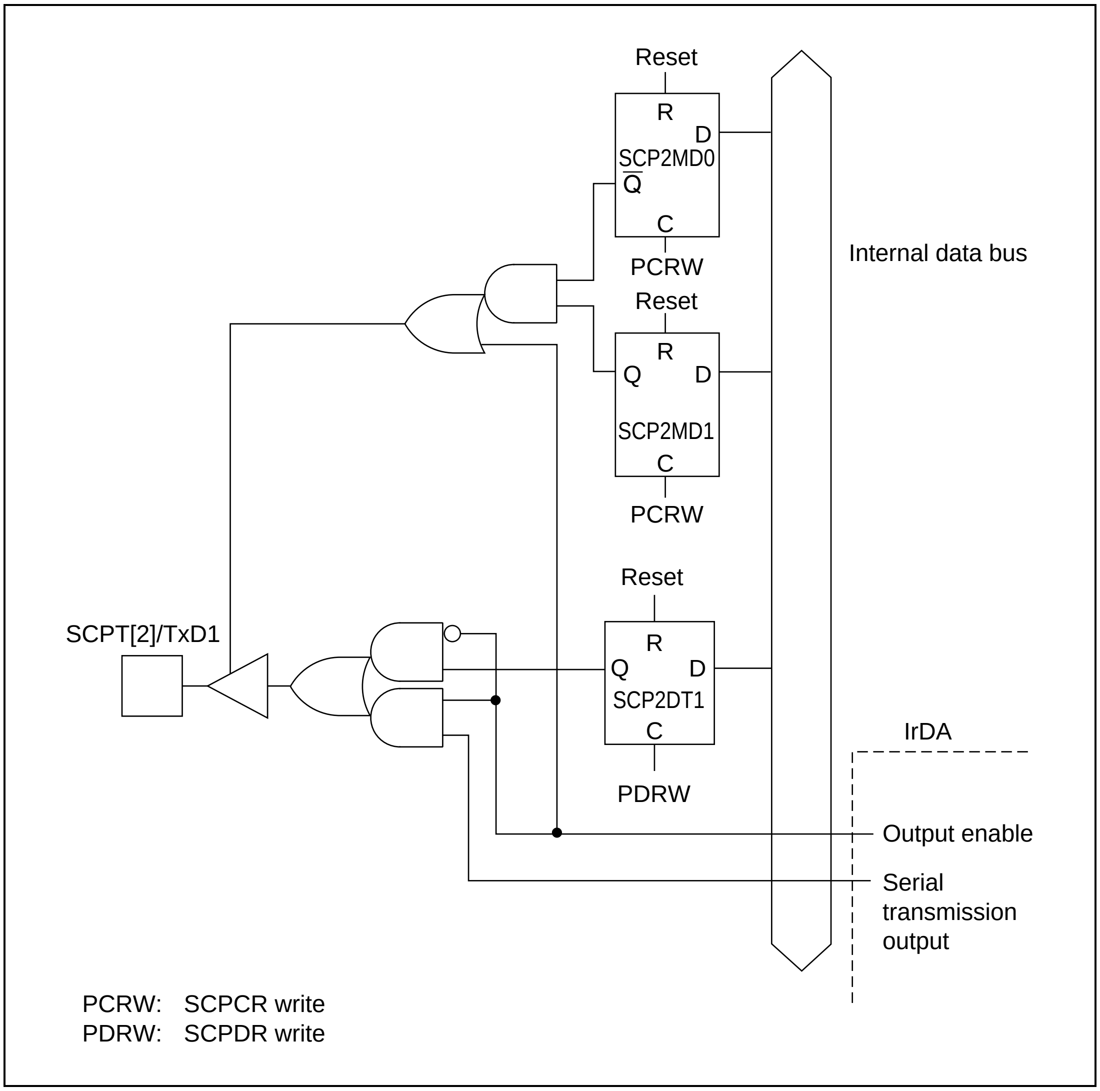


Figure 17.3 SCPT[2]/TxD1 Pin

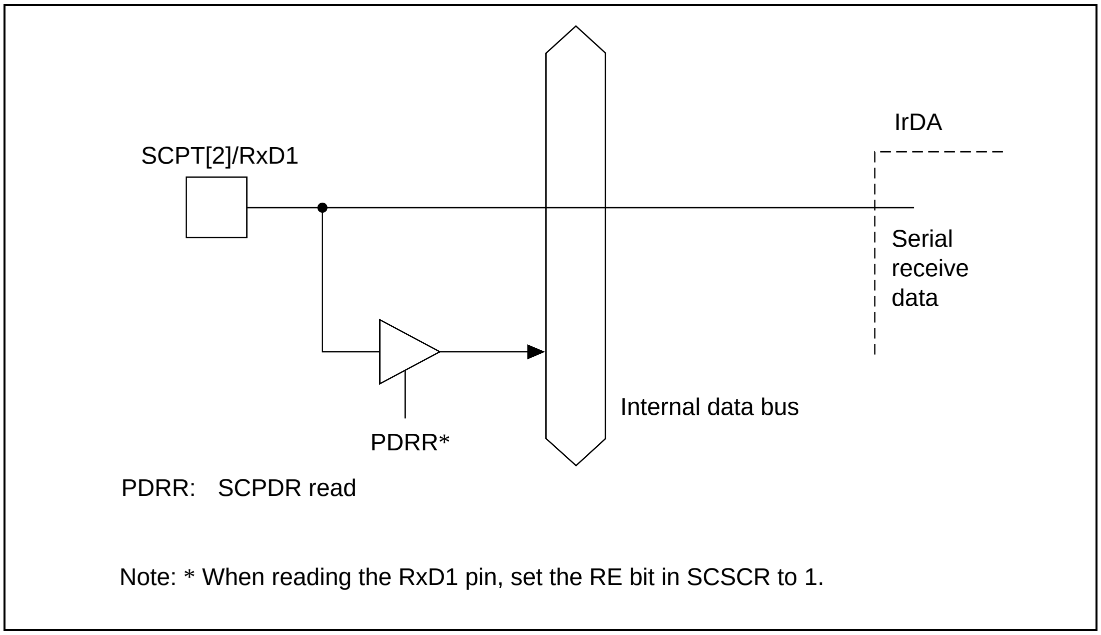


Figure 17.4 SCPT[2]/RxD1 Pin

17.1.3 Pin Configuration

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock pin	SCK1	Input/output	Clock input/output
Receive data pin	RxD1	Input	Receive data input
Transmit data pin	TxD1	Output	Transmit data output

17.1.4 Register Configuration

The IrDA has internal registers shown in table 17.2. By using these registers, an IrDA or an SCIF mode, a data format, and a bit rate can be specified, and a transmit and a receive unit can be controlled.

Table 17.2 Register Configuration

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Serial mode register 1	SCSMR1	R/W	H'00	H'04000140	8 bits
Bit rate register 1	SCBRR1	R/W	H'FF	H'04000142	8 bits
Serial control register 1	SCSCR1	R/W	H'00	H'04000144	8 bits
Transmit FIFO data register 1	SCFTDR1	W	—	H'04000146	8 bits
Serial status register 1	SCSSR1	R/(W)*1	H'0060	H'04000148	16 bits
Receive data FIFO register 1	SCFRDR1	R	Undefined	H'0400014A	8 bits
FIFO control register 1	SCFCR1	R/W	H'00	H'0400014C	8 bits
FIFO data count set register 1	SCFDR1	R	H'0000	H'0400014E	16 bits

Notes: 1. Only 0 can be written to clear the flag.

2. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

17.2 Register Description

Specifications of the registers in the IrDA are the same as those in the SCIF except for the serial mode register described below. Therefore, refer to section 16, Serial Communication Interface with FIFO, for these registers.

17.2.1 Serial Mode Register (SCSMR)

SCSMR is an 8-bit register, which can select an IrDA or an SCIF mode, specify an SCIF serial communication format, select an output pulse width of of IrDA and select a baud rate generator clock source.

This module operates as IrDA by setting the IRMOD bit to 1. At this time, Bits 3 to 6 are fixed to 0. This register functions in the same way as the SCSMR register in SCIF by setting the IRMOD bit to 0; therefore, this module can also operates as SCIF.

SCSMR is initialized to H'00 at power-on reset, manual reset, stop by module standby function, or standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	IRMOD	ICK3	ICK2	ICK1	ICK0	PSEL	CKS1	CKS0
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 7—IrDA mode (IRMOD)

This bit selects whether this module operates as an IrDA serial communication interface or as an SCIF.

Bit 7: IRMOD	Description
0	Operates as an SCIF
1	Operates as an IrDA

Bit 6 to 3—Ir clock select bits (ICK3 -ICK0)

Bit 2—Output pulse width select (PSEL)

Output pulse width select bit (PSEL) selects an output pulse width of IrDA that is 3/16 of bit length for 115kbps or 3/16 of a bit length for selected baud rate.

Ir clock select bits should be set properly to fix an output pulse width to 3/16 of bit length for 115K bps by setting PSEL bit to 1.

Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Description
ICK3	ICK2	ICK1	ICK0	PSEL	pulse width:3/16 of 115kbps bit length
ICK3	ICK2	ICK1	ICK0	1	
don't care				0	pulse width:3/16 of a bit length

It is necessary to generate a fixed clock pulse, IRCLK, by dividing Pϕ clock to 1/2N+2 with a value N determined by setting of ICK3-ICK0.

Example;

Pϕ clock:14.7456 MHz

IRCLK:921.6KHz (fixed)

N:setting of ICK3-ICK0 ($0^{2}N^{2}15$)

$$N \geq \frac{P\phi}{2XIRCLK} - 1 ^{3} 7$$

Accordingly, N is 7.

Bits 1 and 0—Clock select 1 and 0 (CKS1 and CKS0)

This bit selects an internal baud rate generator clock source. Pϕ, Pϕ/4, Pϕ/16, or Pϕ/64 can be selected by setting the CKS1 and CKS0 bits.

Refer to section 16.2.9, Bit Rate Register, for relationships among the clock source, the bit rate register set value, and the baud rate.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ clock (initial value)
0	1	Pϕ/4 clock
1	0	Pϕ/16
1	1	Pϕ/64

Note: Peripheral clock

17.3 Operation Description

The IrDA module can perform a infrared communication conforming to IrDA 1.0 by connecting a infrared transmit/receive unit. A serial communication interface unit includes a 16-stage FIFO buffer in a transmit unit and a receive unit, and therefore CPU overhead can be reduced and a high-speed communication can be successively performed. This module also supports DMAC data transfer. The IrDA module differs from the SCIF in section 16 in that it does not include modem control signals RTS and CTS.

Refer to section 16.3, SCIF Operation, for SCIF mode operation.

17.3.1 Overview

The IrDA module modifies TxD/RxD transmit/receive data waveforms to satisfy the IrDA 1.0 specification of the infrared communication.

In the IrDA 1.0 specification, communication is first performed in the rate of 9600 bps, and the communication rate is changed. However, the communication rate cannot be automatically changed in this module, and therefore, perform communication by checking the communication rate and setting the appropriate rate in this module with software.

Note: In IrDA mode, reception cannot be performed when the TE bit in the serial control register (SCSCR) is set to 1 (enabling transmission). When performing reception, clear the TE bit in SCSCR to 0.

As the SH7709's RxD1 pin is active-high in IrDA mode, a (Schmitt) inverter must be inserted when connecting an active-low IrD module.

The RxD1 pin is active-low in SCIF mode.

17.3.2 Transmit

As for the serial output signal (UART frame) from the SCIF, its waveforms are modified and the signal is converted into the IR frame serial output signal by the IrDA module, as shown in figure 17.5.

When serial data is 0, the 3/16-bit width pulse of the IR frame is generated and output. When serial data is 1, a pulse is not output.

The infrared LED is driven with this signal that was demodulated into 3/16 width.

17.3.3 Receive

The 3/16-bit width pulse of the IR frame that was received is demodulated and converted into the UART frame, as shown in figure 17.5.

Demodulation to 0 is performed for pulse output, and demodulation to 1 is performed for no pulse output.

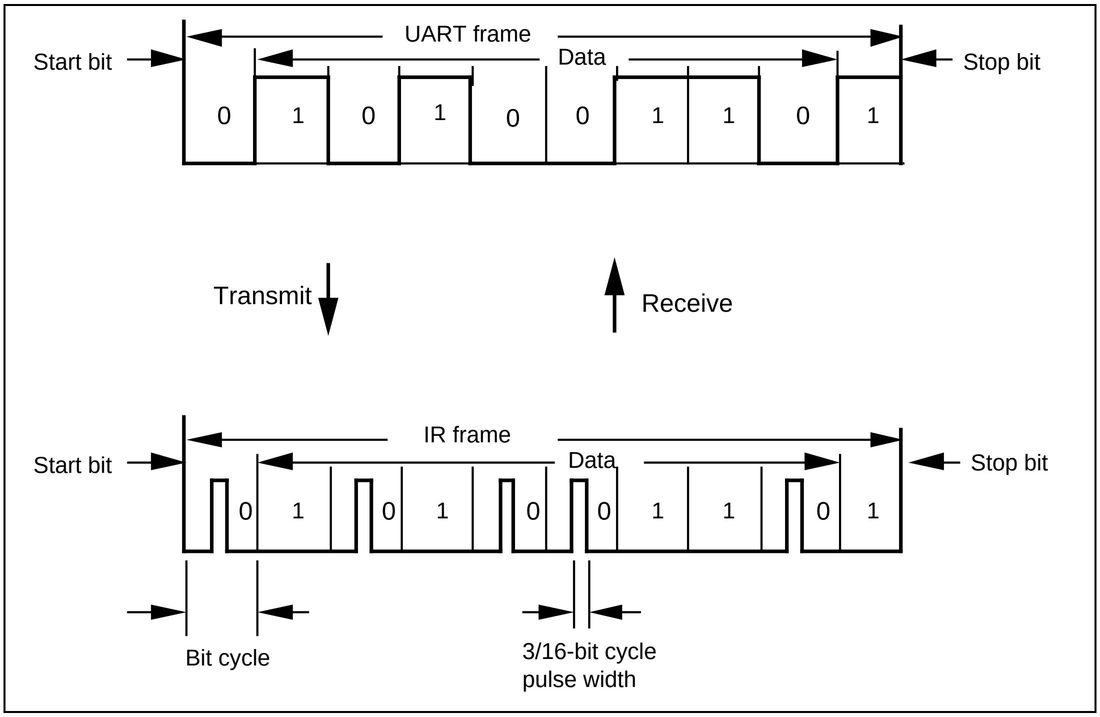


Figure 17.5 Transmit/Receive Operation

Section 18 Pin Function Controller

18.1 Overview

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the direction of input/output. The pin function and input/output direction can be selected for each pin individually without regard to the operating mode of the LSI. Table 18.1 lists the multiplexed pins.

Table18.1 List of Multiplexed Pins

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 input/output (port)	D23 input/output (data bus)
A	PTA6 input/output (port)	D22 input/output (data bus)
A	PTA5 input/output (port)	D21 input/output (data bus)
A	PTA4 input/output (port)	D20 input/output (data bus)
A	PTA3 input/output (port)	D19 input/output (data bus)
A	PTA2 input/output (port)	D18 input/output (data bus)
A	PTA1 input/output (port)	D17 input/output (data bus)
A	PTA0 input/output (port)	D16 input/output (data bus)
B	PTB7 input/output (port)	D31 input/output (data bus)
B	PTB6 input/output (port)	D30 input/output (data bus)
B	PTB5 input/output (port)	D29 input/output (data bus)
B	PTB4 input/output (port)	D28 input/output (data bus)
B	PTB3 input/output (port)	D27 input/output (data bus)
B	PTB2 input/output (port)	D26 input/output (data bus)
B	PTB1 input/output (port)	D25 input/output (data bus)
B	PTB0 input/output (port)	D24 input/output (data bus)
C	PTC7 I/O (port)/PINT7 input (INTC)	—
C	PTC6 I/O (port)/PINT6 input (INTC)	—
C	PTC5 I/O (port)/PINT5 input (INTC)	—
C	PTC4 I/O (port)/PINT4 input (INTC)	—
C	PTC3 I/O (port)/PINT3 input (INTC)	—
C	PTC2 I/O (port)/PINT2 input (INTC)	—
C	PTC1 I/O (port)/PINT1 input (INTC)	—

Table18.1 List of Multiplexed Pins (cont)

Port	Port Function (Related Module)	Other Function (Related Module)
C	PTC0 I/O (port)/PINT0 input (INTC)	—
D	PTD7 I/O (port)	DACK1 output (DMAC)
D	PTD6 input (port)	$\overline{\text{DREQ1}}$ output (DMAC)
D	PTD5 I/O (port)	DACK0 output (DMAC)
D	PTD4 input (port)	$\overline{\text{DREQ0}}$ output (DMAC)
D	PTD3 I/O (port)	$\overline{\text{WAKEUP}}$ output (INTC)
D	PTD2 I/O (port)	$\overline{\text{RESETOUT}}$
D	PTD1 I/O (port)	DRAK0 output (DMAC)
D	PTD0 I/O (port)	DRAK1 output (DMAC)
E	PTE7 I/O (port)	—
E	PTE6 I/O (port)	$\overline{\text{CAS2L}}$ output (BCN)
E	PTE5 I/O (port)	$\overline{\text{CE2B}}$ output (PCMCIA)
E	PTE4 I/O (port)	$\overline{\text{CE2A}}$ output (PCMCIA)
E	PTE3 I/O (port)	$\overline{\text{CAS2H}}$ output (BCN)
E	PTE2 I/O (port)	$\overline{\text{RAS3U}}$ output (BCN)
E	PTE1 I/O (port)	$\overline{\text{RAS2U}}$ output (BCN)
E	PTE0 I/O (port)	—
F	PTF7 input (port)/PINT15 input (INTC)	—
F	PTF6 input (port)/PINT14 input (INTC)	—
F	PTF5 input (port)/PINT13 input (INTC)	—
F	PTF4 input (port)/PINT12 input (INTC)	—
F	PTF3 input (port)/PINT11 input (INTC)	—
F	PTF2 input (port)/PINT10 input (INTC)	—
F	PTF1 input (port)/PINT9 input (INTC)	—
F	PTF0 input (port)/PINT8 input (INTC)	—
G	PTG7 input (port)	$\overline{\text{IOIS16}}$ input (PCMCIA)
G	PTG6 input (port)	—
G	PTG5 input (port)	—
G	PTG4 input (port)	—
G	PTG3 input (port)	—
G	PTG2 input (port)	—

Table18.1 List of Multiplexed Pins (cont)

Port	Port Function (Related Module)	Other Function (Related Module)
G	PTG1 input (port)	—
G	PTG0 input (port)	—
H	PTH7 I/O (port)	TCLK I/O (timer)
H	PTH6 input (port)	—
H	PTH5 input (port)	$\overline{\text{ADTRG}}$ input (ADC)
H	PTH4 input (port) / IRQ4 input (INTC)	$\overline{\text{IRQ4}}$ input (INTC)
H	PTH3 input (port) / IRQ3 input (INTC)	$\overline{\text{IRQ3}}$ input (INTC)
H	PTH2 input (port) / IRQ2 input (INTC)	$\overline{\text{IRQ2}}$ input (INTC)
H	PTH1 input (port) / IRQ1 input (INTC)	$\overline{\text{IRQ1}}$ input (INTC)
H	PTH0 input (port) / IRQ0 input (INTC)	$\overline{\text{IRQ0}}$ input (INTC)
J	PTJ7 input/output (port)	STATUS1 output (SYSC)
J	PTJ6 input/output (port)	STATUS0 output (SYSC)
J	PTJ5 input/output (port)	$\overline{\text{CASHH}}$ output (BCN)
J	PTJ4 input/output (port)	$\overline{\text{CASHL}}$ output (BCN)
J	PTJ3 input/output (port)	$\overline{\text{CASLH}}$ output (BCN)
J	PTJ2 input/output (port)	$\overline{\text{CASLL}}$ output (BCN) / $\overline{\text{CAS}}$ output (BCN)
J	PTJ1 input/output (port)	$\overline{\text{RAS2L}}$ output (BCN)
J	PTJ0 input/output (port)	$\overline{\text{RAS3L}}$ output (BCN)
K	PTK7 input/output (port)	$\overline{\text{WE3}}$ output (BCN) / DQMUU output (BCN) / $\overline{\text{ICIOWR}}$ output (BCN)
K	PTK6 input/output (port)	$\overline{\text{WE2}}$ output (BCN) / DQMUL output (BCN) / $\overline{\text{ICIORD}}$ output (BCN)
K	PTK5 input/output (port)	CKE output (BCN)
K	PTK4 input/output (port)	$\overline{\text{BS}}$ output (BCN)
K	PTK3 input/output (port)	$\overline{\text{CS5}}$ output (BCN) / $\overline{\text{CE1A}}$ output (BCN)
K	PTK2 input/output (port)	$\overline{\text{CS4}}$ output (BCN)
K	PTK1 input/output (port)	$\overline{\text{CS3}}$ output (BCN)
K	PTK0 input/output (port)	$\overline{\text{CS2}}$ output (BCN)
L	PTL7 input (port)	AN7 input (A/D) / DA0 output (D/A)
L	PTL6 input (port)	AN6 input (A/D) / DA1 output (D/A)
L	PTL5 input (port)	AN5 input (A/D

Table18.1 List of Multiplexed Pins (cont)

Port	Port Function (Related Module)	Other Function (Related Module)
L	PTL4 input (port)	AN4 input (A/D)
L	PTL3 input (port)	AN3 input (A/D)
L	PTL2 input (port)	AN2 input (A/D)
L	PTL1 input (port)	AN1 input (A/D)
L	PTL0 input (port)	AN0 input (A/D)
SCPT	SCPT7 input (port) / IRQ5 input (INTC)	CTS2 input (UART CH3) / IRQ5 input (INTC)
SCPT	SCPT6 input/output (port)	RTS2 input (UART ch3)
SCPT	SCPT5 input/output (port)	SCK2 input/output (UART ch 3)
SCPT	SCPT4 input (port)	RxD2 input (UART ch 3)
	SCPT4 output (port)	TxD2 output (UART ch 3)
SCPT	SCPT3 input/output (port)	SCK1 input/output (UART ch 2)
SCPT	SCPT2 input (port)	RxD1 input (UART ch 2)
	SCPT2 output (port)	TxD1 output (UART ch 2)
SCPT	SCPT1 input (port)	SCK0 input/output (UART ch 1)
SCPT	SCPT0 input (port)	RxD0 input (UART ch 1)
	SCPT0 output (port)	TxD0 output (UART ch 1)

Note: SCPT, SCPT, and SCP4 have the same data register to be accessed although they have different input pins and output pins.

18.2 Register Configuration

Table 18.2 summarizes the registers of the pin function controller.

Table18.2 Pin Function Controller Registers

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port A control register	PACR	R/W	H'0000	H'04000100	16
Port B control register	PBCR	R/W	H'0000	H'04000102	16
Port C control register	PCCR	R/W	H'AAAA	H'04000104	16
Port D control register	PDCR	R/W	H'AA8A	H'04000106	16
Port E control register	PECR	R/W	H'AAAA	H'04000108	16
Port F control register	PFCR	R/W	H'AAAA	H'0400010A	16
Port G control register	PGCR	R/W	H'AAAA	H'0400010C	16
Port H control register	PHCR	R/W	H'AAAA	H'0400010E	16
Port J control register	PJCR	R/W	H'0000	H'04000110	16
Port K control register	PKCR	R/W	H'0000	H'04000112	16
Port L control register	PLCR	R/W	H'0000	H'04000114	16
SC port control register	SCPCR	R/W	H'A888	H'04000116	16

Note: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

18.3 Register Descriptions

18.3.1 Port A Control Register (PACR)

Port A Control Register (PACR) is a 16-bit read/write register that selects the pin functions. PACR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	PA4 MD1	PA4 MD0	PA3 MD1	PA3 MD0	PA2 MD1	PA2 MD0	PA1 MD1	PA1 MD0	PA0 MD1	PA0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15,14—PA7 Mode 1, 0 (PA7MD1, PA7MD0)

Bit 13,12—PA6 Mode 1, 0 (PA6MD1, PA6MD0)

Bit 11,10—PA5 Mode 1, 0 (PA5MD1, PA5MD0)

Bit 9,8—PA4 Mode 1, 0 (PA4MD1, PA4MD0)

Bit 7,6—PA3 Mode 1, 0 (PA3MD1, PA3MD0)

Bit 5,4—PA2 Mode 1, 0 (PA2MD1, PA2MD0)

Bit 3,2—PA1 Mode 1, 0 (PA1MD1, PA1MD0)

Bit 1,0—PA0 Mode 1, 0 (PA0MD1, PA0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PAnMD1	**PAnMD0**	**Pin Function**	
0	0	Other function	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

18.3.2 Port B Control Register (PBCR)

Port B Control Register (PBCR) is a 16-bit read/write register that selects the pin functions. PBCR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1	PB4 MD0	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	PB0 MD1	PB0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15,14—PB7 Mode 1, 0 (PB7MD1, PB7MD0)

Bit 13,12—PB6 Mode 1, 0 (PB6MD1, PB6MD0)

Bit 11,10—PB5 Mode 1, 0 (PB5MD1, PB5MD0)

Bit 9,8—PB4 Mode 1, 0 (PB4MD1, PB4MD0)

Bit 7,6—PB3 Mode 1, 0 (PB3MD1, PB3MD0)

Bit 5,4—PB2 Mode 1, 0 (PB2MD1, PB2MD0)

Bit 3,2—PB1 Mode 1, 0 (PB1MD1, PB1MD0)

Bit 1,0—PB0 Mode 1, 0 (PB0MD1, PB0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PBnMD1	**PBnMD0**	**Pin Function**	
0	0	Reserved	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

18.3.3 Port C Control Register (PCCR)

Port C Control Register (PCCR) is a 16-bit read/write register that selects the pin functions. PCCR is initialized to H'AAAA by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	PC5 MD1	PC5 MD0	PC4 MD1	PC4 MD0	PC3 MD1	PC3 MD0	PC2 MD1	PC2 MD0	PC1 MD1	PC1 MD0	PC0 MD1	PC0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	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 15,14—PC7 Mode 1, 0 (PC7MD1, PC7MD0)

Bit 13,12—PB6 Mode 1, 0 (PC6MD1, PC6MD0)

Bit 11,10—PC5 Mode 1, 0 (PC5MD1, PC5MD0)

Bit 9,8—PC4 Mode 1, 0 (PC4MD1, PC4MD0)

Bit 7,6—PC3 Mode 1, 0 (PC3MD1, PC3MD0)

Bit 5,4—PC2 Mode 1, 0 (PC2MD1, PC2MD0)

Bit 3,2—PC1 Mode 1, 0 (PC1MD1, PC1MD0)

Bit 1,0—PC0 Mode 1, 0 (PC0MD1, PC0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PCnMD1	**PCnMD0**	**Pin Function**	
0	0	Reserved	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

18.3.4 Port D Control Register (PDCR)

Port D Control Register (PDCR) is a 16-bit read/write register that selects the pin functions. PDCR is initialized to H'AA8A by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PD7 MD1	PD7 MD0	PD6 MD1	PD6 MD0	PD5 MD1	PD5 MD0	PD4 MD1	PD4 MD0	PD3 MD1	PD3 MD0	PD2 MD1	PD2 MD0	PD1 MD1	PD1 MD0	PD0 MD1	PD0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	0	0	1	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 15,14—PD7 Mode 1, 0 (PD7MD1, PD7MD0)

Bit 11,10—PD5 Mode 1, 0 (PD5MD1, PD5MD0)

Bit 7,6—PD3 Mode 1, 0 (PD3MD1, PD3MD0)

Bit 5,4—PD2 Mode 1, 0 (PD2MD1, PD2MD0)

Bit 3,2—PD1 Mode 1, 0 (PD1MD1, PD1MD0)

Bit 1,0—PD0 Mode 1, 0 (PD0MD1, PD0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–3,5,7)

Bit (2n+1)	Bit 2n		
PDnMD1	**PDnMD0**	**Pin Function**	
0	0	Other function	(initial valne, n=2)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value, n=0,1,3,5,7)
1	1	Port input (Pullup MOS: off)	

Bit 13,12—PD6 Mode 1, 0 (PD6MD1, PD6MD0)

Bit 9,8—PD4 Mode 1, 0 (PD4MD1, PD4MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 4,6)

Bit (2n+1)	Bit 2n		
PDnMD1	**PDnMD0**	**Pin Function**	
0	0	Other function	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

18.3.5 Port E Control Register (PECR)

Port E Control Register (PECR) is a 16-bit read/write register that selects the pin functions. PECR is initialized to H'AAAA by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PE7 MD1	PE7 MD0	PE6 MD1	PE6 MD0	PE5 MD1	PE5 MD0	PE4 MD1	PE4 MD0	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1	PE0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	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 15,14—PE7 Mode 1, 0 (PE7MD1, PE7MD0)

Bit 13,12—PE6 Mode 1, 0 (PE6MD1, PE6MD0)

Bit 11,10—PE5 Mode 1, 0 (PE5MD1, PE5MD0)

Bit 9,8—PE4 Mode 1, 0 (PE4MD1, PE4MD0)

Bit 7,6—PE3 Mode 1, 0 (PE3MD1, PE3MD0)

Bit 5,4—PE2 Mode 1, 0 (PE2MD1, PE2MD0)

Bit 3,2—PE1 Mode 1, 0 (PE1MD1, PE1MD0)

Bit 1,0—PE0 Mode 1, 0 (PE0MD1, PE0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PEnMD1	**PEnMD0**	**Pin Function**	
0	0	Other function	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

18.3.6 Port F Control Register (PFCR)

Port F Control Register (PFCR) is a 16-bit read/write register that selects the pin functions. PFCR is initialized to H'AAAA by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	PF4 MD1	PF4 MD0	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	PF0 MD1	PF0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	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 15,14—PF7 Mode 1, 0 (PF7MD1, PF7MD0)

Bit 13,12—PF6 Mode 1, 0 (PF6MD1, PF6MD0)

Bit 11,10—PF5 Mode 1, 0 (PF5MD1, PF5MD0)

Bit 9,8—PF4 Mode 1, 0 (PF4MD1, PF4MD0)

Bit 7,6—PF3 Mode 1, 0 (PF3MD1, PF3MD0)

Bit 5,4—PF2 Mode 1, 0 (PF2MD1, PF2MD0)

Bit 3,2—PF1 Mode 1, 0 (PF1MD1, PF1MD0)

Bit 1,0—PF0 Mode 1, 0 (PF0MD1, PF0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**	
PFnMD1	**PFnMD0**	**Pin Function**
0	0	Reserved
0	1	Reserved
1	0	Port input (Pullup MOS: on) (Initial value)
1	1	Port input (Pullup MOS: off)

18.3.7 Port G Control Register (PGCR)

Port G Control Register (PGCR) is a 16-bit read/write register that selects the pin functions. PGCR is initialized to H'AAAA by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PG7 MD1	PG7 MD0	PG6 MD1	PG6 MD0	PG5 MD1	PG5 MD0	PG4 MD1	PG4 MD0	PG3 MD1	PG3 MD0	PG2 MD1	PG2 MD0	PG1 MD1	PG1 MD0	PG0 MD1	PG0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	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 15,14—PG7 Mode 1, 0 (PG7MD1, PG7MD0)

Bit 13,12—PG6 Mode 1, 0 (PG6MD1, PG6MD0)

Bit 11,10—PG5 Mode 1, 0 (PG5MD1, PG5MD0)

Bit 9,8—PG4 Mode 1, 0 (PG4MD1, PG4MD0)

Bit 7,6—PG3 Mode 1, 0 (PG3MD1, PG3MD0)

Bit 5,4—PG2 Mode 1, 0 (PG2MD1, PG2MD0)

Bit 3,2—PG1 Mode 1, 0 (PG1MD1, PG1MD0)

Bit 1,0—PG0 Mode 1, 0 (PG0MD1, PG0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PGnMD1	**PGnMD0**	**Pin Function**	
0	0	Other function	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

18.3.8 Port H Control Register (PHCR)

Port H Control Register (PHCR) is a 16-bit read/write register that selects the pin functions. PHCR is initialized to H'AAAA by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PH7 MD1	PH7 MD0	PH6 MD1	PH6 MD0	PH5 MD1	PH5 MD0	PH4 MD1	PH4 MD0	PH3 MD1	PH3 MD0	PH2 MD1	PH2 MD0	PH1 MD1	PH1 MD0	PH0 MD1	PH0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	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 15,14—PH7 Mode 1, 0 (PH7MD1, PH7MD0): These bits select the pin functions and the input pullup MOS control.

Bit 15	Bit 14		
PH7MD1	**PH7MD0**	**Pin Function**	
0	0	Other function	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 13,12—PH6 Mode 1, 0 (PH6MD1, PH6MD0)

Bit 11,10—PH5 Mode 1, 0 (PH5MD1, PH5MD0)

Bit 9,8—PH4 Mode 1, 0 (PH4MD1, PH4MD0)

Bit 7,6—PH3 Mode 1, 0 (PH3MD1, PH3MD0)

Bit 5,4—PH2 Mode 1, 0 (PH2MD1, PH2MD0)

Bit 3,2—PH1 Mode 1, 0 (PH1MD1, PH1MD0)

Bit 1,0—PH0 Mode 1, 0 (PH0MD1, PH0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–6)

Bit (2n+1)	Bit 2n		
PHnMD1	**PHnMD0**	**Pin Function**	
0	0	Other function	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

18.3.9 Port J Control Register (PJCR)

Port J Control Register (PJCR) is a 16-bit read/write register that selects the pin functions. PJCR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PJ7 MD1	PJ7 MD0	PJ6 MD1	PJ6 MD0	PJ5 MD1	PJ5 MD0	PJ4 MD1	PJ4 MD0	PJ3 MD1	PJ3 MD0	PJ2 MD1	PJ2 MD0	PJ1 MD1	PJ1 MD0	PJ0 MD1	PJ0 MD0
Initial value:	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—PJ7 Mode 1, 0 (PJ7MD1, PJ7MD0)

Bit 13,12—PJ6 Mode 1, 0 (PJ6MD1, PJ6MD0)

Bit 11,10—PJ5 Mode 1, 0 (PJ5MD1, PJ5MD0)

Bit 9,8—PJ4 Mode 1, 0 (PJ4MD1, PJ4MD0)

Bit 7,6—PJ3 Mode 1, 0 (PJ3MD1, PJ3MD0)

Bit 5,4—PJ2 Mode 1, 0 (PJ2MD1, PJ2MD0)

Bit 3,2—PJ1 Mode 1, 0 (PJ1MD1, PJ1MD0)

Bit 1,0—PJ0 Mode 1, 0 (PJ0MD1, PJ0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**		
PJnMD1	**PJnMD0**	**Pin Function**	
0	0	Other function	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

18.3.10 Port K Control Register (PKCR)

Port K Control Register (PKCR) is a 16-bit read/write register that selects the pin functions. PKCR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PK7 MD1	PK7 MD0	PK6 MD1	PK6 MD0	PK5 MD1	PK5 MD0	PK4 MD1	PK4 MD0	PK3 MD1	PK3 MD0	PK2 MD1	PK2 MD0	PK1 MD1	PK1 MD0	PK0 MD1	PK0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15,14—PK7 Mode 1, 0 (PK7MD1, PK7MD0)

Bit 13,12—PK6 Mode 1, 0 (PK6MD1, PK6MD0)

Bit 11,10—PK5 Mode 1, 0 (PK5MD1, PK5MD0)

Bit 9,8—PK4 Mode 1, 0 (PK4MD1, PK4MD0)

Bit 7,6—PK3 Mode 1, 0 (PK3MD1, PK3MD0)

Bit 5,4—PK2 Mode 1, 0 (PK2MD1, PK2MD0)

Bit 3,2—PK1 Mode 1, 0 (PK1MD1, PK1MD0)

Bit 1,0—PK0 Mode 1, 0 (PK0MD1, PK0MD0)

These bits select the pin functions and the input pullup MOS control.

(n = 0–7)

Bit (2n+1)	**Bit 2n**	
PKnMD1	**PKnMD0**	**Pin Function**
0	0	Other function (Initial value)
0	1	Port output
1	0	Port input (Pullup MOS: on)
1	1	Port input (Pullup MOS: off)

18.3.11 Port L Control Register (PLCR)

Port L Control Register (PLCR) is a 16-bit read/write register that selects the pin functions. PLCR is initialized to H'0000 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	PL7 MD1	PL7 MD0	PL6 MD1	PL6 MD0	PL5 MD1	PL5 MD0	PL4 MD1	PL4 MD0	PL3 MD1	PL3 MD0	PL2 MD1	PL2 MD0	PL1 MD1	PL1 MD0	PL0 MD1	PL0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15,14—PL7 Mode 1, 0 (PL7MD1, PL7MD0)

Bit 13,12—PL6 Mode 1, 0 (PL6MD1, PL6MD0)

Bit 11,10—PL5 Mode 1, 0 (PL5MD1, PL5MD0)

Bit 9,8—PL4 Mode 1, 0 (PL4MD1, PL4MD0)

Bit 7,6—PL3 Mode 1, 0 (PL3MD1, PL3MD0)

Bit 5,4—PL2 Mode 1, 0 (PL2MD1, PL2MD0)

Bit 3,2—PL1 Mode 1, 0 (PL1MD1, PL1MD0)

Bit 1,0—PL0 Mode 1, 0 (PL0MD1, PL0MD0)

These bits select the pin functions.

(n = 0–7)

Bit (2n+1)	Bit 2n	
PLnMD1	**PLnMD0**	**Pin Function**
0	0	Other function (Initial value)
0	1	Reserved
1	0	Port input
1	1	Port input

18.3.12 Port SC Control Register (SCPCR)

SP Port Control Register (SCPCR) is a 16-bit read/write register that selects the pin functions. The setting of SCPCR is valid only when transmit/receive operation is disabled in the setting of the SCSCR register. SCPCR is initialized to H'A888 by power-on resets; however, it is not initialized for manual resets, software standby mode, or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:	SCP7 MD1	SCP7 MD0	SCP6 MD1	SCP6 MD0	SCP5 MD1	SCP5 MD0	SCP4 MD1	SCP4 MD0	SCP3 MD1	SCP3 MD0	SCP2 MD1	SCP2 MD0	SCP1 MD1	SCP1 MD0	SCP0 MD1	SCP0 MD0
Initial value:	1	0	1	0	1	0	0	0	1	0	0	0	1	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—SCP7 Mode 1, 0 (SCP7MD1, SCP7MD0): These bits select the pin functions and the input pullup MOS control.

Bit 15	Bit 14		
SCP7MD1	**SCP7MD0**	**Pin Function**	
0	0	Other function	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 13, 12—SCP6 Mode 1, 0 (SCP6MD1, SCP6MD0): These bits select the pin functions and the input pullup MOS control.

Bit 13	Bit 12		
SCP6MD1	**SCP6MD0**	**Pin Function**	
0	0	Other function	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 11, 10—SCP5 Mode 1, 0 (SCP5MD1, SCP5MD0): These bits select the pin functions and the input pullup MOS control.

Bit 11	**Bit 10**		
SCP5MD1	**SCP5MD0**	**Pin Function**	
0	0	Other function	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 9, 8—SCP4 Mode 1, 0 (SCP4MD1, SCP4MD0): These bits select the pin functions and the input pullup MOS control.

Bit 9	**Bit 8**		
SCP4MD1	**SCP4MD0**	**Pin Function**	
0	0	Transmit data output 2 (TxD2) Receive data input 2 (RxD2)	(Initial value)
0	1	General output (SCPT[4] output pin) Receive data input 2 (RxD2)	
1	0	SCPT[4] input pin pullup (input pin) Transmit data output 2 (TxD2)	
1	1	General input (SCPT[4] input pin) Transmit data output 2 (TxD2)	

Note: There is no combination of simultaneous input/output of SCPT[4] because one bit (SCP4DT) is accessed using two pins of TxD2 and RxD2.

TxD2 is high-Z if SCSCR, TE is 0 and SCP4D1/SCP4D0 are not 0/1.

Bit 7, 6—SCP3 Mode 1, 0 (SCP3MD1, SCP3MD0): These bits select the pin functions and the input pullup MOS control.

Bit 7	**Bit 6**		
SCP3MD1	**SCP3MD0**	**Pin Function**	
0	0	Other function	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 5, 4—SCP2 Mode 1, 0 (SCP2MD1, SCP2MD0): These bits select the pin functions and the input pullup MOS control.

Bit 5	Bit 4	
SCP2MD1	**SCP2MD0**	**Pin Function**
0	0	Transmit data output 1 (TxD1) Receive data input 1 (RxD1) (Initial value)
0	1	General output (SCPT[2] output pin) Receive data input 1 (RxD1)
1	0	SCPT[2] input pin pullup (input pin) Transmit data output 1 (TxD1)
1	1	General input (SCPT[2] input pin) Transmit data output 1 (TxD1)

Note: There is no combination of simultaneous input/output of SCPT[2] because one bit (SCP4DT) is accessed using two pins of TxD1 and RxD1.
TxD is high-Z if SCSCR, TE is 0 and SCP2D1/SCP2D0 are not 0/1.

Bit 3, 2—SCP1 Mode 1, 0 (SCP1MD1, SCP1MD0): These bits select the pin functions and the input pullup MOS control.

Bit 3	Bit 2	
SCP1MD1	**SCP1MD0**	**Pin Function**
0	0	Other function
0	1	Port output
1	0	Port input (Pullup MOS: on) (Initial value)
1	1	Port input (Pullup MOS: off)

Bit 1, 0—SCP0 Mode 1, 0 (SCP0MD1, SCP0MD0): These bits select the pin functions and the input pullup MOS control.

Bit 1	Bit 0	
SCP0MD1	**SCP0MD0**	**Pin Function**
0	0	Transmit data output 0 (TxD0) Receive data input 0 (RxD0) (Initial value)
0	1	General output (SCPT[0] output pin) Receive data input 0 (RxD0)
1	0	SCPT[0] input pin pullup (input pin) Transmit data output 0 (TxD0)
1	1	General input (SCPT[0] input pin) Transmit data output 0 (TxD0)

Note: There is no combination of simultaneous input/output of SCPT[0] because one bit (SCP0DT) is accessed using two pins of TxD0 and RxD0.
TxD2 is high-Z if SCSCR,TE is 0 and SCP0MD1/SCP0MD0 are not 0/1.

Section 19 I/O Port

19.1 Overview

This LSI has twelve 8-bit ports (ports A to L and SC). All port pins are multiplexed with other pin functions (Pin Function Controller (PFC) maintains the selection of the pin functions and pullup MOS control). Each port has a data register which stores the data to the pins.

19.2 Port A

Port A is an 8-bit input/output port with the pin configuration shown in figure 19.1. Each pin has a input pullup MOS, which is controlled by Port A Control Register (PACR) in PFC.

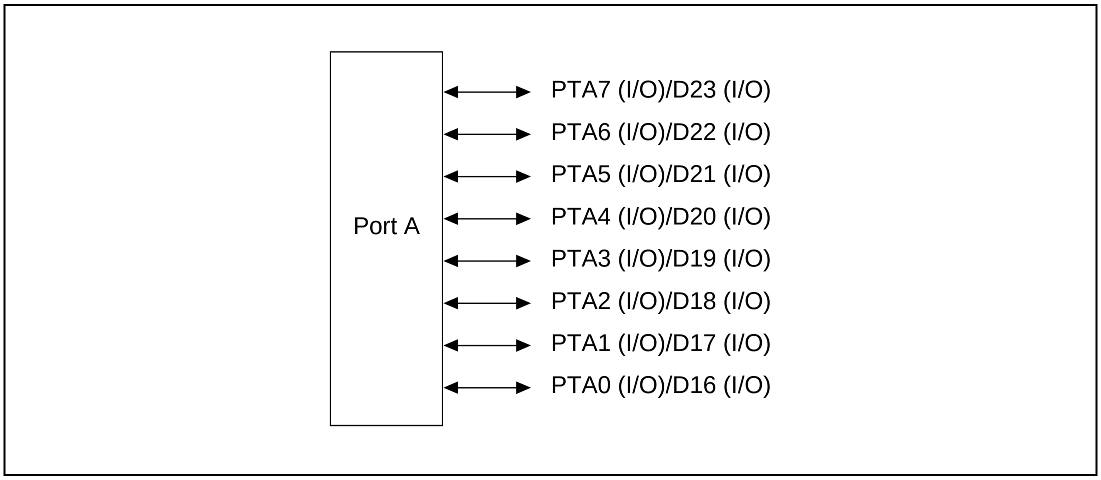


Figure 19.1 Port A

19.2.1 Register Descriptions

Table 19.1 summarizes the registers of port A.

Table 19.1 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'00	H'04000120	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.2.2 Port A Data Register (PADR)

Port A Data Register (PADR) is an 8-bit readable/writable register that stores data for pins PTA7 to PTA0. PA7DT to PA0DT bit corresponds to PTA7 to PTA0 pin. When the pin function is general output port, if the port is read the value of the corresponding PADR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.2 shows the function of PADR.

PADR is initialized to H'00 by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.2 Read/Write Operation of the Port A Data Register (PADR)

PAnMD1	PAnMD0	Pin Status	Read	Write
0	0	Other function	PADR value	Value is written to PADR, but does not affect pin state
	1	Output	PADR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PADR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PADR, but does not affect pin state

(n = 7 to 0)

19.3 Port B

Port B is an 8-bit input/output port with the pin configuration shown in figure 19.2. Each pin has a input pullup MOS, which is controlled by Port B Control Register (PBCR) in PFC.

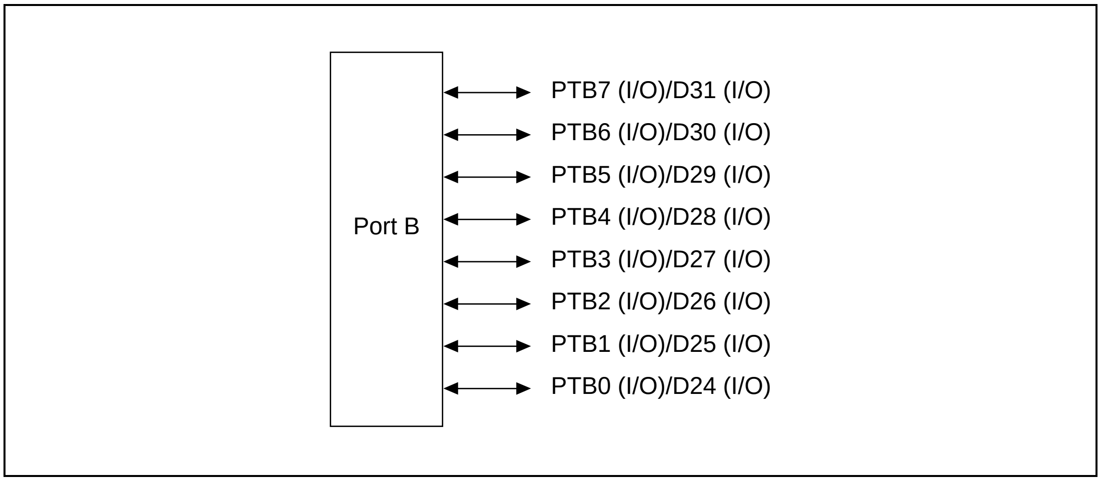


Figure 19.2 Port B

19.3.1 Register Descriptions

Table 19.3 summarizes the registers of port B.

Table 19.3 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'00	H'04000122	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.3.2 Port B Data Register (PBDR)

Port B Data register (PBDR) is an 8-bit readable/writable register that stores data for pins PTB7 to PTB0. PB7DT to PB0DT bit corresponds to PTB7 to PTB0 pin. When the pin function is general output port, if the port is read the value of the corresponding PBDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.4 shows the function of PBDR.

