SH7707

Hardware Manual

HITACHI

ADE-602-137
Rev.1.0
11/25/99
Hitachi, Ltd.
MC-Setsu

Hitachi semiconductor

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

Section 1 Overview and Pin Functions 1
1.1 SH7707 Features 1
1.2 Block Diagram 6
1.3 Pin Description 7
1.3.1 Pin Arrangement 7
1.3.2 Pin Functions 8

Section 2 CPU 17
2.1 Register Configuration 17
2.1.1 Privileged Mode and Banks 17
2.1.2 General Registers 20
2.1.3 System Registers 21
2.1.4 Control Registers 21
2.2 Data Formats 23
2.2.1 Data Format in Registers 23
2.2.2 Data Format in Memory 23
2.3 Instruction Features 24
2.3.1 Execution Environment 24
2.3.2 Addressing Modes 26
2.3.3 Instruction Formats 30
2.4 Instruction Set 33
2.4.1 Instruction Set Classified by Function 33
2.4.2 Instruction Code Map 49
2.5 Processor States and Processor Modes 52
2.5.1 Processor States 52
2.5.2 Processor Modes 53

Section 3 Memory Management Unit (MMU) 55
3.1 Overview 55
3.1.1 Features 55
3.1.2 Role of MMU 55
3.1.3 Virtual Address Space 57
3.1.4 Register Configuration 61
3.2 Register Descriptions 61
3.3 TLB Functions 64
3.3.1 Configuration of the TLB 64
3.3.2 TLB Indexing 66
3.3.3 TLB Address Comparison 67
3.3.4 Page Management Information 69

3.4 MMU Functions 70
3.4.1 MMU Hardware Management 70
3.4.2 MMU Software Management 70
3.4.3 MMU Instruction (LDLTB) 71
3.4.4 Avoiding Synonym Problems 72
3.5 MMU Exceptions 74
3.5.1 TLB Miss Exception 74
3.5.2 TLB Protection Violation Exception 75
3.5.3 TLB Invalid Exception 76
3.5.4 Initial Page Write Exception 77
3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error) 79
3.6 Memory-Mapped TLB 81
3.6.1 Address Array 81
3.6.2 Data Array 82
3.6.3 Usage Examples 84
3.7 Usage Note 84

Section 4 Exception Handling 85
4.1 Overview 85
4.1.1 Features 85
4.1.2 Register Configuration 85
4.2 Exception Handling Function 85
4.2.1 Exception Handling Flow 85
4.2.2 Exception Handling Vector Addresses 86
4.2.3 Acceptance of Exceptions 88
4.2.4 Exception Codes 90
4.2.5 Exception Request Masks 92
4.2.6 Returning from Exception Handling 93
4.3 Register Descriptions 93
4.4 Exception Handler Operation 94
4.4.1 Reset 94
4.4.2 Interrupts 94
4.4.3 General Exceptions 94
4.5 Individual Exception Operations 95
4.5.1 Resets 95
4.5.2 General Exceptions 96
4.5.3 Interrupts 99
4.6 Cautions 100

Section 5 Cache 103
5.1 Overview 103
5.1.1 Features 103

5.1.2 Cache Structure ... 103
5.1.3 Register Configuration ... 105
5.2 Register Description ... 105
5.2.1 Cache Control Register (CCR) ... 105
5.3 Cache Operation ... 106
5.3.1 Searching the Cache ... 106
5.3.2 Read Access ... 108
5.3.3 Write Access ... 108
5.3.4 Write-Back Buffer ... 108
5.3.5 Coherency of Cache and External Memory ... 109
5.3.6 RAM Mode ... 109
5.4 Memory-Mapped Cache ... 109
5.4.1 Address Array ... 109
5.4.2 Data Array ... 110
5.5 Usage Examples ... 112
5.5.1 Invalidating Specific Entries ... 112
5.5.2 Reading the Data of a Specific Entry ... 112

Section 6 Interrupt Controller (INTC) ... 113
6.1 Overview ... 113
6.1.1 Features ... 113
6.1.2 Block Diagram ... 114
6.1.3 Pin Configuration ... 115
6.1.4 Register Configuration ... 116
6.2 Interrupt Sources ... 117
6.2.1 NMI Interrupt ... 117
6.2.2 IRQ Interrupts ... 117
6.2.3 IRL Interrupts ... 118
6.2.4 PINT Interrupts ... 120
6.2.5 On-Chip Supporting Module Interrupts ... 120
6.2.6 Interrupt Exception Handling and Priority ... 121
6.3 INTC Registers ... 128
6.3.1 Interrupt Priority Registers A to F (IPRA–IPRF) ... 128
6.3.2 Interrupt Control Register 0 (ICR0) ... 129
6.3.3 Interrupt Control Register 1 (ICR1) ... 130
6.3.4 Interrupt Control Register 2 (ICR2) ... 133
6.3.5 PINT Interrupt Enable Register (PINTER) ... 134
6.3.6 Interrupt Request Register 0 (IRR0) ... 135
6.3.7 Interrupt Request Register 1 (IRR1) ... 137
6.3.8 Interrupt Request Register 2 (IRR2) ... 139
6.3.9 Interrupt Request Register 3 (IRR3) ... 141
6.3.10 Interrupt Request Register 4 (IRR4) ... 143
6.4 INTC Operation ... 145

6.4.1 Interrupt Sequence 145
6.4.2 Multiple Interrupts 147
6.5 Interrupt Response Time 147

Section 7 User Break Controller (UBC) 151
7.1 Overview 151
7.1.1 Features 151
7.1.2 Block Diagram 151
7.1.3 Register Configuration 153
7.1.4 Break Conditions and Register Settings 153
7.2 UBC Register Functions 154
7.2.1 Break Address Register A (BARA) 154
7.2.2 Break Address Register B (BARB) 154
7.2.3 Break ASID Register A (BASRA) 155
7.2.4 Break ASID Register B (BASRB) 155
7.2.5 Break Address Mask Register A (BAMRA) 155
7.2.6 Break Address Mask Register B (BAMRB) 156
7.2.7 Break Bus Cycle Register A (BBRA) 156
7.2.8 Break Bus Cycle Register B (BBRB) 157
7.2.9 Break Data Register B (BDRB) 158
7.2.10 Break Data Mask Register B (BDMRB) 159
7.2.11 Break Control Register (BRCR) 160
7.3 UBC Operation 162
7.3.1 User Break Operation Flow 162
7.3.2 Instruction Fetch Cycle Break 163
7.3.3 Data Access Cycle Break 164
7.3.4 Saved Program Counter (PC) Value 165
7.3.5 Examples of Use 166
7.3.6 Cautions 168

Section 8 Power-Down Modes 169
8.1 Overview 169
8.1.1 Power-Down Modes 169
8.1.2 Register Configuration 170
8.1.3 Pin Configuration 171
8.2 Registers 171
8.2.1 Standby Control Register (STBCR) 171
8.2.2 Standby Control Register 2 (STBCR2) 172
8.2.3 Standby Control Register 3 (STBCR3) 174
8.3 Sleep Mode 175
8.3.1 Transition to Sleep Mode 175
8.3.2 Canceling Sleep Mode 175
8.4 Standby Mode 176

8.4.1 Transition to Standby Mode 176
8.4.2 Canceling Standby Mode 177
8.4.3 Clock Pause Function 178
8.5 Module Standby Function 179
8.5.1 Transition to Module Standby Function 179
8.5.2 Canceling the Module Standby Function 180
8.6 Timing of STATUS Pin Changes 180
8.6.1 Timing for Resets 180
8.6.2 Timing for Canceling Standby Mode 182
8.6.3 Timing for Canceling Sleep Mode 184

Section 9 On-Chip Oscillation Circuits 187
9.1 Overview 187
9.1.1 Features 187
9.2 Overview of CPG 188
9.2.1 CPG Block Diagram 188
9.2.2 CPG Pin Configuration 190
9.2.3 CPG Register Configuration 190
9.3 Clock Operating Modes 190
9.4 Register Descriptions 198
9.4.1 Frequency Control Register (FRQCR) 198
9.5 Changing the Frequency 201
9.5.1 Changing the Multiplication Ratio 201
9.5.2 Changing the Division Ratio 201
9.6 PLL Standby Function 202
9.6.1 Overview of the PLL Standby Function 202
9.6.2 Usage 202
9.7 Controlling Clock Output 203
9.7.1 Clock Modes 0–1 203
9.7.2 Clock Modes 3–7 203
9.8 Overview of WDT 204
9.8.1 Block Diagram of WDT 204
9.8.2 Register Configurations 205
9.9 WDT Registers 205
9.9.1 Watchdog Timer Counter (WTCNT) 205
9.9.2 Watchdog Timer Control/Status Register (WTCSR) 206
9.9.3 Notes on Register Access 208
9.10 Using the WDT 209
9.10.1 Canceling Standby 209
9.10.2 Changing the Frequency 209
9.10.3 Using Watchdog Timer Mode 210
9.10.4 Using Interval Timer Mode 210
9.11 Notes on Board Design 211

9.11.1 When Using a Crystal Oscillator 211
9.11.2 When Using the PLL 211

Section 10 Bus State Controllers (BSC, BSCP) 213
10.1 Overview 213
10.1.1 Features 213
10.1.2 Block Diagram 215
10.1.3 Pin Configuration 218
10.1.4 Register Configuration 220
10.1.5 Area Overview 221
10.1.6 PCMCIA Support 224
10.2 BSC and BSCP Registers 229
10.2.1 Bus Control Register 1 (BCR1) 229
10.2.2 Bus Control Register 2 (BCR2) 232
10.2.3 Bus Control Register 3 (BCR3) 232
10.2.4 Wait State Control Register 1 (WCR1) 236
10.2.5 Wait State Control Register 2 (WCR2) 237
10.2.6 Individual Memory Control Register (MCR) 240
10.2.7 DRAM Control Register (DCR) 243
10.2.8 PCMCIA Control Register (PCR) 246
10.2.9 Refresh Timer Control/Status Register (RTCSR) 248
10.2.10 Refresh Timer Counter (RTCNT) 250
10.2.11 Refresh Time Constant Register (RTCOR) 250
10.2.12 Refresh Count Register (RFCR) 251
10.2.13 Notes on Accessing Refresh Control Related Registers 251
10.3 BSC/BSCP Operation 252
10.3.1 Endian/Access Size and Data Alignment 252
10.3.2 Description of Areas 256
10.3.3 Basic Interface 259
10.3.4 DRAM Interface 266
10.3.5 Burst ROM Interface 282
10.3.6 PCMCIA Interface 285
10.3.7 Waits between Access Cycles 294
10.3.8 Bus Arbitration 295

Section 11 PC Card Controller (PCC) 297
11.1 Overview 297
11.1.1 Features 297
11.1.2 Block Diagram 298
11.1.3 Register Configuration 299
11.1.4 PCMCIA Support 300
11.2 Register Descriptions 303
11.2.1 Area 6 Interface Status Register (PCC0ISR) 303