PBDR is initialized to H'00 by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	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

Table 19.4 Read/Write Operation of the Port B Data Register (PBDR)

PBnMD1	PBnMD0	Pin Status	Read	Write
0	0	Other function	PBDR value	Value is written to PBDR, but does not affect pin state
	1	Output	PBDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PBDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PBDR, but does not affect pin state

(n = 7 to 0)

19.4 Port C

Port C is an 8-bit input/output port with the pin configuration shown in figure 19.3. Each pin has a input pullup MOS, which is controlled by Port C Control Register (PCCR) in PFC.

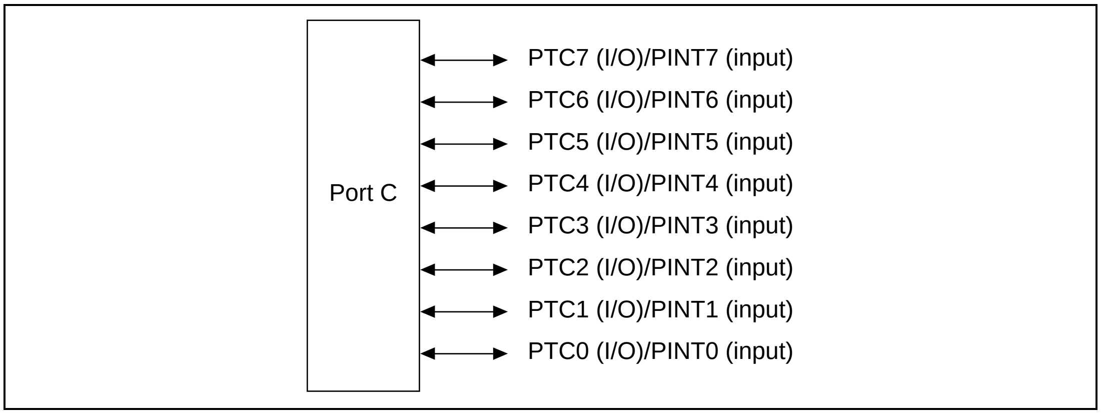


Figure 19.3 Port C

19.4.1 Register Descriptions

Table 19.5 summarizes the registers of port C.

Table 19.5 Register Description

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'FF	H'04000124	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.4.2 Port C Data Register (PCDR)

Port C Data register (PCDR) is an 8-bit readable/writable register that stores data for pins PTC7 to PTC0. PC7DT to PC0DT bit corresponds to PTC7 to PTC0 pin. When the pin function is general output port, if the port is read the value of the corresponding PCDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.6 shows the function of PCDR.

PCDR is initialized to H'FF by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
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

Table 19.6 Read/Write Operation of the Port C Data Register (PCDR)

PCnMD1	PCnMD0	Pin Status	Read	Write
0	0	Reserved	—	—
	1	Output	PCDR value	Value written is output by pin
1	0	Input (Pullup MOS on)	Pin status	Can write to PCDR, but it has no effect on pin status.
	1	Input (Pullup MOS off)	Pin status	Can write to PCDR, but it has no effect on pin status.

(n = 7 to 0)

19.5 Port D

Port D is a 2-bit input/output and 2-bit input port with the pin configuration shown in figure 19.3 Each pin has a input pullup MOS, which is controlled by Port D Control Register (PDCR) in PFC.

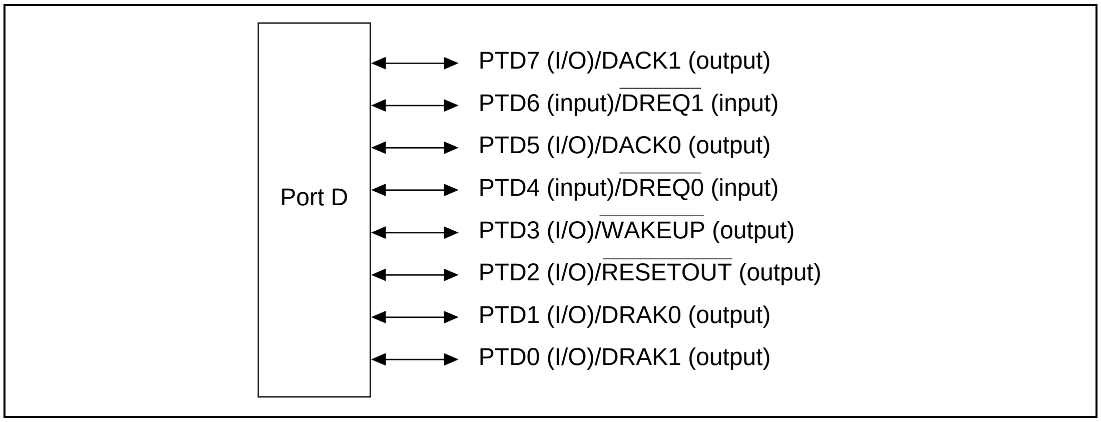

Port D
PTD7 (I/O)/DACK1 (output)
PTD6 (input)/$\overline{\text{DREQ1}}$ (input)
PTD5 (I/O)/DACK0 (output)
PTD4 (input)/$\overline{\text{DREQ0}}$ (input)
PTD3 (I/O)/$\overline{\text{WAKEUP}}$ (output)
PTD2 (I/O)/$\overline{\text{RESETOUT}}$ (output)
PTD1 (I/O)/DRAK0 (output)
PTD0 (I/O)/DRAK1 (output)

Figure 19.4   Port D

19.5.1 Register Descriptions

Table 19.7 summarizes the registers of port D.

Table 19.7   Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port D data register	PDDR	R/W or R	H'FB	H'04000126	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.5.2 Port D Data Register (PDDR)

Port D Data register (PDDR) is a 6-bit readable/writable and 2-bit readable register that stores data for pins PTD7 to PTD0. PD7DT to PD0DT bit corresponds to PTD7 to PTD0 pin. When the pin function is general output port, if the port is read the value of the corresponding PDDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.8 shows the function of PDDR.

PDDR is initialized to H'00 by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Note that the low level is read if bits 6 and 4 are read except in general-purpose input.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	1	1	1	1	1	0	1	1
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 19.8 Read/Write Operation of the Port D Data Register (PDDR)

PDnMD1	PDnMD0	Pin Status	Read	Write
0	0	Other function	PDDR value	Value is written to PDDR, but does not affect pin state
	1	Output	PDDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PDDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PDDR, but does not affect pin state

(n = 7, 5, 3, 2, 1, 0)

PDnMD1	PDnMD0	Pin Status	Read	Write
0	0	Other function	Low level	Ignored (no effect on pin status)
	1	Reserved	Low level	Ignored (no effect on pin status)
1	0	Input (Pullup MOS on)	Pin state	Ignored (no effect on pin status)
	1	Input (Pullup MOS off)	Pin state	Ignored (no effect on pin status)

(n = 6, 4)

Bits 3 to 0—Reserved: These bits are always read as 0, and should only be written with 0.

19.6 Port E

Port E is an 8-bit input/output port with the pin configuration shown in figure 19.5. Each pin has a input pullup MOS, which is controlled by Port E Control Register (PECR) in PFC.

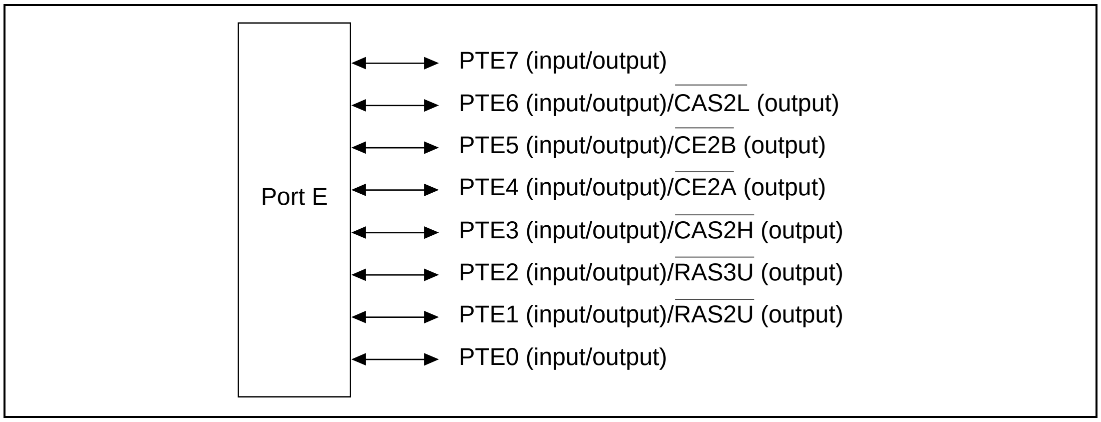

Port E
PTE7 (input/output)
PTE6 (input/output)/$\overline{\text{CAS2L}}$ (output)
PTE5 (input/output)/$\overline{\text{CE2B}}$ (output)
PTE4 (input/output)/$\overline{\text{CE2A}}$ (output)
PTE3 (input/output)/$\overline{\text{CAS2H}}$ (output)
PTE2 (input/output)/$\overline{\text{RAS3U}}$ (output)
PTE1 (input/output)/$\overline{\text{RAS2U}}$ (output)
PTE0 (input/output)

Figure 19.5 Port E

19.6.1 Register Descriptions

Table 19.9 summarizes the registers of port E.

Table 19.9 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'FF	H'04000128	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.6.2 Port E Data Register (PEDR)

Port E Data register (PEDR) is an 8-bit readable/writable register that stores data for pins PTE7 to PTE0. PE7DT to PE0DT bit corresponds to PTE7 to PTE0 pin. When the pin function is general output port, if the port is read the value of the corresponding PEDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.10 shows the function of PEDR.

PEDR is initialized to H'00 by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
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

Table 19.10 Read/Write Operation of the Port E Data Register (PEDR)

PEnMD1	PEnMD0	Pin Status	Read	Write
0	0	Other function	PEDR value	Can write to PEDR, but it has no effect on pin status.
	1	Output	PEDR value	Value written is output by pin
1	0	Input (Pullup MOS on)	Pin status	Can write to PEDR, but it has no effect on pin status.
	1	Input (Pullup MOS off)	Pin status	Can write to PEDR, but it has no effect on pin status.

(n = 7 to 0)

19.7 Port F

Port F is an 8-bit input port with the pin configuration shown in figure 19.6. Each pin has a input pullup MOS, which is controlled by Port F Control Register (PFCR) in PFC.

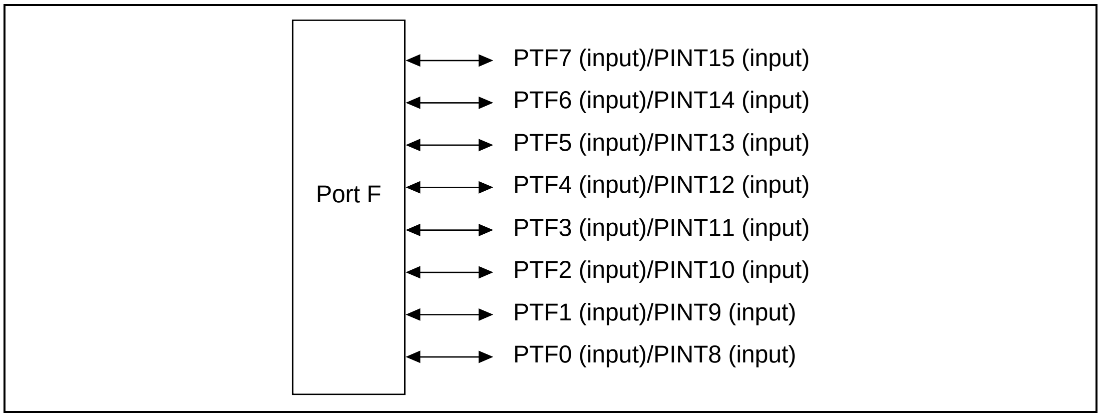

Port F
PTF7 (input)/PINT15 (input)
PTF6 (input)/PINT14 (input)
PTF5 (input)/PINT13 (input)
PTF4 (input)/PINT12 (input)
PTF3 (input)/PINT11 (input)
PTF2 (input)/PINT10 (input)
PTF1 (input)/PINT9 (input)
PTF0 (input)/PINT8 (input)

Figure 19.6 Port F

19.7.1 Register Descriptions

Table 19.11 summarizes the registers of port F.

Table 19.11 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R	H'FF	H'0400012A	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.7.2 Port F Data Register (PFDR)

Port F Data register (PFDR) is an 8-bit readable register that stores data for pins PTF7 to PTF0. PF7DT to PF0DT bit corresponds to PTF7 to PTF0 pin. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.12 shows the function of PFDR.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	1	1	1	1	0	1	2	1
R/W:	R	R	R	R	R	R	R	R

Table 19.12 Read/Write Operation of the Port F Data Register (PFDR)

PFnMD1	PFnMD0	Pin Status	Read	Write
0	0	Reserved	H’00	Ignored (no effect on pin status)
	1	Reserved	H’00	Ignored (no effect on pin status)
1	0	Input (Pullup MOS on)	Pin status	Ignored (no effect on pin status)
	1	Input (Pullup MOS off)	Pin status	Ignored (no effect on pin status)

(n = 7 to 0)

19.8 Port G

Port G is an 8-bit input port with the pin configuration shown in figure 19.7. Each pin has a input pullup MOS, which is controlled by Port G Control Register (PGCR) in PFC.

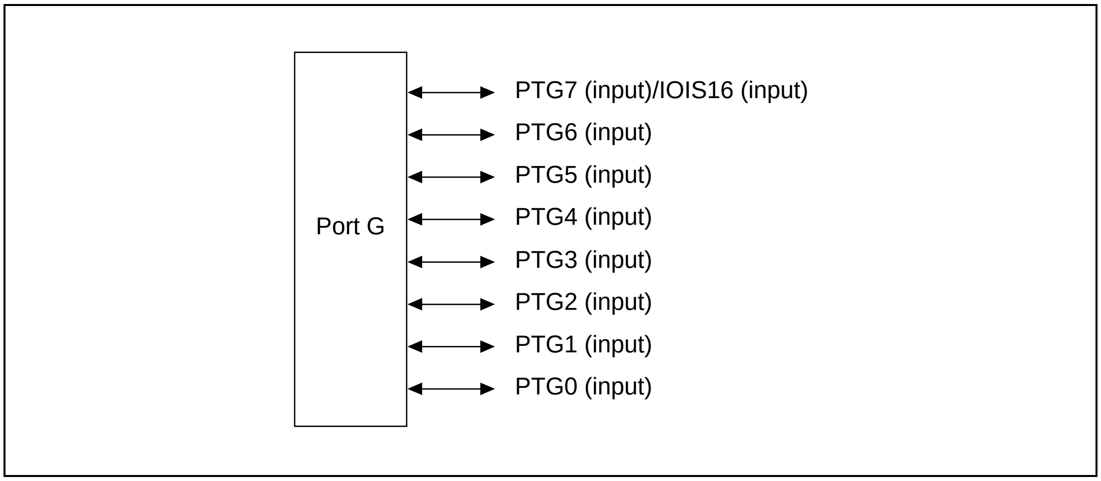


Figure 19.7 Port G

19.8.1 Register Descriptions

Table 19.13 summarizes the registers of port G.

Table 19.13 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port G data register	PGDR	R	H'FF	H'0400012C	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.8.2 Port G Data Register (PGDR)

Port G Data register (PGDR) is an 8-bit readable register that stores data for pins PTG7 to PTG0. PG7DT to PG0DT bit corresponds to PTG7 to PTG0 pin. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.14 shows the function of PGDR.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R

Table 19.14 Read/Write Operation of the Port G Data Register (PGDR)

PGnMD1	PGnMD0	Pin Status	Read	Write
0	0	Other function	H'00	Ignored (no effect on pin status)
	1	Reserved	H'00	Ignored (no effect on pin status)
1	0	Input (Pullup MOS on)	Pin status	Ignored (no effect on pin status)
	1	Input (Pullup MOS off)	Pin status	Ignored (no effect on pin status)

(n = 7 to 0)

19.9 Port H

Port H is an 1-bit input/output and 5-bit input port with the pin configuration shown in figure 19.8. Each pin has a input pullup MOS, which is controlled by Port H Control Register (PHCR) in PFC.

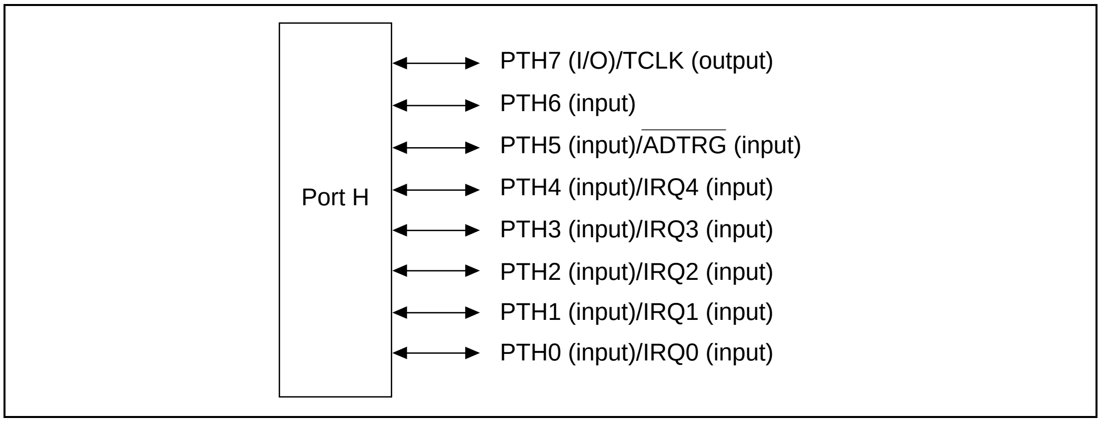

Port H
PTH7 (I/O)/TCLK (output)
PTH6 (input)
PTH5 (input)/$\overline{\text{ADTRG}}$ (input)
PTH4 (input)/IRQ4 (input)
PTH3 (input)/IRQ3 (input)
PTH2 (input)/IRQ2 (input)
PTH1 (input)/IRQ1 (input)
PTH0 (input)/IRQ0 (input)

Figure 19.8 Port H

19.9.1 Register Descriptions

Table 19.15 summarizes the registers of port H.

Table 19.15 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R/W or R	H'FF	H'0400012E	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.9.2 Port H Data Register (PHDR)

Port H Data register (PHDR) is an 1-bit readable/writable and 7-bit readable register that stores data for pins PTH7 to PTH0. PH7DT to PH0DT bit corresponds to PTH7 to PTH0 pin. When the pin function is general output port, if the port is read the value of the corresponding PCDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.16 shows the function of PHDR.

PHDR is initialized to H'00 by Power-on reset. It retains its previous value in standby mode, sleep mode and by manual reset.

Note that the low level is read if bits 6 to 0 are read except in general-purpose input.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R	R	R	R	R

Table 19.16 Read/Write Operation of the Port H Data Register (PHDR)

PHnMD1	PHnMD0	Pin Status	Read	Write
0	0	Other function	PHDR value	Value is written to PHDR, but does not affect pin state
	1	Output	PHDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PHDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PHDR, but does not affect pin state

(n = 7)

PHnMD1	PHnMD0	Pin Status	Read	Write
0	0	Other function	Low level	Ignored (no effect on pin status)
	1	Reserved	Low level	Ignored (no effect on pin status)
1	0	Input (Pullup MOS on)	Pin state	Ignored (no effect on pin status)
	1	Input (Pullup MOS off)	Pin state	Ignored (no effect on pin status)

(n = 6 to 0)

19.10 Port J

Port J is an 8-bit input/output port with the pin configuration shown in figure 19.9. Each pin has a input pullup MOS, which is controlled by Port J Control Register (PJCR) in PFC.

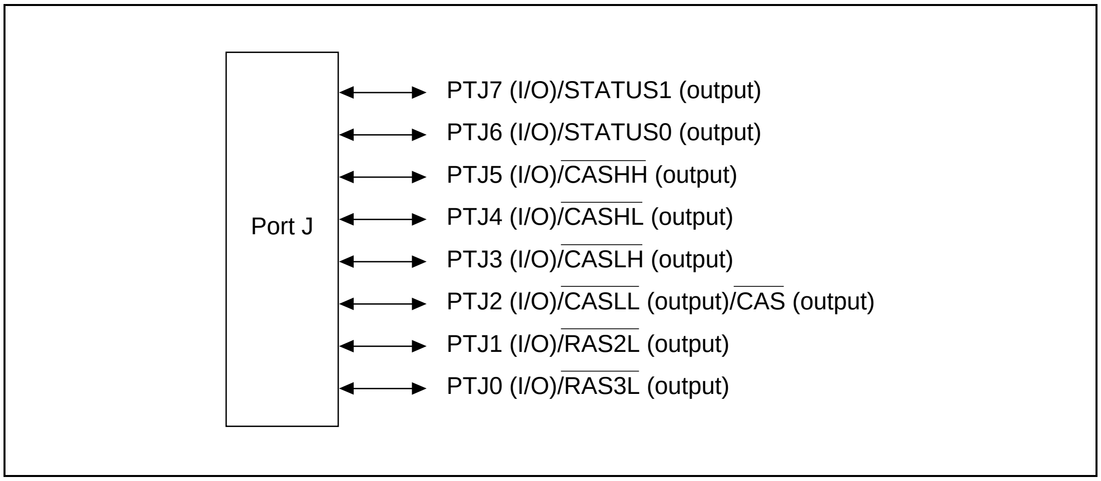

Port J
PTJ7 (I/O)/STATUS1 (output)
PTJ6 (I/O)/STATUS0 (output)
PTJ5 (I/O)/$\overline{\text{CASHH}}$ (output)
PTJ4 (I/O)/$\overline{\text{CASHL}}$ (output)
PTJ3 (I/O)/$\overline{\text{CASLH}}$ (output)
PTJ2 (I/O)/$\overline{\text{CASLL}}$ (output)/$\overline{\text{CAS}}$ (output)
PTJ1 (I/O)/$\overline{\text{RAS2L}}$ (output)
PTJ0 (I/O)/$\overline{\text{RAS3L}}$ (output)

Figure 19.9 Port J

19.10.1 Register Descriptions

Table 19.17 summarizes the registers of port J.

Table 19.17 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port J data register	PJDR	R/W	H’0000	H’04000130	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.10.2 Port J Data Register (PJDR)

Port J Data register (PJDR) is an 8-bit readable/writable register that stores data for pins PTJ7 to PTJ0. PJ7DT to PJ0DT bit corresponds to PTJ7 to PTJ0 pin. When the pin function is general output port, if the port is read the value of the corresponding PJDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.18 shows the function of PJDR.

PJDR is initialized to H'00 by Power-on reset. It retains its previous value in software standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT
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

Table 19.18 Read/Write Operation of the Port J Data Register (PJDR)

PJnMD1	PJnMD0	Pin Status	Read	Write
0	0	Other function	PJDR value	Value is written to PJDR, but does not affect pin state
	1	Output	PJDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PJDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PJDR, but does not affect pin state

(n = 7 to 0)

19.11 Port K

Port K is an 8-bit input/output port with the pin configuration shown in figure 19.10. Each pin has a input pullup MOS, which is controlled by Port K Control Register (PKCR) in PFC.

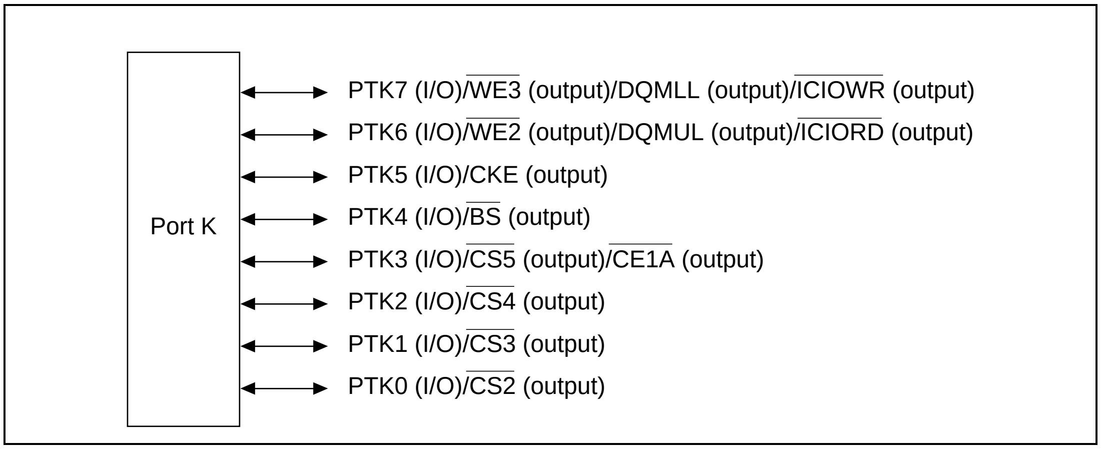

Port K
PTK7 (I/O)/$\overline{WE3}$ (output)/DQMLL (output)/$\overline{ICIOWR}$ (output)
PTK6 (I/O)/$\overline{WE2}$ (output)/DQMUL (output)/$\overline{ICIORD}$ (output)
PTK5 (I/O)/CKE (output)
PTK4 (I/O)/$\overline{BS}$ (output)
PTK3 (I/O)/$\overline{CS5}$ (output)/$\overline{CE1A}$ (output)
PTK2 (I/O)/$\overline{CS4}$ (output)
PTK1 (I/O)/$\overline{CS3}$ (output)
PTK0 (I/O)/$\overline{CS2}$ (output)

Figure 19.10 Port K

19.11.1 Register Descriptions

Table 19.19 summarizes the registers of port K.

Table 19.19 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port K data register	PKDR	R/W	H'0000	H'04000132	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.11.2 Port K Data Register (PKDR)

Port K Data register (PKDR) is an 8-bit readable/writable register that stores data for pins PTK7 to PTK0. PK7DT to PK0DT bit corresponds to PTK7 to PTK0 pin. When the pin function is general output port, if the port is read the value of the corresponding PKDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.20 shows the function of PKDR.

PKDR is initialized to H'00 by Power-on reset or hardware standby. It retains its previous value in software standby mode, sleep mode and by manual reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT
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

Table 19.20 Read/Write Operation of the Port K Data Register (PKDR)

PKnMD1	PKnMD0	Pin Status	Read	Write
0	0	Other function	PKDR value	Value is written to PKDR, but does not affect pin state
	1	Output	PKDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to PKDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to PKDR, but does not affect pin state

(n = 7 to 0)

19.12 Port L

Port L is an 8-bit input port with the pin configuration shown in figure 19.11.

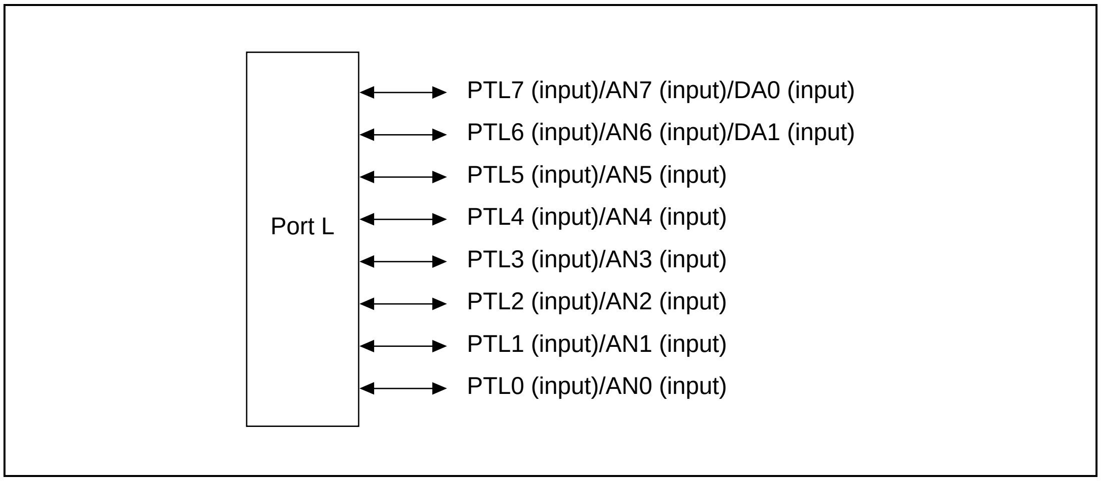


Figure 19.11 Port L

19.12.1 Register Descriptions

Table 19.21 summarizes the registers of port L.

Table 19.21 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
Port L data register	PLDR	R/W	H'0000	H'04000134	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.12.2 Port L Data Register (PLDR)

Port L Data register (PLDR) is an 8-bit readable register that stores data for pins PTL7 to PTL0. PL7DT to PL0DT bit corresponds to PTL7 to PTL0 pin. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.22 shows the function of PLDR.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	PL2DT	PL1DT	PL0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Table 19.22 Read/Write Operation of the Port L Data Register (PLDR)

PLnMD1	PLnMD0	Pin Status	Read	Write
0	0	Other function	H'00	Ignored (no effect on pin status)
	1	Reserves	H'00	Ignored (no effect on pin status)
1	0	Input	Pin state	Ignored (no effect on pin status)
	1	Input	Pin state	Ignored (no effect on pin status)

(n =7 to 0)

19.13 SC Port

SC port is a 4-bit input/output,3-bit output and 4-bit input port with the pin configuration shown in figure 19.12. Each pin has a input pullup MOS, which is controlled by SC port Control Register (SCPCR) in PFC.

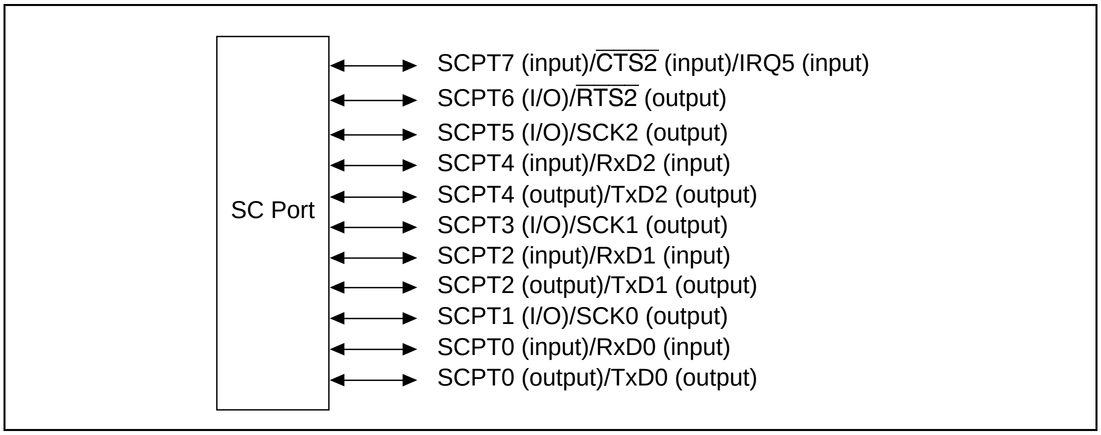

SC Port
SCPT7 (input)/$\overline{\text{CTS2}}$ (input)/IRQ5 (input)
SCPT6 (I/O)/$\overline{\text{RTS2}}$ (output)
SCPT5 (I/O)/SCK2 (output)
SCPT4 (input)/RxD2 (input)
SCPT4 (output)/TxD2 (output)
SCPT3 (I/O)/SCK1 (output)
SCPT2 (input)/RxD1 (input)
SCPT2 (output)/TxD1 (output)
SCPT1 (I/O)/SCK0 (output)
SCPT0 (input)/RxD0 (input)
SCPT0 (output)/TxD0 (output)

Figure 19.12 SC Port

19.13.1 Register Descriptions

Table 19.23 summarizes the registers of SC port.

Table 19.23 Register Descriptions

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
SC Port data register	SCPDR	R/W or R	H'0000	H'4000136	8

Note: This register is located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that this register is not cached.

19.13.2 Port SC Data Register (SCPDR)

Port SC Data register (SCPDR) is a 7-bit readable/writable and 1-bit readable register that stores data for pins SCPT7 to SCPT0. SCP7DT to SCP0DT bit corresponds to SCPT7 to SCPT0 pin. When the pin function is general output port, if the port is read the value of the corresponding SCPDR bit is returned directly. When the function is general input port, if the port is read the corresponding pin level is read. Table 19.24 shows the function of SCPDR.

SCPDR is initialized to H'00 by Power-on reset or hardware standby. It retains its previous value in software standby mode, sleep mode and by manual reset.

Note that the low level is read if bit 7 is read except in general-purpose input.

Bit:	7	6	5	4	3	2	1	0
Bit name:	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT
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

Table 19.24 Read/Write Operation of the SC Port Data Register (SCPDR)

SCPnMD1	SCPnMD0	Pin Status	Read	Write
0	0	Other function	SCPDR value	Value is written to SCPDR, but does not affect pin state
	1	Output	SCPDR value	Write value is output from pin
1	0	Input (Pullup MOS on)	Pin state	Value is written to SCPDR, but does not affect pin state
	1	Input (Pullup MOS off)	Pin state	Value is written to SCPDR, but does not affect pin state

(n = 6 to 0)

SCP7MD1	SCP7MD0	Pin Status	Read	Write
0	0	Other function	Low level	Ignored (no effect on pin status)
	1	Reserved	Low level	Ignored (no effect on pin status)
1	0	Input (Pullup MOS on)	Pin state	Ignored (no effect on pin status)
	1	Input (Pullup MOS off)	Pin state	Ignored (no effect on pin status)

Section 20 A/D Converter

20.1 Overview

This LSI includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

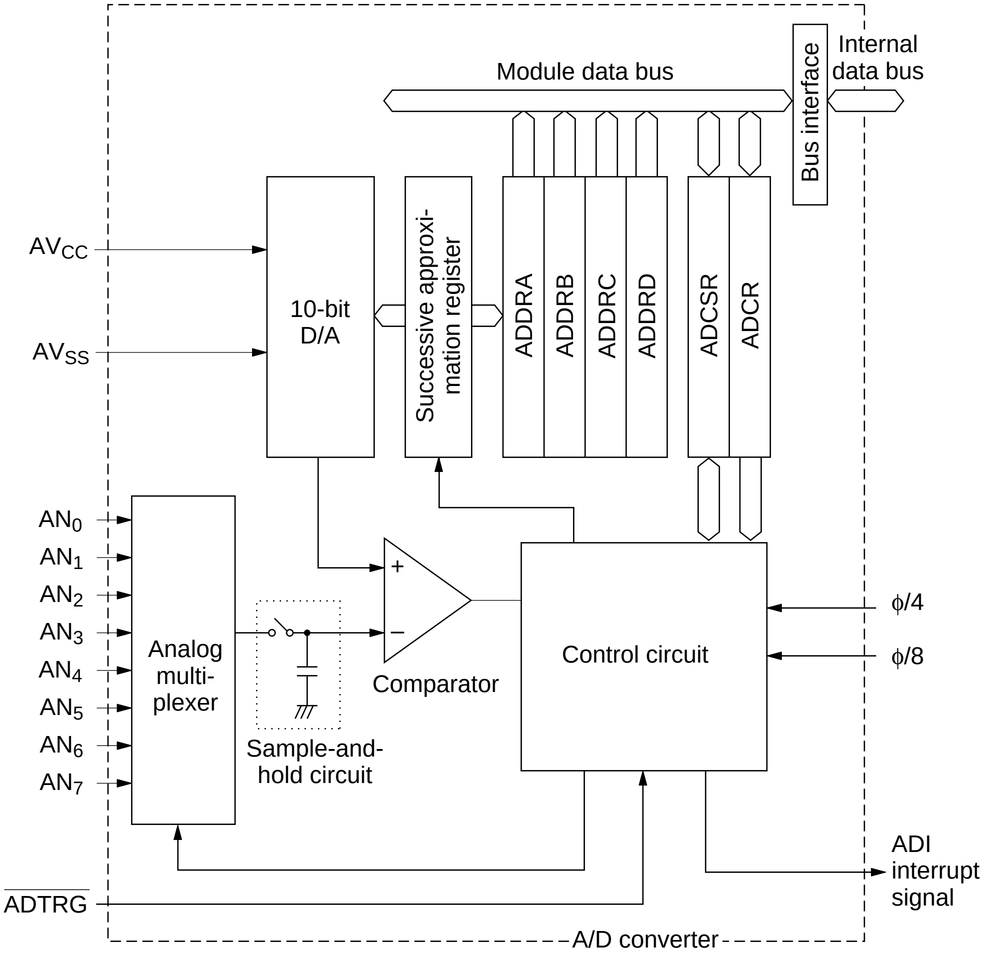
20.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- High-speed conversion
- Conversion time: maximum 8.9µs per channel (with 15 Mhz peripheral clock)
- Two conversion modes
- Single mode: A/D conversion of one channel
- Multi mode: A/D conversion on one to four channels
- Four 16-bit data registers
- A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at end of conversion
- At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

20.1.2 Block Diagram

Figure 20.1 shows a block diagram of the A/D converter.



ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 20.1 A/D Converter Block Diagram

20.1.3 Input Pins

Table 20.1 summarizes the A/D converter’s input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. AVcc also functions as the A/D converter reference voltage.

Table 20.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin6	AN_6	Input	
Analog input pin7	AN_7	Input	
Analog power supply pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

20.1.4 Register Configuration

Table 20.2 summarizes the A/D converter's registers.

Table 20.2 A/D Converter Registers

Register Name	Abbr.	R/W	Initial Value	Address	Access Size
A/D data register A (high)	ADDRAH	R	H'00	H'04000080	16, 8
A/D data register A (low)	ADDRAL	R	H'00	H'04000082	8
A/D data register B (high)	ADDRBH	R	H'00	H'04000084	16, 8
A/D data register B (low)	ADDRBL	R	H'00	H'04000086	8
A/D data register C (high)	ADDRCH	R	H'00	H'04000088	16, 8
A/D data register C (low)	ADDRCL	R	H'00	H'0400008A	8
A/D data register D (high)	ADDRDH	R	H'00	H'0400008C	16, 8
A/D data register D (low)	ADDRDL	R	H'00	H'0400008E	8
A/D control/status register	ADCSR	R/(W) *	H'00	H'4000090	8
A/D control register	ADCR	R/W	H'3F	H'04000092	8

Notes: 1. Only 0 can be written in bit 7, to clear the flag.

2. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

20.2 Register Descriptions

20.2.1 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 of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that always read 0. Table 20.3 indicates the pairings of analog input channels and A/D data registers.

The CPU can always read the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 20.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8
ADDRn:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
ADDRn:	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

n = A–D

Table 20.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD

20.2.2 A/D Control/Status Register (ADCSR)

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ADF	ADIE	ADST	MULTI	CKS	CH2	CH1	CH0
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

Note: Write 0 to clear the flag.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7:ADF	Description	
0	[Clearing condition]	(Initial value)
	(1) Cleared by reading ADF while ADF = 1, then writing 0 in ADF	
	(2) Cleared when DMAC is activated by ADI interrupt and ADDR is read	
1	[Setting conditions]	
	Single mode: A/D conversion ends	
	Multi mode: A/D conversion ends in all selected channels	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6:ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5:ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends.	
	Multi mode: A/D conversion starts; ADST is cleared to 0 when conversion on the selected channel.	

Bit 4—Multi Mode (MULTI): Selects single mode or multi mode. For further information on operation in these modes, see section 22.4, Operation.

Bit 4:SCAN	Description	
0	Single mode	(Initial value)
1	Multi mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

Bit 3:CKS	Description	
0	Conversion time = 266 states (maximum)	(Initial value)
1	Conversion time = 134 states (maximum)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Multi Mode
0	0	0	AN0 (Initial value)	AN0
		1	AN1	AN0, AN1
	1	0	AN2	AN0 to AN2
		1	AN3	AN0 to AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

20.2.3 A/D Control Register (ADCR)

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'3F by a reset and in standby mode.

Bit 7, 6—Trigger Enable (TRGE1, TRGE0): Enables or disables external triggering of A/D conversion.

The TRGE1 and TRGE0 bits should only be set when conversion is not in progress.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TRGE1	TRGE0	—	—	—	—	—	—
Initial value:	0	0	1	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

Bit 7: TRGE1	Bit 6: TRGE0	Description
0	0	When an external trigger is input, the A/D conversion does not start (initial value)
0	1	
1	0	
1	1	The A/D conversion starts at the falling edge of an input signal from the external trigger pin (ADTRG).

20.3 Bus Master Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the bus master by an 8-bit data bus. Therefore, although the upper byte can be accessed directly by the bus master, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the bus master and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the bus master.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 20.2 shows the data flow for access to an A/D data register.

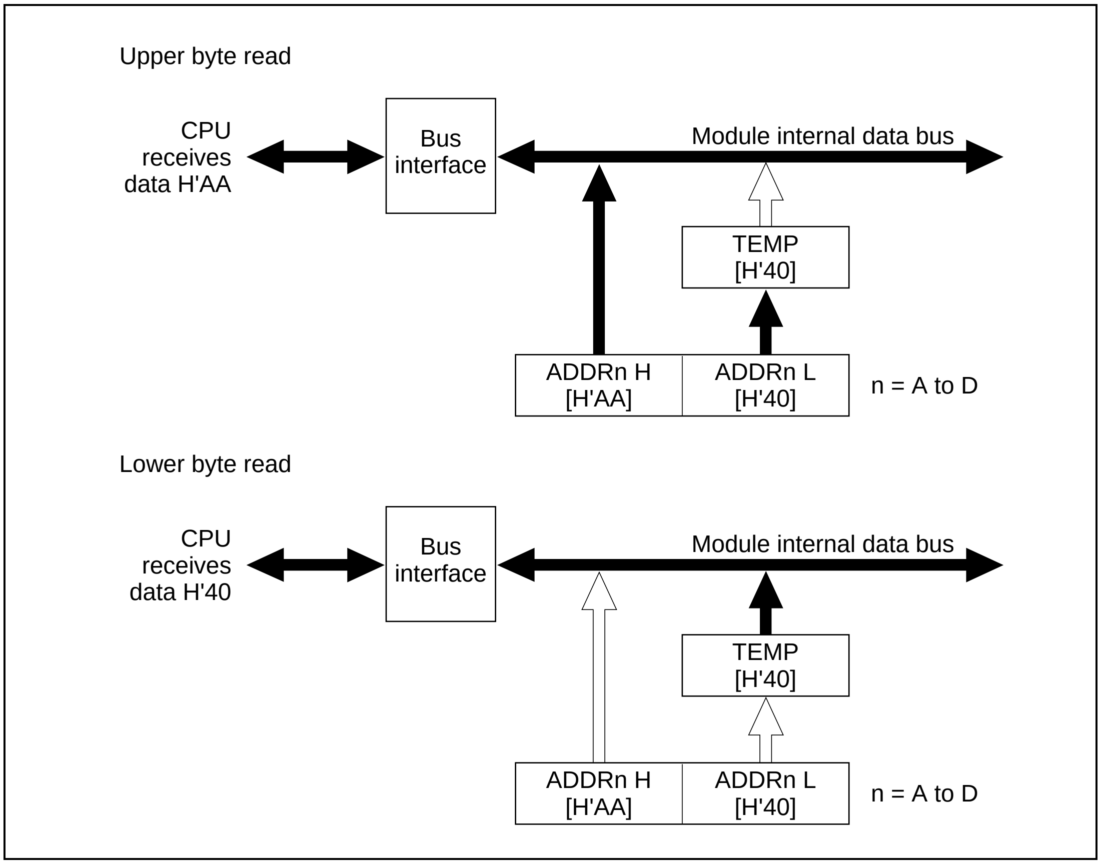


Figure 20.2 A/D Data Register Access Operation (Reading H'AA40)

20.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

20.4.1 Single Mode (MULTI = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

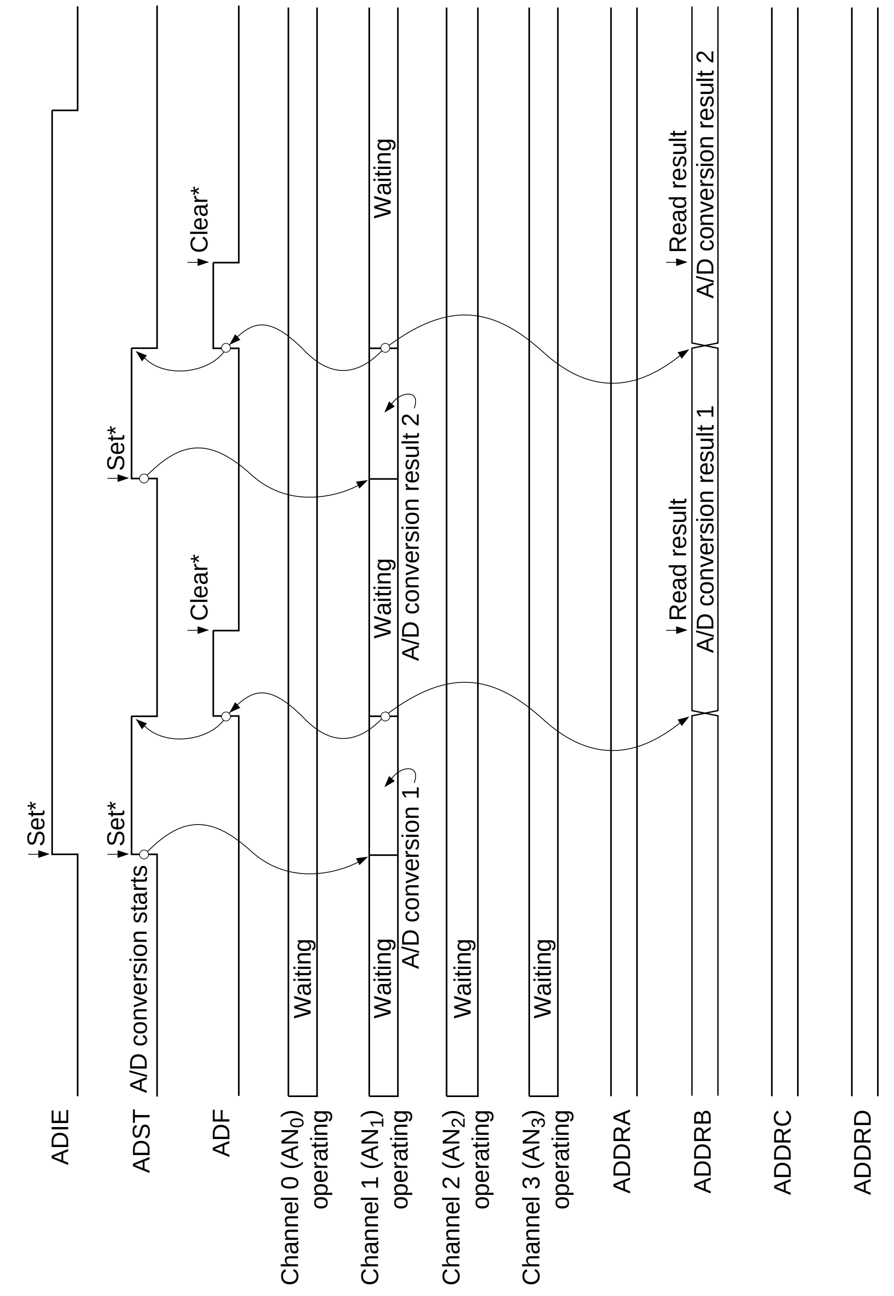
When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 20.3 shows a timing diagram for this example.

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag 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 is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends.



Note: Downward arrows (↓) indicate instruction execution.