11.2.2 Area 6 General Control Register (PCC0GCR) 306
11.2.3 Area 6 Card Status Change Register (PCC0CSCR) 308
11.2.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER) 311
11.2.5 Area 5 Interface Status Register (PCC1ISR) 314
11.2.6 Area 5 General Control Register (PCC1GCR) 316
11.2.7 Area 5 Card Status Change Register (PCC1CSCR) 318
11.2.8 Area 6 Card Status Change Interrupt Enable Register (PCC1CSCIER) 320
11.3 Operation 322
11.3.1 PC card Connection Specifications (Interface Diagram, Pin Correspondence) 322
11.3.2 PC Card Interface Timing 326
11.3.3 Usage Notes 331

Section 12 Direct Memory Access Controller (DMAC) 333
12.1 Overview 333
12.1.1 Features 333
12.1. 2 Block Diagram 335
12.1.3 Pin Configuration 336
12.1.4 Register Configuration 337
12.2 Register Descriptions 338
12.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3) 338
12.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3) 339
12.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3) 340
12.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3) 341
12.2.5 DMA Operation Register (DMAOR) 346
12.3 Operation 348
12.3.1 DMA Transfer Flow 348
12.3.2 DMA Transfer Requests 350
12.3.3 Channel Priority Order 352
12.3.4 DMA Transfer Types 355
12.3.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing 364
12.3.6 Source Address Reload Function 370
12.3.7 DMA Transfer Ending Conditions 372
12.4 Compare Match Timer (CMT) 374
12.4.1 Overview 374
12.4.2 Register Configuration 375
12.4.3 Register Descriptions 375
12.4.4 Operation 378
12.4.5 Compare Match 379
12.5 Examples of Use 381
12.5.1 Example of DMA Transfer between On-Chip IRDA and External Memory 381
12.5.2 Example of DMA Transfer between AD0 and External Memory (Address Reload On) 381

12.5.3 Example of DMA Transfer between External Memory and SCIF Transmitter (SCIT2) (Indirect Address On) 383
12.6 Cautions 385

Section 13 Timer (TMU) 387
13.1 Overview 387
13.1.1 Features 387
13.1.2 Block Diagram 388
13.1.3 Pin Configuration 389
13.1.4 Register Configuration 389
13.2 TMU Registers 390
13.2.1 Timer Output Control Register (TOCR) 390
13.2.2 Timer Start Register (TSTR) 391
13.2.3 Timer Control Register (TCR) 392
13.2.4 Timer Constant Register (TCOR) 395
13.2.5 Timer Counters (TCNT) 396
13.2.6 Input Capture Register (TCPR2) 397
13.3 TMU Operation 398
13.3.1 Overview 398
13.3.2 Basic Functions 398
13.4 Interrupts 402
13.4.1 Timing of Status Flag Setting 402
13.4.2 Timing of Status Flag Clearing 403
13.4.3 Interrupt Sources and Priorities 403
13.5 Usage Notes 404
13.5.1 Writing to Registers 404
13.5.2 Reading Registers 404

Section 14 Real-Time Clock (RTC) 405
14.1 Overview 405
14.1.1 Features 405
14.1.2 Block Diagram 405
14.1.3 Pin Configuration 407
14.1.4 RTC Register Configuration 408
14.2 RTC Registers 408
14.2.1 64-Hz Counter (R64CNT) 408
14.2.2 Second Counter (RSECCNT) 409
14.2.3 Minute Counter (RMINCNT) 409
14.2.4 Hour Counter (RHRCNT) 410
14.2.5 Day of Week Counter (RWKCNT) 410
14.2.6 Date Counter (RDAYCNT) 411
14.2.7 Month Counter (RMONCNT) 412
14.2.8 Year Counter (RYRCNT) 412

14.2.9 Second Alarm Register (RSECAR) ... 413
14.2.10 Minute Alarm Register (RMINAR) ... 413
14.2.11 Hour Alarm Register (RHRAR) ... 414
14.2.12 Day of Week Alarm Register (RWKAR) ... 414
14.2.13 Date Alarm Register (RDAYAR) ... 415
14.2.14 Month Alarm Register (RMONAR) ... 416
14.2.15 RTC Control Register 1 (RCR1) ... 416
14.2.16 RTC Control Register 2 (RCR2) ... 418
14.3 RTC Operation ... 419
14.3.1 Initial Settings of Registers after Power-On ... 419
14.3.2 Setting the Time ... 419
14.3.3 Reading the Time ... 421
14.3.4 Alarm Function ... 422
14.3.5 Crystal Oscillator Circuit ... 422

Section 15 Serial Communication Interface (SCI) ... 425
15.1 Overview ... 425
15.1.1 Features ... 425
15.1.2 Block Diagram ... 426
15.1.3 Pin Configuration ... 429
15.1.4 Register Configuration ... 430
15.2 Register Descriptions ... 430
15.2.1 Receive Shift Register ... 430
15.2.2 Receive Data Register ... 430
15.2.3 Transmit Shift Register ... 431
15.2.4 Transmit Data Register ... 431
15.2.5 Serial Mode Register ... 431
15.2.6 Serial Control Register ... 434
15.2.7 Serial Status Register ... 437
15.2.8 Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR) ... 441
15.2.9 Bit Rate Register (SCBRR) ... 442
15.3 Operation ... 451
15.3.1 Overview ... 451
15.3.2 Operation in Asynchronous Mode ... 453
15.3.3 Multiprocessor Communication ... 463
15.3.4 Synchronous Operation ... 471
15.4 SCI Interrupt Sources ... 481
15.5 Usage Notes ... 481

Section 16 Smart Card Interface ... 485
16.1 Overview ... 485
16.1.1 Features ... 485
16.1.2 Block Diagram ... 486

16.1.3 Pin Configuration 487
16.1.4 Smart Card Interface Register Configuration 487
16.2 Register Descriptions 487
16.2.1 Smart Card Mode Register (SCSCMR) 488
16.2.2 Serial Status Register (SCSSR) 489
16.3 Operation 490
16.3.1 Overview 490
16.3.2 Pin Connections 491
16.3.3 Data Format 492
16.3.4 Register Settings 493
16.3.5 Clock 495
16.3.6 Data Transmission and Reception 498
16.4 Usage Notes 504
16.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode 504
16.4.2 Retransmission (Receive and Transmit Modes) 506