Figure 20.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

20.4.2 Multi Mode (MULTI = 1)

In multi mode, conversion is performed once in turn for each analog input in a group of one or more channels. When the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, or AN4 when CH2 = 1).

When more than one channel has been selected, A/D conversion starts on the second channel (AN1 or AN5) as soon as conversion ends on the first channel.

A/D conversion is performed once on each of the selected channels. The conversion results are transferred to and stored in the ADDR register for each channel.

To prevent incorrect operation, A/D conversion should be halted by clearing the ADST bit to 0 before changing the mode or analog input channels. After the change is made, the first channel is selected and A/D conversion is restarted by setting the ADST bit to 1 (the mode or channel change and setting of the ADST bit can be carried out simultaneously).

An example of the A/D conversion operation in multi mode when three channels (AN0—AN2) in group 0 are selected is described below. Figure 20.4 shows the operation timing.

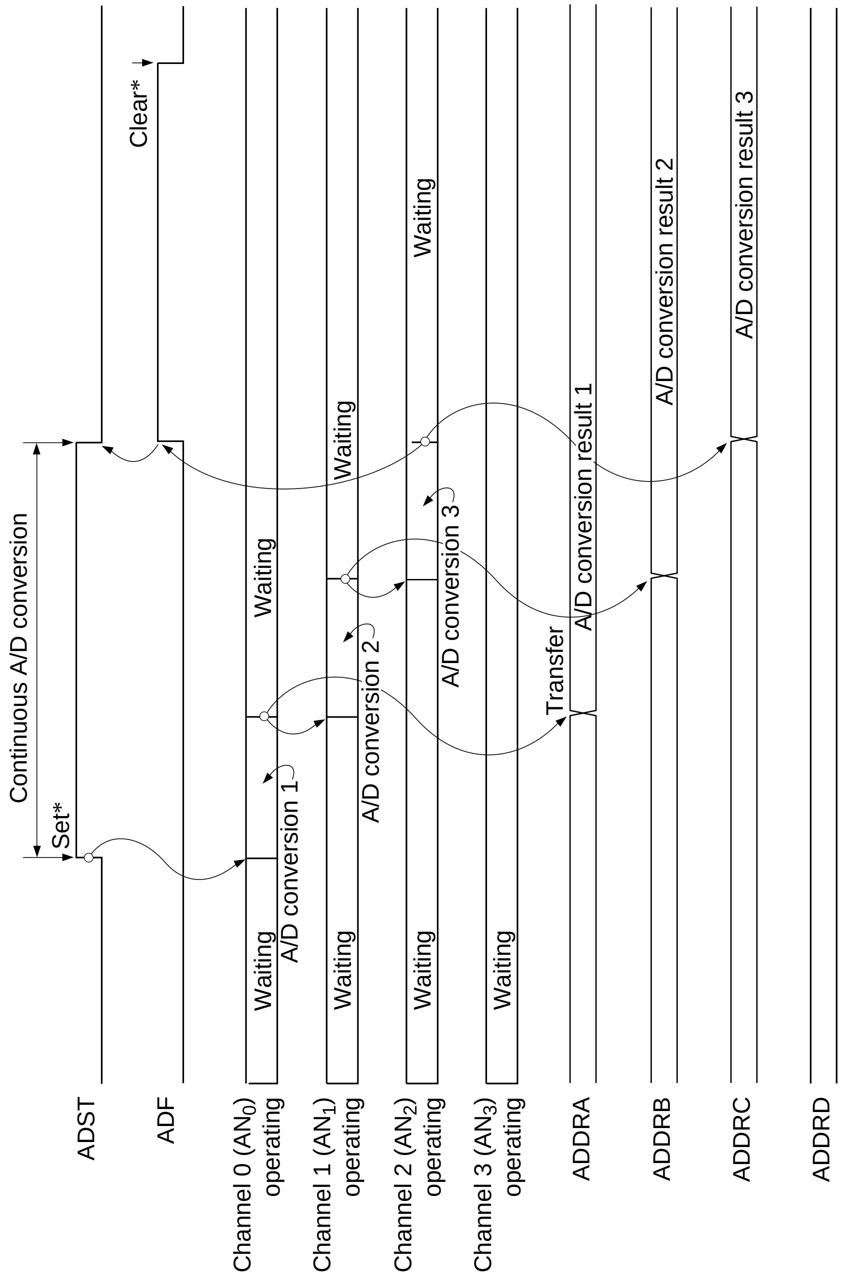
(1) Multi mode is selected (MULTI = 1) as the operating mode, channel group 2 is selected (CH2 = 0), analog input channels AN0—AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).

(2) A/D conversion starts on the first channel (AN0), and when completed, the result is transferred to ADDRA. Conversion then starts automatically on the second channel (AN1).

(3) Conversion proceeds in the same way through the third channel (AN2).

(4) When conversion is completed for all the selected channels (AN0—AN2), ADF is set to 1, and A/D conversion ends.

If the ADIE bit is 1, an ADI interrupt is requested when A/D conversion ends.



Note: * Downward arrow indicates instruction executed by software.

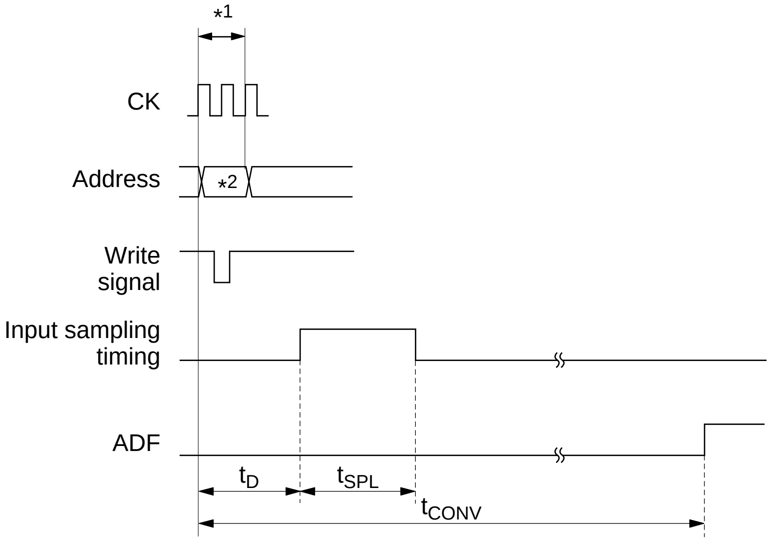
Figure 20.4 Example of A/D Converter Operation (Multi Mode, Channels AN$_0$ to AN$_2$ Selected)

20.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time tD after the ADST bit is set to 1, then starts conversion. Figure 20.5 shows the A/D conversion timing. Table 20.4 indicates the A/D conversion time.

As indicated in figure 20.5, the A/D conversion time includes tD and the input sampling time. The length of tD varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 20.4.

In scan mode, the values given in table 20.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.



t_D A/D start delay

t_{SPL} Input sampling time

t_{CONV} A/D conversion time

Notes: 1. ADSCR write cycle

2. ADSCR address

Figure 20.5 A/D Conversion Timing

Table 20.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	Min	Typ	Max	Min	Typ	Max
Synchronization delay	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	65	—	—	32	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Note: Values in the table are numbers of states.

20.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE1, TRGE0 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 20.6 shows the timing.

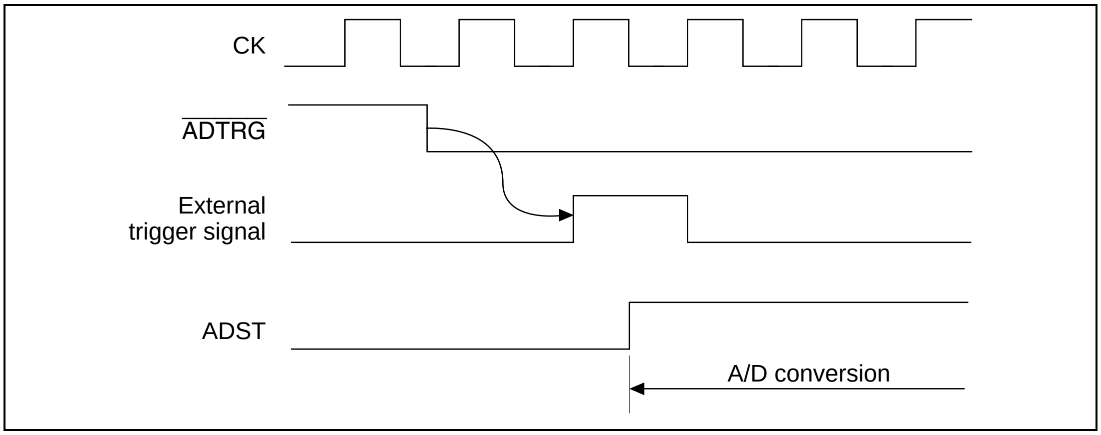


Figure 20.6 External Trigger Input Timing

20.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

20.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel to its analog reference value 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:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below using figure 20.7. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in the figure) to 000000001 (001 in the figure)(figure 20.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure)(figure 20.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 20.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 20.7, item (4)). Note that it does not include offset, full-scale or quantization error.

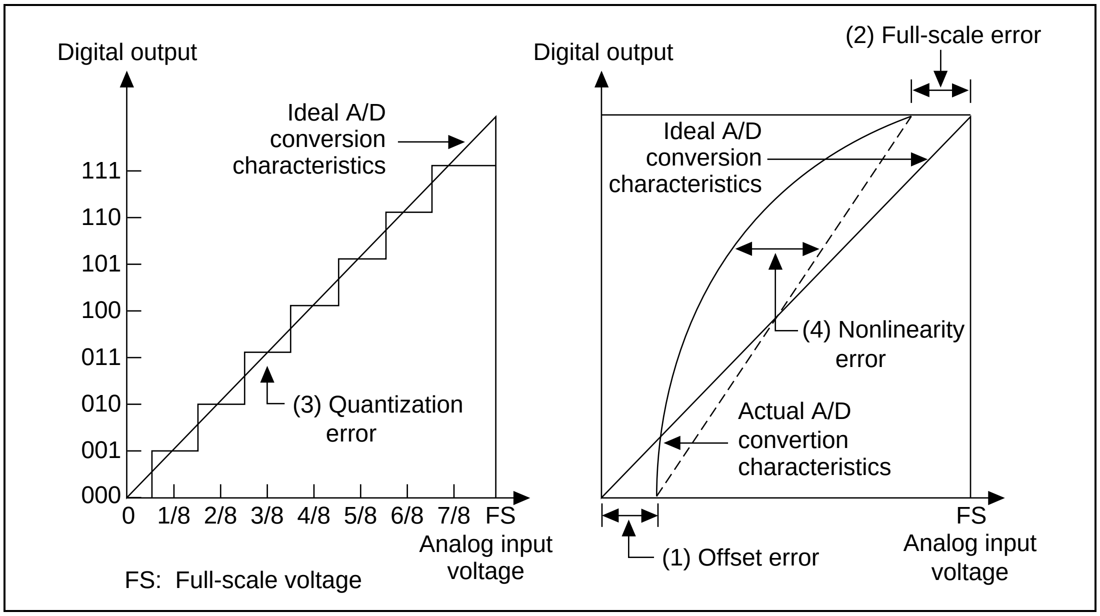


Figure 20.7 Definitions of A/D Conversion Accuracy

20.7 A/D Converter Usage Notes

When using the A/D converter, note the points listed in section 202.7.1 below.

20.7.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins ANn should be in the range AV_{SS} ² ANn ² AV_{CC}.
- Relationships of AV_{CC} and AV_{SS} to V_{CC} and V_{SS}: AV_{CC}, AV_{SS}, V_{CC} and V_{SS} should be related as follows: $AV_{CC} = V_{CC}$ ± 0.3 V and $AV_{SS} = V_{SS}$.

20.7.2 Handling of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0–AN7), connect an input protection circuit like the one shown in figure 20.8. The circuit shown also includes an RC filter to suppress noise. This circuit is shown as an example: The circuit constants should be selected according to actual application conditions. Table 20.5 list the analog input pin specifications and figure 20.9 shows an equivalent circuit diagram of the analog input ports.

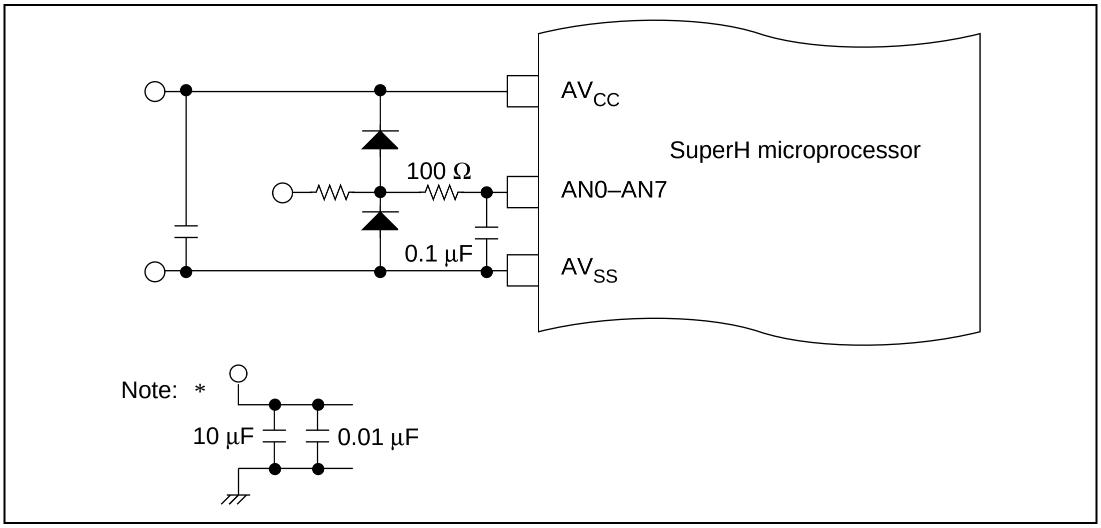


Figure 20.8 Example of Analog Input Protection Circuit

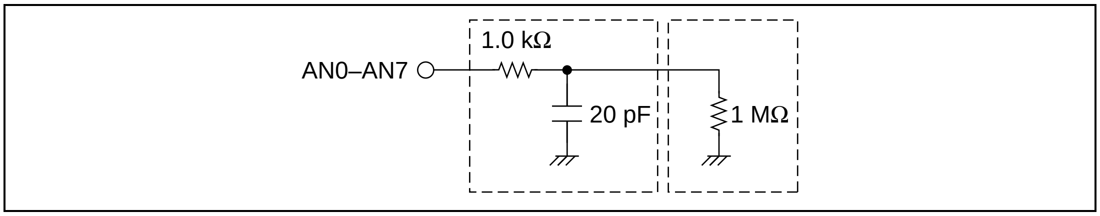


Figure 20.9 Analog Input Pin Equivalent Circuit

Table 20.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

Section 21 D/A Converter

21.1 Overview

This LSI includes a D/A converter with two channels.

21.1.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10µs (with 20-pF capacitive load)
- Output voltage: 0 V toAVcc

21.1.2 Block Diagram

Figure 21.1 shows a block diagram of the D/A converter.

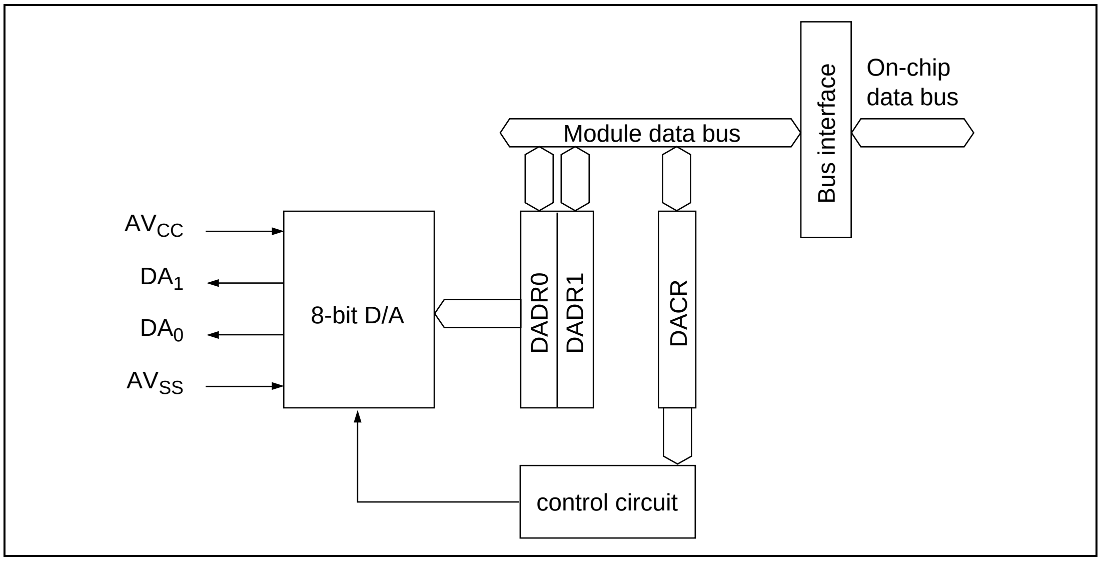


Figure 21.1 D/A Converter Block Diagram

21.1.3 Input/Output Pins

Table 21.1 summarizes the D/A converter’s input and output pins.

Table 21.1 D/A Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog output pin 0	DA_0	Output	Analog output, channel 0
Analog output pin 1	DA_1	Output	Analog output, channel 1

21.1.4 Register Configuration

Table 21.2 summarizes the D/A converter’s registers.

Table 21.2 D/A Converter Registers

Register Name	Abbr.	R/W	Initial Value	Address*
D/A data register 0	DADR0	R/W	H'00	H'040000A0
D/A data register 1	DADR1	R/W	H'00	H'040000A2
D/A control register	DACR	R/W	H'1F	H'040000A4

Notes: 1. Lower 16 bits of the address

2. These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

21.2 Register Descriptions

21.2.1 D/A Data Registers 0 and 1 (DADR0/1)

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers that store the data to be converted. When analog output is enabled, the D/A data register values are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset.

Bit:	7	6	5	4	3	2	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

21.2.2 D/A Control Register (DACR)

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter. DACR is initialized to H'1F by a reset.

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit 7-D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7:DAOE1	Description	
0	DA_1 analog output is disabled	(initial value)
1	Channel-1 D/A conversion and DA_1 analog output are enabled	

Bit 6-D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6:DAOE0	Description	
0	DA_0 analog output is disabled	(initial value)
1	Channel-0 D/A conversion and DA_0 analog output are enabled	

Bit 5-D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, D/A conversion is controlled independently in channels 0 and 1. When this LSI enters standby mode while D/A conversion is enabled, the D/A output is held and the analog power supply current is equivalent to that during D/A conversion. To reduce the analog power supply current in standby mode, clear the DAOE0 and the DAOE1 bits and disable the D/A output.

Bit 7:DAOE1	Bit 6:DAOE0	Bit 5:DAE	Description
0	0	-	D/A conversion is disabled in channels 0 and 1
0	1	0	D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1
0	1	1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1
1	0	1	D/A conversion is enabled in channels 0 and 1
1	1	-	D/A conversion is enabled in channels 0 and 1

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during A/D and D/A conversion.

Bits 4 to 0-Reserved: Read-only bits, always read as 1.

21.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figure 23.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an output pin. The converted result is output after the conversion time. The output value is (DADR0 contents/256) * AV_{CC}. Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

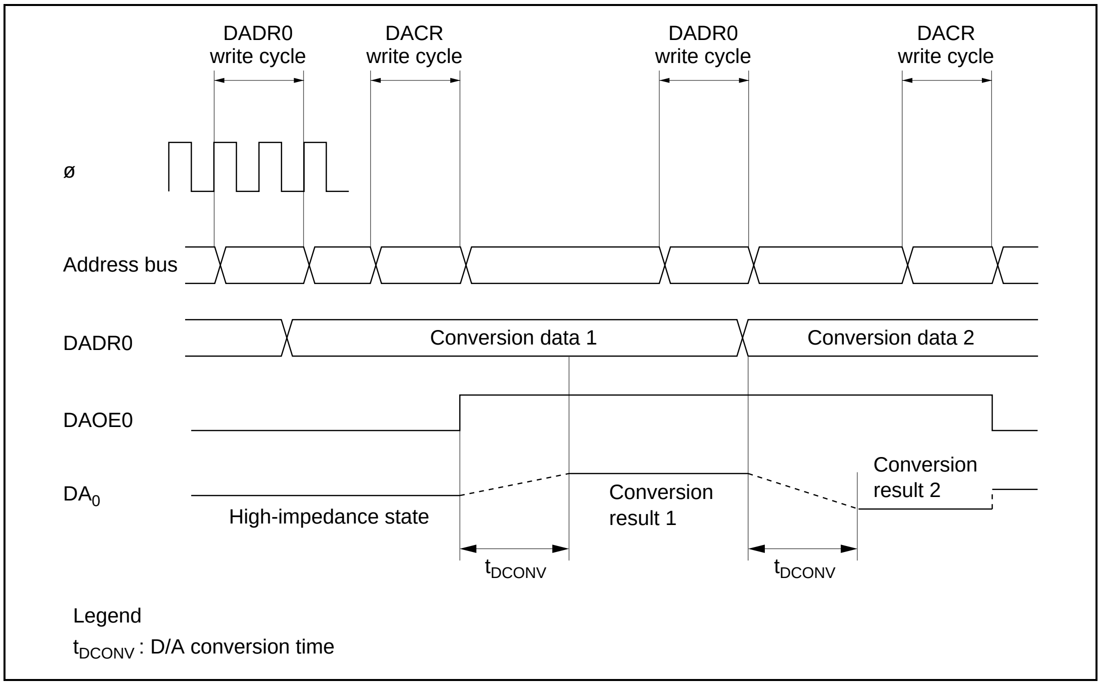


Figure 21.2 Example of D/A Converter Operation

Section 22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Table 22.1 shows tha absolute maximum ratings.

Table 22.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage	Vcc	– 0.3 to 4.6	V
Input voltage (except port L)	Vin	– 0.3 to Vcc + 0.3	V
Input voltage (port L)	Vin	– 0.3 to Avcc + 0.3	V
Analog power supply voltage	AVcc	– 0.3 to 4.6	V
Analog input voltage	V_{AN}	– 0.3 to Avcc + 0.3	V
Operating temperature	Topr	– 20 to 75	°C
Storage temperature	Tstr	– 55 to 125	°C

Caution: Operating the chip in excess of the absolute maximum rating may result in permanent damage.

22.2 DC Characteristics

Tables 22.2 and 22.3 list DC characteristics.

Table 22.2 DC Characteristics
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Power supply voltage	Vcc	3.0	3.3	3.6	V	During normal operation, sleep mode, standby mode
		2.0	3.3	3.6		RTC operating voltage in standby mode

Table 22.2 DC Characteristics (cont)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Current dissipation	Normal operation	Icc	–	100[*1]	240[*1]	mA	Vcc = 3.3 V
			–	–	–		*1: 80 Mhz (Bø: 40 MHz)
			–	–	–		*2: 40 MHz
	In sleep mode[*4]		–	–	–		*3: 20 MHz
			–	35[*2]	45[*2]		*4: no external bus cycles
			–	18[*3]	23[*3]		except refresh cycles
	In standby mode		–	15	30	µA	Ta = 25°C (RTC on)
			–	–	400		Ta > 50°C (RTC on)
			–	5	15		Ta = 25°C (RTC off)
			–	–	300		Ta > 50°C (RTC off)
Input high voltage	$\overline{\text{RESET}}$, NMI	VIH	Vcc × 0.9	–	Vcc + 0.3	V	
	$\overline{\text{BREQ}}$, $\overline{\text{IRQ5}}$–		Vcc–0.5	–	Vcc + 0.3		In standby mode
	$\overline{\text{IRQ0}}$, MD5–MD0		Vcc–0.7	–	Vcc + 0.3		Normal operation
	EXTAL, CKIO		Vcc–0.7	–	Vcc + 0.3		
	Port L		2.0	–	AVcc + 0.3		
	Other input pins		2.0	–	Vcc + 0.3		

Table 22.2 DC Characteristics (cont)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input low voltag e	$\overline{\text{RESET}}$, NMI	VIL	–0.3	–	Vcc × 0.1	V	
	$\overline{\text{BREQ}}$, IRQ5–		–0.3	–	0.5		In standby mode
	IRQ0, MD5–MD0		–0.3	–	Vcc × 0.2		Normal operation
	Port L		–0.3	–	AVcc × 0.2		
	Other input pins		–0.3	–	Vcc × 0.2		
Input leak current	All input pins	I Iin I	–	–	1.0	µA	Vin = 0.5 to Vcc–0.5 V
Three-state leak current	I/O, all output pins (off condition)	I Isti I	–	–	1.0	µA	Vin = 0.5 to Vcc–0.5 V
Output high voltage	All output pins	VOH	2.4	–	–	V	Vcc = 3.0 V, IOH = –200µA
			2.0	–	–		Vcc = 3.0 V, IOH = –2 mA
Output low voltage	All output pins	VOL	–	–	0.55		Vcc = 3.6 V, IOL = 1.6 mA
Pull-up resistance	Port pin	Ppull	30	60	120	KΩ	
Pin capacity	All pins	C	–	–	20	PF	
Analog power supply voltage		AVcc	3.0	3.3	3.6	V	

Table 22.2 DC Characteristics (cont)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Analog power supply	During A/D conversion	AIcc	–	0.8	2	mA	
current	During A/D and D/A conversion		–	2.4	6	mA	
	Idle		–	0.01	5.0	µA	
RAM standby voltage		VRAM	2.0	–	–	V	

Notes: 1. Regardless of whether PLL or RTC is used, connect PLLV$_{cc}$ and RTCV$_{cc}$ to V$_{cc}$, and PLLV$_{ss}$ and RTCVss to V$_{ss}$.

2. AV$_{cc}$ must be under condition of V$_{cc}$ – 0.3V ≤ AV$_{cc}$ ≤ V$_{cc}$ + 0.3V. If the A/D and D/A converters are not used, do not leave the AV$_{cc}$ and AV$_{ss}$ pins open. Connect AV$_{cc}$ to V$_{cc}$, and connect AV$_{ss}$ to V$_{ss}$.

3. Current dissipation values shown are the values at which all output pins are without load under conditions of VIHmin = V$_{cc}$ – 0.5 V, VILmax = 0.5 V.

Table 22.3 Permitted Output Current Values
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	IOL	–	–	2.0	mA
Output low-level permissible current (total)	_ IOL	–	–	120	mA
Output high-level permissible current (per pin)	–IOH	–	–	2.0	mA
Output high-level permissible current (total)	_ (–IOH)	–	–	40	mA

Caution: To ensure LSI reliability, do not exceed the value for output current given in Table 22.3.

22.3 AC Characteristics

In general, inputting for this LSI should be clock synchronous. Keep to the setup and hold times for each input signal unless otherwise specified.

Table 22.4 Maximum Operating Frequencies
(Vcc = 3.3V ± 0.3V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Remark
Operating frequency	CPU, cache, TLB	f	1	–	80	MHz	
	External bus		1	–	40		
	Peripheral module		0.25	–	30		

22.3.1 Clock Timing

Table 22.5 Clock Timing (1)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 20 Mhz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	fEX	5	20	MHz	22.1
EXTAL clock input cycle time	tEXcyc	50	200	ns	
EXTAL clock input low pulse width	tEXL	8[*1] or 12[*2]	–	ns	
EXTAL clock input high pulse width	tEXH	8[*1] or 12[*2]	–	ns	
EXTAL clock input rise time	tEXR	–	4	ns	
EXTAL clock input fall time	tEXF	–	4	ns	
CKIO clock input frequency	fCKI	16	20	MHz	22.2
CKIO clock input cycle time	tCKIcyc	50	66.7	ns	
CKIO clock input low pulse width	tCKIL	8	–	ns	
CKIO clock input high pulse width	tCKIH	8	–	ns	
CKIO clock input rise time	tCKIR	–	4	ns	
CKIO clock input fall time	tCKIF	–	4	ns	
CKIO clock output frequency	fOP	5	20	MHz	22.3
CKIO clock output cycle time	tcyc	50	200	ns	
CKIO clock output low pulse width	tCKOL	20	–	ns	
CKIO clock output high pulse width	tCKOH	20	–	ns	
CKIO clock output rise time	tCKOR	–	7	ns	
CKIO clock output fall time	tCKOF	–	7	ns	
Power-on oscillation settling time	tOSC1	10	–	ms	22.4
$\overline{\text{BREQ}}$ setup time	tRESBRQS	20	–	ns	
$\overline{\text{BREQ}}$ reset hold time	tBREQRH	0	–	ns	
$\overline{\text{RESETP}}$ setup time	tREPS	20	–	ns	22.4, 22.5
$\overline{\text{RESETM}}$ setup time	tREMS	0	–	ns	
$\overline{\text{RESETP}}$ assert time	tREPW	20	–	ns	
$\overline{\text{RESETM}}$ assert time	tREMW	20	–	tcyc	
Standby return oscillation settling time 1	tOSC2	10	–	ms	22.5
Standby return oscillation settling time 2	tOSC3	10	–	ms	22.6

Table 22.5 Clock Timing (2)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 40 MHz)

Item	Symbol	Min	Max	Unit	Figure
Standby return oscillation settling time 3	tOSC4	11	–	ms	22.7
PLL synchronization settling time	tPLL	100	–	us	22.8, 22.9, 22.10
IRQ interrupt determination time (RTC used and standby mode)	tIRQSTB	100	–	us	22.10

Notes: 1. With PLL circuit 1 operating.
2. With PLL circuit 2 not used.

Table 22.5 Clock Timing (3)

(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 40 Mhz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	fEX	5	40	MHz	22.1
EXTAL clock input cycle time	tEXcyc	25	200	ns	
EXTAL clock input low pulse width	tEXL	7[*1] or 10[*2]	–	ns	
EXTAL clock input high pulse width	tEXH	7[*1] or 10[*2]	–	ns	
EXTAL clock input rise time	tEXR	–	4	ns	
EXTAL clock input fall time	tEXF	–	4	ns	
CKIO clock input frequency	fCKI	16	40	MHz	22.2
CKIO clock input cycle time	tCKIcyc	25	66.7	ns	
CKIO clock input low pulse width	tCKIL	7	–	ns	
CKIO clock input high pulse width	tCKIH	7	–	ns	
CKIO clock input rise time	tCKIR	–	3	ns	
CKIO clock input fall time	tCKIF	–	3	ns	

Table 22.5 Clock Timing (4)
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C,
Maximum External Bus Operating Frequency: 40 Mhz)

Item	Symbol	Min	Max	Unit	Figure
CKIO clock output frequency	fOP	5	40	MHz	22.3
CKIO clock output cycle time	tcyc	25	200	ns	
CKIO clock output low pulse width	tCKOL	8	–	ns	
CKIO clock output high pulse width	tCKOH	8		ns	
CKIO clock output rise time	tCKOR	–	6	ns	
CKIO clock output fall time	tCKOF	–	6	ns	
Power-on oscillation settling time	tOSC1	10	–	ms	22.4, 22.5
$\overline{\text{RESETP}}$ setup time	tRESPS	20	–	ns	
$\overline{\text{RESETM}}$ setup time	tRESMS	0	–	ns	
$\overline{\text{RESETP}}$ assert time	tRESPW	20	–	ns	
$\overline{\text{RESETM}}$ assert time	tRESMW	20	–	ns	
Standby return oscillation settling time 1	tOSC2	10	–	ms	22.5
Standby return oscillation settling time 2	tOSC3	10	–	ms	22.6
Standby return oscillation settling time 3	tOSC4	11	–	ms	22.7
PLL synchronization settling time	tPLL	100	–	us	22.8, 22.9, 22.10
IRQ interrupt determination time (RTC used and standby mode)	tIRQSTB	100	–	us	22.10

Notes: 1. With PLL circuit 1 operating.
2. With PLL circuit 2 not used.

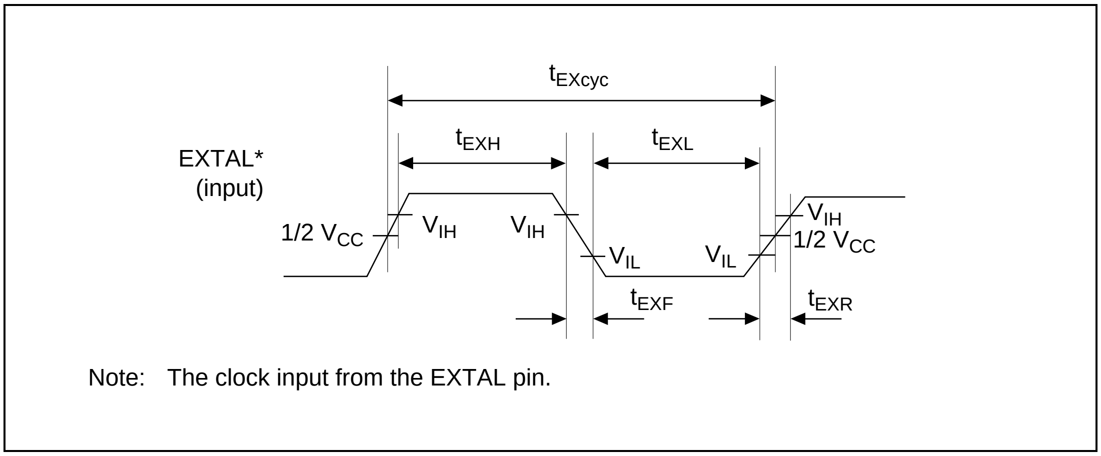


Figure 22.1 EXTAL Clock Input Timing

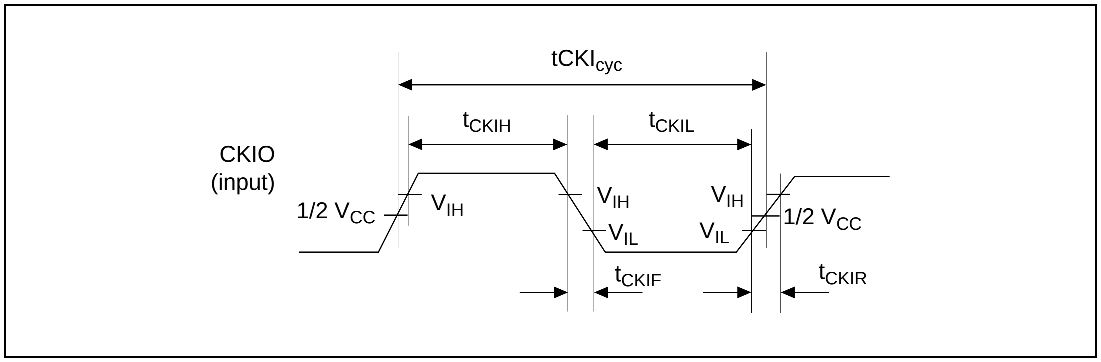


Figure 22.2 CKIO Clock Input Timing

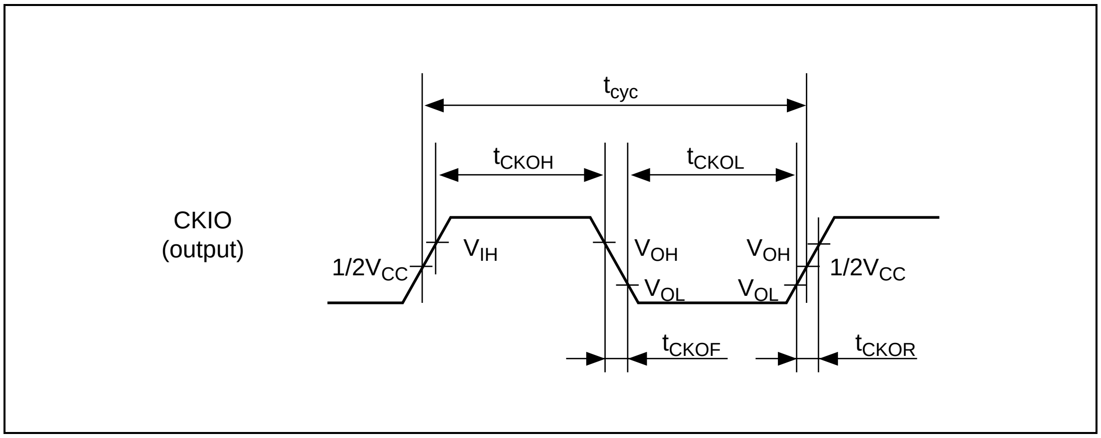
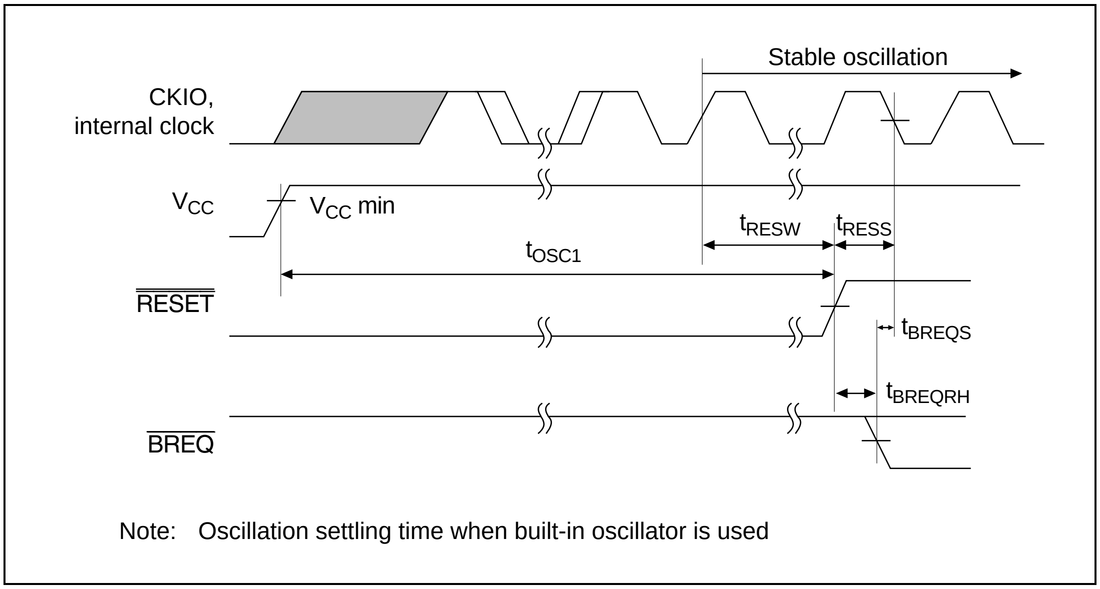
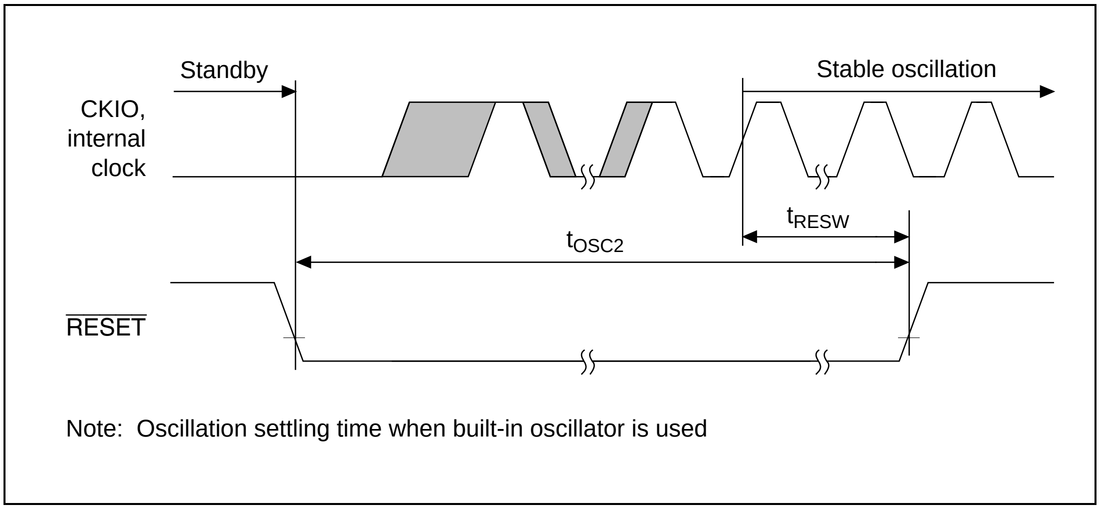


Figure 22.3 CKIO Clock Output Timing



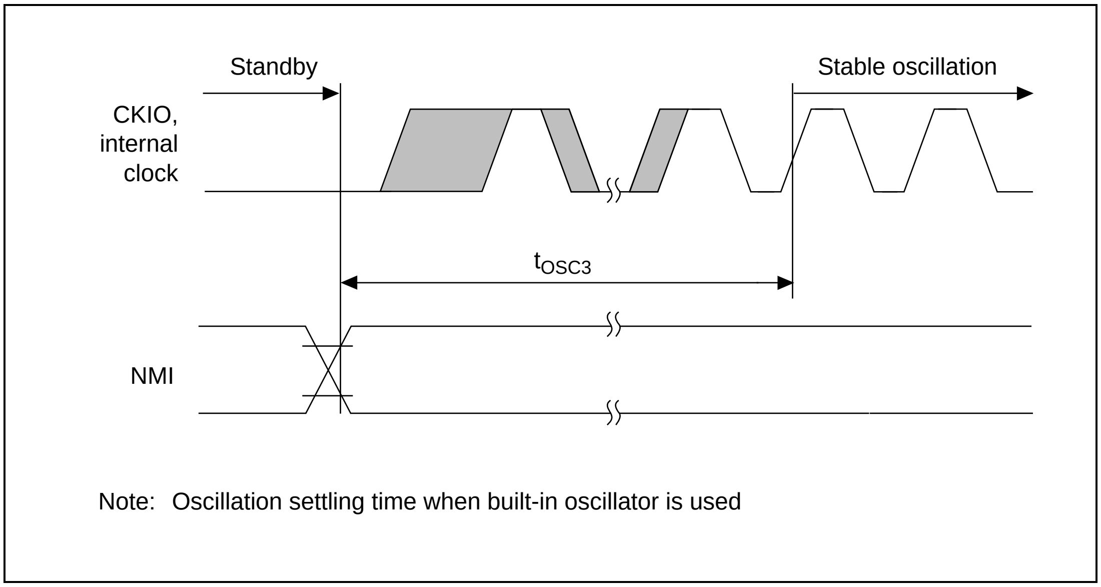
Note: Oscillation settling time when built-in oscillator is used

Figure 22.4 Power-on Oscillation Settling Time



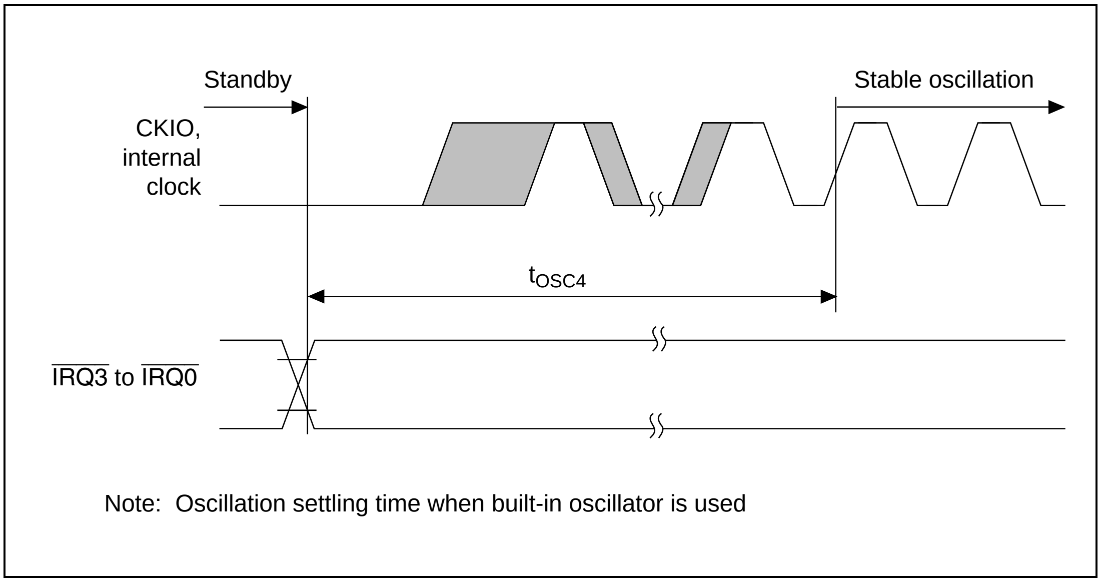
Note: Oscillation settling time when built-in oscillator is used

Figure 22.5 Oscillation Settling Time at Standby Return (Return by Reset)



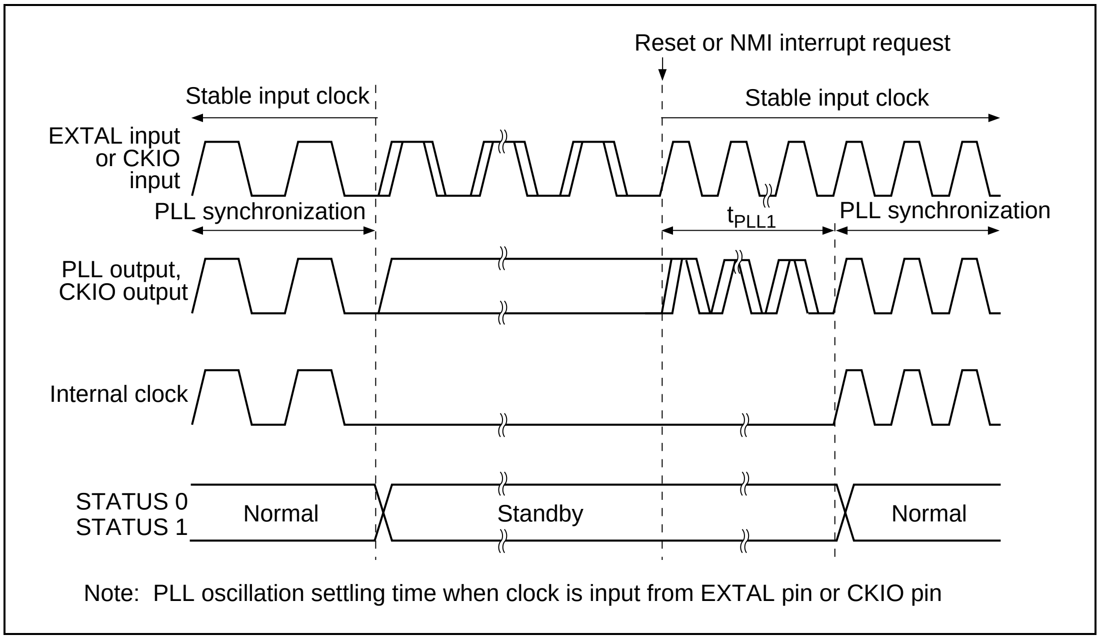
Note: Oscillation settling time when built-in oscillator is used

Figure 22.6 Oscillation Settling Time at Standby Return (Return by NMI)



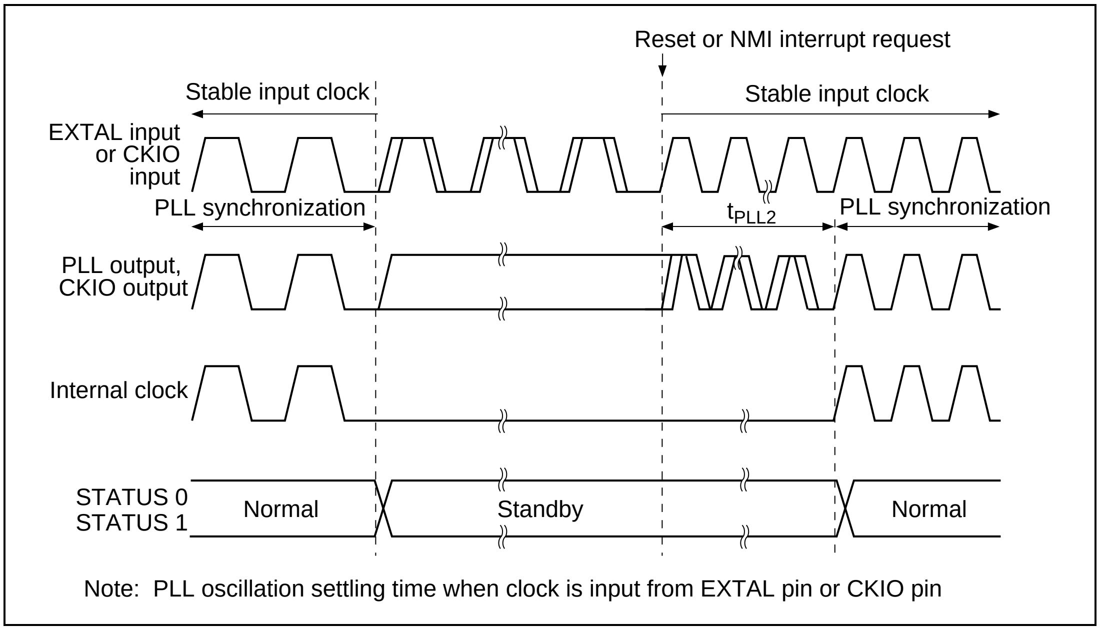
Note: Oscillation settling time when built-in oscillator is used

Figure 22.7 Oscillation Settling Time at Standby Return
(Return by $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$)



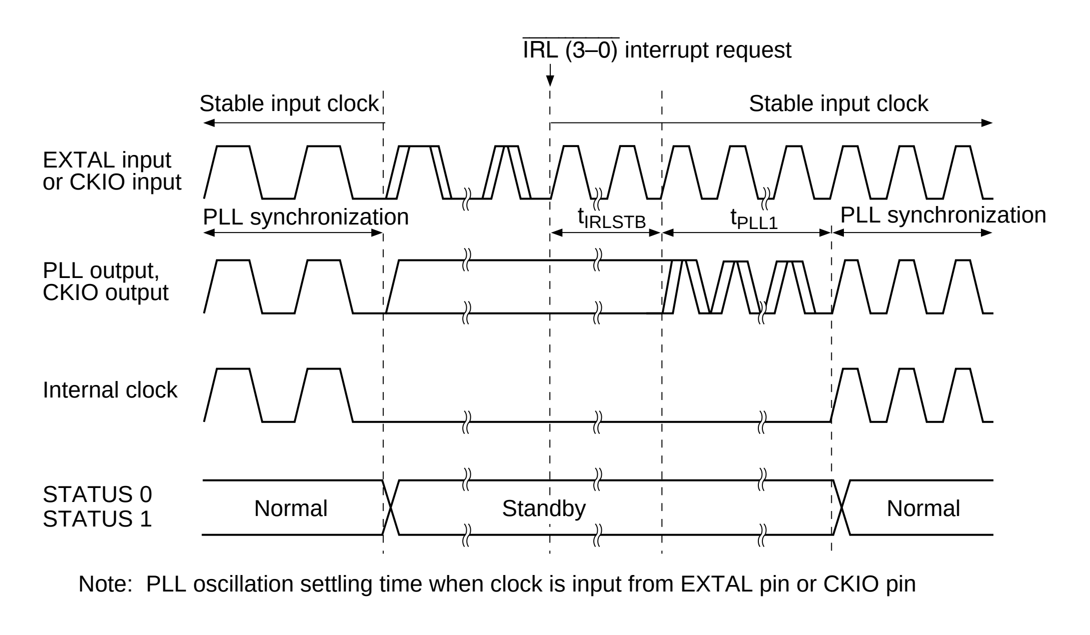
Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 22.8 PLL Synchronization Settling Time by Reset or NMI



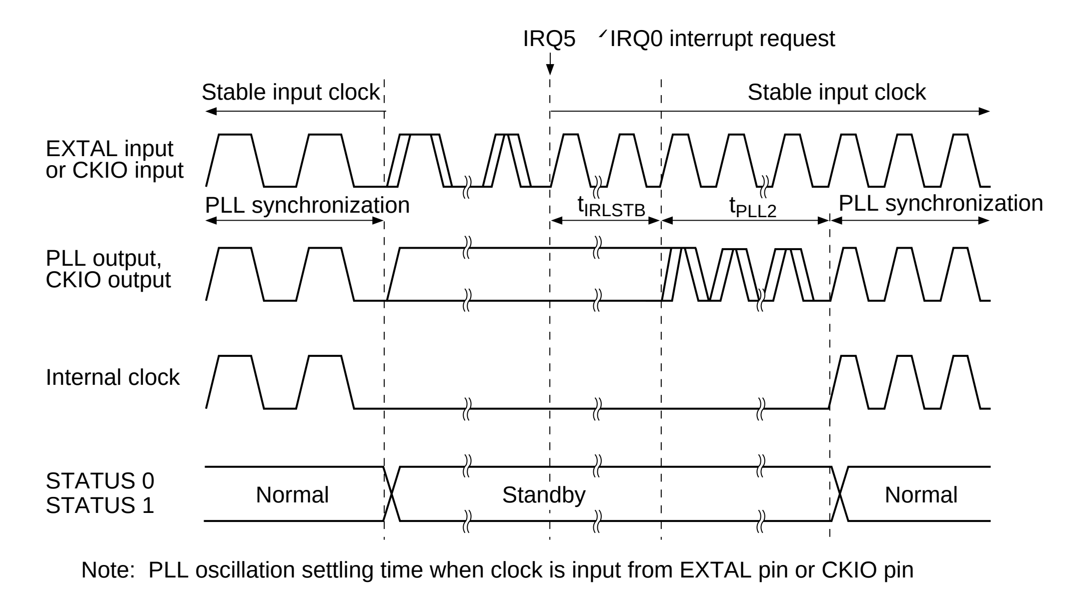
Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 22.9 PLL Synchronization Settling Time by Reset or NMI



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 22.10 PLL Synchronization Settling Time by IRQ/IRL Interrupt



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 22.11 PLL Synchronization Settling Time by IRQ/IRL Interrupt

22.3.2 Control Signal Timing

Table 22.6 Control Signal Timing (Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item	Symbol	–40 *2		Unit	Figure
		Min	Max		
$\overline{\text{RESETP}}$ pulse width	tRESPW	20 *4	—	tcyc	
$\overline{\text{RESETP}}$ setup time*1	tRESPS	23	—	ns	
$\overline{\text{RESETP}}$ hold time	tRESPH	2	—	ns	22.12
$\overline{\text{RESETM}}$ pulse width	tRESMW	12 *3	—	tcyc	22.13
$\overline{\text{RESETM}}$ setup time	tRESMS	3	—	ns	22.15
$\overline{\text{RESETM}}$ hold time	tRESMH	34	—	ns	
$\overline{\text{BREQ}}$ setup time	tBREQS	12	—	ns	22.15
$\overline{\text{BREQ}}$ hold time	tBREQH	3	—	ns	
NMI setup time*1	tNMIS	10	—	ns	
$\overline{\text{IRQ5}}$–$\overline{\text{IRQ0}}$ setup time*1	tIRQS	10	—	ns	22.13
NMI hold time	tNMIH	4	—	ns	22.14
$\overline{\text{IRQ5}}$–$\overline{\text{IRQ0}}$ hold time	tIRQH	4	—	ns	
$\overline{\text{IRQOUT}}$ delay time	tIRQOD	—	14	ns	
$\overline{\text{BACK}}$ delay time	tBACKD	—	14	ns	
STATUS1, STATUS0 delay time	tSTD	—	18	ns	
Bus tri-state delay time 1	tBOFF1	0	33	ns	22.15
Bus tri-state delay time 2	tBOFF2	0	33	ns	22.16
Bus buffer on time 1	tBON1	0	33	ns	
Bus buffer on time 2	tBON2	0	33	ns	

Notes: 1. RESET, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock fall when the setup shown is used. When the setup cannot be used, detection can be delayed until the next clock falls.

2. The upper limit of the external bus clock is 40 MHz.

3. In the standby mode, tRESMW=tOSC2 (10ms). In the sleep mode, RESETM must be kept low until STATUS (0-1) change to reset (HH). When the clock multiplication ratio being changed, RESETM must be kept low until STATUS (0-1) change to reset (HH).

4. In the standby mode, tRESPW=tOSC2 (10ms). When the multiplication factor is changed, tRESPW=tPLL1 (100us).

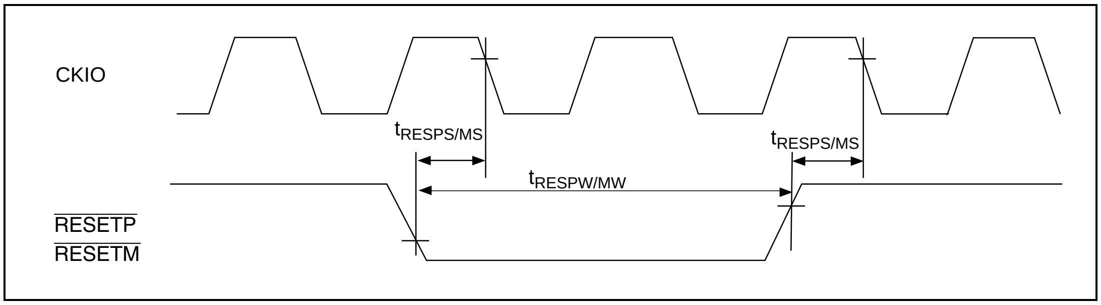


Figure 22.12 Reset Input Timing

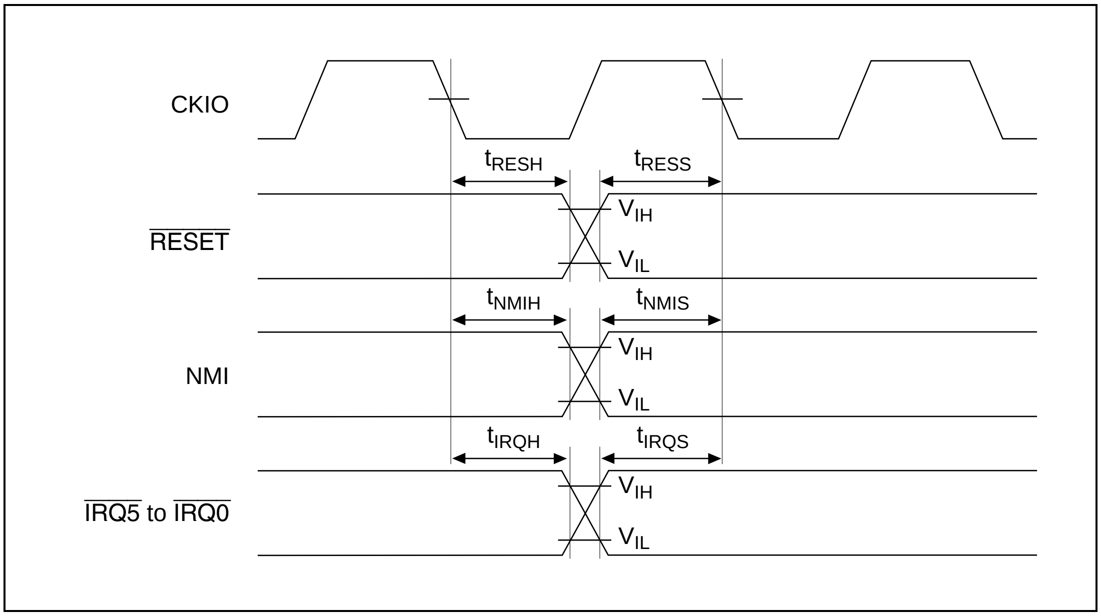


Figure 22.13 Interrupt Signal Input Timing

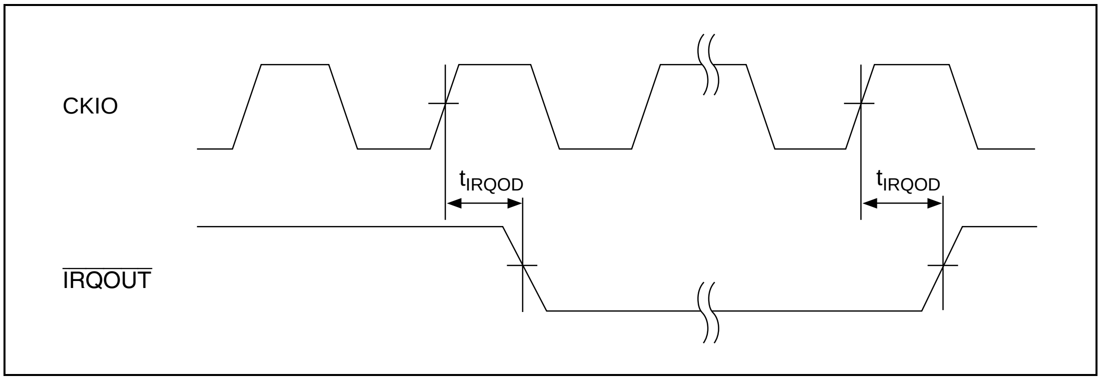


Figure 22.14 $\overline{\text{IRQOUT}}$ Timing

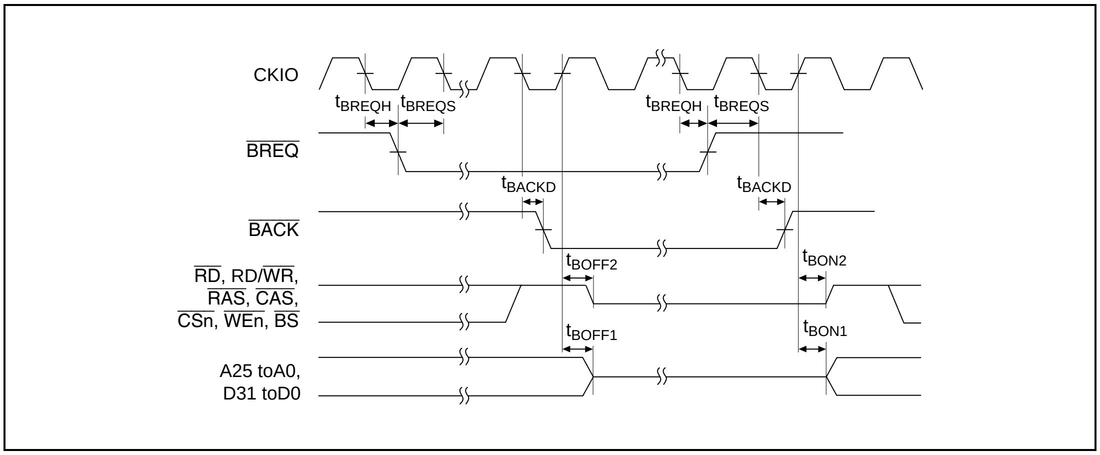


Figure 22.15 Bus Release Timing

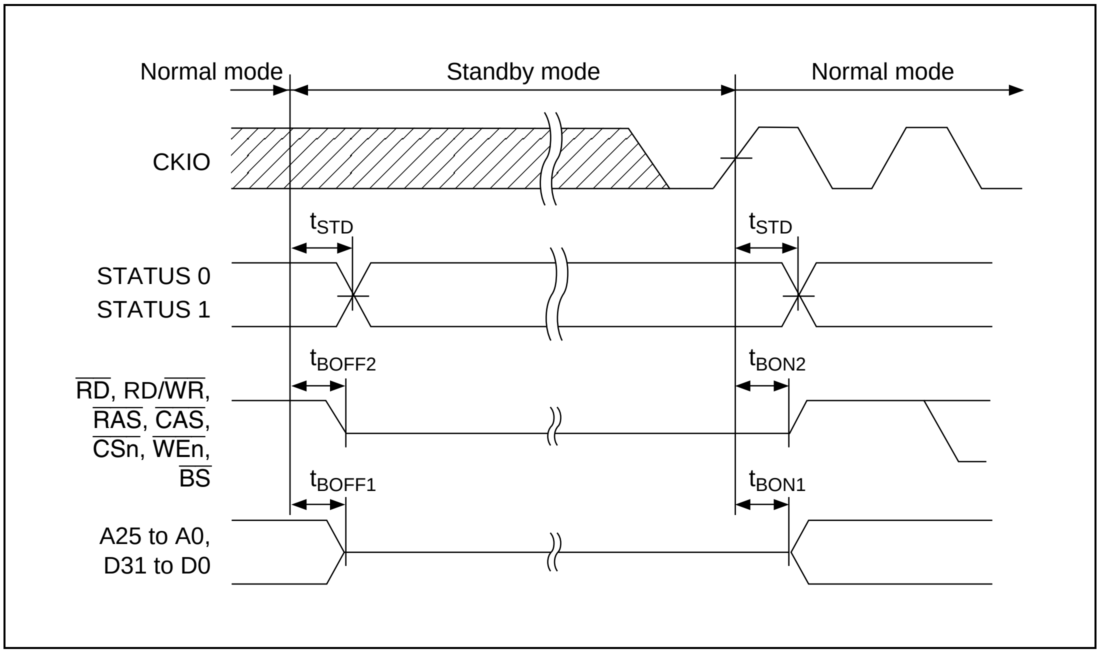


Figure 22.16 Pin Drive Timing at Standby

Table 22.7 Bus Timing (1)
(Clock Modes 0/1/2/3/4/7, Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item	Symbol	–40 *1		Unit	Figure
		Min	Max		
Address delay time	tAD	1	15	ns	22.17–22.30, 22.36–22.43, 22.46 - 22.53
Address setup time	tAS	0	–	ns	22.23–22.30
Address hold time	tAH	10	–	ns	22.17–22.30
$\overline{\text{BS}}$ delay time	tBSD	–	14	ns	22.17–22.30, 22.36–22.42, 22.46 - 22.53
$\overline{\text{CS}}$ delay time 1	tCSD1	1	14	ns	22.17–22.30, 22.36–22.42, 22.46 - 22.53
$\overline{\text{CS}}$ delay time 2	tCSD2	1	14	ns	22.17–22.22
Read write delay time	tRWD	1	14	ns	22.17–22.30, 22.36–22.42, 22.46 - 22.53
Read write hold time	tRWH	0	–	ns	22.17–22.30
Read strobe delay time	tRSD	–	14	ns	22.17–22.22, 22.47–22.53
Read data setup time 1	tRDS1	12	–	ns	22.17–22.26, 22.47–22.53
Read data setup time 2	tRDS2	8	–	ns	22.27–22.30, 22.36 - 22.39
Read data setup time 3	tRDS3	12	–	ns	22.31
Read data setup time 4	tRDS4	12	–	ns	22.32
Read data hold time 1	tRDH1	0	–	ns	22.17–22.26, 22.47–22.53
Read data hold time 2	tRDH2	3	–	ns	22.36–22.39
Read data setup time 2	tRDS2	8	–	ns	22.27–22.30, 22.36 - 22.39
Read data setup time 3	tRDS3	12	–	ns	22.31
Read data setup time 4	tRDS4	12	–	ns	22.32
Read data hold time 1	tRDH1	0	–	ns	22.17–22.26, 22.47–22.53
Read data hold time 2	tRDH2	3	–	ns	22.36–22.39
Read data hold time 3	tRDH3	0	–	ns	22.31
Read data hold time 4	tRDH4	6	–	ns	22.32
Write enable delay time	tWED	–	14	ns	22.17–22.19, 22.47 - 22.48
Write data delay time 1	tWDD1	–	17	ns	22.17–22.19, 22.47- 22.48, 22.51–22.53
Write data delay time 2	tWDD2	–	15	ns	22.23–22.30, 22.40 - 22.43
Write data setup time	tWDS	0	–	ns	22.23–22.30

Table 22.7 Bus Timing (1) (cont)
(Clock Modes 0/1/2/3/4/7, Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3V, AVcc = Vcc ± 0.3V, Ta = –20 to 75°C)

Item	Symbol	–40 [*1]		Unit	Figure
		Min	Max		
Write data hold time 1	tWDH1	2	–	ns	22.17–22.19, 22.23–22.30, 22.47–22.53
Write data hold time 2	tWDH2	2	–	ns	22.40-22.43
Write data hold time 3	tWDH3	2	–	ns	22.17–22.19, 22.23–22.30
Write data hold time 4	tWDH4	2	–	ns	22.47–22.53
$\overline{\text{WAIT}}$ setup time[*2]	tWTS	12	–	ns	22.18–22.22, 22.48, 22.50, 22.52, 22.53
$\overline{\text{WAIT}}$ hold time[*2]	tWTH	6	–	ns	22.18–22.22, 22.48, 22.50, 22.52, 22.53
$\overline{\text{RAS}}$ delay time 1	tRASD1	1	15	ns	22.23–22.33
$\overline{\text{RAS}}$ delay time 2	tRASD2	1	13	ns	22.27–22.30
$\overline{\text{CAS}}$ delay time 1	tCASD1	1	15	ns	22.23–22.33
$\overline{\text{CAS}}$ delay time 2	tCASD2	1	13	ns	22.36-22.46
$\overline{\text{CAS}}$ delay time 3	tCASD3	1	15	ns	22.31-22.32
$\overline{\text{CAS}}$ delay time 4	tCASD4	1	15	ns	22.31-22.32
$\overline{\text{DQM}}$ $\overline{\text{delay}}$ $\overline{\text{time}}$	tDQMD	1	12	ns	22.36-22.43
$\overline{\text{ICIORD}}$ delay time	tICRSD	–	14	ns	22.51–22.53
$\overline{\text{ICIOWR}}$ delay time	tICWSD	–	14	ns	22.51–22.53
$\overline{\text{IOIS16}}$ setup time	tIO16S	12	–	ns	22.52, 22.53
$\overline{\text{IOIS16}}$ hold time	tIO16H	6	–	ns	22.52, 22.53
$\overline{\text{DACK}}$ delay time 1	tDAKD1	–	14	ns	22.17–22.32, 22.47–22.53
$\overline{\text{DACK}}$ delay time 2	tDAKD2	–	15	ns	22.31, 22.32

Notes: 1. The upper limit of the external bus clock is 40 MHz.

2. $\overline{\text{WAIT}}$ is synchronous. Misoperation may result if the setup and hold times are not observed.

22.3.3 Basic Timing

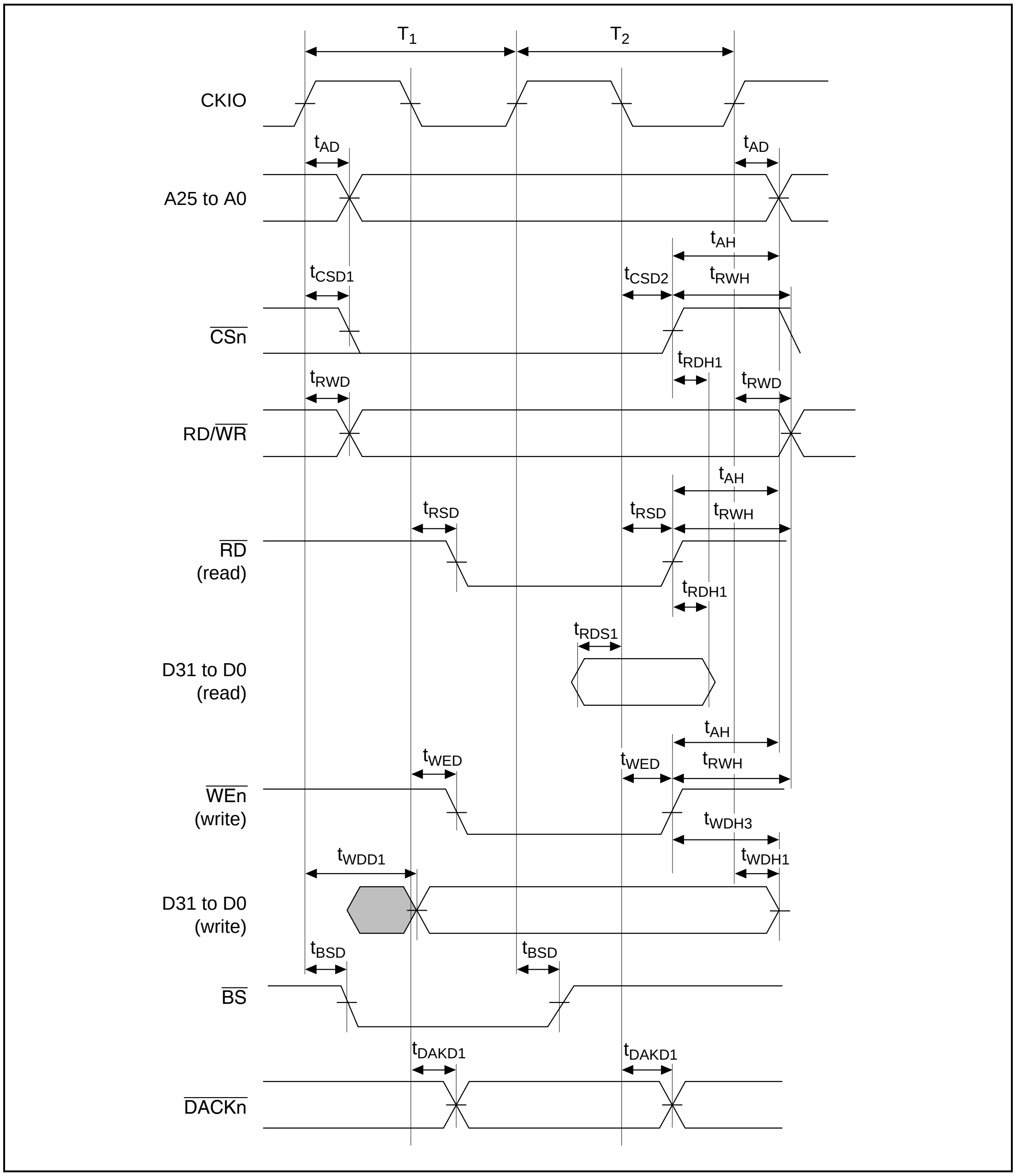


Figure 22.17 Basic Bus Cycle (No Wait)

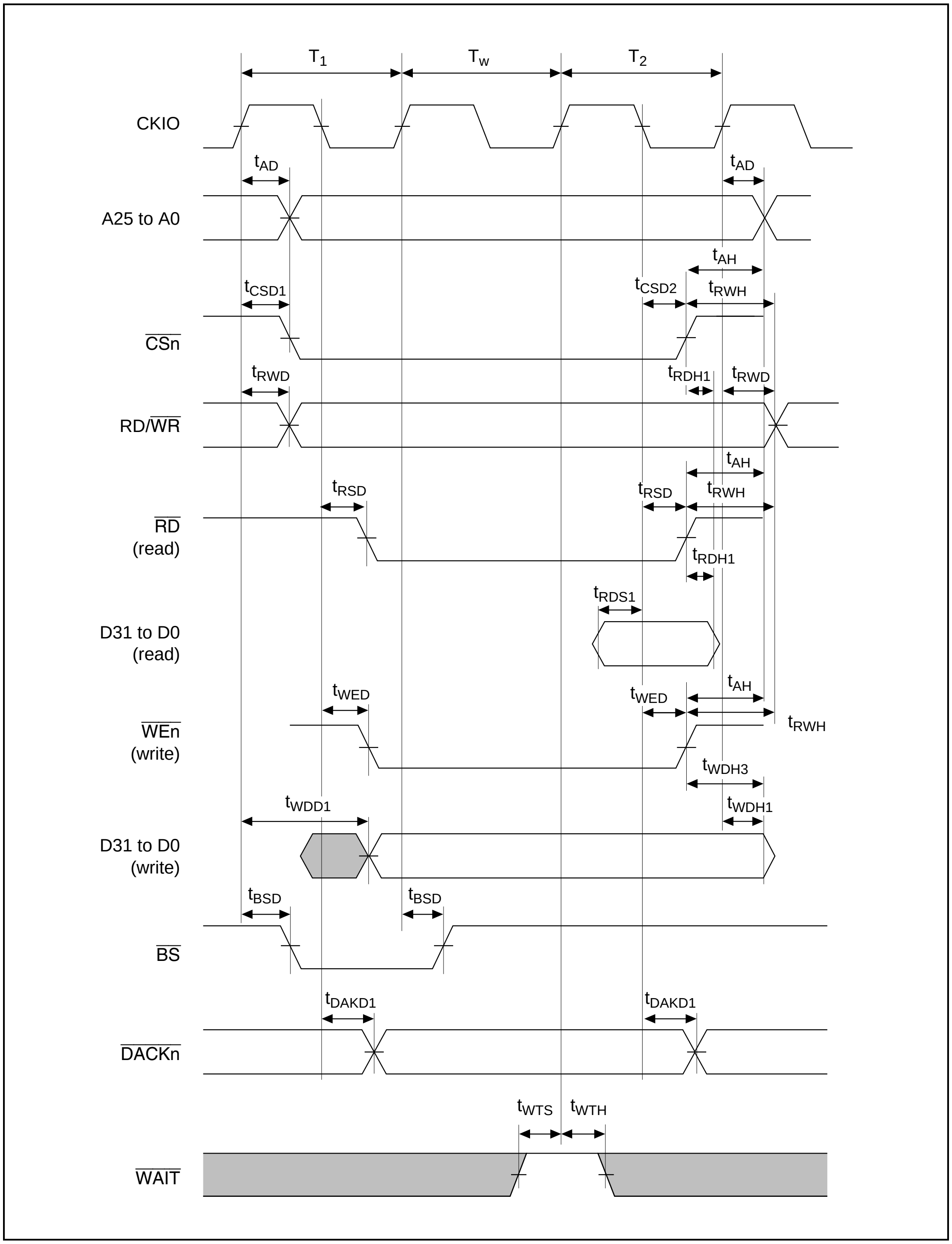


Figure 22.18 Basic Bus Cycle (One Wait)

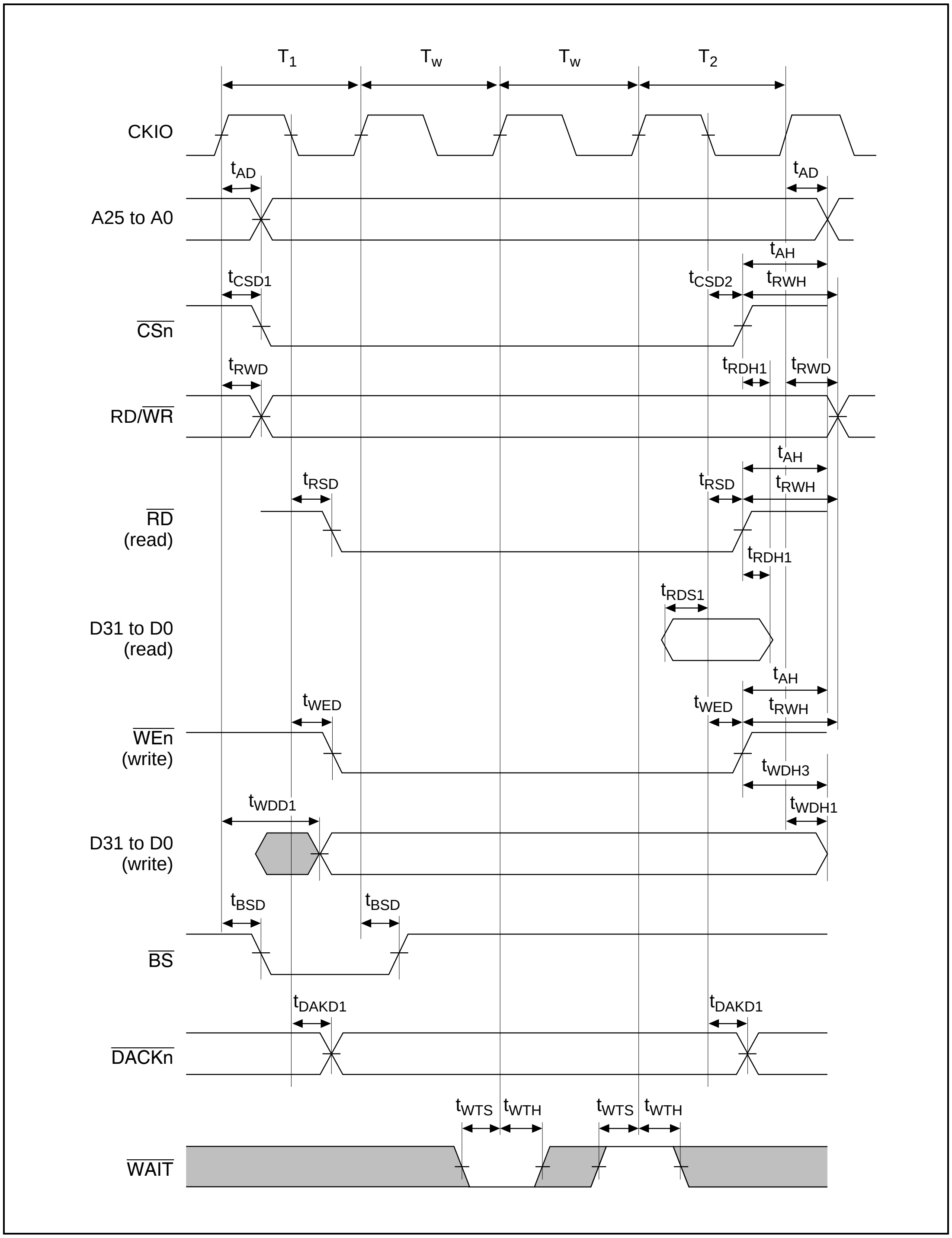
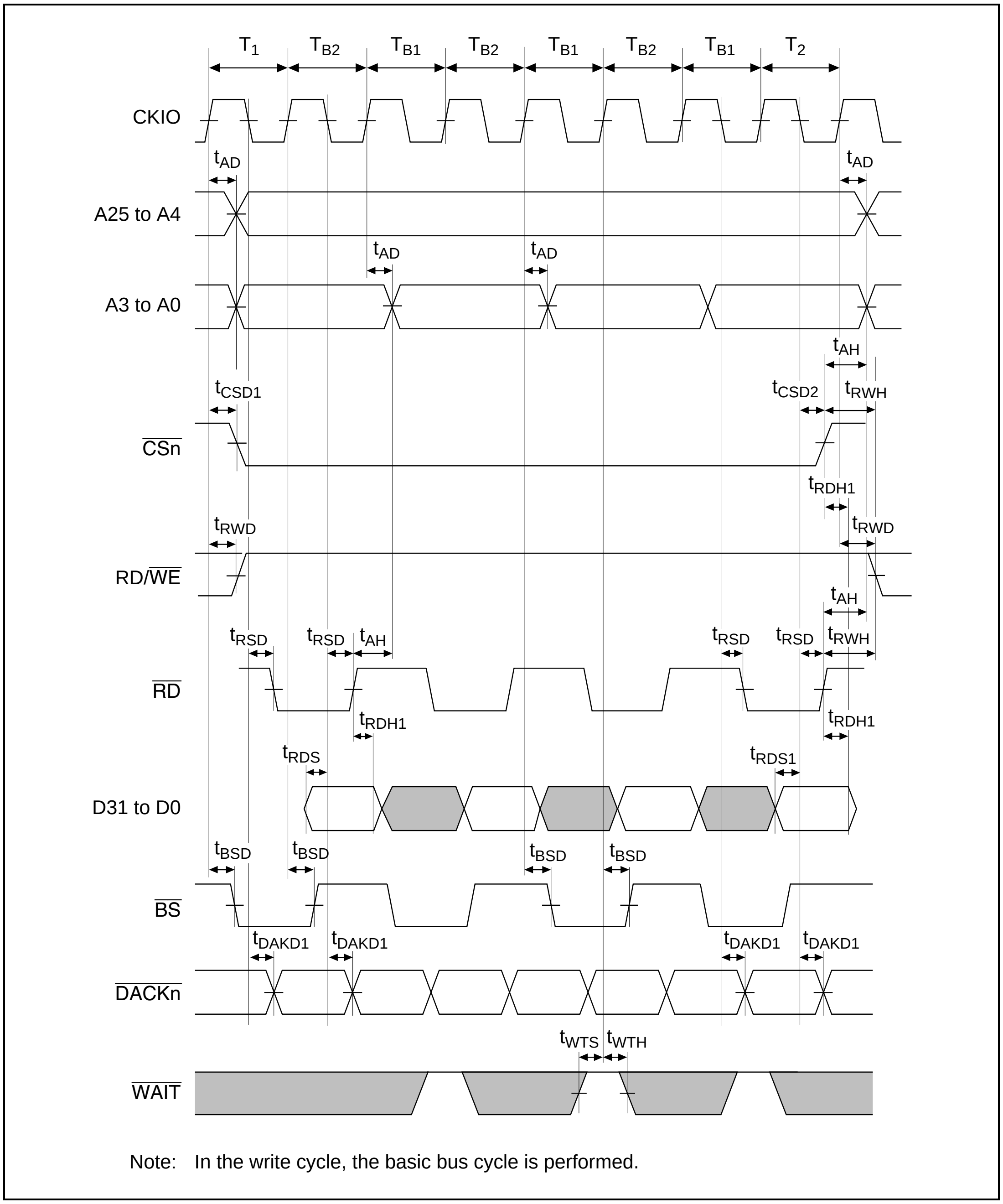


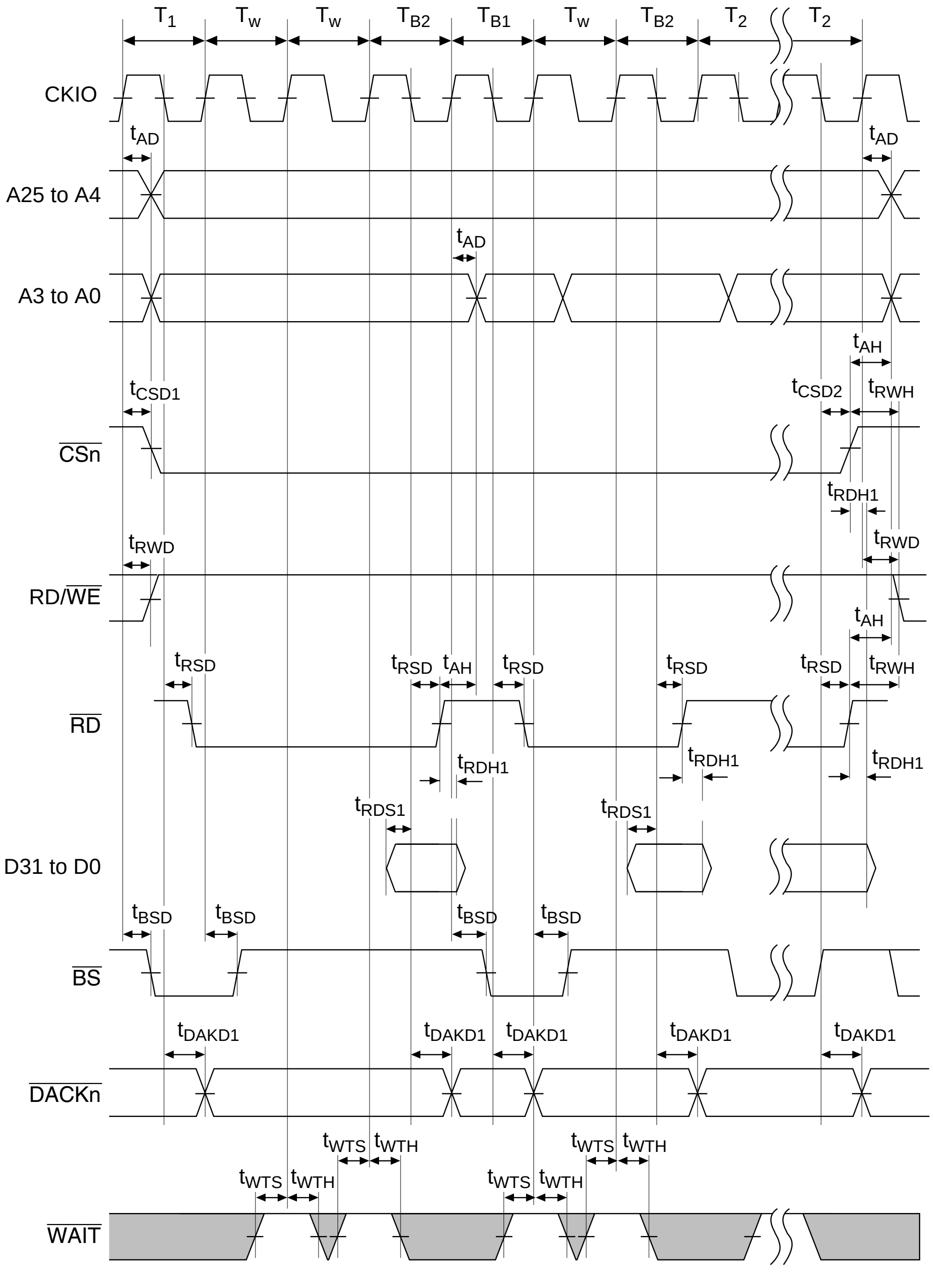
Figure 22.19 Basic Bus Cycle (External Wait)

22.3.4 Burst ROM Timing



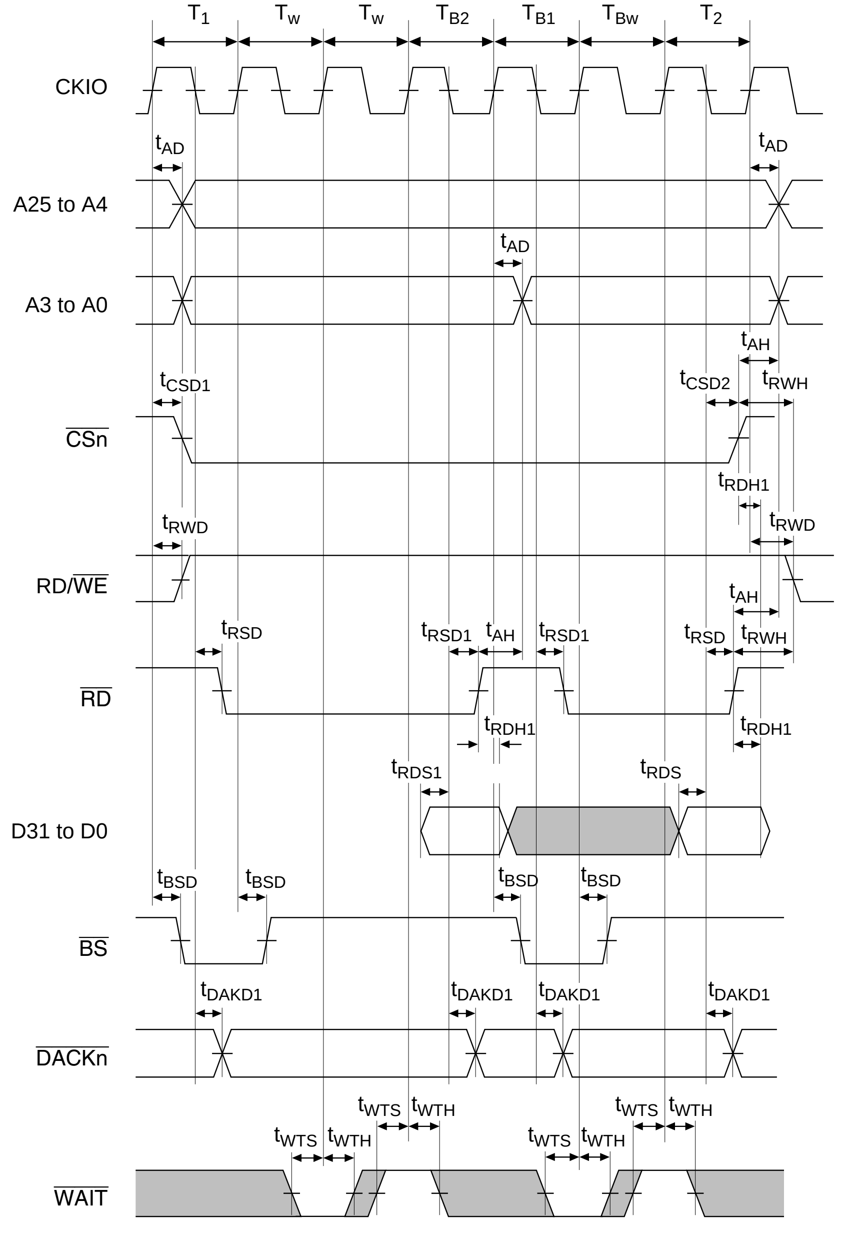
Note: In the write cycle, the basic bus cycle is performed.