Section 17 Serial Communication Interface with FIFO (SCIF) 509
17.1 Overview 509
17.1.1 Features 509
17.1.2 Block Diagram 510
17.1.3 Pin Configuration 513
17.1.4 Register Configuration 514
17.2 Register Descriptions 514
17.2.1 Receive Shift Register 2 (SCRSR2) 514
17.2.2 Receive FIFO Data Register 2 (SCFRDR2) 514
17.2.3 Transmit Shift Register 2 (SCTSR2) 515
17.2.4 Transmit FIFO Data Register 2 (SCFTDR2) 515
17.2.5 Serial Mode Register 2 (SCSMR2) 515
17.2.6 Serial Control Register 2 (SCSCR2) 517
17.2.7 Serial Status Register 2 (SCSSR2) 519
17.2.8 Bit Rate Register 2 (SCBRR2) 523
17.2.9 FIFO Control Register 2 (SCFCR2) 531
17.2.10 FIFO Data Count Register 2 (SCFDR2) 533
17.3 Operation 533
17.3.1 Overview 533
17.3.2 Serial Operation 535
17.4 SCIF Interrupts 545
17.5 Usage Notes 546

Section 18 IRDA 549
18.1 Overview 549
18.1.1 Features 549
18.1.2 Block Diagram 550

18.1.3 Pin Configuration 553
18.1.4 Register Configuration 554
18.2 Register Description 555
18.2.1 Serial Mode Register 1 (SCSMR1) 555
18.3 Operation 557
18.3.1 Overview 557
18.3.2 Transmission 558
18.3.3 Reception 558

Section 19 Pin Function Controller 559
19.1 Overview 559
19.2 Register Configuration 563
19.3 Register Descriptions 564
19.3.1 Port A Control Register (PACR) 564
19.3.2 Port B Control Register (PBCR) 565
19.3.3 Port C Control Register (PCCR) 566
19.3.4 Port D Control Register (PDCR) 567
19.3.5 Port E Control Register (PECR) 569
19.3.6 Port F Control Register (PFCR) 570
19.3.7 Port G Control Register (PGCR) 571
19.3.8 Port H Control Register (PHCR) 572
19.3.9 Port J Control Register (PJCR) 574
19.3.10 Port K Control Register (PKCR) 575
19.3.11 Port L Control Register (PLCR) 576
19.3.12 Port SC Control Register (SCPCR) 577

Section 20 I/O Port 583
20.1 Overview 583
20.2 Port A 583
20.2.1 Register Description 583
20.2.2 Port A Data Register (PADR) 584
20.3 Port B 585
20.3.1 Register Description 585
20.3.2 Port B Data Register (PBDR) 586
20.4 Port C 587
20.4.1 Register Description 587
20.4.2 Port C Data Register (PCDR) 588
20.5 Port D 589
20.5.1 Register Description 589
20.5.2 Port D Data Register (PDDR) 590
20.6 Port E 591
20.6.1 Register Description 591
20.6.2 Port E Data Register (PEDR) 592

20.7 Port F 593
20.7.1 Register Description 593
20.7.2 Port F Data Register (PFDR) 594
20.8 Port G 595
20.8.1 Register Description 595
20.8.2 Port G Data Register (PGDR) 596
20.9 Port H 597
20.9.1 Register Description 597
20.9.2 Port H Data Register (PHDR) 598
20.10 Port J 599
20.10.1 Register Description 599
20.10.2 Port J Data Register (PJDR) 600
20.11 Port K 601
20.11.1 Register Description 601
20.11.2 Port K Data Register (PKDR) 602
20.12 Port L 603
20.12.1 Register Description 603
20.12.2 Port L Data Register (PLDR) 604
20.13 SC Port 605
20.13.1 Register Description 605
20.13.2 SC Port Data Register (SCPDR) 606

Section 21 LCD Controller 609
21.1 Overview 609
21.1.1 Features 609
21.1.2 Display Modes 610
21.1.3 Block Diagram 611
21.1.4 Pin Configuration 612
21.1.5 Register Configuration 613
21.2 Register Descriptions 619
21.2.1 Address Register (LCDAR) 619
21.2.2 Display Control Registers (LCDDR) 620
21.2.3 Palette Registers (LCDPR) 633
21.2.4 DMA Control Registers (LCDDMR) 636
21.3 Operation 639
21.3.1 Gradation Processing 639
21.3.2 Endian 641
21.3.3 Palette Register Settings 648
21.3.4 Determining the DIV Set Value (Clock Ratio) 650
21.3.5 LCD Data Output 654
21.3.6 Module Stop Function 659

Section 22 A/D Converter 661
22.1 Overview 661
22.1.1 Features 661
22.1.2 Block Diagram 662
22.1.3 Input Pins 663
22.1.4 Register Configuration 664
22.2 Register Descriptions 664
22.2.1 A/D Data Registers A to D (ADDRA to ADDRD) 664
22.2.2 A/D Control/Status Register (ADCSR) 665
22.2.3 A/D Control Register (ADCR) 667
22.3 Bus Master Interface 668
22.4 Operation 669
22.4.1 Single Mode (MULTI = 0) 669
22.4.2 Multi Mode (MULTI = 1) 672
22.4.3 Input Sampling and A/D Conversion Time 674
22.4.4 External Trigger Input Timing 675
22.5 Interrupts 676
22.6 Definitions of A/D Conversion Accuracy 676
22.7 A/D Converter Usage Notes 677
22.7.1 Setting Analog Input Voltage 677
22.7.2 Handling of Analog Input Pins 677