Figure 22.20 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 22.21 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 22.22 Burst ROM Bus Cycle (External Wait)

22.3.5 DRAM Timing

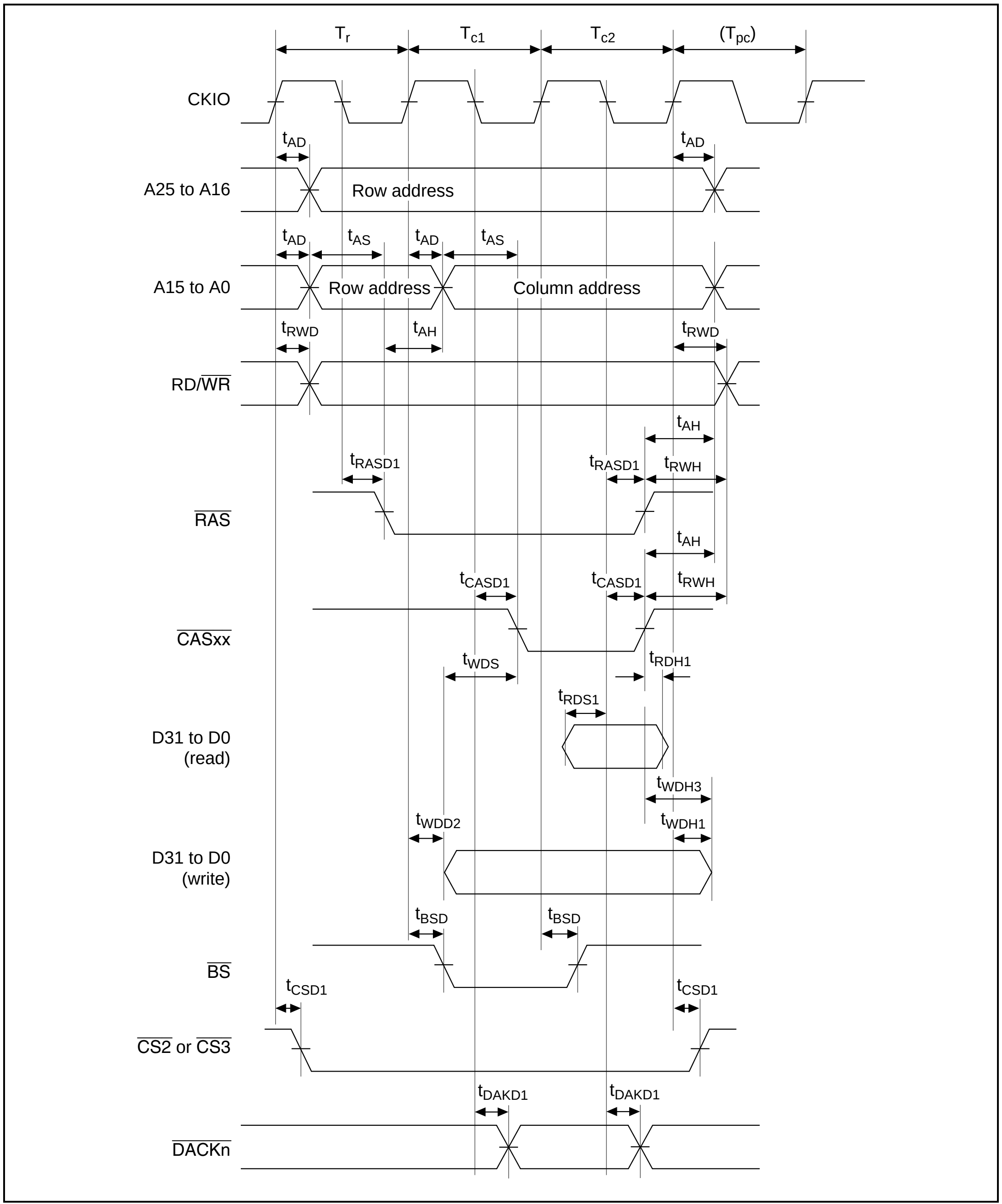


Figure 22.23 DRAM Bus Cycle (TRCD = 0, AnW = 1, TPC = 0)

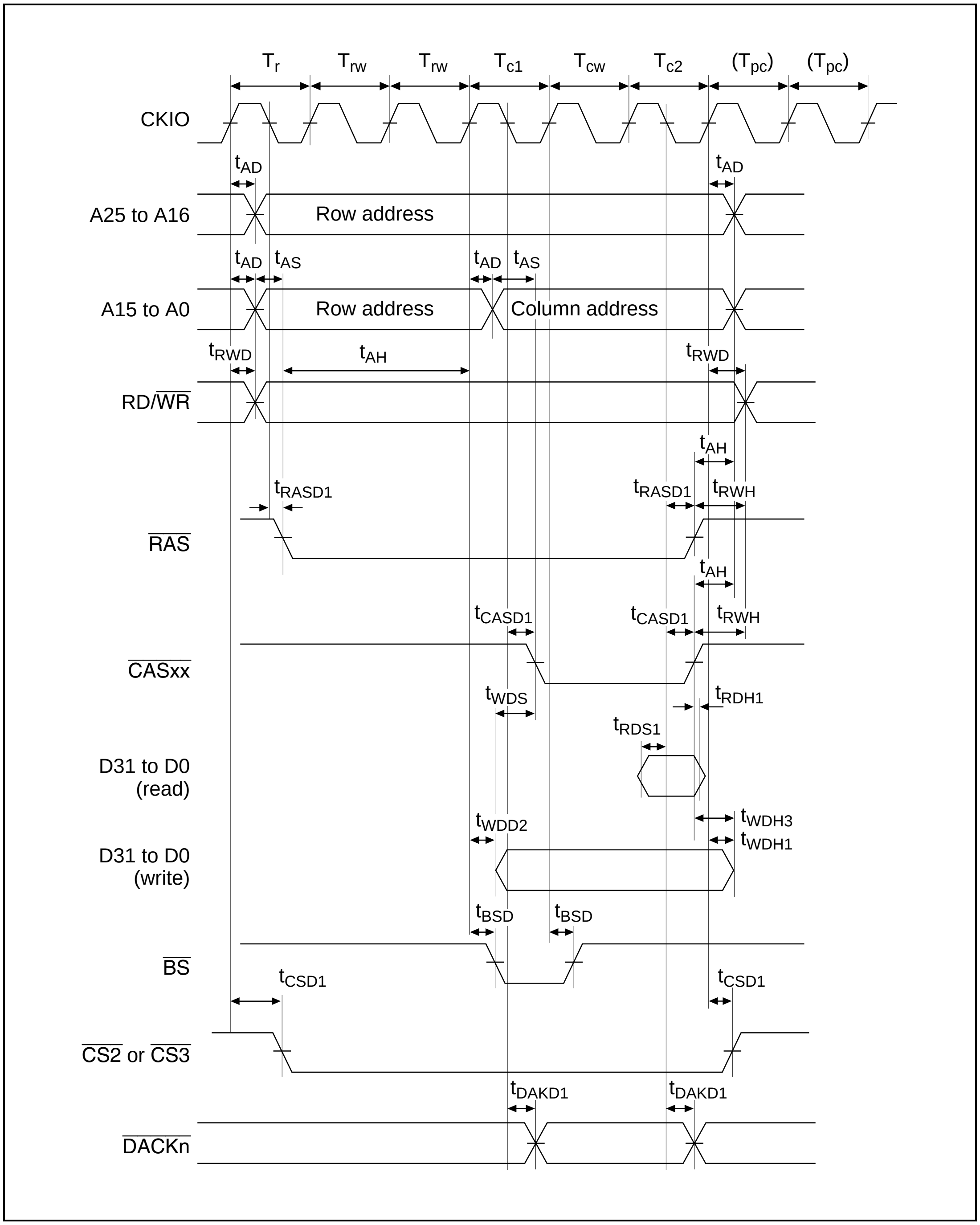


Figure 22.24 DRAM Bus Cycle (TRCD = 2, AnW = 2, TPC = 1)

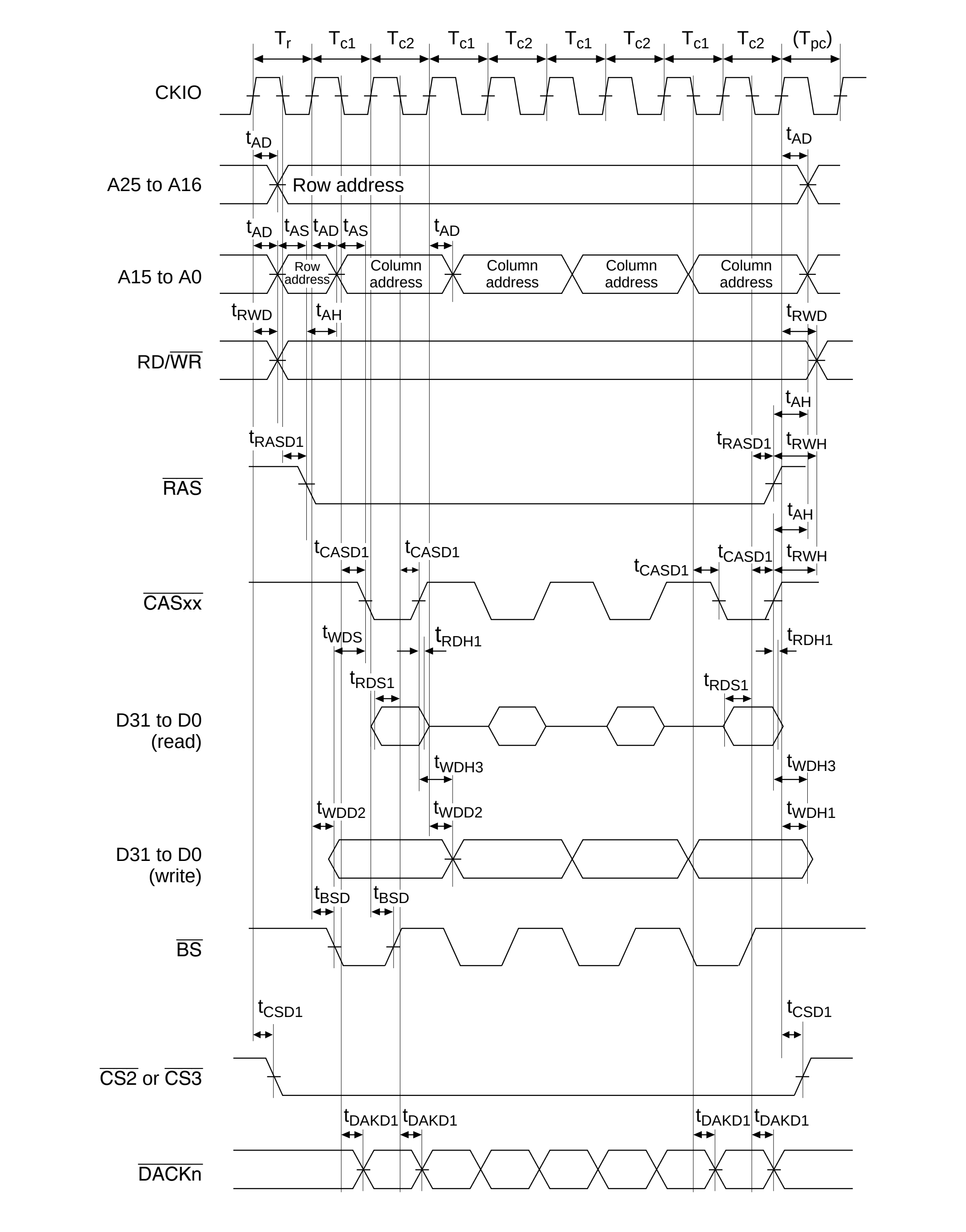


Figure 22.25 DRAM Burst Bus Cycle (TRCD = 0, AnW = 1, TPC = 0)

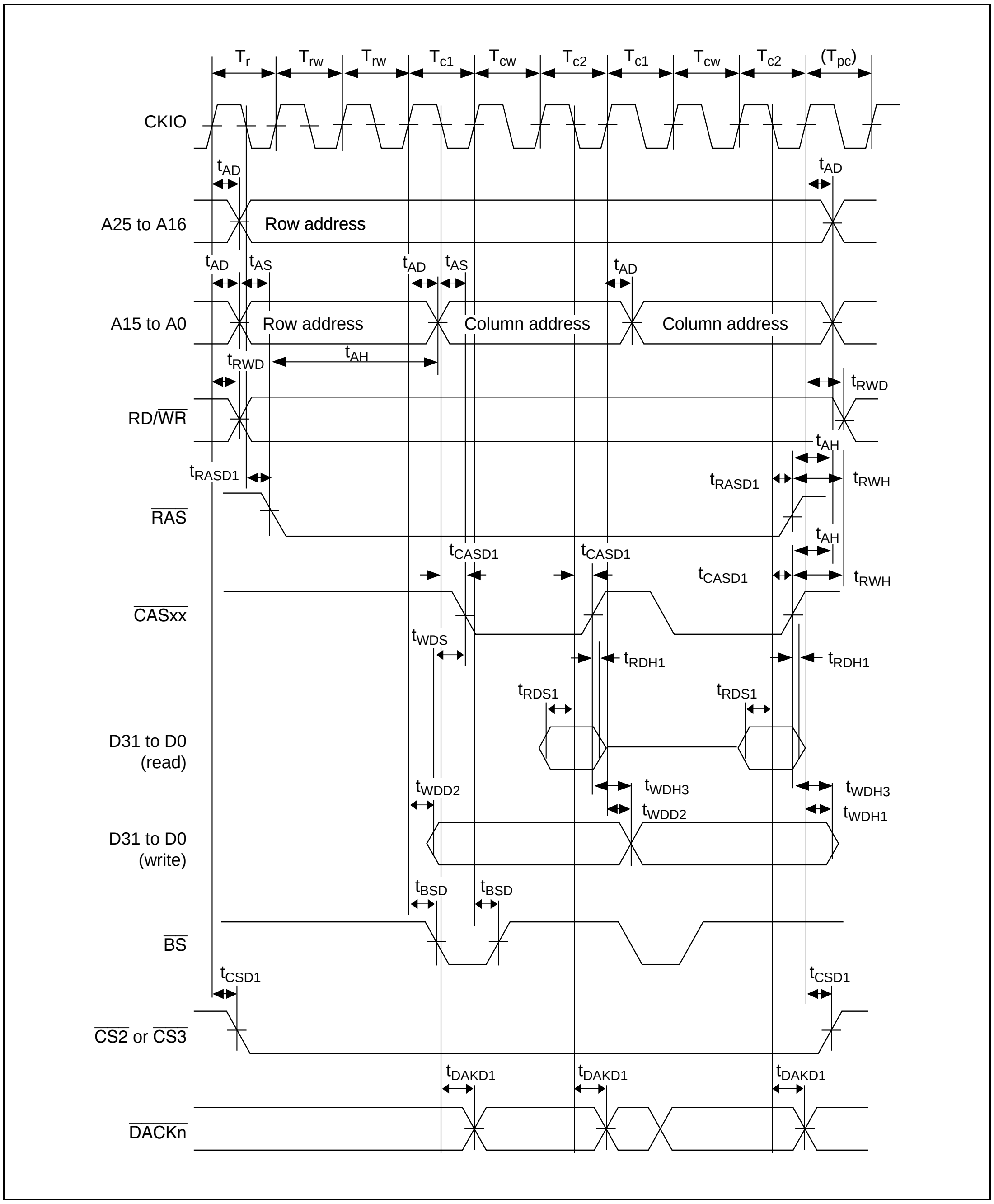


Figure 22.26 DRAM Burst Bus Cycle (TRCD = 2, AnW = 2, TPC = 0)

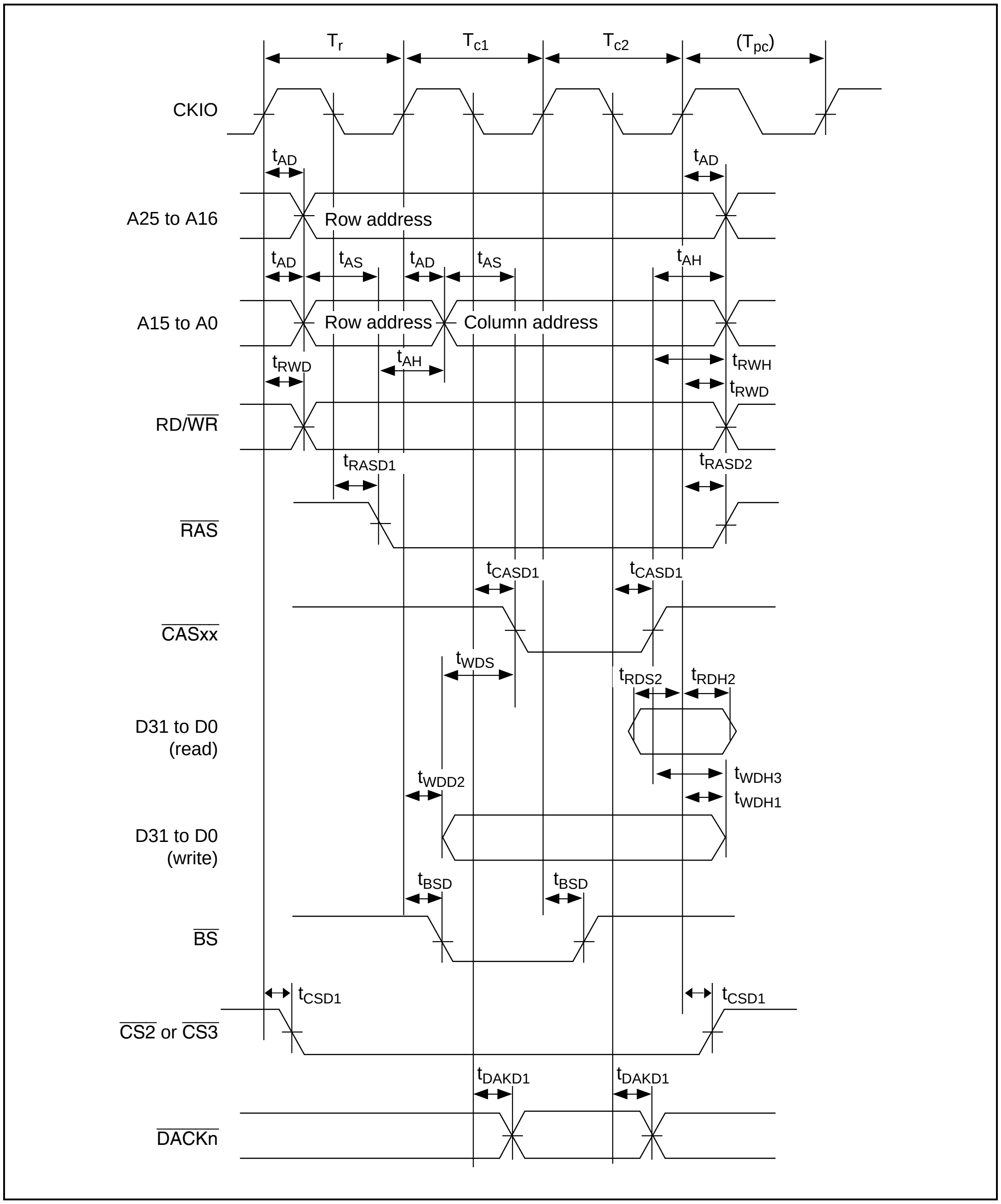


Figure 22.27 DRAM Bus Cycle (EDO mode, TRCD = 0, AnW = 1, TPC = 0)

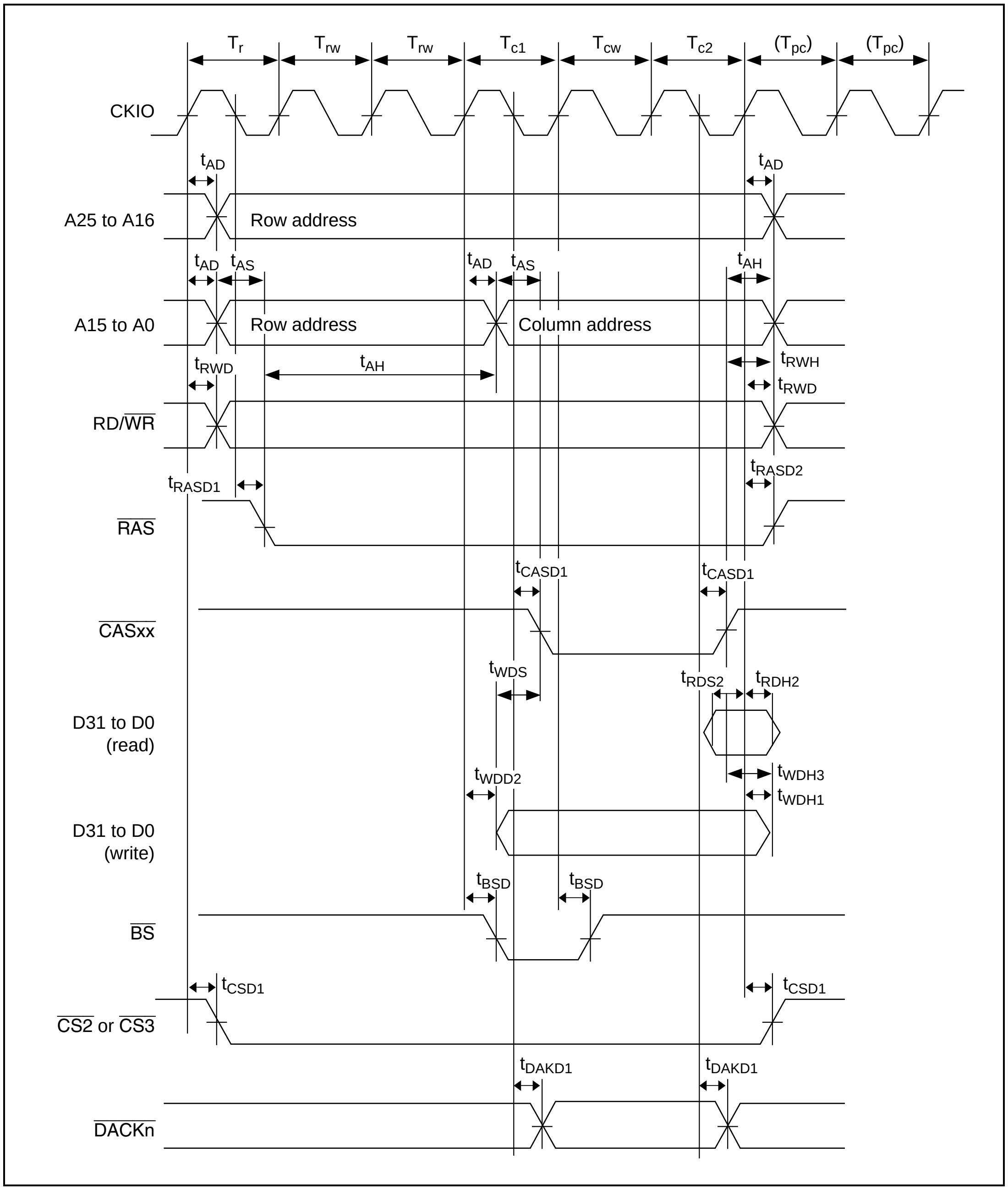


Figure 22.28 DRAM Bus Cycle (EDO mode, TRCD = 2, AnW = 2, TPC = 1)

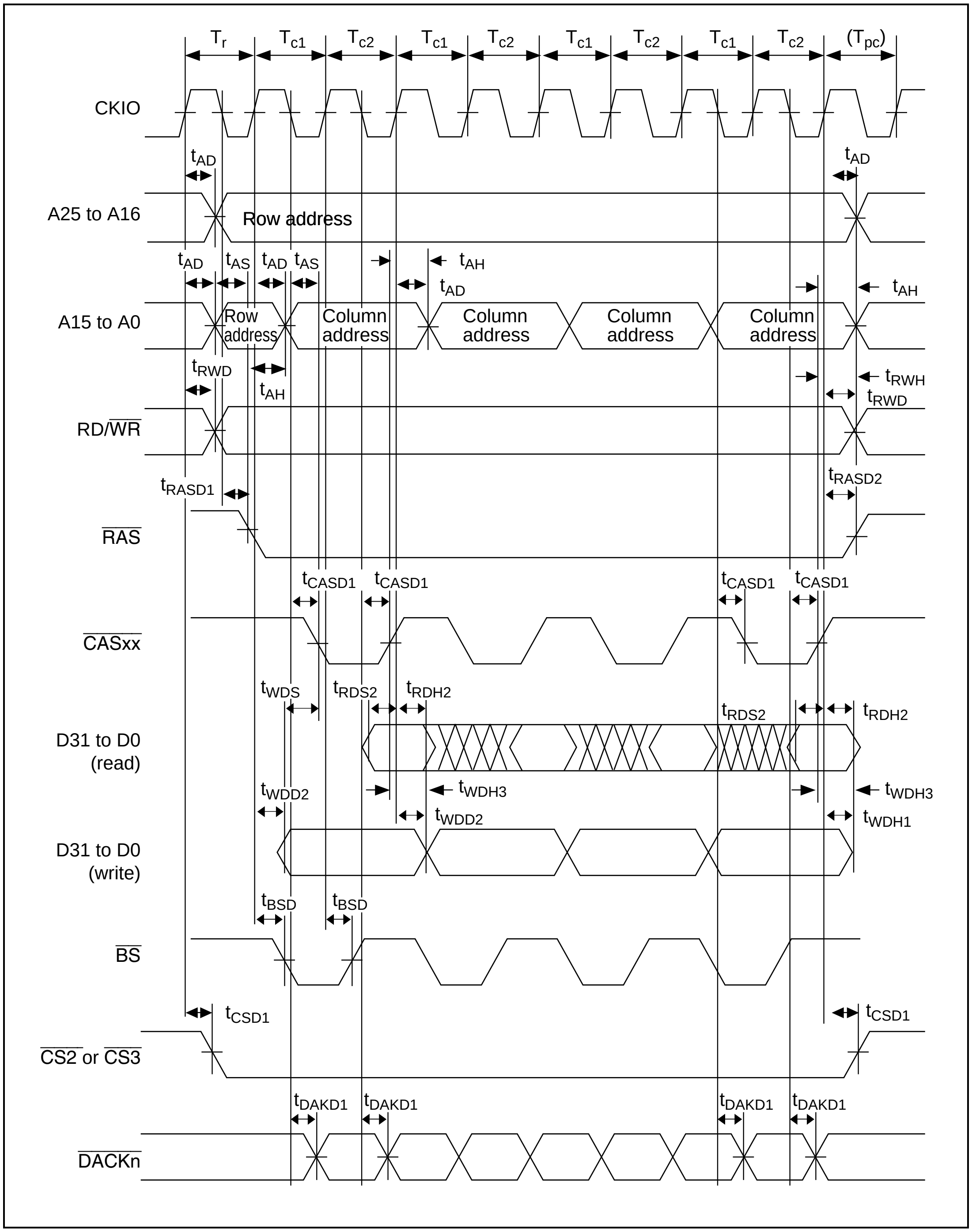


Figure 22.29 DRAM Burst Bus Cycle (EDO mode, TRCD = 0, AnW = 1, TPC = 0)

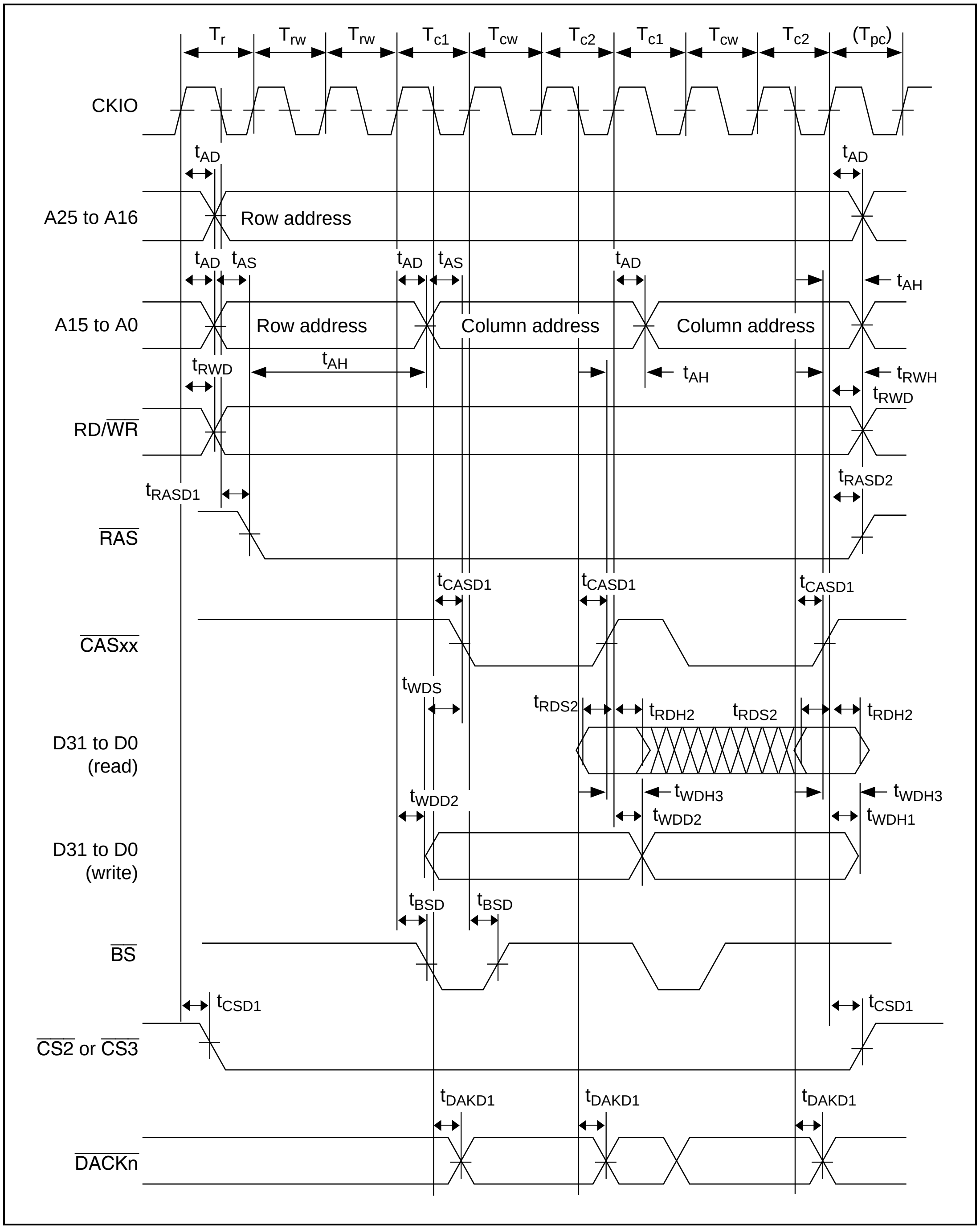


Figure 22.30 DRAM Burst Bus Cycle
(EDO mode, TRCD = 2, AnW = 2, TPC = 0)

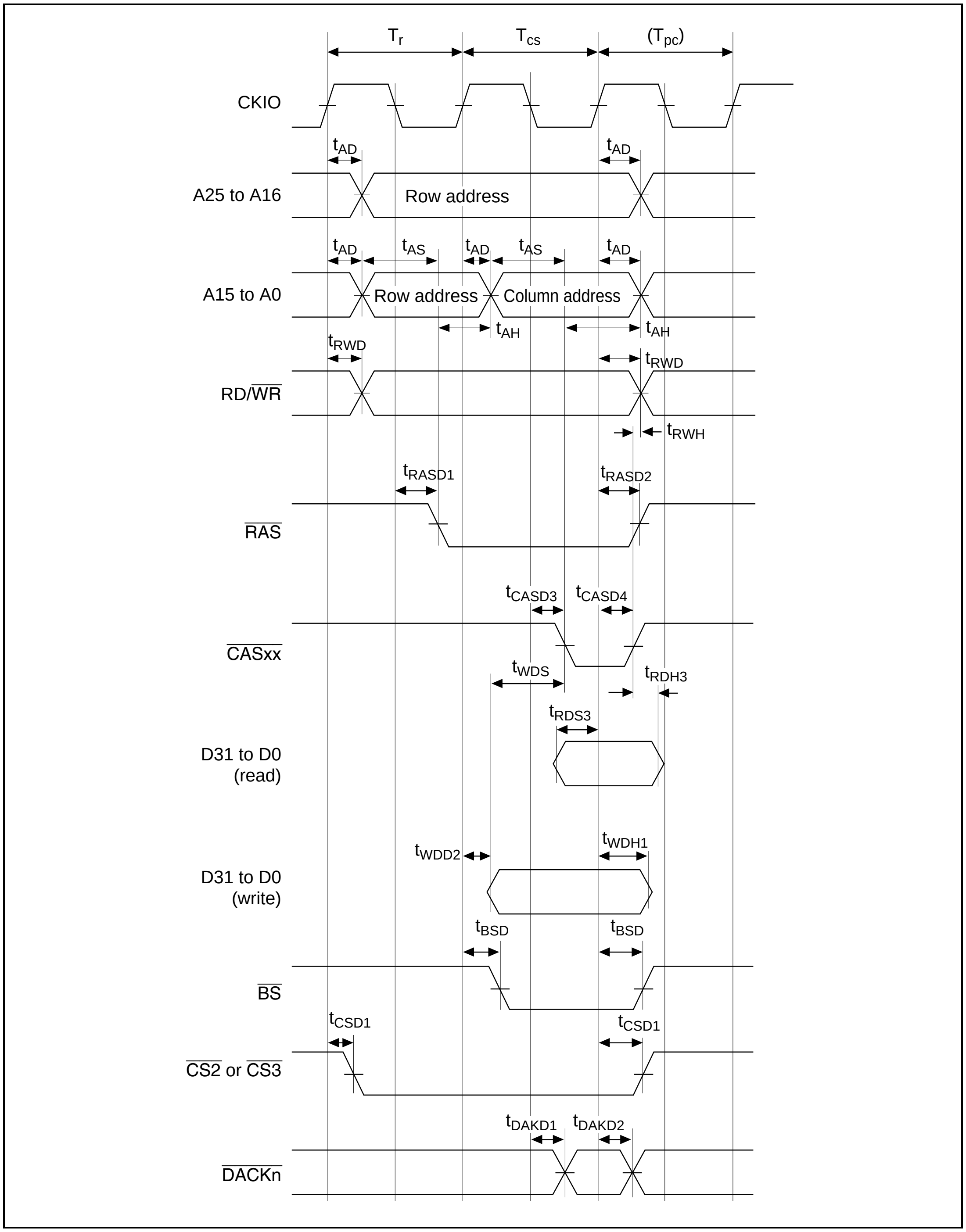


Figure 22.31 DRAM Short-Pitch Access Timing

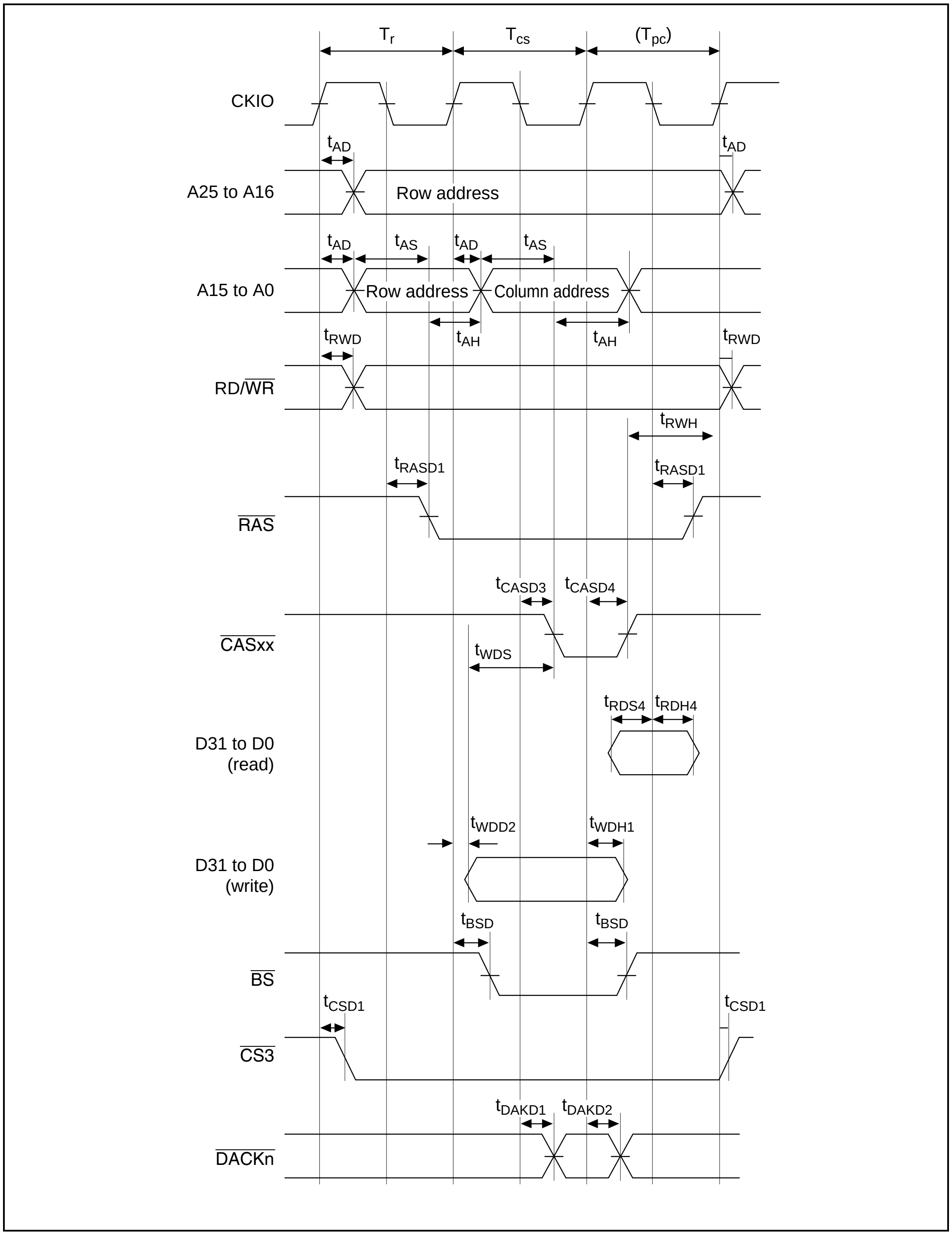


Figure 22.32 DRAM Short-Pitch Access Timing (EDO mode)

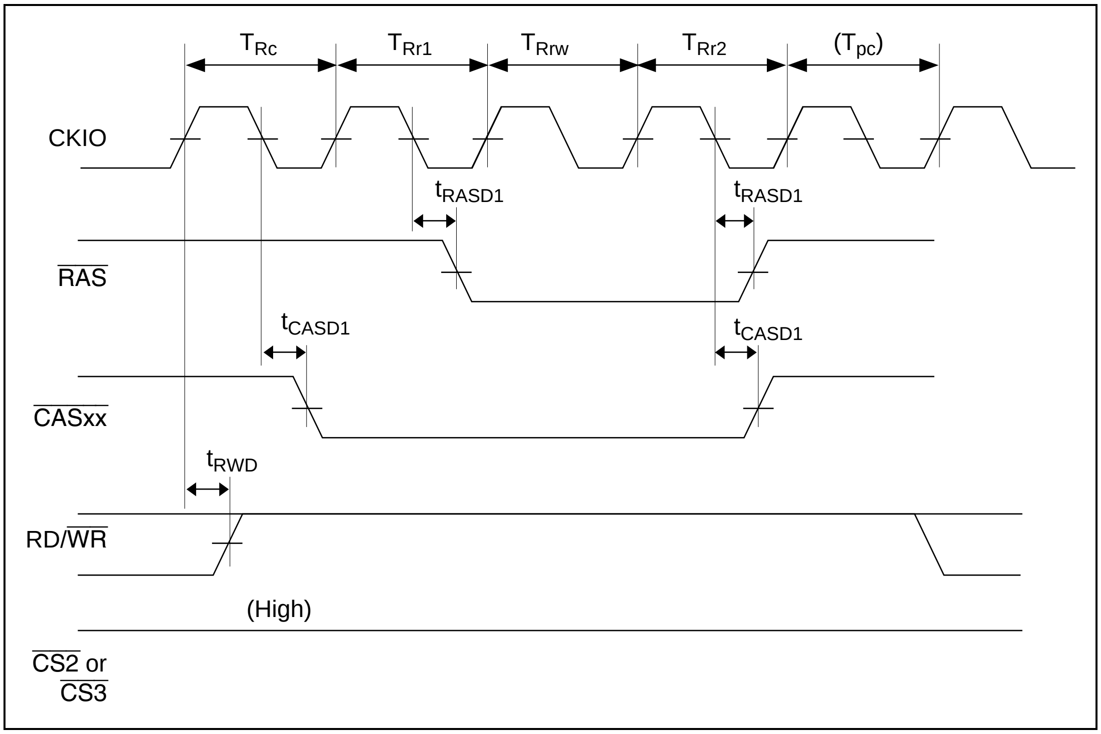


Figure 22.33 DRAM CAS-before-RAS Refresh Cycle (TRAS = 0, TPC = 0)

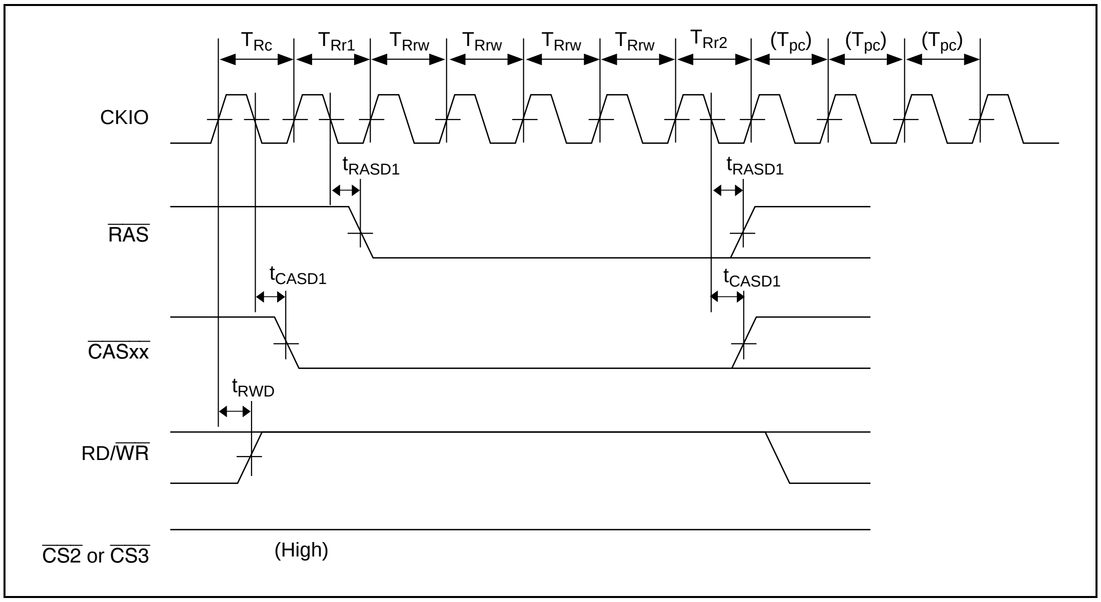


Figure 22.34 DRAM CAS-before-RAS Refresh Cycle (TRAS = 3, TPC = 2)

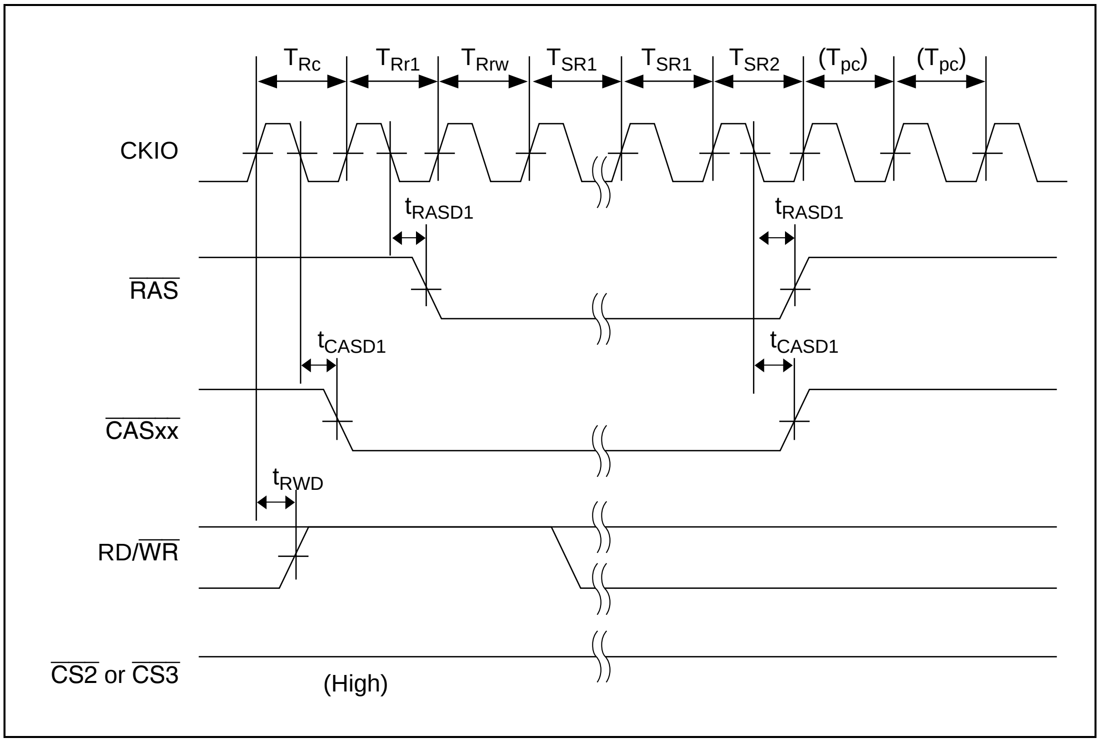


Figure 22.35 DRAM Self-refresh Cycle (TPC = 0)

22.3.6 Synchronous DRAM Timing

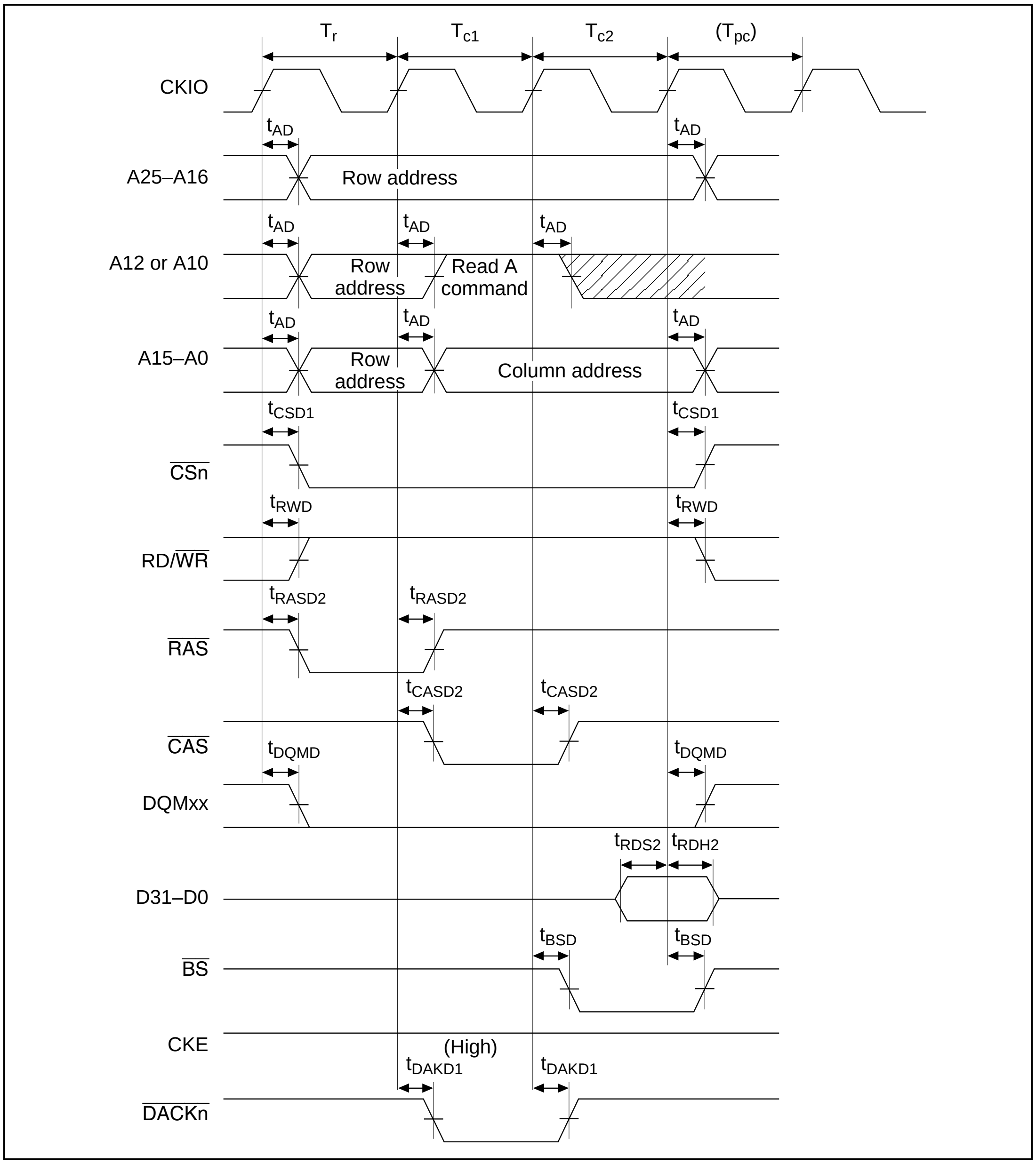


Figure 22.36 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0)

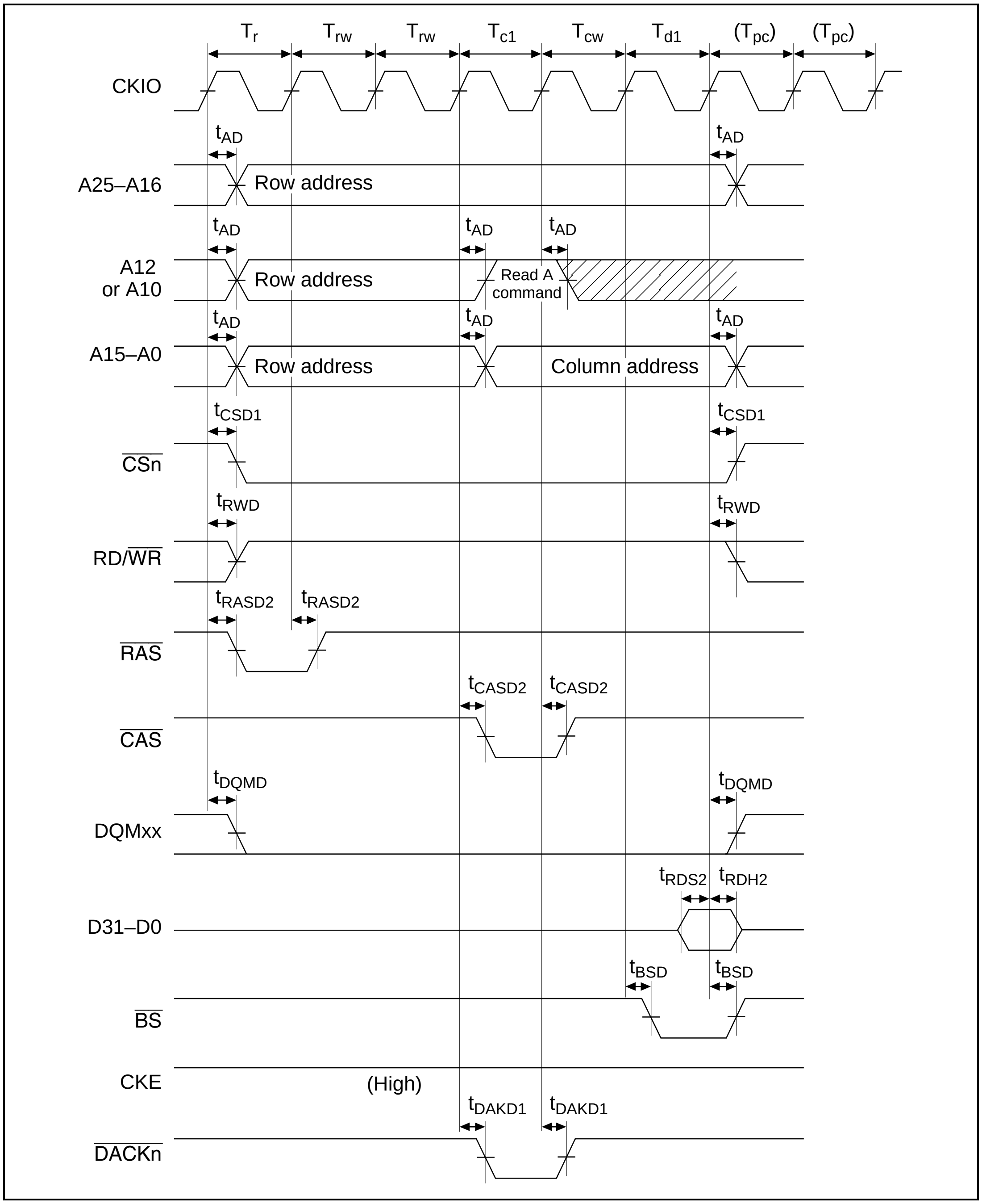


Figure 22.37 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1)

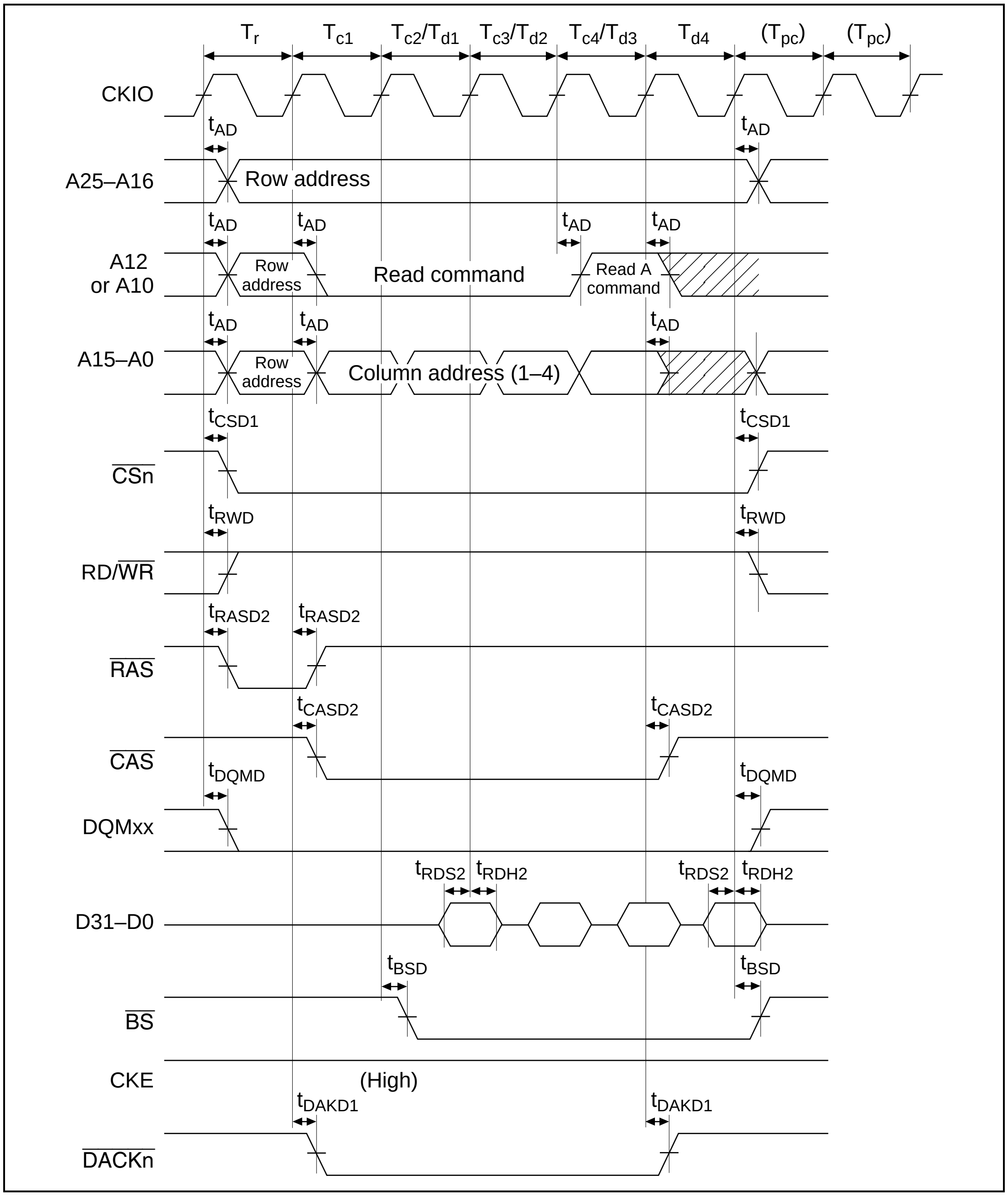


Figure 22.38 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

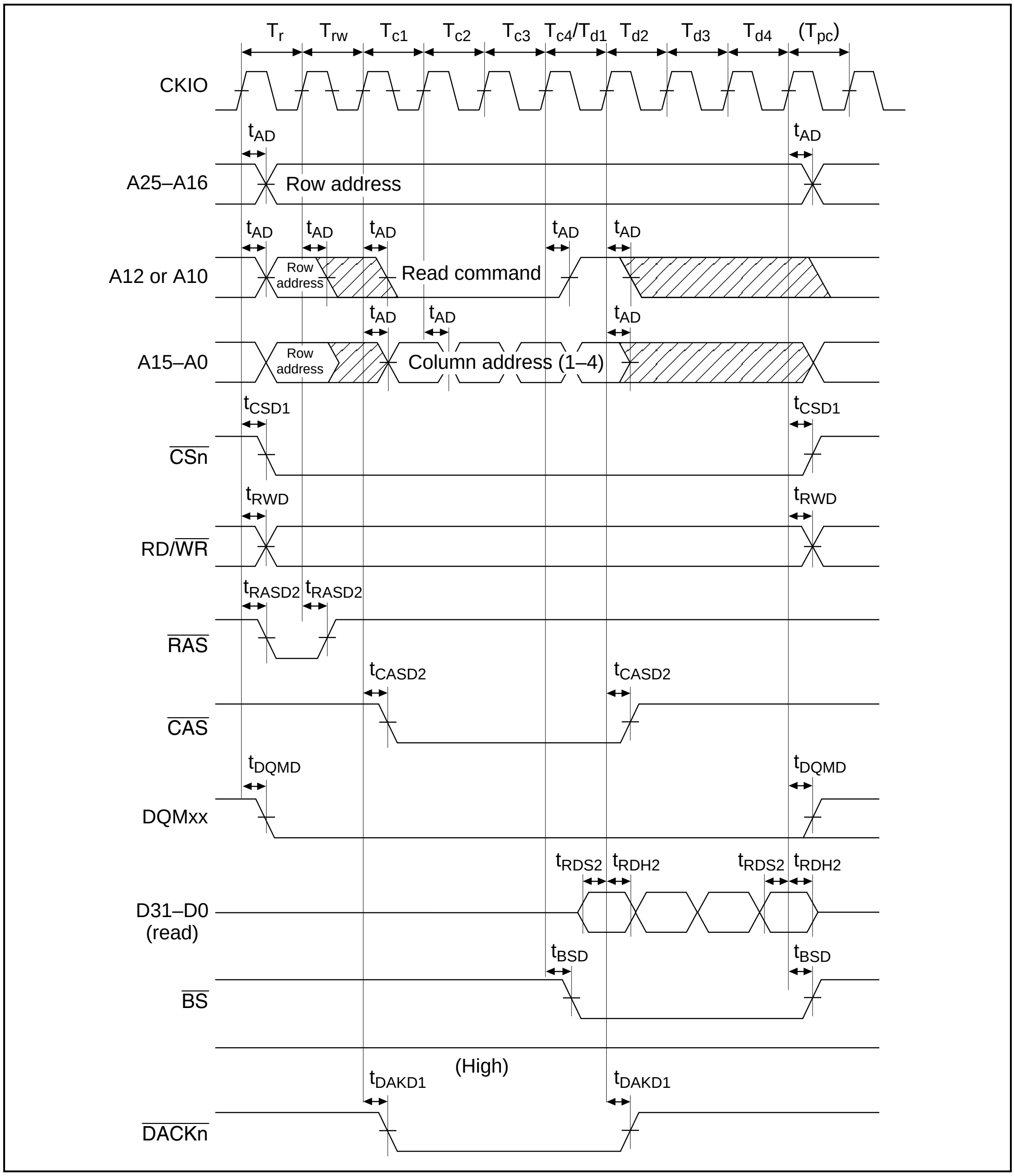


Figure 22.39 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

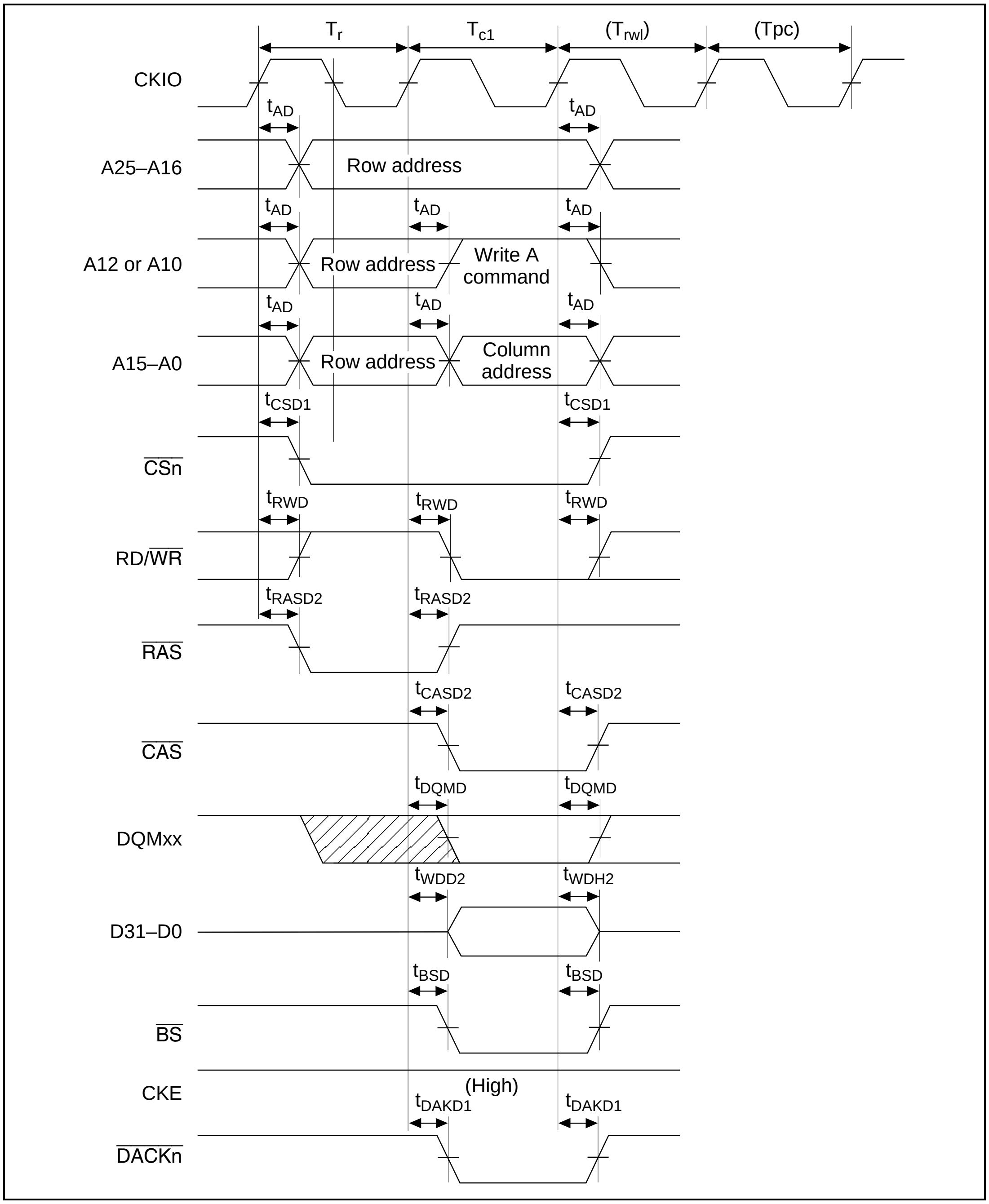


Figure 22.40 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0)

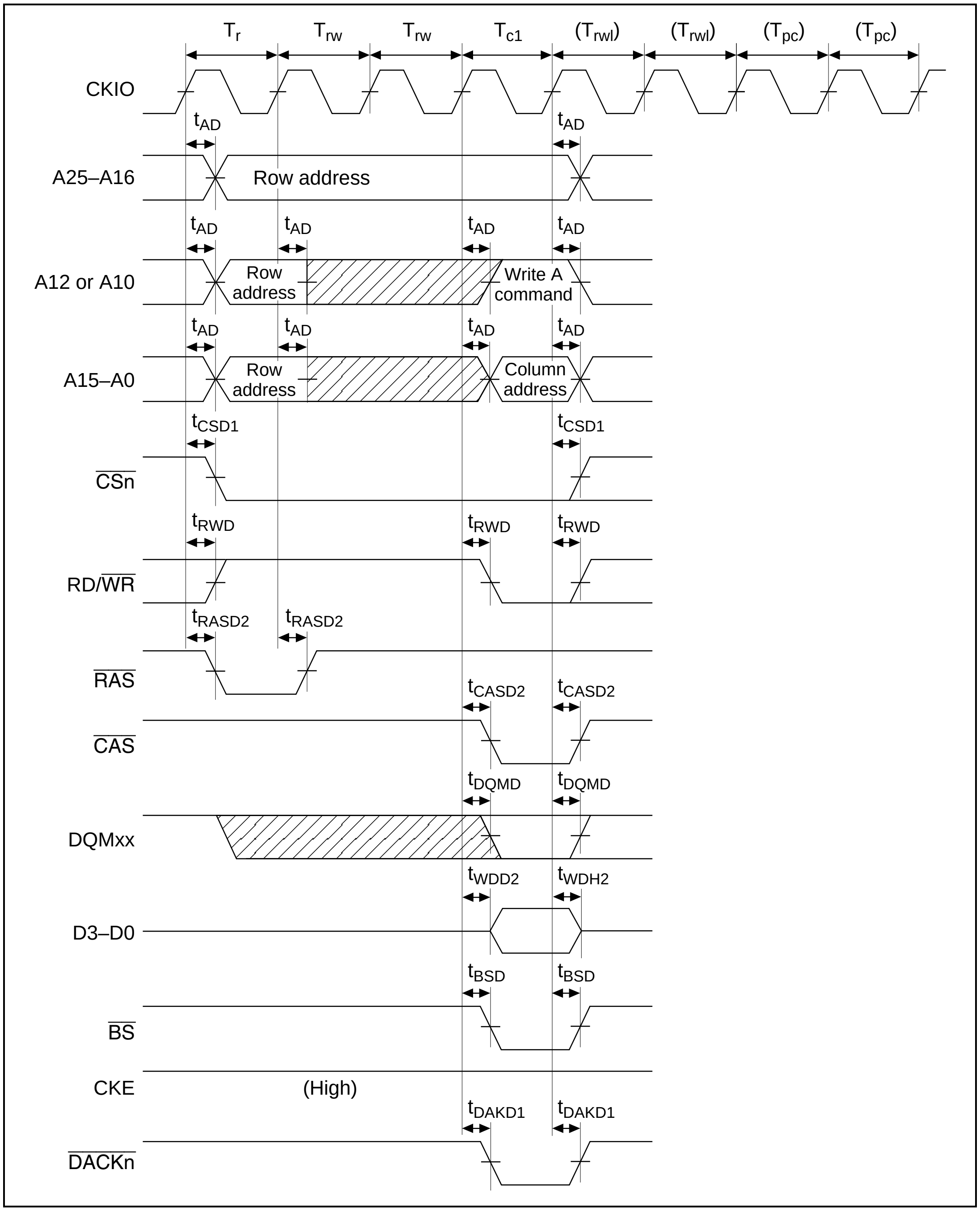


Figure 22.41 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1)

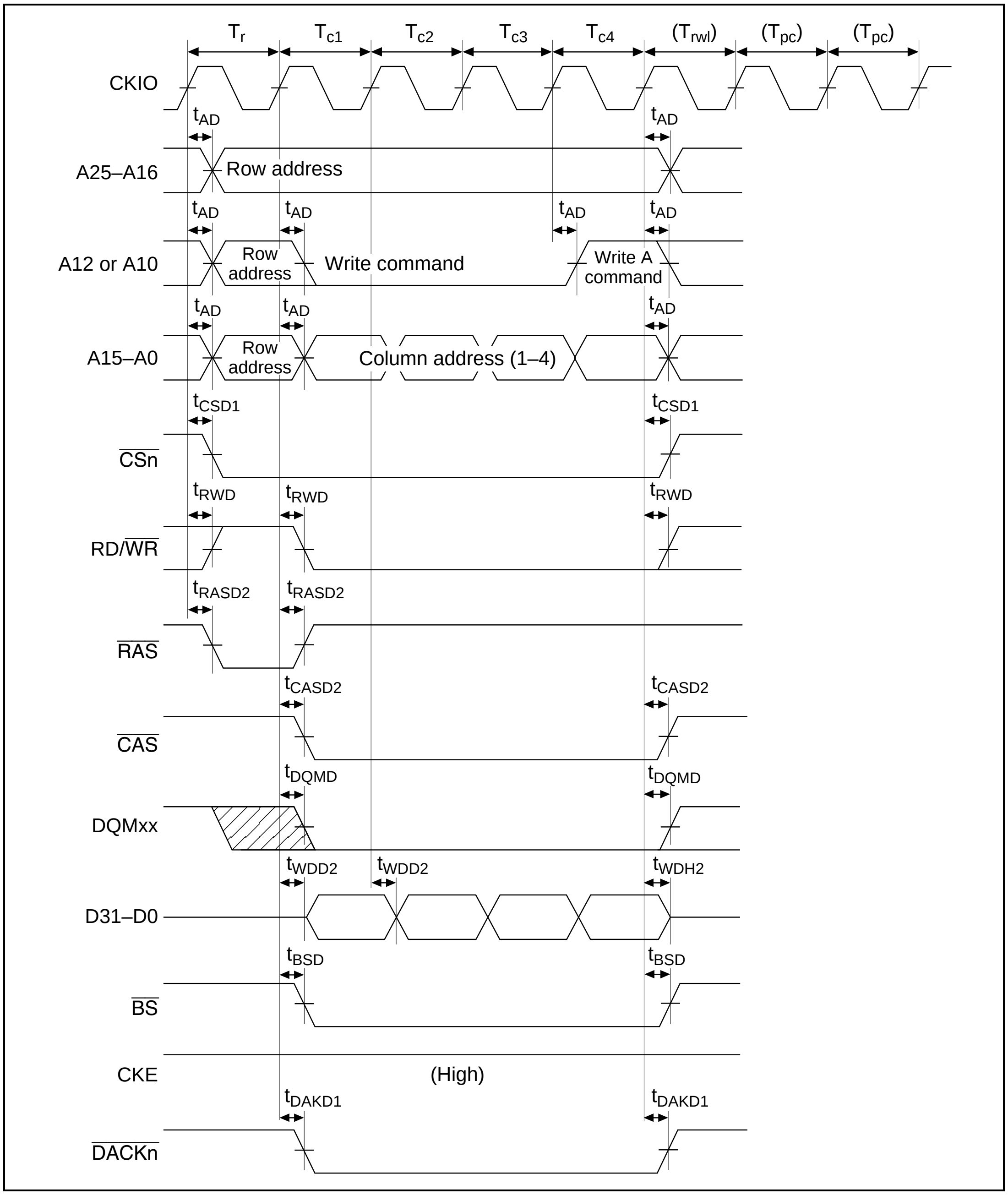


Figure 22.42 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

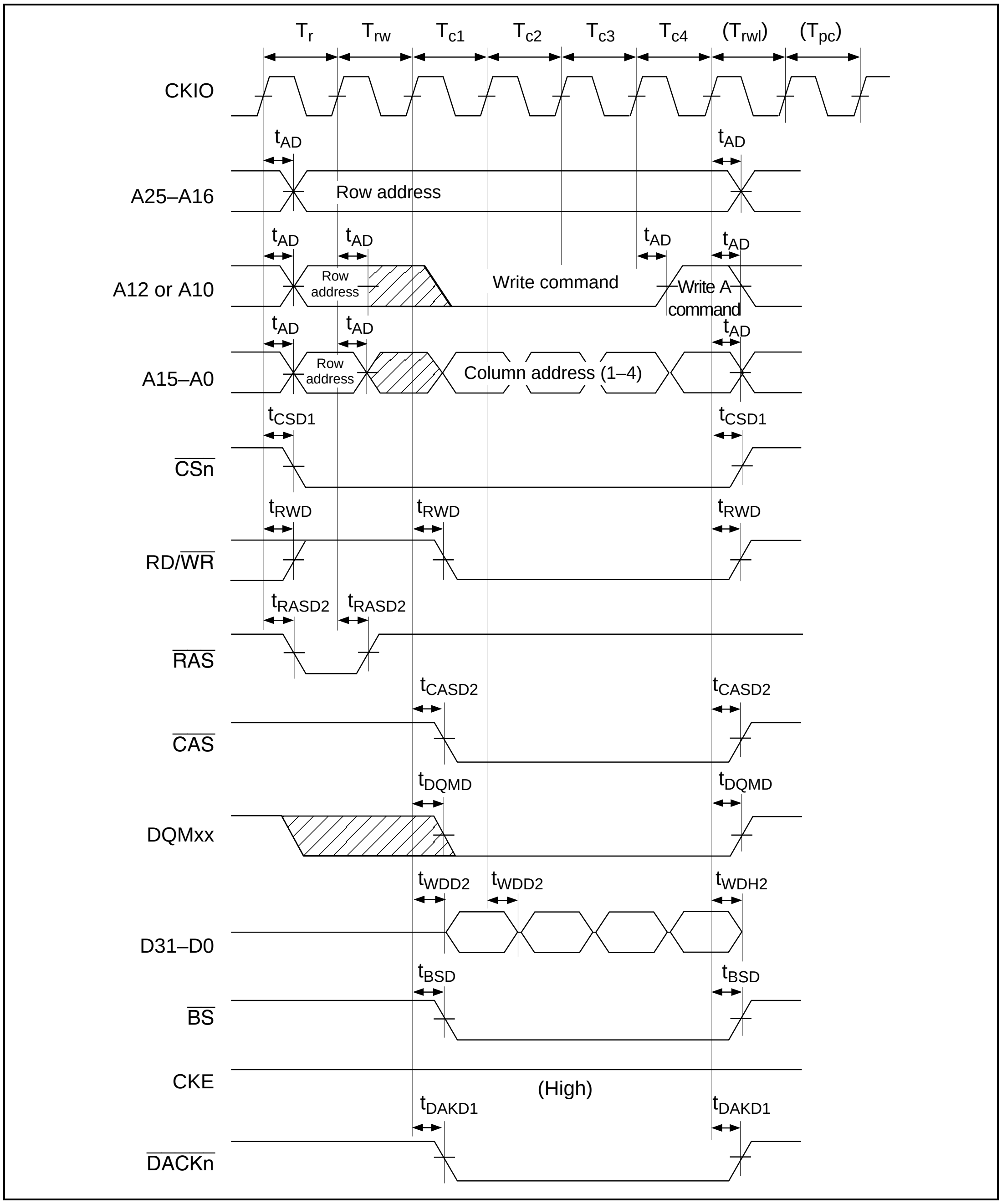


Figure 22.43 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

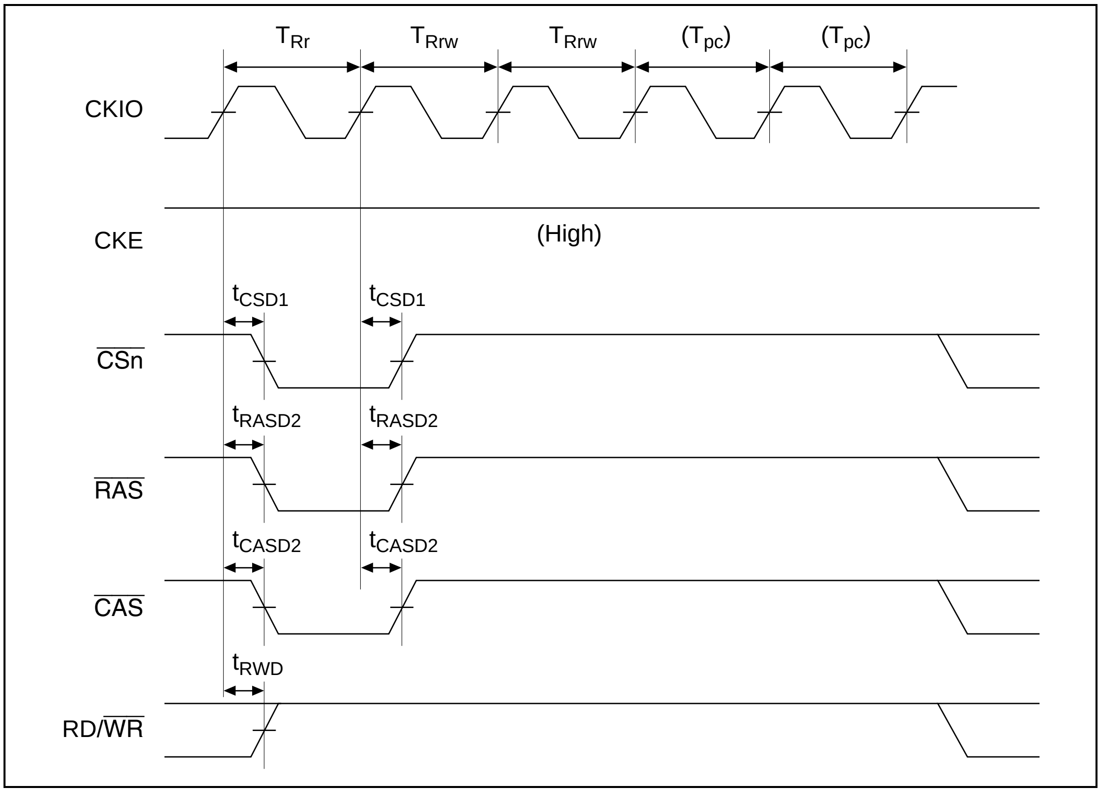


Figure 22.44 Synchronous DRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

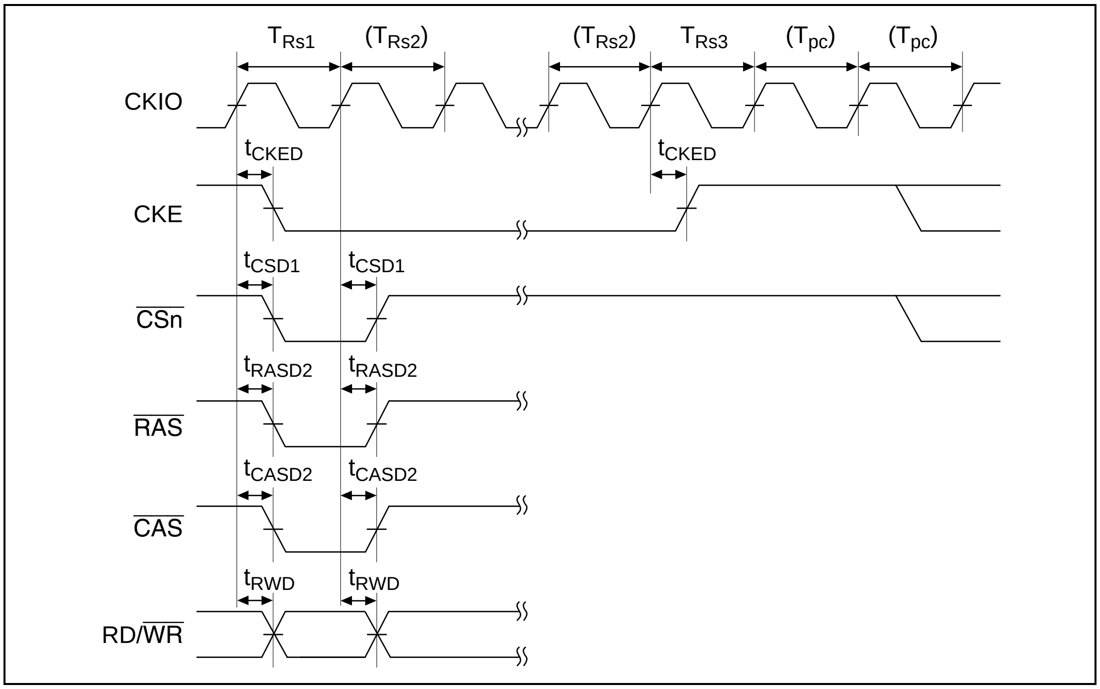


Figure 22.45 Synchronous DRAM Self-Refresh Cycle (TPC = 0)

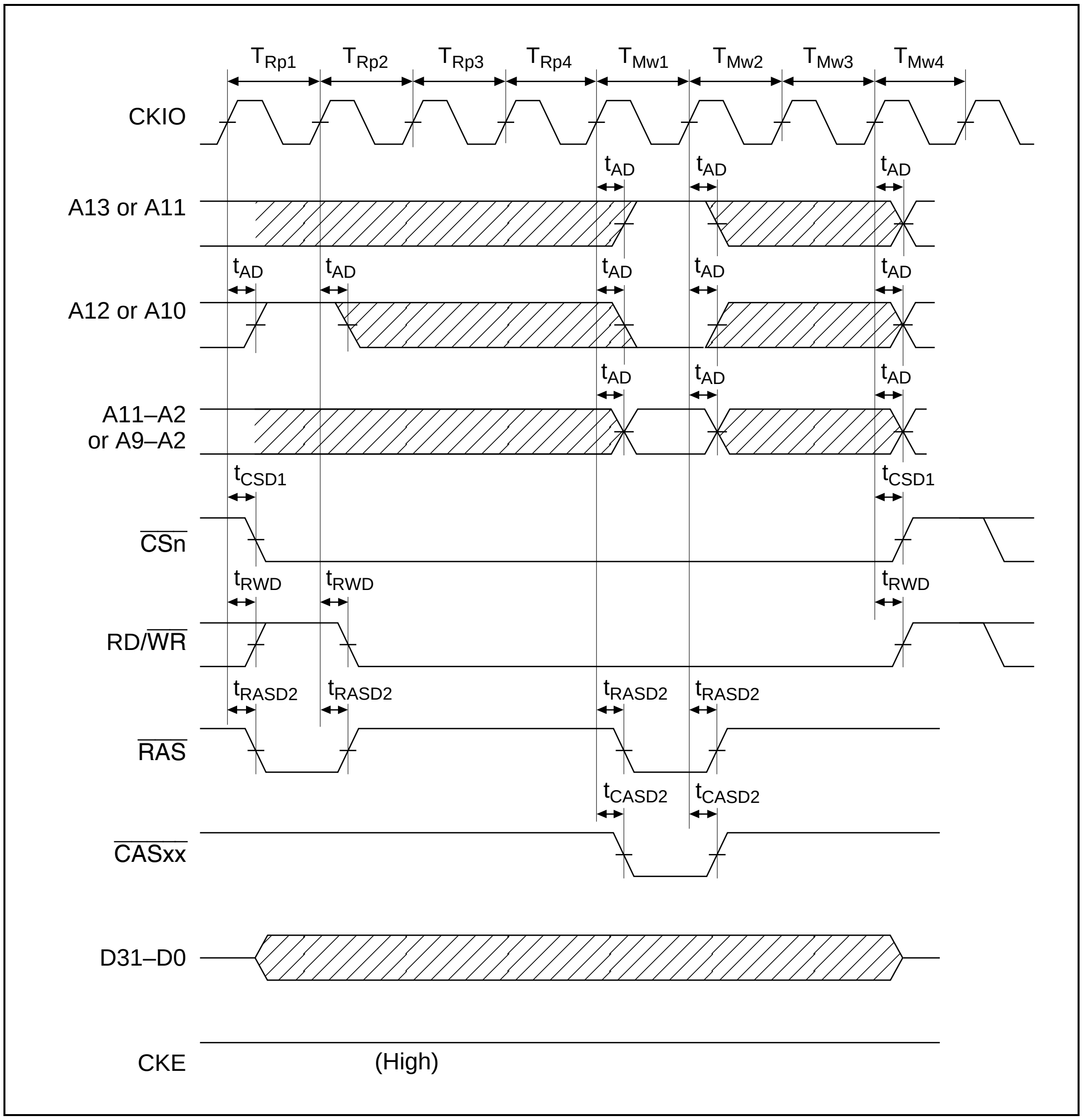


Figure 22.46 Synchronous DRAM Mode Register Write Cycle

22.3.7 PCMCIA Timing

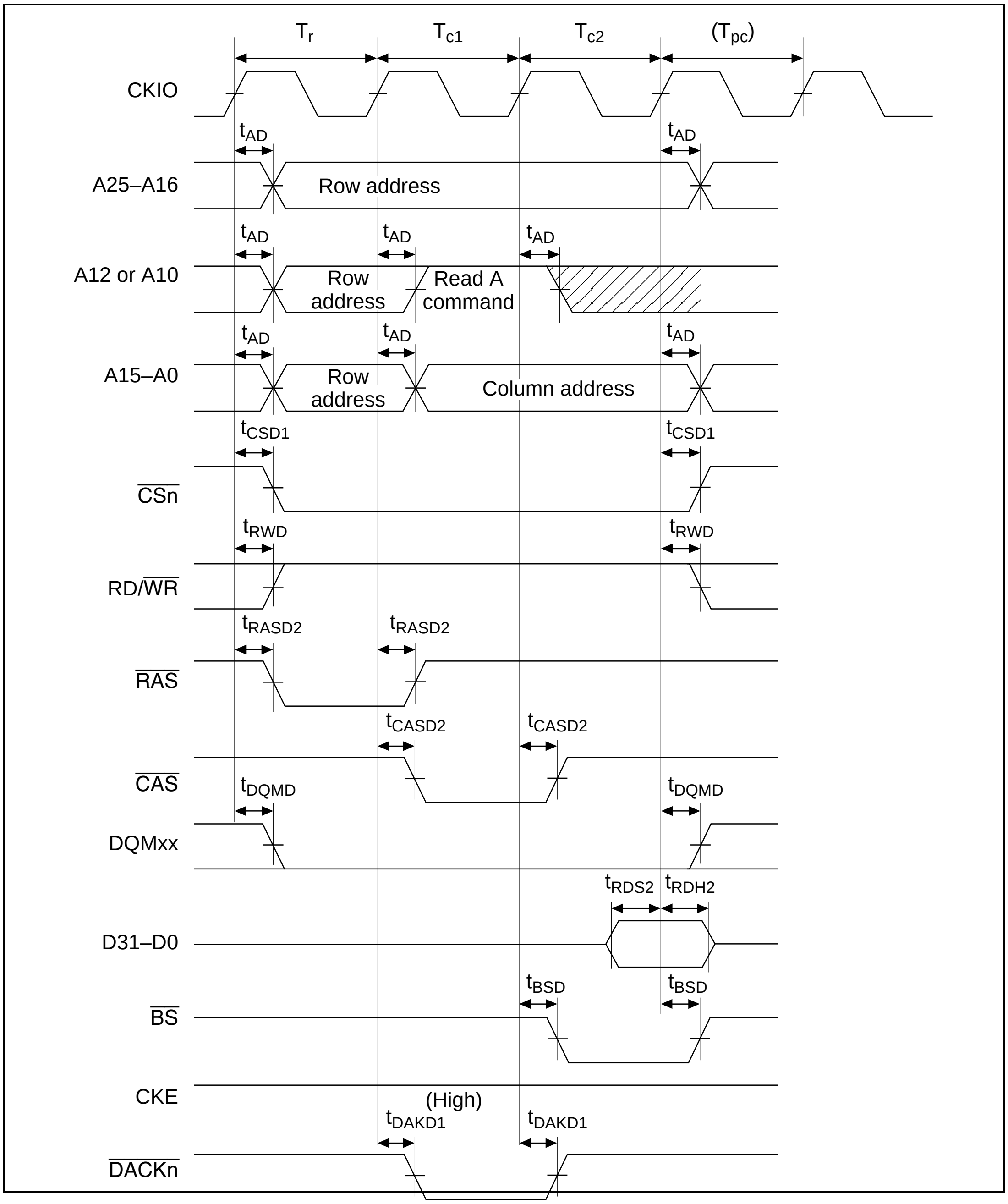


Figure 22.47 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

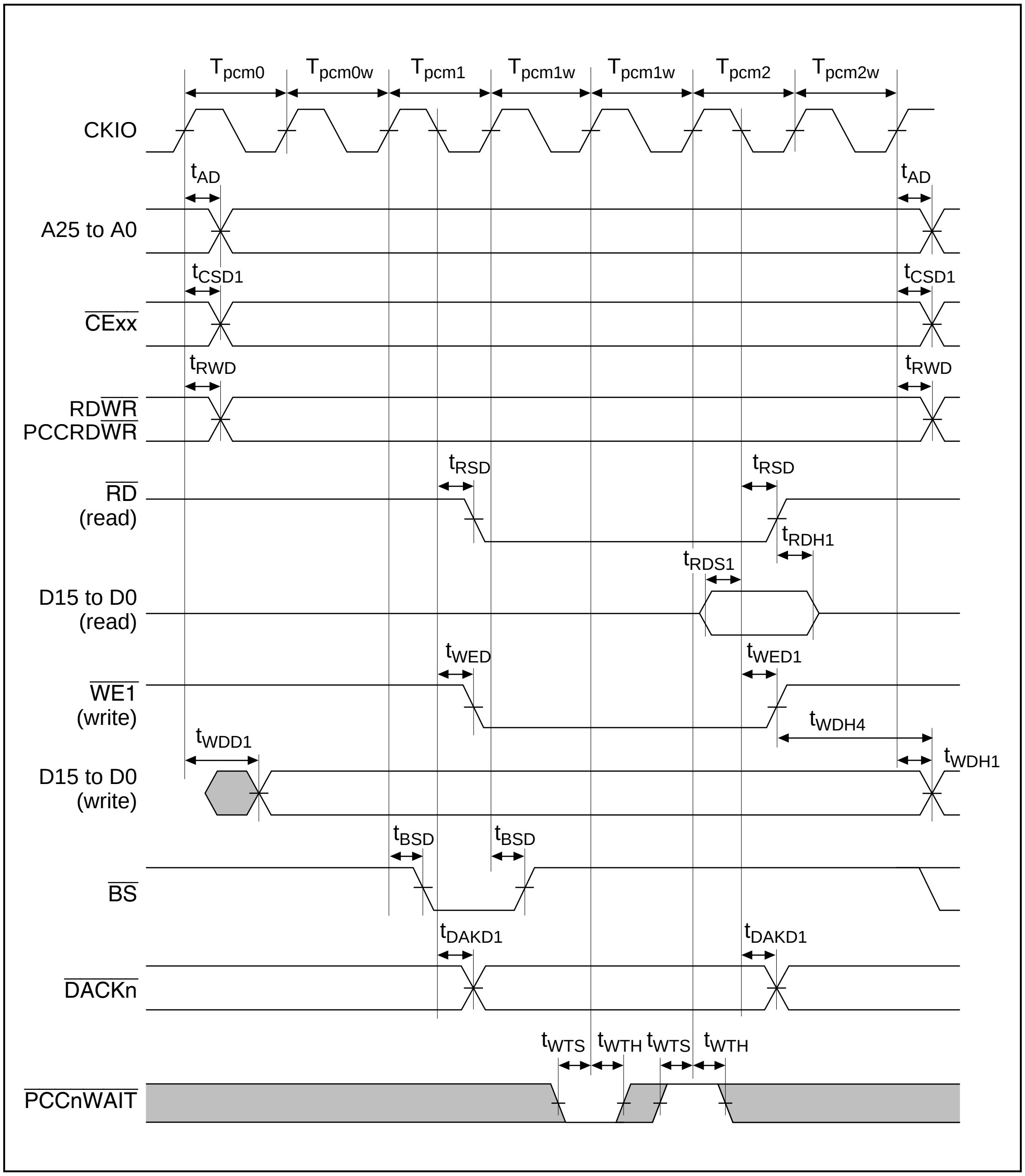


Figure 22.48 PCMCIA Memory Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait)

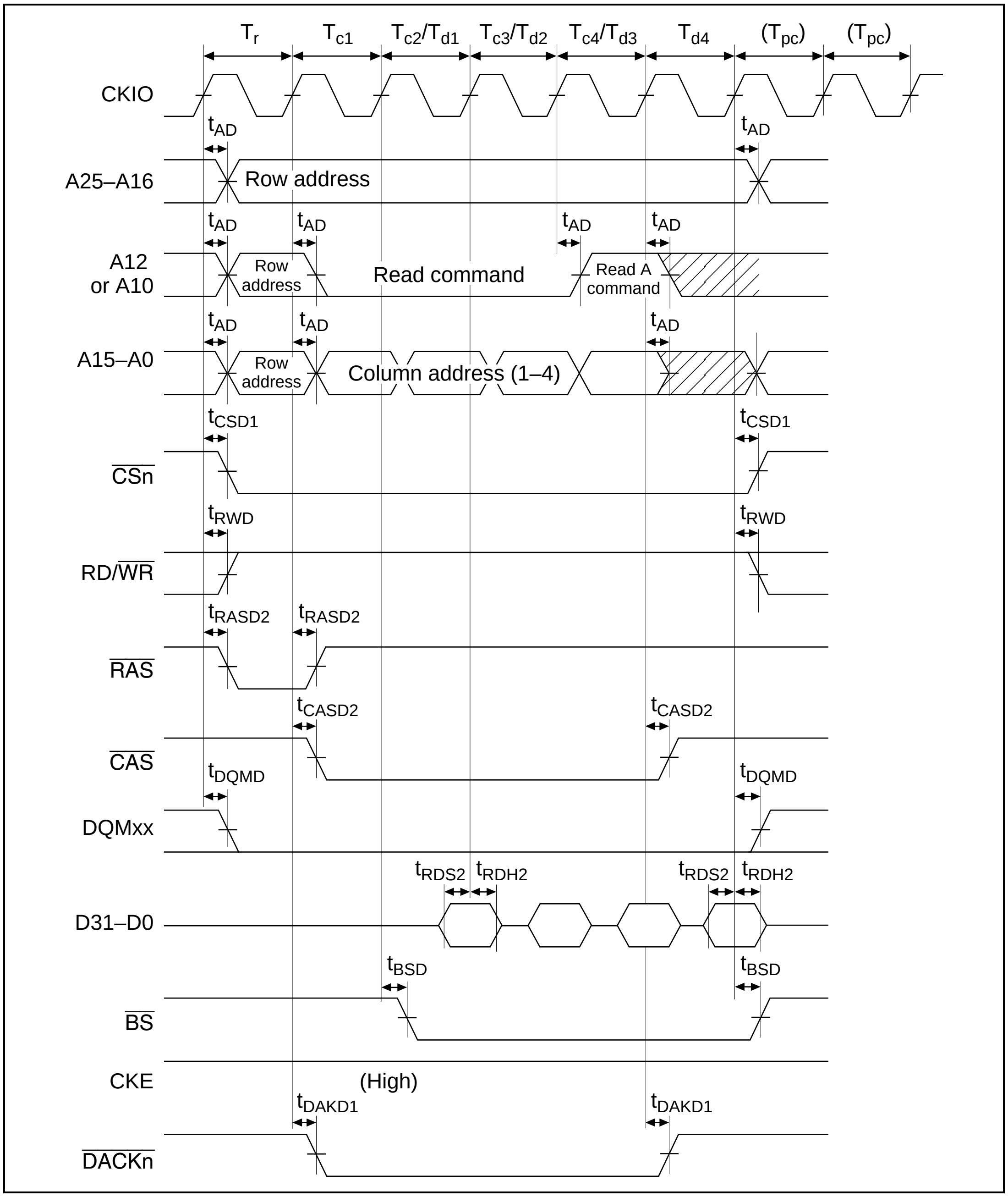


Figure 22.49 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)