Section 23 D/A Converter 679
23.1 Overview 679
23.1.1 Features 679
23.1.2 Block Diagram 679
23.1.3 Input/Output Pins 680
23.1.4 Register Configuration 680
23.2 Register Descriptions 681
23.2.1 D/A Data Registers 0 and 1 (DADR0/1) 681
23.2.2 D/A Control Register (DACR) 681
23.3 Operation 683

Section 24 Electrical Characteristics 685
24.1 Absolute Maximum Ratings 685
24.2 DC Characteristics 686
24.3 AC Characteristics 688
24.3.1 Clock Timing 689
24.3.2 Control Signal Timing 698
24.3.3 AC Bus Timing Specifications 702
24.3.4 Basic Timing 706
24.3.5 Burst ROM Timing 709

24.3.6 DRAM Timing 712
24.3.7 PCMCIA Timing 726
24.3.8 Peripheral Module Signal Timing 733
24.3.9 AC Characteristics Measurement Conditions 751
24.4 A/D Conversion Characteristics 753
24.5 D/A Conversion Characteristics 753

Appendix A Pin Functions 755
A.1 Pin States 755
A.2 Pin Specifications 759
A.3 Handling of Unused Pins 764
A.4 Pin States in Access to Each Address Space 765

Appendix B Control Registers 789
B.1 Register Address Map 789

Appendix C Load Time Variation Due to Load Capacitance 816

Appendix D Package Dimensions 817

Section 1 Overview and Pin Functions

1.1 SH7707 Features

The SH7707 is a single-chip RISC (reduced instruction set computer) microcomputer that integrates a Hitachi-original RISC-type SuperH architecture CPU core that has an on-chip multiplier, cache memory, and a memory management unit, together with on-chip supporting functions required for system configuration such as a timer, a realtime clock, an interrupt controller, and a serial communication interface. The SH7707 includes data protection and virtual memory functions, and was designed by building a memory management unit onto an SuperH RISC engine family (SH7000 or SH7600 Series).

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 various kinds of memory. The SH7707 microcomputer also supports an LCD controller, an infrared communication function, a PCMCIA interface, an A/D converter, a D/A converter, and a PLL which multiplies an 11-MHz or 15-MHz main clock from a 32-kHz clock pulse generator for the real-time clock.

A powerful built-in power management function keeps power consumption low, even during high-speed operation. The SH7707 can run at four times the frequency of the system bus operating speed, providing both high speed and low power consumption.

The features of the SH7707 are listed in table 1.1.

Table 1.1 SH7707 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 SH7600 Series) — 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 (384-Mbyte actual memory space) • Space identifier ASID: 8 bits, 256 logical address spaces • 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 using PLL synchronization • High-multiplication PLL using the RTC 32 kHz frequency as its source oscillation • Processing states: — Power-on reset state — Manual reset state — Exception-handling state — Program execution state — Power-down state — Bus-released state • Power-down modes: — Sleep mode — Standby mode — Module standby mode • On-chip clock pulse generator consists of a 2-MHz to 20-MHz crystal oscillator circuit, 32-kHz generator, and PLL circuits which multiply the frequency

Table 1.1 SH7707 Features (cont)

Item	Features
Memory management unit	• 4 Gbytes of address space, 256 address spaces (8-bit ASID) • Page unit sharing • Supports multiple page sizes: 1 or 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 instructions/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–IRQ4, IRQ3/$\overline{\text{IRL3}}$–IRQ0/$\overline{\text{IRL0}}$), • On-chip supporting module 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 controllers	• 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, and burst ROM — Support for PCMCIA interface (2 channels) — Output of chip select signal (CS0, CS2–CS6) for corresponding area • DRAM refresh function • DRAM burst access function • DRAM short-pitch access function • Usable as either big- or little-endian machine

Table 1.1 SH7707 Features (cont)