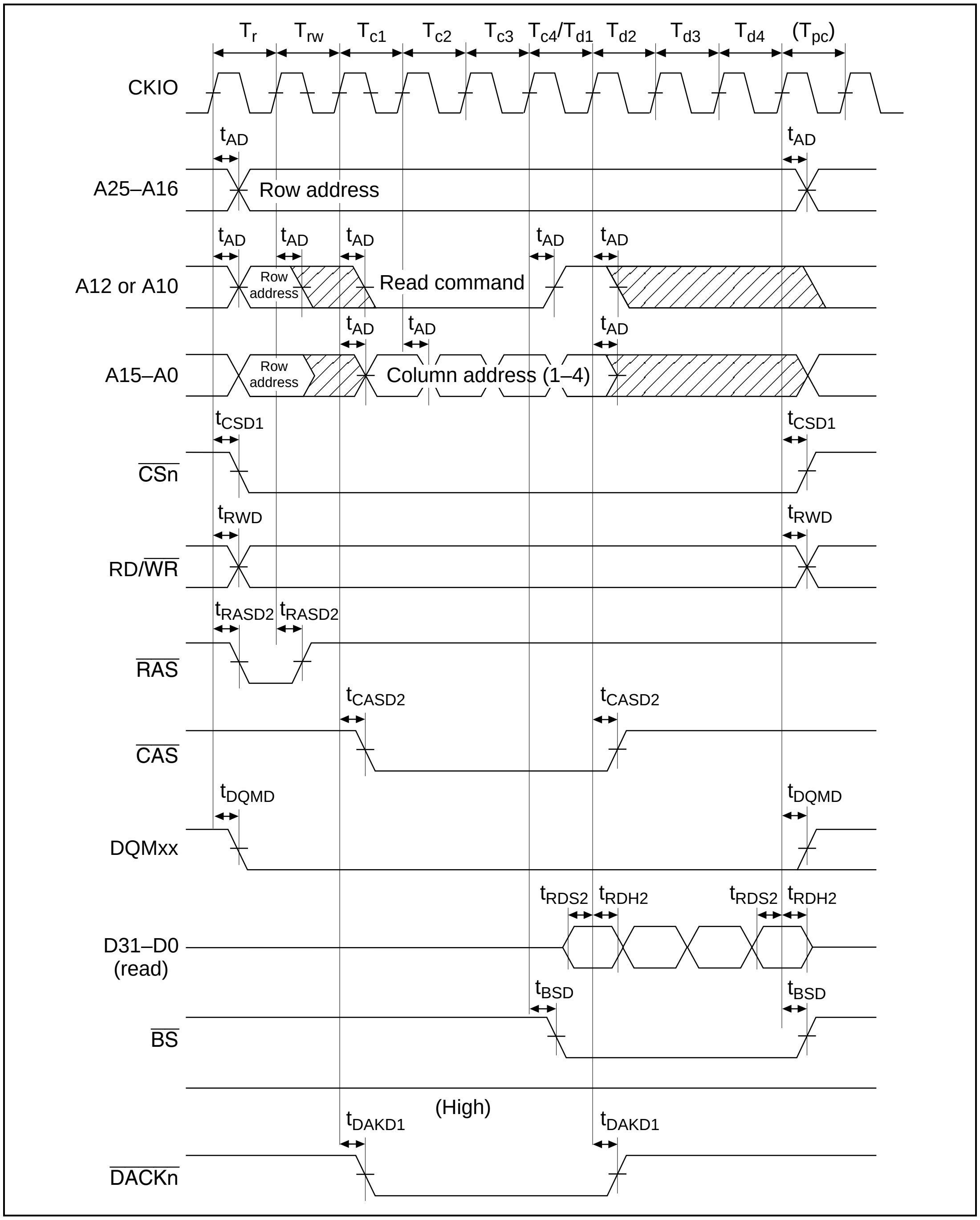


Figure 22.50 PCMCIA Memory Bus Cycle
(Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3)

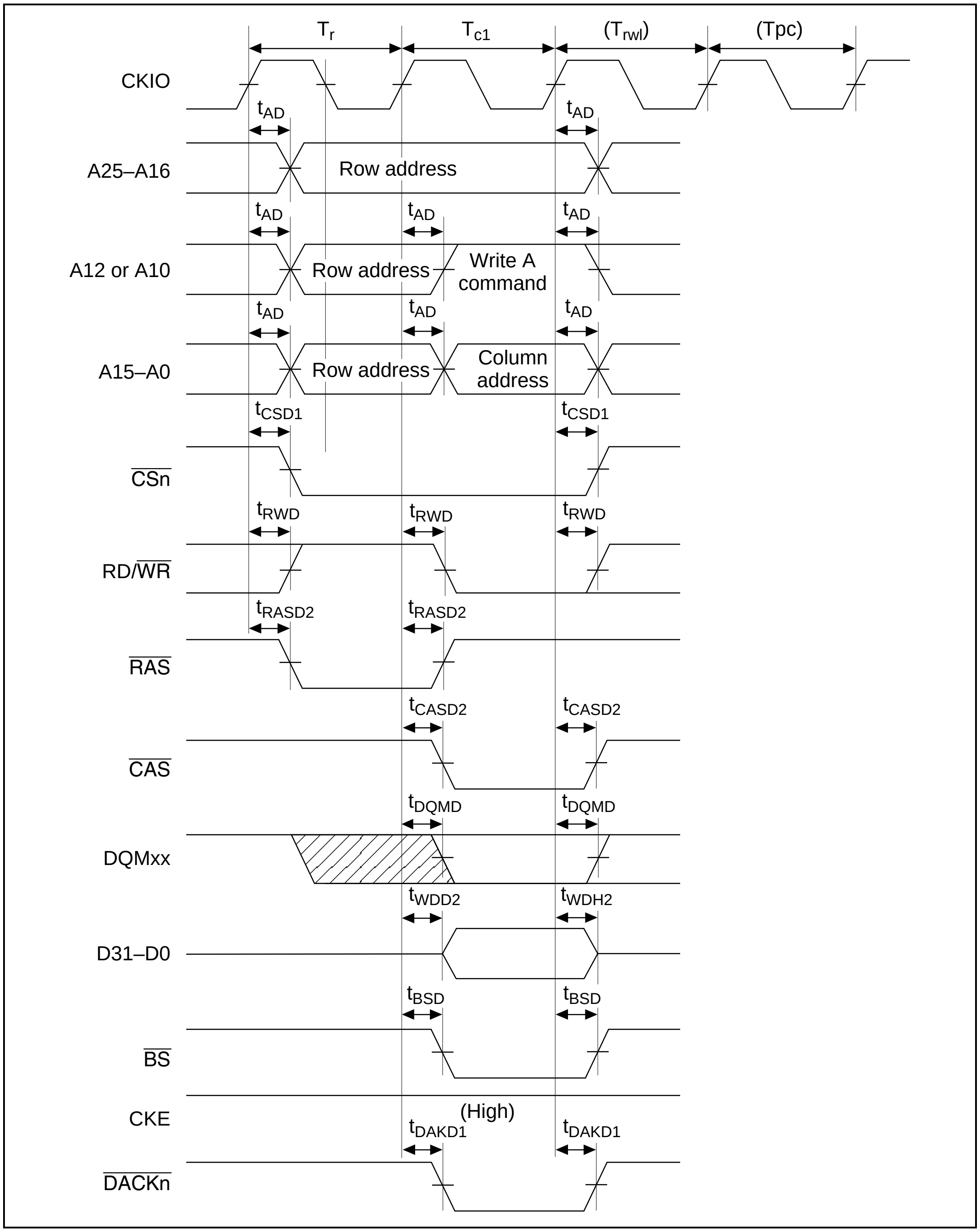


Figure 22.51 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

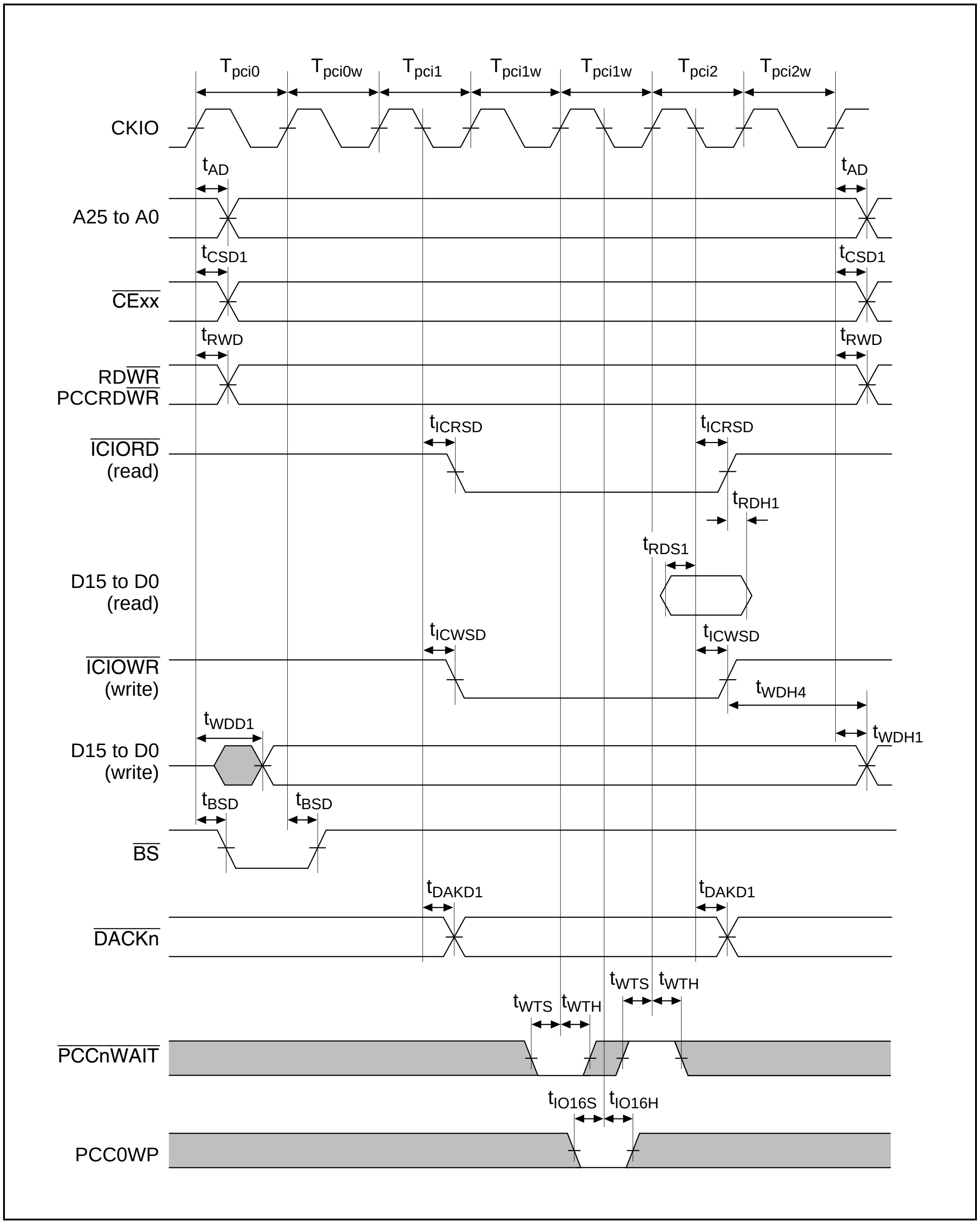


Figure 22.52 PCMCIA I/O Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait)

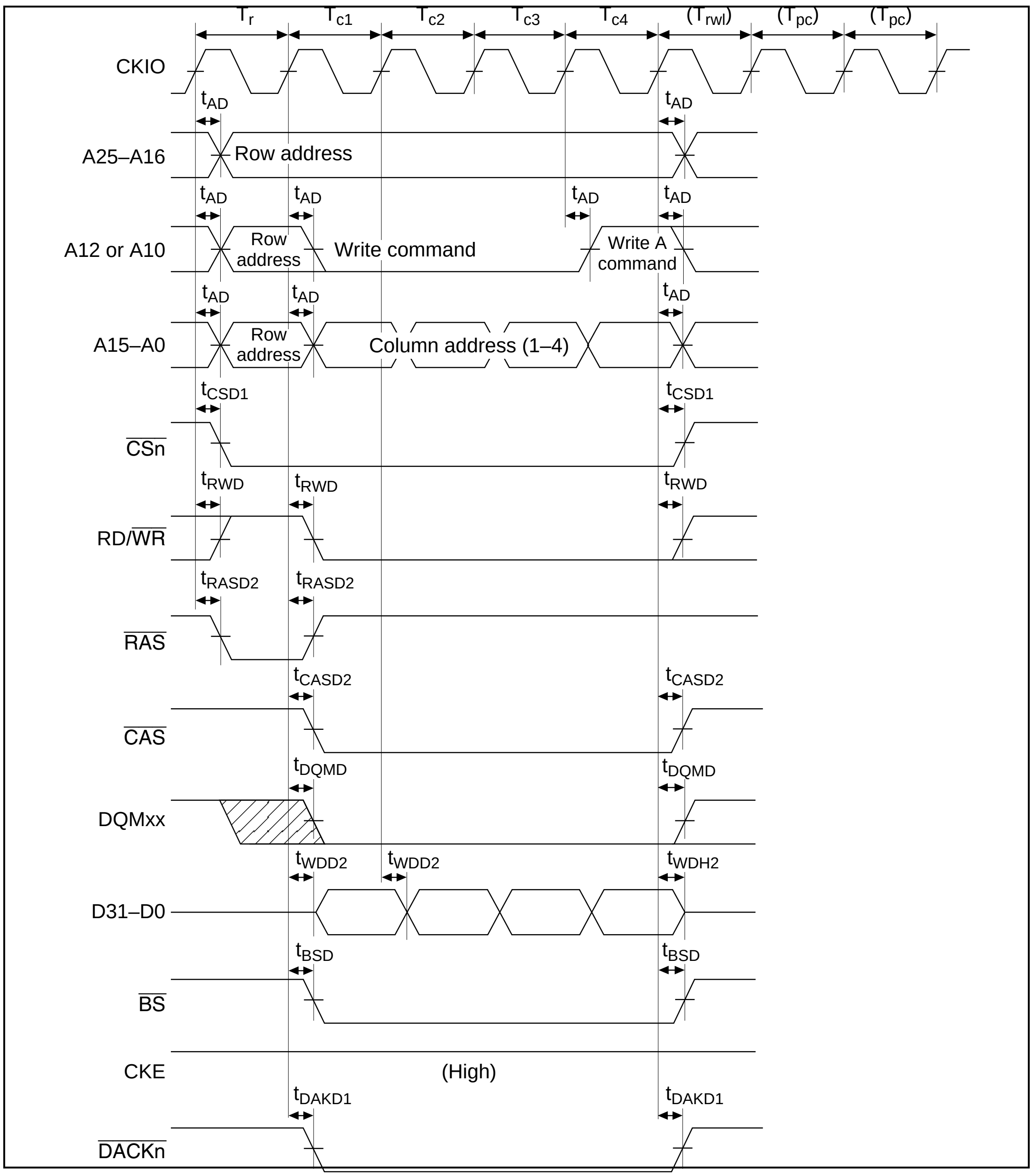


Figure 22.53 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, One Wait, Bus Sizing)

22.3.8 Peripheral Module Signal Timing

Table 22.8 Peripheral Module Signal Timing
(Vcc = 3.3 ± 0.3 V, AVcc = 3.3 ± 0.3 V, AVcc = Vcc ± 0.3 V, Ta = –20 to 75°C)

				–40			
Module	**Item**		**Symbol**	**Min**	**Max**	**Unit**	**Figure**
TMU, RTC	Timer input setup time		tTCLKS	15	—	ns	22.54
	Timer clock input setup time		tTCKS	15	—		22.55
	Timer clock pulse width	Edge specification	tTCKWH	1.5	—	tcyc	
		Both edge specification	tTCKWL	2.5	—		
	Oscillation settling time		tROSC	—	3	S	22.56
SCI	Input clock cycle	Asynchronization	tSCYC	4	—	tcyc	22.57
		Clocked synchronization		6	—		22.58
	Input clock rise time		tSCKR	—	1.5		22.57
	Input clock fall time		tSCKF	—	1.5		
	Input clock pulse width		tSCKW	0.4	0.6	tscyc	
	Transmission data delay time		tTXD	—	100	ns	22.58
	Receive data setup time (clocked synchronization)		tRXS	100	—		
	Receive data hold time (clocked synchronization)		tRXH	100	—		
	RTS delay time		tRTSD	—	100		
	CTS setup time (clocked synchronization)		tCTSS	100	—		
	CTS hold time (clocked synchronization)		tCTSH	100	—		
Port	Output data delay time		tPORTD	—	17	ns	22.59
	Input data setup time		tPORTS1	15	—		
	Input data hold time		tPORTH1	8	—		
	input data setup time		tPORTS2	17	—		
	Input data hold time		tPORTH2	10	—		
DMAC	$\overline{\text{DREQ}}$ setup time		tDRQS	12	—	ns	22.60
	$\overline{\text{DREQ}}$ hold time		tDRQH	8	—		
	$\overline{\text{DRAK}}$ delay time		tDRAKD	14	—		22.61

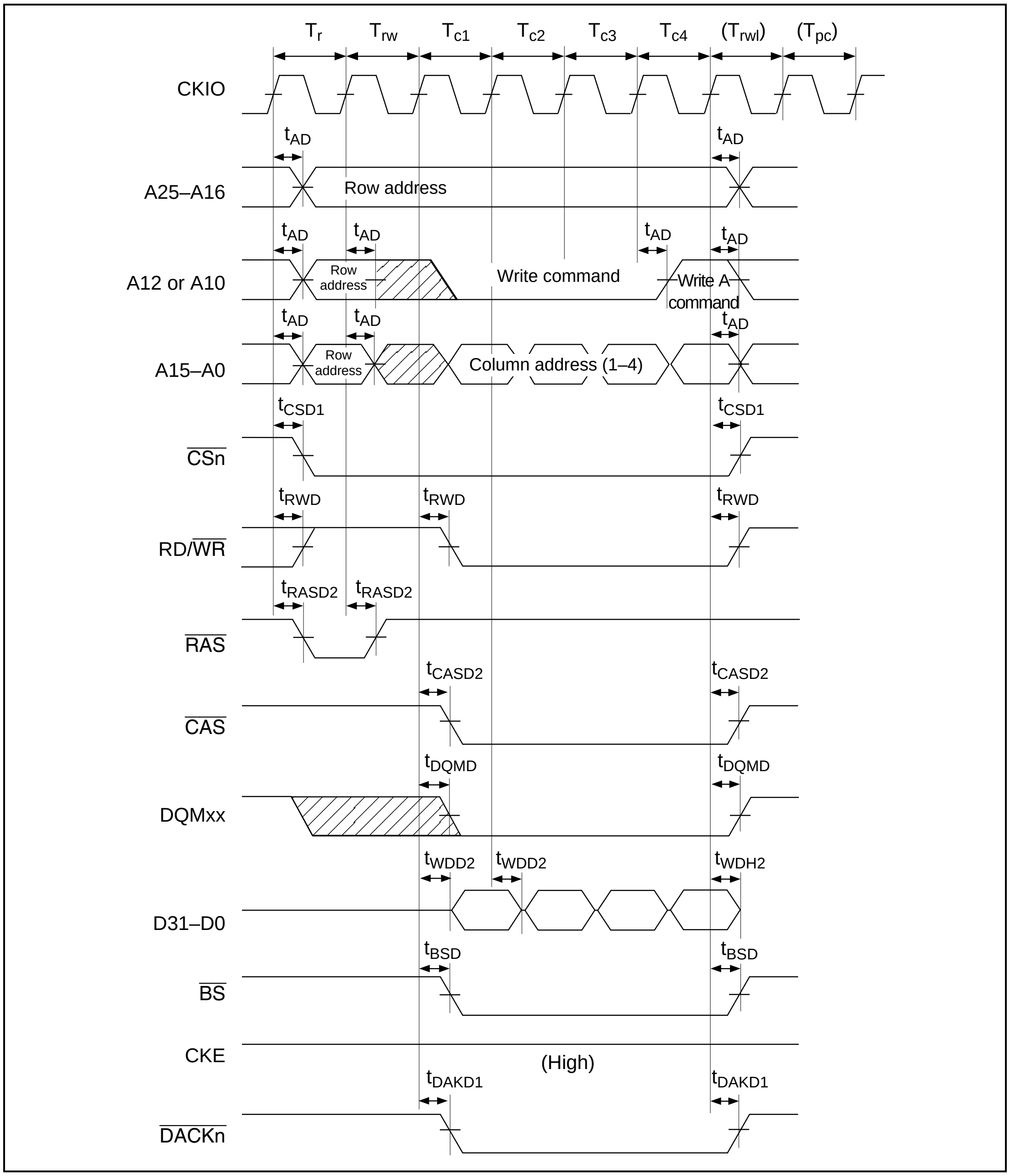


Figure 22.54 TCLK Input Timing

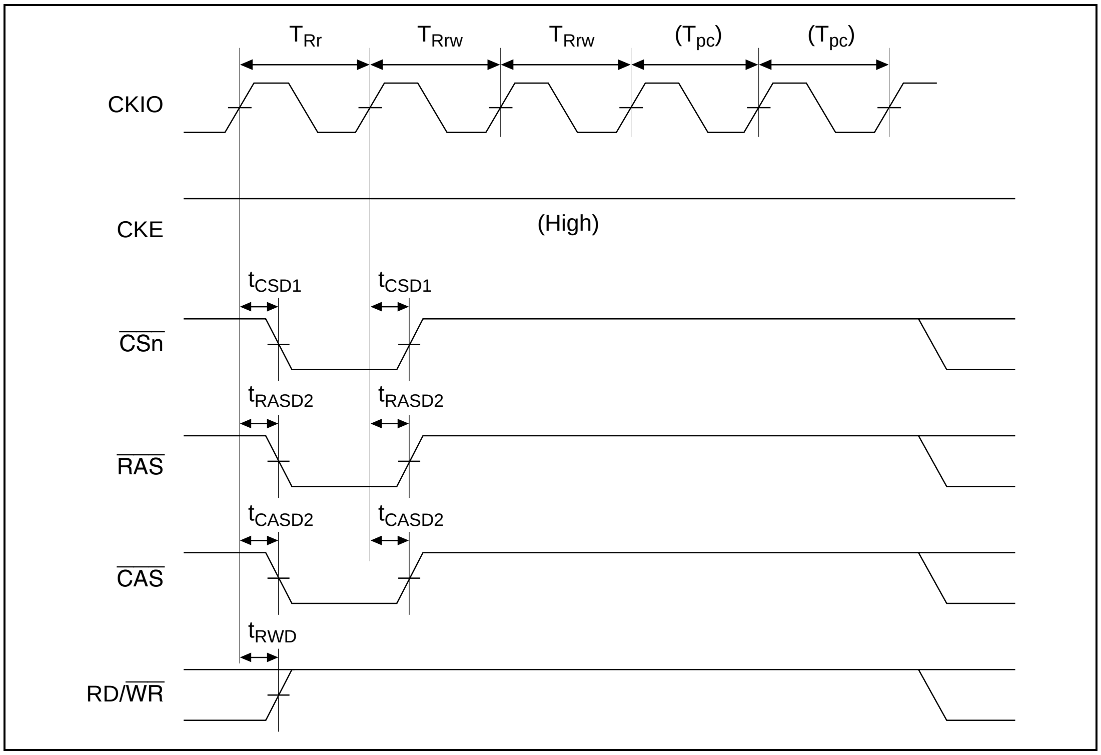


Figure 22.55 TCLK Clock Input Timing

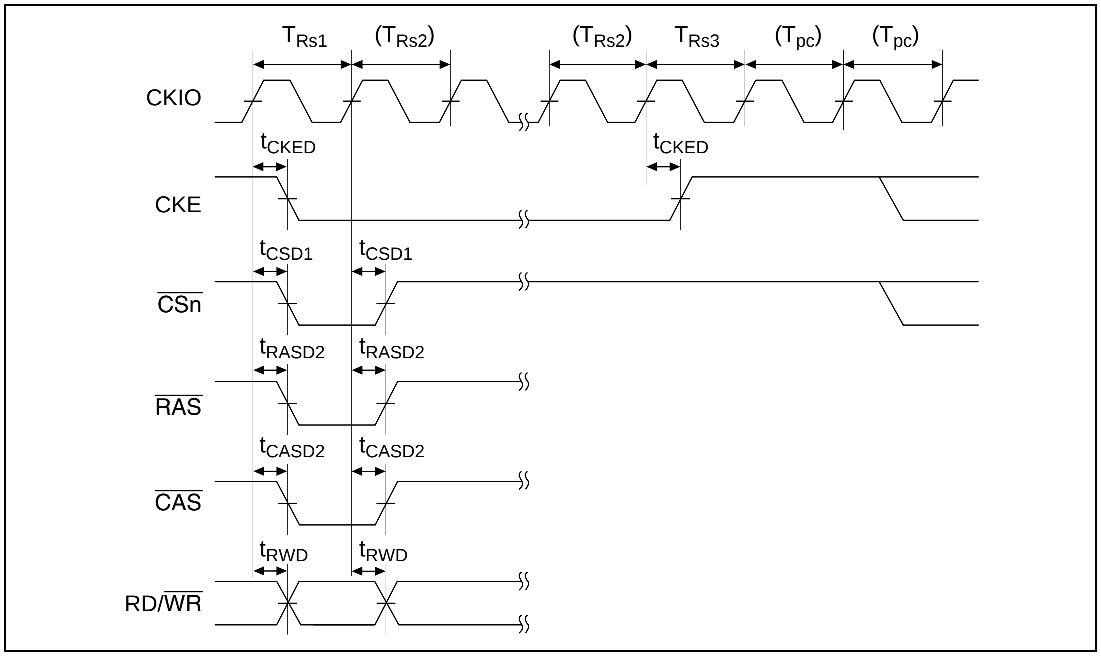


Figure 22.56 Oscillation Settling Time at RTC Crystal Oscillator Power-on

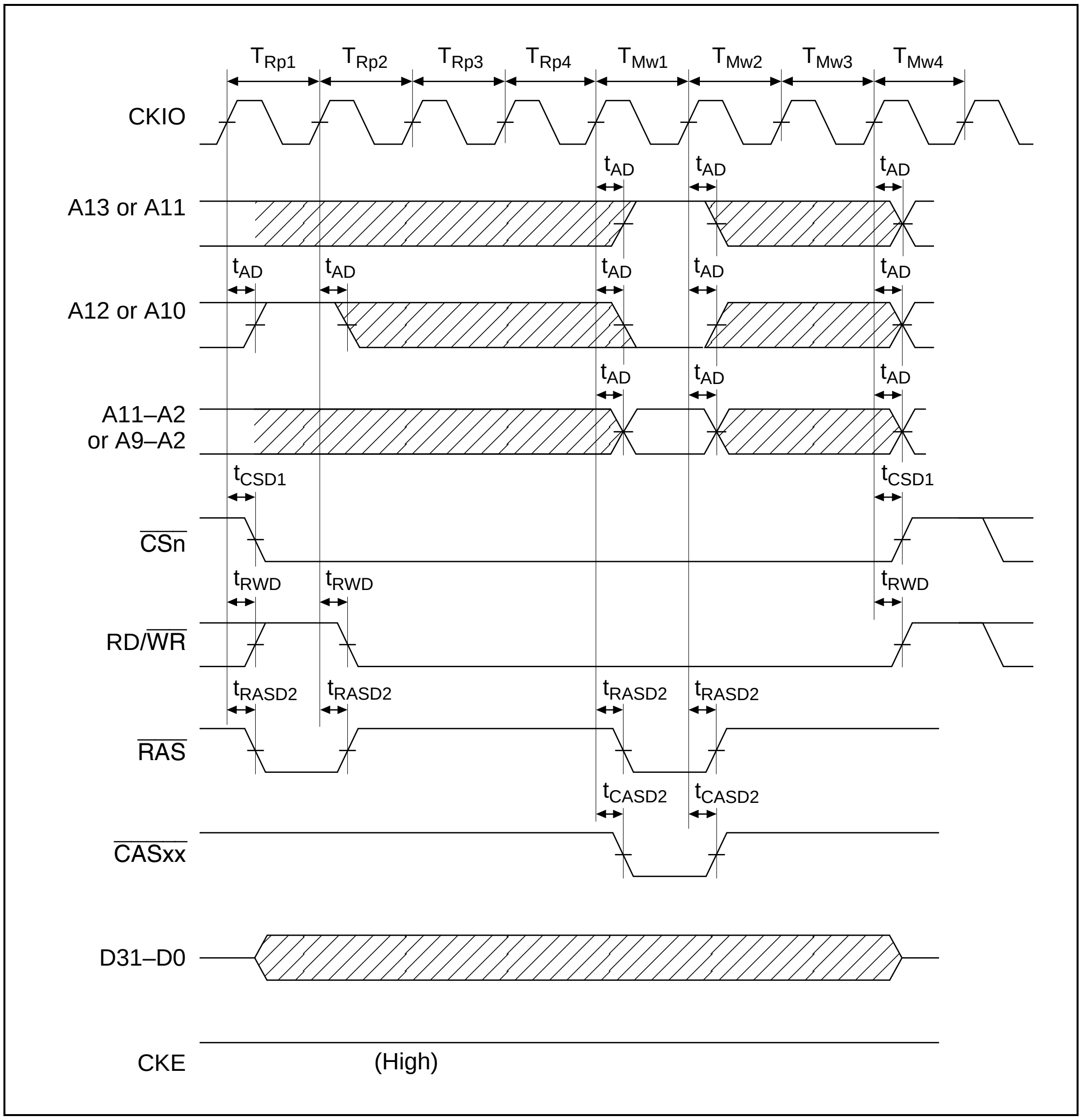


Figure 22.57 SCK Input Clock Timing

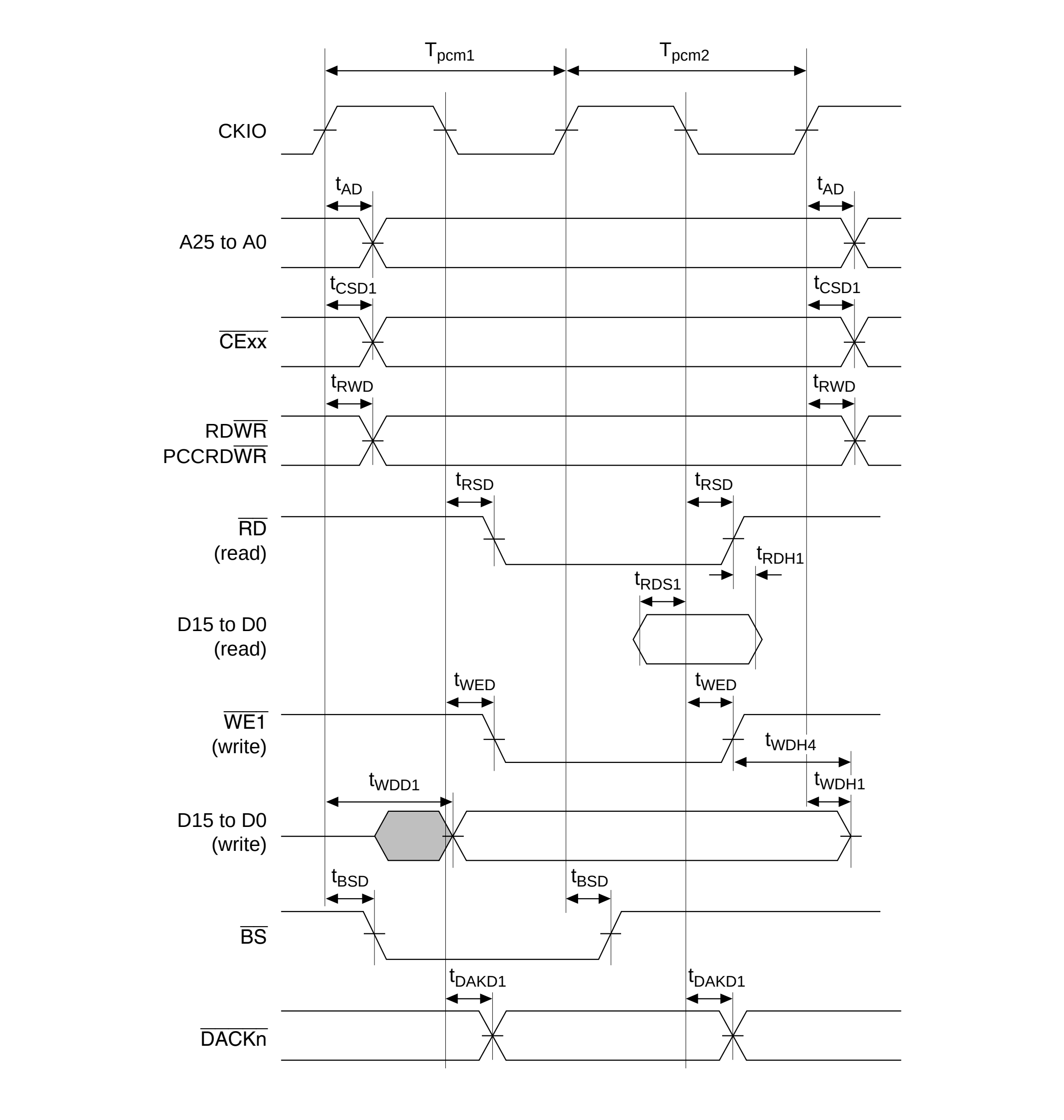


Figure 22.58 SCI I/O Timing in Clocked Synchronous Mode

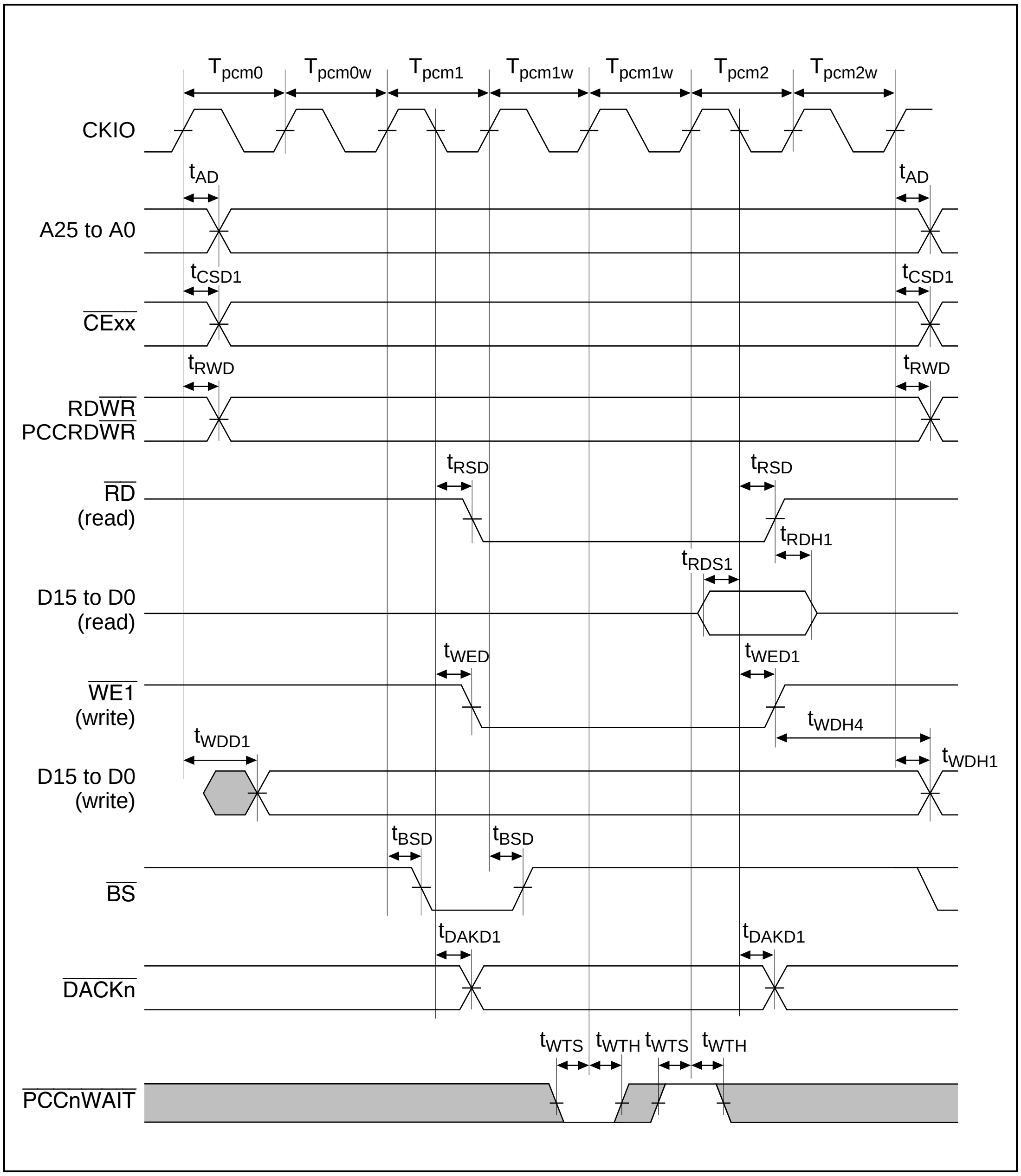
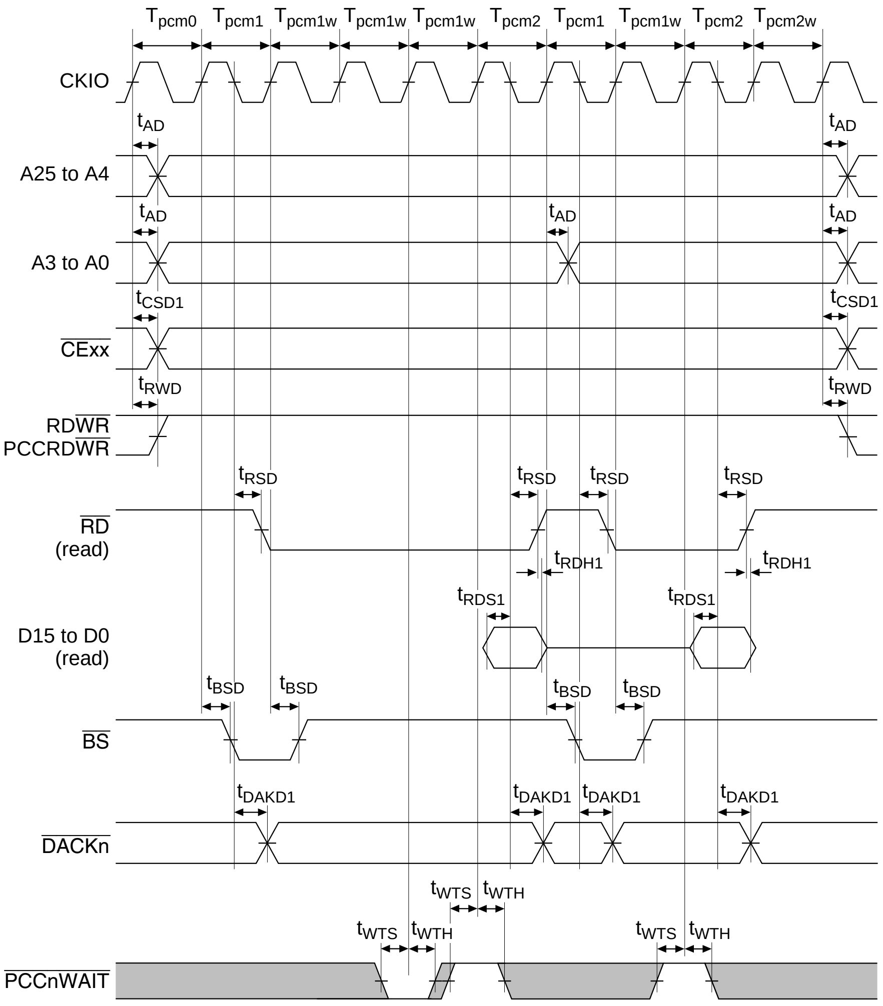


Figure 22.59 I/O Port Timing



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 22.60 DREQ Input Timing

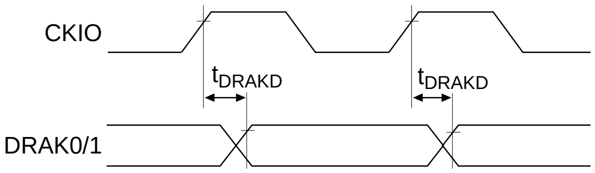
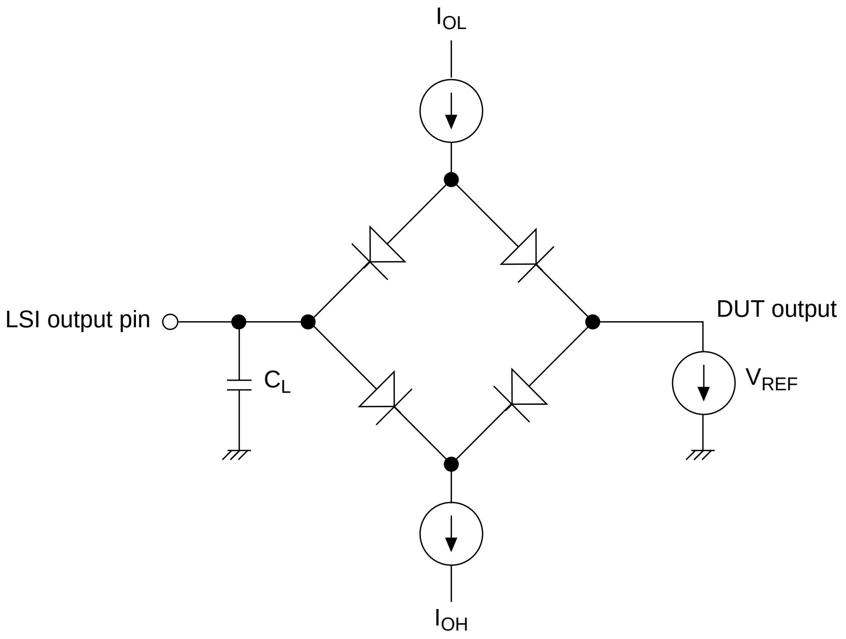


Figure 22.61 DRAK Input Timing

22.3.9 AC Characteristics Measurement Conditions

- I/O signal reference level: 1.5 V (Vcc = 3.3 ± 0.3 V)
- Input pulse level: Vss to 3.0 V (where $\overline{\text{RESET}}$, $\overline{\text{BREQ}}$, NMI, $\overline{\text{IRQ5}}$–$\overline{\text{IRQ0}}$, CKIO, and MD5–MD0 are within Vss to Vcc)
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.

30 pF: CKIO, $\overline{\text{RAS}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$-$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$

50 pF: All other pins

2. I_{OL} and I_{OH} are the values shown in Table 22.3

Figure 22.62 Output Load Circuit

22.4 A/D Conversion Characteristics

Table 22.9 lists the A/D conversion characteristics.

Table 22.9 A/D Conversion Characteristics
(Vcc = 3.3 ± 0.3V, AVcc = 3.3 ± 0.3V, AVcc =Vcc ± 0.3V,
Ta = –20 to 75°C)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bits
Conversion time	—	—	8.9	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	KΩ
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±2.0	LSB
Full-scale error	—	—	±2.0	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±4.0	LSB

22.5 D/A Conversion Characteristics

Table 22.10 lists the D/A conversion characteristics.

Table 22.10 D/A Conversion Characteristics
(Vcc = 3.3 ± 0.3V, AVcc = 3.3 ± 0.3V, AVcc =Vcc ± 0.3V,
Ta = –20 to 75°C)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20pF capacitive load
Absolute accuracy	—	±2.5	±4.0	LSB	2MΩ resistive load

Appendix A Pin Functions

A.1 Pin States

Table A.1 shows pin states during resets, power-down states, and the bus-released states.

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State

		Reset		**Power-Down**		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus Released**
Clock	EXTAL	I	I	I	I	I
	XTAL	O*[1]	O*[1]	O*[1]	O*[1]	O*[1]
	CKIO	IO*[1]	IO*[1]	IO*[1]	IO*[1]	IO*[1]
	EXTAL2	I	I	I	I	I
	XTAL2	O	O	O	O	O
	CAP1, CAP2	—	—	—	—	—
System control	RESETP	I	I	I	I	I
	RESETM	I	I	I	I	I
	BREQ	I	I	I	I	
	BACK	O	O	O	O	L
	MD[5:0]	I	I	I	I	I
	CA	I	I	I	I	I
	STATUS[1:0]/PTJ[7:6]	O	OP*[3]	OP*[3]	OP*[3]	OP*[3]
Interrupt	IRL[3:0]/IRQ[3:0]/ PTH[3:0]	V*[8]	I	I	I	I
	IRQ4/ PTH[4]	V*[8]	I	I	I	I
	NMI	I	I	I	I	I
	PINT[15:8]/PTF[7:0]	V	I	IZ	I	I
	PINT[7:0]/PTC[7:0]	V	P	IK	P	P
	IRQOUT	O	O	O	O	O
Address bus	A[25:0]	Z	O	ZL*[10]	O	Z
Data bus	D[15:0]	Z	I	Z	IO	Z
	D[23:16]/PTA[7:0]	Z	IP*[3]	ZK*[3]	IOP*[3]	ZP*[3]
	D[31:24]/PTB[7;0]	Z	IP*[3]	ZK*[3]	IOP*[3]	ZP*[3]

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus Released**
Bus control	CS0	H	O	ZH*[11]	O	Z
	CS[2:4]/PTK[0:2]	H	OP*[3]	ZH*[11]K*[3]	OP*[3]	ZP*[3]
	CS5/CE1A/PTK[3]	H	OP*[3]	ZH*[11]K*[3]	OP*[3]	ZP*[3]
	CS6/CE1B	H	O	ZH*[11]	O	Z
	BS/PTK[4]	H	OP*[3]	ZH*[11]K*[3]	OP*[3]	ZP*[3]
	RAS3L/PTJ[0]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	RAS2L/PTJ[1]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	RAS3U/PTE[2]	V	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	RAS2U/PTE[1]	V	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CAS2L/PTE[6]	V	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CAS2H/PTE[3]	V	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CASLL/CAS/PTJ[2]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CASLH/PTJ[3]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CASHL/ PTJ[4]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	CASHH/ PTJ[5]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	WE0/DQMLL	H	O	ZH*[11]	O	Z
	WE1/DQMLU/WE	H	O	ZH*[11]	O	Z
	WE2/DQMUL/ICIORD/ PTK[6]	H	OP*[3]	ZH*[11]K*[3]	OP*[3]	ZP*[3]
	WE3/DQMUU/ICIOWR/ PTK[7]	H	OP*[3]	ZH*[11]K*[3]	OP*[3]	ZP*[3]
	RDWR	H	O	ZH*[11]	O	Z
	RD	H	O	ZH*[11]	O	Z
	CKE/PTK[5]	H	OP*[3]	OK*[3]	OP*[3]	OP*[3]
	WAIT	Z	I	Z	I	Z
DMAC	DREQ0/PTD[4]	V	ZI*[7]	Z	I	I
	DACK0/PTD[5]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	DRAK0/PTD[1]	V	OP*[3]	ZH*[11]K*[3]	OP*[3]	OP*[3]
	DREQ1/PTD[6]	V	ZI*[7]	Z	I	I
	DACK1/PTD[7]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	DRAK1/PTD[0]	V	OP*[3]	ZH*[11]K*[3]	OP*[3]	OP*[3]

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus Released**
Timer	TCLK/PTH[7]	V	ZP	IOP*[5]	IOP*[5]	IOP*[5]
SCI/ SmartCARD without FIFO	RxD0/SCPT[0]	Z	ZI*[7]	Z	IZ*[6]	IZ*[6]
	TxD0/SCPT[0]	Z	ZO*[7]	ZK*[3]	OZ*[6]	OZ*[6]
	SCK0/SCPT[1]	V	ZP*[3]	ZK*[3]	IOP*[5]	IOP*[5]
SCIF/IrDA with FIFO	RxD1/SCPT[2]	Z	ZI*[7]	Z	IZ*[6]	IZ*[6]
	TxD1/SCPT[2]	Z	ZO*[7]	ZK*[3]	OZ*[6]	OZ*[6]
	SCK1/SCPT[3]	V	ZP*[3]	ZK*[3]	IOP*[5]	IOP*[5]
SCIF with FIFO	RxD2/SCPT[4]	Z	ZI*[7]	Z	IZ*[6]	IZ*[6]
	TxD2/SCPT[4]	Z	ZO*[7]	ZK*[3]	OZ*[6]	OZ*[6]
	SCK2/SCPT[5]	V	ZP*[3]	ZK*[3]	IOP*[5]	IOP*[5]
	RTS2/SCPT[6]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	CTS2/IRQ5/SCPT[7]	V*[8]	ZI*[7]	I	I	I
Port	PTE[7:6]	V	P*[3]	K*[3]	P*[3]	P*[3]
	CE2B/PTE[5]	V	OP*[3]	ZH*[11] K*[3]	OP*[3]	ZP*[3]
	CE2A/PTE[4]	V	OP*[3]	ZH*[11] K*[3]	OP*[3]	ZP*[3]
	PTE[0]	V	P*[3]	K*[3]	P*[3]	P*[3]
	IOIS16/PTG[7]	V	I	Z	I	I
	PTG[6:0]	V	I	Z	I	I
	PHT[6]	V	I	Z	I	I
	ADTRG/PTH[5]	V*[8]	I	IZ	I	I
	WAKEUP/PTD[3]	V	OP*[3]	OK*[3]	OP*[3]	ZP*[3]
	RESETOUT/PTD[2]	O	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
Analog	AN[5:0]/PTL[5:0]	Z	ZI*[7]	Z	I	I
	AN[6:7]/DA[1:0]/PTL[6:7]	Z	ZI*[7]	I	IO*[9]	IO*[9]

I: Input
O: Output
H: High-level output
L: Low-level output
Z: High impedance
P: Input or output depending on register setting
K: Input pin is high impedance, output pin holds the state
V: Input/output buffer off, pull-up MOS on

Notes:
1. Dependent on the clock mode (MD2-MD0 setting)
2. Z when the port function is used.
3. K or P when the port function is used.
4. When the port function is used, K or P. When the port function is not used, Z or O depending on register setting.
5. When the port function is used, K or P. When the port function is not used, I or O depending on register setting.
6. Dependent on register setting
7. I or O when the port function is used.
8. Input schmidt buffers and pull-up MOS of IRQ[5:0] and ADTRG are on; other inputs are off.
9. When the port function is used, I.When the port function is not used, I or O, depending on register setting.
10. In the standby mode, Z or L depending on register setting.
11. In the standby mode, Z or H depending on register setting.

A.2 Pin Specifications

Table A.2 shows the pin specifications.

Table A.2 Pin Specifications

Pin	Pin No.	I/O	Function
MD5	197	I	Operating mode pin (Endian mode)
MD4, MD3	196, 195	I	Operating mode pin (Area 0 bus-width)
MD2 to MD0	2, 1, 144	I	Operating mode pin (clock mode)
$\overline{\text{RAS3L}}$/PTJ[0]	106	I/O	RAS for area3 lower 32MB / I/O port
$\overline{\text{RAS2L}}$/PTJ[1]	107	I/O	RAS for area2 lower 32MB / I/O port
$\overline{\text{CE2A}}$/PTE[4]	103	I/O	PCMCIA CE2A / I/O port
$\overline{\text{CE2B}}$/PTE[5]	104	I/O	PCMCIA CE2B / I/O port
RXD0/SCPT[0]	171	I	Serial port 0 data input / input port
RXD1/SCPT[2]	172	I	Serial port 1 data input / input port
RXD2/SCPT[4]	174	I	Serial port 2 data input / input port
TXD0/SCPT[0]	164	O	Serial port 0 data output/ output port
TXD1/SCPT[2]	166	O	Serial port 1 data output/ output port
TXD2/SCPT[4]	168	O	Serial port 2 data output/ output port
SCK0/SCPT[1]	165	I/O	Serial port 0 clock input/output / I/O port
SCK1/SCPT[3]	167	I/O	Serial port 1 clock input/output / I/O port
SCK2/SCPT[5]	169	I/O	Serial port 2 clock input/output / I/O port
RTS2/SCPT[6]	170	I/O	Serial port 2 transmit request / I/O port
STATUS1/PTJ[7]	158	I/O	Processor status / I/O port
STATUS0/PTJ[6]	157	I/O	Processor status / I/O port
A25 to A0	86, 84, 82, 80, 78 to 72, 70, 68 to 60, 58, 56 to 53	O	Address bus
D31 to D24/ PTB[7] to PTB[0]	13 to 18, 20, 22	I/O	Data bus / I/O port

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
D23 to D16/ PTA[7] to PTA[0]	23 to 26, 28, 30 to 32	I/O	Data bus / I/O port
D15 to D0	34, 36 to 44, 46, 48 to 52	I/O	Data bus
PTC[7:0]/ PINT[7:0]	177 to 180,185 to 188	I/O	I/O port / port interrupt request
WAKEUP/PTD[3]	182	I/O	WAKEUP / I/O port
RESETOUT/ PTD[2]	184	I/O	RESETOUT / I/O port
DRAK0/PTD[1]	189	I/O	DMA control pin / I/O port
DRAK1/PTD[0]	190	I/O	DMA control pin / I/O port
$\overline{\text{DREQ0}}$/PTD[4]	191	I	DMA transfer request 0 / input port
$\overline{\text{DREQ1}}$/PTD[6]	192	I	DMA transfer request 1 / input port
AN[5:0]/PTL[5:0]	204 to 199	I	Analog input pin / input port
AN[7:6]/DA[1:0]/ PTL[7:6]	207, 206	I/O	Analog input/output pin / input port
$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	102	O	Chip select 6 / PCMCIA CE1B
$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/ PTK[3]	101	I/O	Chip select 5 / PCMCIA CE2B / I/O port

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
$\overline{\text{CS4}}$/PTK[2]	100	I/O	Chip select 4 / I/O port
$\overline{\text{CS3}}$/PTK[1]	99	I/O	Chip select 3 / I/O port
$\overline{\text{CS2}}$/PTK[0]	98	I/O	Chip select 2 / I/O port
$\overline{\text{CS0}}$	96	O	Chip select 0
$\overline{\text{BS}}$/PTK[4]	87	I/O	Bus cycle start / I/O port
$\overline{\text{CASHH}}$/PTJ[5]	113	I/O	D31–D24 selection CAS / I/O port
$\overline{\text{CASHL}}$/PTJ[4]	112	I/O	D23–D16 selection CAS / I/O port
$\overline{\text{CASLH}}$/PTJ[3]	110	I/O	D15–D8 selection CAS / I/O port
$\overline{\text{CASLL}}$/ CAS/PTJ[2]	108	I/O	D7–D0 selection CAS / CAS(SDRAM) / I/O port
$\overline{\text{DACK0}}$/PTD[5]	114	I/O	DMA transfer strobe 0 / I/O port
$\overline{\text{DACK1}}$/PTD[7]	115	I/O	DMA transfer strobe 1 / I/O port
$\overline{\text{RD}}$	88	O	Read strobe pin
$\overline{\text{WE0}}$/ DQMLL	89	O	D7–D0 select signal/ DQM(SDRAM)
$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	90	O	D15–D8 select signal / DQM(SDRAM)/ PCMCIA $\overline{\text{WE}}$ signal
$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$/PTK[6]	91	I/O	D23–D16 select signal / DQM(SDRAM) / PCMCIA $\overline{\text{IORD}}$ signal / I/O port
$\overline{\text{WE3}}$/DQMUU /$\overline{\text{ICIOWR}}$/PTK[7]	92	I/O	D31–D24 select signal /DQM(SDRAM) / PCMCIA $\overline{\text{IOWR}}$ signal / I/O port
RD/$\overline{\text{WR}}$	93	O	Read/write switchover signal
PTE[7]	94	I/O	I/O port
$\overline{\text{CAS2L}}$/PTE[6]	116	I/O	CAS for D7-D0(area 2 DRAM) / I/O port
$\overline{\text{CAS2H}}$/PTE[3]	117	I/O	CAS for D15-D8(area 2 DRAM) / I/O port
$\overline{\text{RAS3U}}$/PTE[2]	118	I/O	RAS for area3 upper 32MB/ I/O port
$\overline{\text{RAS2U}}$/PTE[1]	119	I/O	RAS for area2 upper 32MB /I/O port
PTE[0]	120	I/O	I/O port
$\overline{\text{RESETM}}$	124	I	Manual reset input

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
PTH[5]/ADTRG	125	I	input port / ADC trigger request
IOIS16/PTG[7]	126	I	I/O is 16 for PC card / input port
PTG[6]	127	I	input port
PTG[5]	128	I	input port
PTG[4]	129	I	input port
PTG[3]	130	I	input port
PTG[2]	131	I	input port
PTG[1]	133	I	input port
PTG[0]	135	I	input port
PTF[7:0]/ PINT[15:8]	136 to 143	I	input port / port interrupt reguest
PTH[6]	151	I	input port
$\overline{\text{WAIT}}$	123	I	Hardware wait request
$\overline{\text{BREQ}}$	122	I	Bus request
$\overline{\text{BACK}}$	121	O	Bus acknowledge
$\overline{\text{IRQOUT}}$	160	O	Interrupt / refresh request notification

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
$\overline{\text{RESETP}}$	193	I	Power-on reset input
NMI	7	I	Nonmaskable interrupt request
IRQ[3:0]/IRL[3:0]/ PTH[3:0]	11 to 8	I	External interrupt requests / external interrupt sources / input port
IRQ4/ PTH[4]	12	I	External interrupt request / input port
CTS2/IRQ5/ SCPT[7]	176	I	Serial port 2 transmit enable / external interrupt request / input port
TCLK/PTH[7]	159	I/O	Clock input / output (for TMU/RTC) / I/O port
EXTAL	156	I	External clock/crystal resonator pin
XTAL	155	O	Crystal resonator pin
CAP1	146	—	External capacitance pin (for PLL1)
CAP2	149	—	External capacitance pin (for PLL2)
CKIO	162	I/O	System clock input/output
XTAL2	4	O	Crystal resonator pin (for on-chip RTC)
EXTAL2	5	I	Crystal resonator pin (for on-chip RTC)
CKE/PTK[5]	105	I/O	CK enable for SDRAM /I/O port
V_{cc}	21, 29, 35, 47, 59, 71, 81, 85, 97, 111, 134, 154, 163, 175, 183, 194	Power supply	Power supply (3.3 V)
V_{cc} (RTC)	3	Power supply	RTC oscillator power supply (3.3 V)
V_{cc} (PLL)	145, 150	Power supply	PLL power supply (3.3 V)
AV_{cc}	205	Power supply	Analog power supply (3.3 V)
V_{ss}	19, 27, 33, 45, 57, 69, 79, 83, 95, 109, 132, 153, 161, 173, 181	Power supply	Power supply (0 V)

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
V_{SS} (RTC)	6	Power supply	RTC oscillator power supply (0 V)
V_{SS} (PLL)	147, 148, 152	Power supply	PLL power supply (0 V)
AV_{SS}	198, 208	Power supply	Analog power supply (0 V)

Note: Power must be supplied constantly to all power supply pins.

A.3 Handling of Unused Pins

- When RTC is not used
  - EXTAL2: Pull up
  - XTAL2: Leave unconnected
  - V_{CC} (RTC Oscillator): Power supply (3.3 V)
  - V_{SS} (RTC Oscillator): Power supply (0 V)
- When PLL1 is not used
  - CAP1: Leave unconnected
  - V_{CC} (PLL): Power supply (3.3 V)
  - V_{SS} (PLL): Power supply (0 V)
- When PLL2 is not used
  - CAP2: Leave unconnected
  - V_{CC} (PLL): Power supply (3.3 V)
  - V_{SS} (PLL): Power supply (0 V)
- When on-chip crystal oscillator is not used
  - XTAL: Leave unconnected
- When EXTAL terminal is not used
  - EXTAL: Pull up
- When A/D converter is not used
  - AN[7:0]: Leave unconnected
  - AV_{CC}: Power supply (3.3 V)
  - AV_{SS}: Power supply (0 V)

A.4 Pin States in Access to Each Address Space

Table A.3 Pin States (Normal Memory/Little-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High
	W	Low	Low	High	Low
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High
	W	High	High	Low	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL /PTK[6]	R	High	High	High	High
	W	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU /PTK[7]	R	High	High	High	High
	W	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled^{*1}	Enabled^{*1}	Enabled^{*1}	Enabled^{*1}
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z^{*2}	Invalid data	Valid data	Valid data
D31 to D16		High-Z^{*2}	High-Z^{*2}	High-Z^{*2}	High-Z^{*2}

Table A.3 Pin States (Normal Memory/Little-Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High	High	High	High
/PTK[6]	W	High	High	Low	High	High	Low	Low
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High	High	High	High
/PTK[7]	W	High	High	High	Low	High	Low	Low
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.3 Pin States (Normal Memory/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.4 Pin States (Normal Memory/Big-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High
	W	Low	High	Low	Low
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High
	W	High	Low	High	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High
/PTK[6]	W	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High
/PTK[7]	W	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z[*2]	Valid data	Invalid data	Valid data
D31 to D16		High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]

Table A.4 Pin States (Normal Memory/Big-Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3U}$/PTE[2]		High	High	High	High	High	High	High
$\overline{RAS3L}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{CASLL}$/$\overline{CAS}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{CASLH}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{CASHL}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{CASHH}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{CAS2L}$/PTE[6]		High	High	High	High	High	High	High
$\overline{CAS2H}$/PTE[3]		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{WE1}$/$\overline{WE}$/DQMLU	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{WE2}$/$\overline{ICIORD}$/DQMUL /PTK[6]	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{WE3}$/$\overline{ICIOWR}$/DQMUU /PTK[7]	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{CE2A}$/PTE[4]		High	High	High	High	High	High	High
$\overline{CE2B}$/PTE[5]		High	High	High	High	High	High	High
$\overline{RAS2U}$/PTE[1]		High	High	High	High	High	High	High
$\overline{RAS2L}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.4 Pin States (Normal Memory/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.5 Pin States (Burst ROM/Little-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		**Byte/Word/Longword Access**	**Byte Access (Address 2n)**	**Byte Access (Address 2n + 1)**	**Word/Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High
/PTK[6]	W	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High
/PTK[7]	W	—	—	—	—
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.5 Pin States (Burst ROM/Little-Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High	High	High	High
/PTK[6]	W	—	—	—	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High	High	High	High
/PTK[7]	W	—	—	—	—	—	—	—
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled^{*1}	Enabled^{*1}	Enabled^{*1}	Enabled^{*1}	Enabled^{*1}	Enabled^{*1}	Enabled^{*1}
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.5 Pin States (Burst ROM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.6 Pin States (Burst ROM/Big-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\mathrm{CS6}}$ to $\overline{\mathrm{CS2}}$, $\overline{\mathrm{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RD}}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{\mathrm{WR}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\mathrm{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RAS3U}}$/PTE[2]		High	High	High	High
$\overline{\mathrm{RAS3L}}$/PTJ[0]		High	High	High	High
$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{CAS}}$/PTJ[2]		High	High	High	High
$\overline{\mathrm{CASLH}}$/PTJ[3]		High	High	High	High
$\overline{\mathrm{CASHL}}$/PTJ[4]		High	High	High	High
$\overline{\mathrm{CASHH}}$/PTJ[5]		High	High	High	High
$\overline{\mathrm{CAS2L}}$/PTE[6]		High	High	High	High
$\overline{\mathrm{CAS2H}}$/PTE[3]		High	High	High	High
$\overline{\mathrm{WE0}}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{\mathrm{WE1}}$/$\overline{\mathrm{WE}}$/DQMLU	R	High	High	High	High
	W	—	—	—	—
$\overline{\mathrm{WE2}}$/$\overline{\mathrm{ICIORD}}$/DQMUL	R	High	High	High	High
/PTK[6]	W	—	—	—	—
$\overline{\mathrm{WE3}}$/$\overline{\mathrm{ICIOWR}}$/DQMUU	R	High	High	High	High
/PTK[7]	W	—	—	—	—
$\overline{\mathrm{CE2A}}$/PTE[4]		High	High	High	High
$\overline{\mathrm{CE2B}}$/PTE[5]		High	High	High	High
$\overline{\mathrm{RAS2U}}$/PTE[1]		High	High	High	High
$\overline{\mathrm{RAS2L}}$/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{WAIT}}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{\mathrm{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.6 Pin States (Burst ROM/Big-Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High	High	High	High
/PTK[6]	W	—	—	—	—	—	—	—
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High	High	High	High
/PTK[7]	W	—	—	—	—	—	—	—
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.6 Pin States (Burst ROM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.7 Pin States (DRAM/Little-Endian)

		16-Bit Bus Width (Area 3)			16-Bit Bus Width (Area 2)		
Pin		Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		Low	High	Low	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	Low	Low	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	Low	High	Low
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	Low	Low
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL /PTK[6]	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU /PTK[7]	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D31 to D16		High-Z^{*2}	High-Z^{*2}	High-Z^{*2}	High-Z^{*2}	High-Z^{*2}	High-Z^{*2}

Table A.7 Pin States (DRAM/Little-Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]	High/Low[*1]
$\overline{\text{RAS3L}}$/PTJ[0]		Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]	Low/High[*1]
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		Low	High	High	High	Low	High	Low
$\overline{\text{CASLH}}$/PTJ[3]		High	Low	High	High	Low	High	Low
$\overline{\text{CASHL}}$/PTJ[4]		High	High	Low	High	High	Low	Low
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	Low	High	Low	Low
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL /PTK[6]	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU /PTK[7]	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.7 Pin States (DRAM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Note: 1. Lower 32MB access / Upper 32MB access.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.8 Pin States (DRAM/Big-Endian)

		16-Bit Bus Width (Area 3)			16-Bit Bus Width (Area 2)		
Pin		Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High/Low[*1]	High/Low[*1]	High/Low[*1]	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		Low/High[*1]	Low/High[*1]	Low/High[*1]	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	Low	Low	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		Low	High	Low	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	Low	Low
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	Low	High	Low
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL /PTK[6]	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU /PTK[7]	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High/Low[*1]	High/Low[*1]	High/Low[*1]
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	Low/High[*1]	Low/High[*1]	Low/High[*1]
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data
D31 to D16		High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]

Table A.8 Pin States (DRAM/Big-Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\mathrm{CS6}}$ to $\overline{\mathrm{CS2}}$, $\overline{\mathrm{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\mathrm{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RAS3U}}$/PTE[2]		High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}
$\overline{\mathrm{RAS3L}}$/PTJ[0]		Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}
$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{CAS}}$/PTJ[2]		High	High	High	Low	High	Low	Low
$\overline{\mathrm{CASLH}}$/PTJ[3]		High	High	Low	High	High	Low	Low
$\overline{\mathrm{CASHL}}$/PTJ[4]		High	Low	High	High	Low	High	Low
$\overline{\mathrm{CASHH}}$/PTJ[5]		Low	High	High	High	Low	High	Low
$\overline{\mathrm{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\mathrm{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High
$\overline{\mathrm{WE0}}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\mathrm{WE1}}$/$\overline{\mathrm{WE}}$/DQMLU	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\mathrm{WE2}}$/$\overline{\mathrm{ICIORD}}$/DQMUL	R	High	High	High	High	High	High	High
/PTK[6]	W	High	High	High	High	High	High	High
$\overline{\mathrm{WE3}}$/$\overline{\mathrm{ICIOWR}}$/DQMUU	R	High	High	High	High	High	High	High
/PTK[7]	W	High	High	High	High	High	High	High
$\overline{\mathrm{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\mathrm{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\mathrm{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\mathrm{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.8 Pin States (DRAM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Note: 1. Lower 32MB access / Upper 32MB access.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.9 Pin States (Synchronous DRAM/Little-Endian)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{RD}}$/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}	High/Low^{*1}
$\overline{\text{RAS3L}}$/PTJ[0]		Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}	Low/High^{*1}
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/PTJ[2]		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTE[3]		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
DQMLU/$\overline{\text{WE1}}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		High^{*2}	High^{*2}	High^{*2}	High^{*2}	High^{*2}	High^{*2}	High^{*2}
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address, command	Address, command	Address, command	Address, command	Address, command	Address, command	Address, command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.9 Pin States (Synchronous DRAM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Lower 32MB access/ Upper 32MB access

2. Normally high. Low in self-refreshing.

Table A.10 Pin States (Synchronous DRAM/Big-Endian)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$, $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{\text{RD}}$/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1
$\overline{\text{RAS3L}}$/PTJ[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/PTJ[2]		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTE[6]		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTE[3]		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2	High*2
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address, command	Address, command	Address, command	Address, command	Address, command	Address, command	Address, command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.10 Pin States (Synchronous DRAM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Lower 32MB access/ Upper 32MB access

2. Normally high. Low in self-refreshing.

Table A.11 Pin States (PCMCIA/Little-Endian)

		PCMCIA Memory Interface(Area 5)			
		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High
	W	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High
/PTK[6]	W	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High
/PTK[7]	W	High	High	High	High
$\overline{\text{CE2A}}$/PTE[4]		High	High	Low	Low
$\overline{\text{CE2B}}$/PTE[5]		High	High	High	High
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High
$\overline{\text{RAS2L}}$/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z^{*2}	Invalid data	Valid data	Valid data
D31 to D16		High-Z^{*2}	High-Z^{*2}	High-Z^{*2}	High-Z^{*2}

Table A.11 Pin States (PCMCIA/Little-Endian) (cont)

		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Long-word Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High	High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL	R	High	High	High	High	Low	Low	Low	Low
/PTK[6]	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU	R	High	High	High	High	High	High	High	High
/PTK[7]	W	High	High	High	High	Low	Low	Low	Low
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	Low	Low	High	High	Low	Low
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High	High
$\overline{\text{RAS2}}$/PTJ[1]		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.11 Pin States (PCMCIA/Little-Endian) (cont)

	PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
	8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin	Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long word Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Long-word Access
D7 to D0	Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8	High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data	Valid data
D31 to D16	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Table A.12 Pin States (PCMCIA/Big-Endian)

		PCMCIA Memory Interface(Area 5)			
		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
CS6 to CS2, CS0		Enabled	Enabled	High	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/WR	R	High	High	High	High
	W	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled
RAS3U/PTE[2]		High	High	High	High
RAS3L/PTJ[0]		High	High	High	High
CASLL/CAS/PTJ[2]		High	High	High	High
CASLH/PTJ[3]		High	High	High	High
CASHL/PTJ[4]		High	High	High	High
CASHH/PTJ[5]		High	High	High	High
CAS2L/PTE[6]		High	High	High	High
CAS2H/PTE[3]		High	High	High	High
WE0/DQMLL	R	High	High	High	High
	W	High	High	High	High
WE1/WE/DQMLU	R	High	High	High	High
	W	Low	Low	Low	Low
WE2/ICIORD/DQMUL	R	High	High	High	High
/PTK[6]	W	High	High	High	High
WE3/ICIOWR/DQMUU	R	High	High	High	High
/PTK[7]	W	High	High	High	High
CE2A/PTE[4]		High	High	Low	Low
CE2B/PTE[5]		High	High	High	High
RAS2U/PTE[1]		High	High	High	High
RAS2/PTJ[1]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
WAIT		Disabled	Disabled	Disabled	Disabled
IOIS16		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z[*2]	Invalid data	Valid data	Valid data
D31 to D16		High-Z[*2]	High-Z[*2]	High-Z[*2]	High-Z[*2]

Table A.12 Pin States (PCMCIA/Big-Endian) (cont)

		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Long-word Access
$\overline{\text{CS6}}$ to $\overline{\text{CS2}}$, $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS3U}}$/PTE[2]		High	High	High	High	High	High	High	High
$\overline{\text{RAS3L}}$/PTJ[0]		High	High	High	High	High	High	High	High
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/PTJ[2]		High	High	High	High	High	High	High	High
$\overline{\text{CASLH}}$/PTJ[3]		High	High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/PTJ[4]		High	High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/PTJ[5]		High	High	High	High	High	High	High	High
$\overline{\text{CAS2L}}$/PTE[6]		High	High	High	High	High	High	High	High
$\overline{\text{CAS2H}}$/PTE[3]		High	High	High	High	High	High	High	High
$\overline{\text{WE0}}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE1}}$/$\overline{\text{WE}}$/DQMLU	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$/DQMUL /PTK[6]	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$/DQMUU /PTK[7]	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{\text{CE2A}}$/PTE[4]		High	High	High	High	High	High	High	High
$\overline{\text{CE2B}}$/PTE[5]		High	High	Low	Low	High	High	Low	Low
$\overline{\text{RAS2U}}$/PTE[1]		High	High	High	High	High	High	High	High
$\overline{\text{RAS2}}$/PTJ[1]		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.12 Pin States (PCMCIA/Big-Endian) (cont)

	PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
	8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin	Byte/ Word/ Long-word Access	Byte Access (Ad dress2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Long-word Access
D7 to D0	Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8	High-Z*[2]	Valid data	Invalid data	Valid data	High-Z*[2]	Valid data	Invalid data	Valid data
D31 to D16	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]	High-Z*[2]

Notes: 1. When WCR2 register wait setting is 0, disabled.

2. Unused data pins should be switched to the port function, or pulled up or down.

Appendix B Memory-Mapped Control Registers

B.1 Register Address Map

Table B.1 Memory-Mapped Control Registers

Control Register	Module *[1]	Bus *[2]	Address *[4]	Size (Bits)	Access Size (Bits) *[3]
PTEH	CCN	S	H'FFFFFFF0	32	32
PTEL	CCN	S	H'FFFFFFF4	32	32
TTB	CCN	S	H'FFFFFFF8	32	32
TEA	CCN	S	H'FFFFFFFC	32	32
MMUCR	CCN	S	H'FFFFFFE0	32	32
BASRA	CCN	S	H'FFFFFFE4	32	32
BASRB	CCN	S	H'FFFFFFE8	32	32
CCR	CCN	S	H'FFFFFFEC	32	32
TRA	CCN	S	H'FFFFFFD0	32	32
EXPEVT	CCN	S	H'FFFFFFD4	32	32
INTEVT	CCN	S	H'FFFFFFD8	32	32
BARA	UBC	S	H'FFFFFFB0	32	32
BAMRA	UBC	S	H'FFFFFFB4	8	8
BBRA	UBC	S	H'FFFFFFB8	16	16
BARB	UBC	S	H'FFFFFFA0	32	32
BAMRB	UBC	S	H'FFFFFFA4	8	8
BBRB	UBC	S	H'FFFFFFA8	16	16
BDRB	UBC	S	H'FFFFFF90	32	32
BDMRB	UBC	S	H'FFFFFF94	32	32
BRCR	UBC	S	H'FFFFFF98	16	16
FRQCR	CPG	S	H'FFFFFF80	16	16
STBCR	CPG	S	H'FFFFFF82	8	8
STBCR2	CPG	S	H'FFFFFF88	8	8
WTCNT	CPG	S	H'FFFFFF84	8	16
WTCSR	CPG	S	H'FFFFFF86	8	16
BCR1	BCN	C	H'FFFFFF60	16	16
BCR2	BCN	C	H'FFFFFF62	16	16

Table B.1 Memory-Mapped Control Registers (cont)

Control Register	Module *[1]	Bus *[2]	Address *[4]	Size (Bits)	Access Size (Bits) *[3]
WCR1	BCN	C	H'FFFFFF64	16	16
WCR2	BCN	C	H'FFFFFF66	16	16
MCR	BCN	C	H'FFFFFF68	16	16
DCR	BCN	C	H'FFFFFF6A	16	16
PCR	BCN	C	H'FFFFFF6C	16	16
RTCSR	BCN	C	H'FFFFFF6E	16	16
RTCNT	BCN	C	H'FFFFFF70	16	16
RTCOR	BCN	C	H'FFFFFF72	16	16
RFCR	BCN	C	H'FFFFFF74	16	16
BCR3	BCN	C	H'FFFFFF7E	16	16
SDMR	BCN	C	H'FFFFD000 – H'FFFFEFFE	—	8
R64CNT	RTC	P	H'FFFFFEC0	8	8
RSECCNT	RTC	P	H'FFFFFEC2	8	8
RMINCNT	RTC	P	H'FFFFFEC4	8	8
RHRCNT	RTC	P	H'FFFFFEC6	8	8
RWKCNT	RTC	P	H'FFFFFEC8	8	8
RDAYCNT	RTC	P	H'FFFFFECA	8	8
RMONCNT	RTC	P	H'FFFFFECC	8	8
RYRCNT	RTC	P	H'FFFFFECE	8	8
RSECAR	RTC	P	H'FFFFFED0	8	8
RMINAR	RTC	P	H'FFFFFED2	8	8
RHRAR	RTC	P	H'FFFFFED4	8	8
RWKAR	RTC	P	H'FFFFFED6	8	8
RDAYAR	RTC	P	H'FFFFFED8	8	8
RMONAR	RTC	P	H'FFFFFEDA	8	8
RCR1	RTC	P	H'FFFFFEDC	8	8
RCR2	RTC	P	H'FFFFFEDE	8	8
ICR0	INTC	P	H'FFFFFEE0	16	16
IPRA	INTC	P	H'FFFFFEE2	16	16
IPRB	INTC	P	H'FFFFFEE4	16	16
TOCR	TMU	P	H'FFFFFE90	8	8

Table B.1 Memory-Mapped Control Registers (cont)

Control Register	Module *[1]	Bus *[2]	Address *[4]	Size (Bits)	Access Size (Bits) *[3]
TSTR	TMU	P	H'FFFFFE92	8	8
TCOR0	TMU	P	H'FFFFFE94	32	32
TCNT0	TMU	P	H'FFFFFE98	32	32
TCR0	TMU	P	H'FFFFFE9C	16	16
TCOR1	TMU	P	H'FFFFFEA0	32	32
TCNT1	TMU	P	H'FFFFFEA4	32	32
TCR1	TMU	P	H'FFFFFEA8	16	16
TCOR2	TMU	P	H'FFFFFEAC	32	32
TCNT2	TMU	P	H'FFFFFEB0	32	32
TCR2	TMU	P	H'FFFFFEB4	16	16
TCPR2	TMU	P	H'FFFFFEB8	32	32
SCSMR	SCI	P	H'FFFFFE80	8	8
SCBRR	SCI	P	H'FFFFFE82	8	8
SCSCR	SCI	P	H'FFFFFE84	8	8
SCTDR	SCI	P	H'FFFFFE86	8	8
SCSSR	SCI	P	H'FFFFFE88	8	8
SCRDR	SCI	P	H'FFFFFE8A	8	8

Table B.1 Memory-Mapped Control Registers (cont)

Control Register	Module *[1]	Bus *[2]	Address *[4]	Size (Bits)	Access Size (Bits) *[3]
INTEVT2	INT	P	H’04000000	32	32
IRR0	INT	P	H’04000004	16	8
IRR1	INT	P	H’04000006	16	8
IRR2	INT	P	H’04000008	16	8
ICR1	INT	P	H’04000010	16	16
ICR2	INT	P	H’04000012	16	16
INTER	INT	P	H’04000014	16	16
IPRC	INT	P	H’04000016	16	16
IPRD	INT	P	H’04000018	16	16
IPRE	INT	P	H’0400001A	16	16
SAR0	DMAC	P	H’04000020	32	16,32
DAR0	DMAC	P	H’04000024	32	16,32
DMATCR0	DMAC	P	H’04000028	32	16,32
CHCR0	DMAC	P	H’0400002C	32	8,16,32
SAR1	DMAC	P	H’04000030	32	16,32
DAR1	DMAC	P	H’04000034	32	16,32
DMATCR1	DMAC	P	H’04000038	32	16,32
CHCR1	DMAC	P	H’0400003C	32	8,16,32
SAR2	DMAC	P	H’04000040	32	16,32
DAR2	DMAC	P	H’04000044	32	16,32
DMATCR2	DMAC	P	H’04000048	32	16,32
CHCR2	DMAC	P	H’0400004C	32	8,16,32
SAR3	DMAC	P	H’04000050	32	16,32
DAR3	DMAC	P	H’04000054	32	16,32
DMATCR3	DMAC	P	H’04000058	32	16,32
CHCR3	DMAC	P	H’0400005C	32	8,16,32
DMAOR	DMAC	P	H’04000060	16	8,16
CMSTR	CMT	P	H’04000070	16	8,16,32
CMCSR	CMT	P	H’04000072	16	8,16,32

Table B.1 Memory-Mapped Control Registers (cont)

Control Register	Module *[1]	Bus *[2]	Address*[4]	Size (Bits)	Access Size (Bits) *[3]
CMCNT	CMT	P	H'04000074	16	8,16,32
CMCOR	CMT	P	H'04000076	16	8,16,32
ADDRAH	A/D	P	H'04000080	8	8
ADDRAL	A/D	P	H'04000082	8	8
ADDRBH	A/D	P	H'04000084	8	8
ADDRBL	A/D	P	H'04000086	8	8
ADDRCH	A/D	P	H'04000088	8	8
ADDRCL	A/D	P	H'0400008A	8	8
ADDRDH	A/D	P	H'0400008C	8	8
ADDRDL	A/D	P	H'0400008E	8	8
ADCSR	A/D	P	H'04000090	8	8
ADCR	A/D	P	H'04000092	8	8
DADR0	D/A	P	H'040000A0	8	8
DADR1	D/A	P	H'040000A2	8	8
DACR	D/A	P	H'040000A4	8	8
PACR	PORT	P	H'04000100	16	16
PBCR	PORT	P	H'04000102	16	16
PCCR	PORT	P	H'04000104	16	16
PDCR	PORT	P	H'04000106	16	16
PECR	PORT	P	H'04000108	16	16
PFCR	PORT	P	H'0400010A	16	16
PGCR	PORT	P	H'0400010C	16	16
PHCR	PORT	P	H'0400010E	16	16
PJCR	PORT	P	H'04000110	16	16
PKCR	PORT	P	H'04000112	16	16
PLCR	PORT	P	H'04000114	16	16
SCPCR	PORT	P	H'04000116	16	16
PADR	PORT	P	H'04000120	8	8
PBDR	PORT	P	H'04000122	8	8
PCDR	PORT	P	H'04000124	8	8

Table B.1 Memory-Mapped Control Registers (cont)

Control Register	Module *[1]	Bus *[2]	Address *[4]	Size (Bits)	Access Size (Bits) *[3]
PDDR	PORT	P	4000126	8	8
PEDR	PORT	P	4000128	8	8
PFDR	PORT	P	400012A	8	8
PGDR	PORT	P	400012C	8	8
PHDR	PORT	P	400012E	8	8
PJDR	PORT	P	4000130	8	8
PKDR	PORT	P	4000132	8	8
PLDR	PORT	P	4000134	8	8
SCPDR	PORT	P	4000136	8	8
SCSMR1	IrDA	P	4000140	8	8
SCBRR1	IrDA	P	4000142	8	8
SCSCR1	IrDA	P	4000144	8	8
SCFTDR1	IrDA	P	4000146	8	8
SCSSR1	IrDA	P	4000148	16	16
SCFRDR1	IrDA	P	400014A	8	8
SCFCR1	IrDA	P	400014C	8	8
SCFDR1	IrDA	P	400014E	16	16
SCSMR2	SCIF	P	4000150	8	8
SCBRR2	SCIF	P	4000152	8	8
SCSCR2	SCIF	P	4000154	8	8
SCFTDR2	SCIF	P	4000156	8	8
SCSSR2	SCIF	P	4000158	16	16
SCFRDR2	SCIF	P	400015A	8	8
SCFCR2	SCIF	P	400015C	8	8
SCFDR2	SCIF	P	400015E	16	16

Notes: 1. Module:

CCN:	Cache controller	UBC:	User break controller
CPG:	Clock pulse generator	BCN:	Bus controller
RTC:	Realtime clock	INTC:	Interrupt controller
TMU:	Timer unit	SCI:	Serial communication interface

2. Internal bus:
S: Connects CPU, CCN, cache, TLB, multiplier, and UBC
C: Connects BCN and cache
P: Connects BCN and peripheral modules RTC, INTC, TMU, and SCI

3. Access size for the control registers. If sizes other than those listed are used, the register will not be correctly accessed.

4. If an area 1 control register is not to be subject to address translation by the MMU, set the first 3 bits of the logical address to 101 to locate the register in the P2 area.

B.2 Registers bits

Table B.2 Registers Bits

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
SDMR	—	—	—	—					BCN
SCSMR	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
SCBRR									SCI
SCSCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI
SCTDR									SCI
SCSSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI
SCRDR									SCI
SCSCMR	—	—	—	—	SDIR	SINV	—	SMIF	SCI
TOCR	—	—	—	—	—	—	—	TCOE	TMU
TSTR	—	—	—	—	—	STR2	STR1	STR0	TMU
TCOR0									TMU
TCNT0									TMU
TCR0	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR1									TMU
TCNT1									TMU

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
TCR1	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR2									TMU
TCNT2									TMU
TCR2	—	—	—	—	—	—	ICPF	UNF	TMU
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCPR2									TMU
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	RTC
RSECCNT	—	10sec			1sec				RTC
RMINCNT	—	10min			1min				RTC
RHRCNT	—	—	10hour		1hour				RTC
RWKCNT	—	—	—	—	—	day of week			RTC
RDAYCNT	—	—	10days		1day				RTC
RMONCNT	—	—	—	10 months	1month				RTC
RYRCNT	10years				1year				RTC
RSECAR	ENB	10sec			1sec				RTC
RMINAR	ENB	10min			1min				RTC
RHRAR	ENB	—	10hour		1hour				RTC
RWKAR	ENB	—	—	—	—	day of week			RTC
RDAYAR	ENB	—	10days		1day				RTC
RMONAR	ENB	—	—	10 months	1month				RTC

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
RCR1	CF	—	—	CIE	AIE	—	—	AF	RTC
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	RTC
ICR0	NML	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IPRA	TMU0				TMU1				INTC
	TMU2				RTC				
IPRB	WDT				REF				INTC
	SCI				—	—	—	—	
BCR1	—	—	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1	BCN
	A5BST0	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	A5PCM	A6PCM	
BCR2	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0	BCN
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—	
BCR3	EXTEND	—	TPC31	TPC30	RCD31	RCD30	TRAS31	TRAS30	BCN
	—	—	TPC21	TPC20	RCD21	RCD20	TRAS21	TRAS20	
WCR1	—	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0	BCN
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0	
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	BCN
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0	
MCR	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0	BCN
	—	BE	SZ	AMX1	AMX0	RFSH	RMODE	EDOMODE	
DCR	TPC1	TPC0	RCD1	RCD0	—	—	TRAS1	TRAS0	BCN
	—	BE	—	AMX1	AMX0	RFSH	RMODE	—	
PCR	—	—	—	—	—	—	—	—	BCN
	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH1	A6TEH0	
RTCSR	—	—	—	—	—	—	—	—	BCN
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS	
RTCNT	—	—	—	—	—	—	—	—	BCN
RTCOR	—	—	—	—	—	—	—	—	BCN

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
RFCR	—	—	—	—	—	—	—	—	BCN
FRQCR	STC2	IFC2	PFC2	—	—	—	SLPFRQ	CKOEN	CPG
	PLLEN	PSTBY	STC1	STC0	IFC1	IFC0	PFC1	PFC0	
STBCR	STBY	—	—	—	—	MSTP2	MSTP1	MSTP0	CPG
STBCR2	MSTPSLP0	—	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	CPG
WTCNT									CPG
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	CPG
BDRB									UBC
BDMRB									UBC
BRCR	CMFA	CMFB	—	—	—	PCBA	—	—	UBC
	DBEB	PCBB	—	—	SEQ	—	—	—	
BARB									UBC
BAMRB	—	—	—	—	—	BASM	BAM	BAM	UBC
BBRB	—	—	—	—	—	—	—	—	UBC
	—	—	ID	ID	RW	RW	SZ	SZ	
BARA									UBC

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
BARA									UBC
BAMRA	—	—	—	—	—	BASM	BAM	BAM	UBC
BBRA	—	—	—	—	—	—	—	—	UBC
	—	—	ID	ID	RW	RW	SZ	SZ	
TRA	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—			
							—	—	
EXPEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
INTEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
MMUCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
BASRA									UBC
BASRB									UBC
CCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	RA	0	CF	CB	WT	CE	
PTEH									CCN
							—	—	

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
PTEL									CCN
							—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									CCN
TEA									CCN
INTEVT2									INT
IRR0									INT
	PINT0R	PINT1R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	
IRR1									INT
	TXI1R	BRI1R	RXI1R	ERI1R	DEI3R	DEI2R	DEI1R	DEI0R	
IRR2									INT
	—	—	—	ADIR	TXI2R	BRI2R	RXI2R	ERI2R	
ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S	INT
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
ICR2	INT15S	INT14S	INT13S	INT12S	INT11S	INT10S	INT9S	INT8S	INT
	INT7S	INT6S	INT5S	INT4S	INT3S	INT2S	INT1S	INT0S	
INTER	INT15E	INT14E	INT13E	INT12E	INT11E	INT10E	INT9E	INT8E	INT
	INT7E	INT6E	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E	
IPRC	IRQ3’s level				IRQ2’s level				INT
	IRQ1’s level				IRQ0’s level				

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
IPRD	PINT0 to 7’s level				PINT8 to 15’s level				INT
	IRQ5’s level				IRQ4’s level				
IPRE	DMAC’s level				IrDA’s level				INT
	SCIF’s level				A/D’s level				
									INT
SAR0									DMAC
DAR0									DMAC
DMATCR0	—	—	—	—	—	—	—	—	DMAC
CHCR0	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	RO	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR1									DMAC
DAR1									DMAC

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
DMATCR1	—	—	—	—	—	—	—	—	DMAC
CHCR1	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	RO	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR2									DMAC
DAR2									DMAC
DMATCR2	—	—	—	—	—	—	—	—	DMAC
CHCR2	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR3									DMAC

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
DAR3									DMAC
DMATCR3	—	—	—	—	—	—	—	—	DMAC
CHCR3	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	RO	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
DMAOR	—	—	—	—	—	—	PR1	PR0	DMAC
	—	—	—	—	—	AE	NMIF	DME	
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	—	STR	
CMCSR	—	—	—	—	—	—	—	—	CMT
	CMF	—	—	—	—	—	CKS1	CKS0	
CMCNT									CMT
CMCOR									CMT
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	A/D
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRBL	AD1	AD0	—	—	—	—	—	—	A/D
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRCL	AD1	AD0	—	—	—	—	—	—	A/D
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRDL	AD1	AD0	—	—	—	—	—	—	A/D
ADCSR	ADF	ADE	ADST	SCAN	CKS	CH2	CH1	CH0	A/D

Table B.2 Registers Bits (cont)

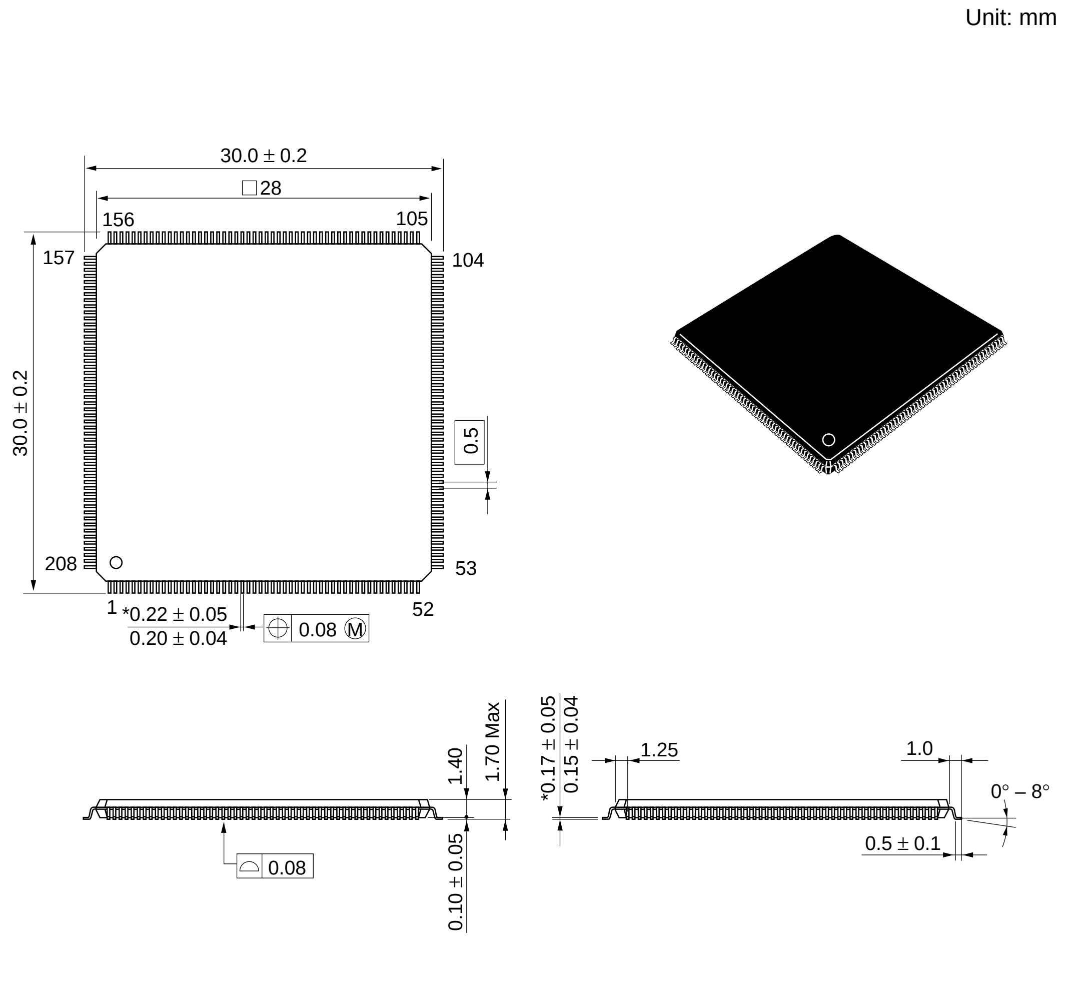
Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
ADCR	TRGE	—	—	—	—	—	—	—	A/D
DADR0									D/A
DADR1									D/A
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	D/A
DASTCR		—	—	—	—	—	—	DASTE	D/A
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	PORT
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	PORT
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	
PCDR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	PORT
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	PORT
	—	—	—	—	—	—	—	—	
PECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0	PORT
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
PFCR	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0	PORT
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0	
PGCR	PG7MD1	PG7MD0	PG6MD1	PG6MD0	PG5MD1	PG5MD0	PG4MD1	PG4MD0	PORT
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0	
PHCR	PH7MD1	PH7MD0	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0	PORT
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0	

Table B.2 Registers Bits (cont)

Register	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Module
PJCR	PJ7MD1	PJ7MD0	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4MD0	PORT
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0	
PKCR	PK7MD1	PK7MD0	PK6MD1	PK6MD0	PK5MD1	PK5MD0	PK4MD1	PK4MD0	PORT
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK0MD0	
PLCR	PL7MD1	PL7MD0	PL6MD1	PL6MD0	PL5MD1	PL5MD0	PL4MD1	PL4MD0	PORT
	PL3MD1	PL3MD0	PL2MD1	PL2MD0	PL1MD1	PL1MD0	PL0MD1	PL0MD0	
SCPCR	SCP7MD1	SCP7MD0	SCP6MD1	SCP6MD0	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0	PORT
	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCP0MD0	
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	PORT
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	PORT
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	PORT
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	PORT
PEDR	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	PORT
PFDR	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT	PORT
PGDR	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT	PORT
PHDR	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT	PORT
PJDR	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT	PORT
PKDR	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	PK1DT	PK0DT	PORT
PLDR	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	PL2DT	PL1DT	PL0DT	PORT
SCPDR	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT	PORT

Appendix C   Package Dimensions

Figure C.1 shows the SH7709 package dimensions.



*Dimension including the plating thickness
Base material dimension

Figure C.1   Package Dimensions(FP-208C)