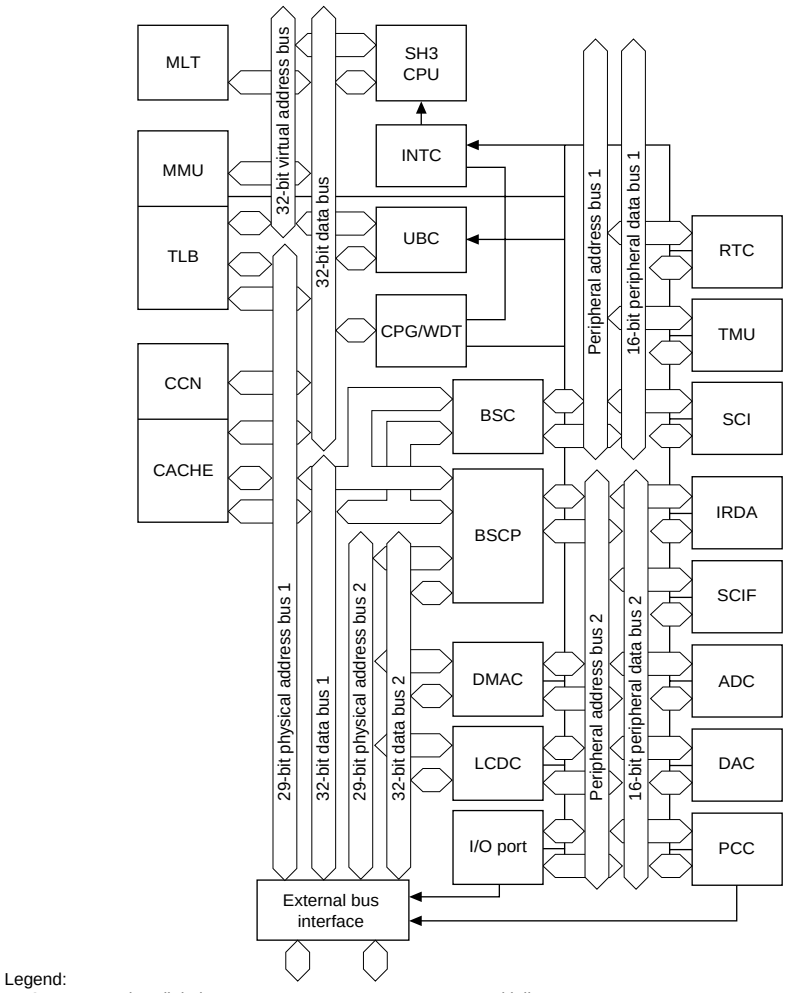
Item	Features
Timer	• 3-channel auto-reload type 32-bit timer • Input capture function • 6 types of counter input clocks can be selected • Maximum resolution: 2 MHz
Real-time clock	• Built-in clock, calendar functions, and alarm functions • On-chip 32-kHz crystal oscillator circuit with a maximum resolution (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 transmission/reception • DMA transfer capability • IrDA: interface based on the IrDA 1.0 system
Serial communication interface 2	• 16-byte FIFO for transmission/reception • DMA transfer capability • Hardware flow control
LCD controller	• Resolution — Max. 640 × 480, register programmable • LCD driver interface — 4-bit (STN monochrome, single screen) — 8-bit (STN monochrome, single screen) — 4-bit × 2 (STN monochrome, dual screens) — 6-bit parallel (color TFT/TFD, 2 bits each for R/G/B) — 8-bit (color STN) • LCD clock output — CL1 (latch clock), CL2 (shift clock), and FLM (first line marker) • Other output signals on some LCD driver interfaces only — DTM (display timing) and M (LCD driving signal alternation) • 2-channel dedicated DMAC

Table 1.1 SH7707 Features (cont)

Item	Features
DMAC	• Four channels • Transfer in physical address space • Transfer data width: 1/2/4 byte(s) • Maximum transfer count: 16 M (16,777,216) • Dual address mode • DMA request — External pin (DREQ): channel 0, 1 — On-chip modules (IRDA, SCIF, ADC, CMT): all channels — Auto-request: all channels • Cycle-steal mode or burst mode
I/O port	• 16 bits (with 16-bit external bus) • Maximum 99 bits by pin multiplexing
PCMCIA	• Supports 2 channels • Rev. 2.1 • Control signal output
A/D converter	• 10 bits ± 4 LSB, 8 channels • Conversion time: 10 µs • Input range: 0–AV_{CC} (max. 3.6 V)
D/A converter	• 8 bits ± 4 LSB, 2 channels • Conversion time: 10 µs • Output range: 0–AV_{CC} (max. 3.6 V)
Package	• 208-pin plastic QFP (FP-208A) • 216-pin CSP (CSP-216)

1.2 Block Diagram

Figure 1.1 shows a functional block diagram of the SH7707.



Legend:

ADC:	Analog-digital converter	MLT:	Multiplier
BSC:	Bus state controller 1	MMU:	Memory management unit
BSCP:	Bus state controller 2	PCC:	PC card controller
CACHE:	Cache memory	RTC:	Real-time clock
CCN:	Cache memory controller	SCI:	Serial communication interface (with smart card interface)
CPG/WDT:	Clock pulse generator/watchdog timer	IRDA:	Serial communication interface (with IRDA)
CPU:	Central processing unit	SCIF:	Serial communication interface (with FIFO)
DAC:	Digital-analog converter	TLB:	Translation look-aside buffer
DMAC:	Direct memory access controller	TMU:	Timer unit
INTC:	Interrupt controller	UBC:	User break controller
LCDC:	Liquid crystal display controller		

Figure 1.1 SH7707 Functional Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

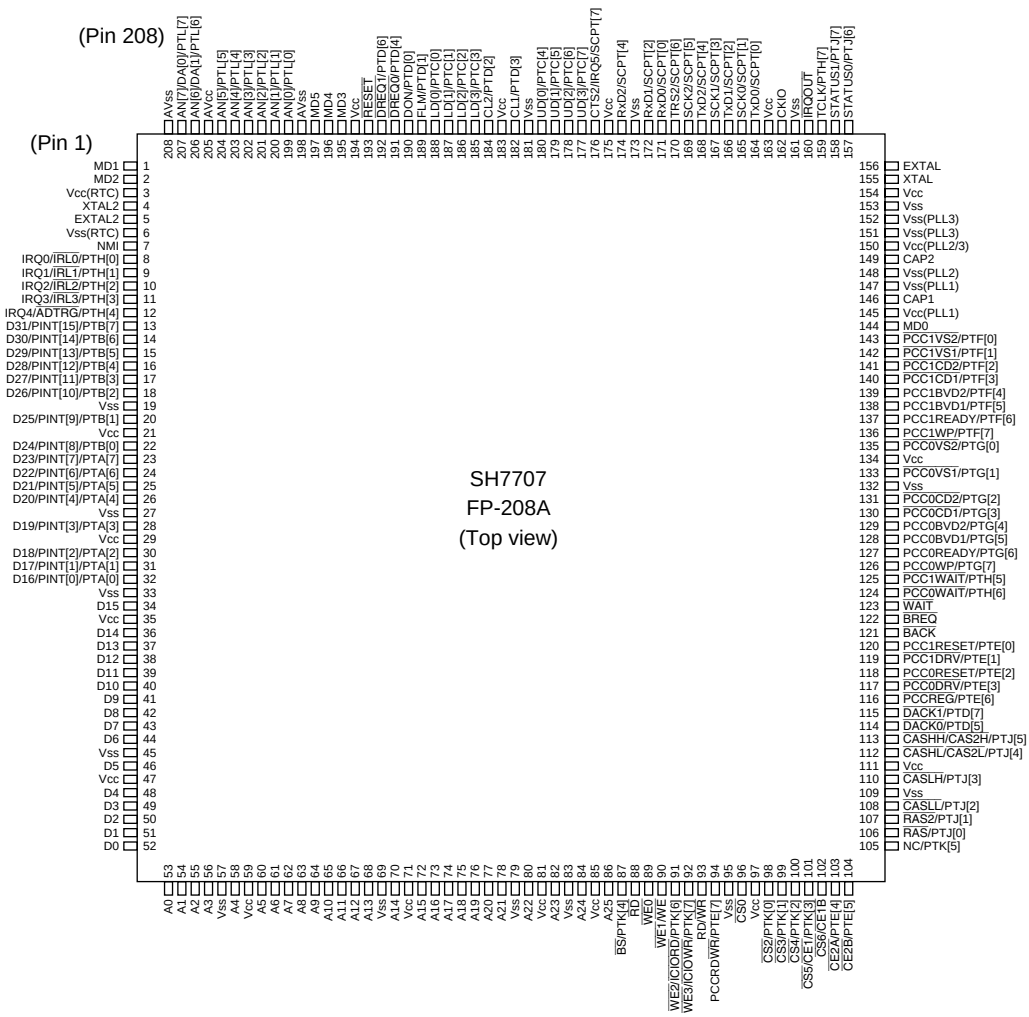
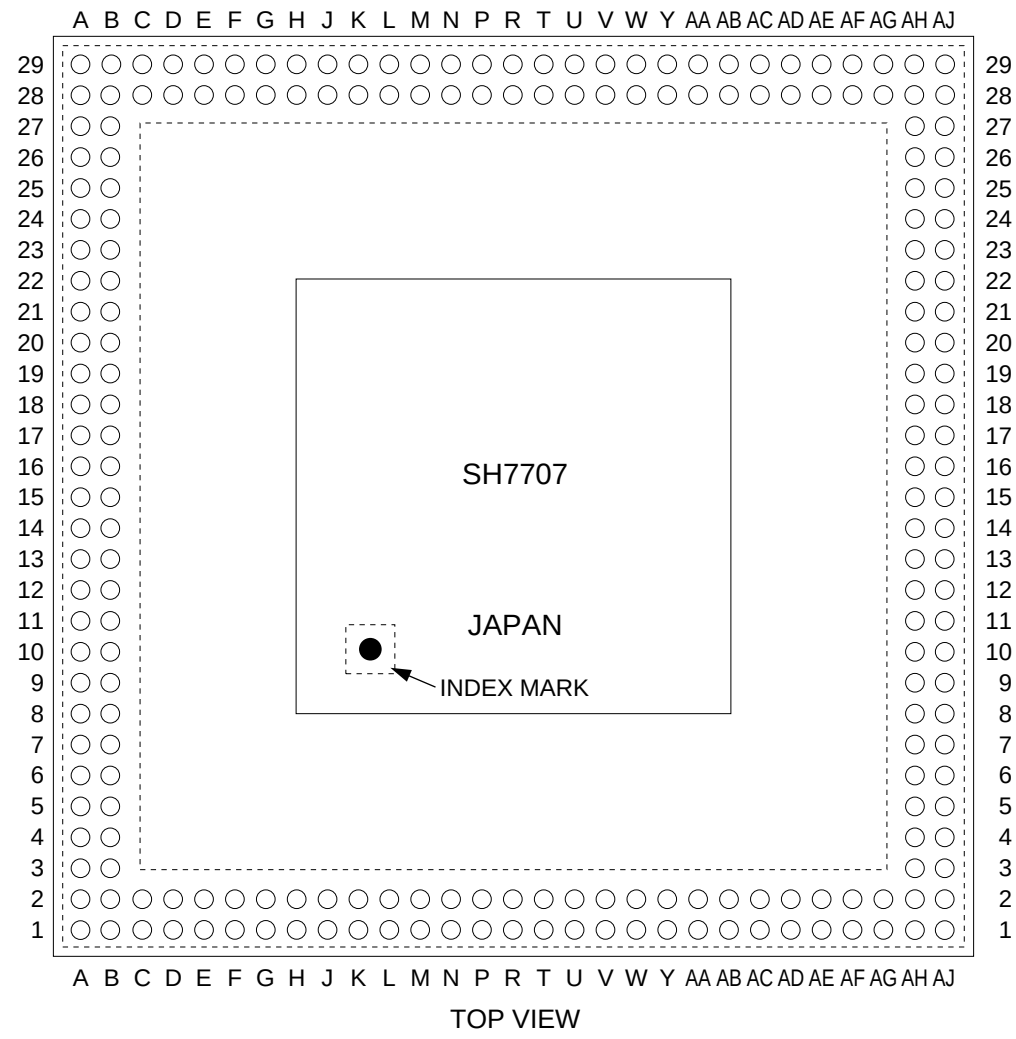


Figure 1.2 (a) Pin Arrangement (208-Pin Plastic QFP)



Note: The terminal area surrounded by dotted line is drawn in perspective.

Figure 1.2 (b) Pin Arrangement (216-Pin CSP)

1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
1	B02	MD1	I	Mode pin
2	A02	MD2	I	Mode pin
3	B03	Vcc(RTC)	Power*1	Dedicated power supply for RTC oscillator (3.3 V)
4	A03	XTAL2	O	Crystal resonator pin (for on-chip RTC)
5	B04	EXTAL2	I	Crystal resonator pin (for on-chip RTC)
6	A04	Vss(RTC)	Power*1	Dedicated power supply for RTC oscillator (0 V)
7	B05	NMI	I	Nonmaskable interrupt request
8	A05	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	I	External interrupt request / external interrupt source / input port
9	B06	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	I	External interrupt request / external interrupt source / input port
10	A06	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	I	External interrupt request / external interrupt source / input port
11	B07	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	I	External interrupt request / external interrupt source / input port
12	A07	IRQ4/$\overline{\text{ADTRG}}$/PTH[4]	I	External interrupt request / ADC trigger request / input port
13	B08	D31/PINT[15]/PTB[7]	I/O	Data bus / interrupt request pin / I/O port
14	A08	D30/PINT[14]/PTB[6]	I/O	Data bus / interrupt request pin / I/O port
15	B09	D29/PINT[13]/PTB[5]	I/O	Data bus / interrupt request pin / I/O port
16	A09	D28/PINT[12]/PTB[4]	I/O	Data bus / interrupt request pin / I/O port
17	B10	D27/PINT[11]/PTB[3]	I/O	Data bus / interrupt request pin / I/O port
18	A10	D26/PINT[10]/PTB[2]	I/O	Data bus / interrupt request pin / I/O port
19	B11	Vss	Power*2	Power supply (0 V)

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
20	A11	D25/PINT[9]/PTB[1]	I/O	Data bus / interrupt request pin / I/O port
21	B12	Vcc	Power*[2]	Power supply (3.3 V)
22	A12	D24/PINT[8]/PTB[0]	I/O	Data bus / interrupt request pin / I/O port
23	B13	D23/PINT[7]/PTA[7]	I/O	Data bus / interrupt request pin / I/O port
24	A13	D22/PINT[6]/PTA[6]	I/O	Data bus / interrupt request pin / I/O port
25	B14	D21/PINT[5]/PTA[5]	I/O	Data bus / interrupt request pin / I/O port
26	A14	D20/PINT[4]/PTA[4]	I/O	Data bus / interrupt request pin / I/O port
27	B15	Vss	Power*[2]	Power supply (0 V)
28	A15	D19/PINT[3]/PTA[3]	I/O	Data bus / interrupt request pin / I/O port
29	B16	Vcc	Power*[2]	Power supply (3.3 V)
30	A16	D18/PINT[2]/PTA[2]	I/O	Data bus / interrupt request pin / I/O port
31	B17	D17/PINT[1]/PTA[1]	I/O	Data bus / interrupt request pin / I/O port
32	A17	D16/PINT[0]/PTA[0]	I/O	Data bus / interrupt request pin / I/O port
33	B18	Vss	Power*[2]	Power supply (0 V)
34	A18	D15	I/O	Data bus
35	B19	Vcc	Power*[2]	Power supply (3.3 V)
36	A19	D14	I/O	Data bus
37	B20	D13	I/O	Data bus
38	A20	D12	I/O	Data bus
39	B21	D11	I/O	Data bus
40	A21	D10	I/O	Data bus
41	B22	D9	I/O	Data bus
42	A22	D8	I/O	Data bus
43	B23	D7	I/O	Data bus

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
44	A23	D6	I/O	Data bus
45	B24	Vss	Power*2	Power supply (0 V)
46	A24	D5	I/O	Data bus
47	B25	Vcc	Power*2	Power supply (3.3 V)
48	A25	D4	I/O	Data bus
49	B26	D3	I/O	Data bus
50	A26	D2	I/O	Data bus
51	B27	D1	I/O	Data bus
52	A27	D0	I/O	Data bus
53	B28	A0	O	Address bus
54	B29	A1	O	Address bus
55	C28	A2	O	Address bus
56	C29	A3	O	Address bus
57	D28	Vss	Power*2	Power supply (0 V)
58	D29	A4	O	Address bus
59	E28	Vcc	Power*2	Power supply (3.3 V)
60	E29	A5	O	Address bus
61	F28	A6	O	Address bus
62	F29	A7	O	Address bus
63	G28	A8	O	Address bus
64	G29	A9	O	Address bus
65	H28	A10	O	Address bus
66	H29	A11	O	Address bus
67	J28	A12	O	Address bus
68	J29	A13	O	Address bus
69	K28	Vss	Power*2	Power supply (0 V)
70	K29	A14	O	Address bus
71	L28	Vcc	Power*2	Power supply (3.3 V)
72	L29	A15	O	Address bus
73	M28	A16	O	Address bus
74	M29	A17	O	Address bus

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
75	N28	A18	O	Address bus
76	N29	A19	O	Address bus
77	P28	A20	O	Address bus
78	P29	A21	O	Address bus
79	R28	Vss	Power*2	Power supply (0 V)
80	R29	A22	O	Address bus
81	T28	Vcc	Power*2	Power supply (3.3 V)
82	T29	A23	O	Address bus
83	U28	Vss	Power*2	Power supply (0 V)
84	U29	A24	O	Address bus
85	V28	Vcc	Power*2	Power supply (3.3 V)
86	V29	A25	O	Address bus
87	W28	$\overline{BS}$/PTK[4]	I/O	Bus cycle start / I/O port
88	W29	$\overline{RD}$	O	Read strobe pin
89	Y28	$\overline{WE0}$	O	D7–D0 select signal
90	Y29	$\overline{WE1}$/$\overline{WE}$	O	D15–D8 select signal / PCMCIA $\overline{WE}$ signal
91	AA28	$\overline{WE2}$/$\overline{ICIORD}$/PTK[6]	I/O	D23–D16 select signal / PCMCIA $\overline{IORD}$ signal / I/O port
92	AA29	$\overline{WE3}$/$\overline{ICIOWR}$/PTK[7]	I/O	D31–D24 select signal / PCMCIA $\overline{IOWR}$ signal / I/O port
93	AB28	RD/$\overline{WR}$	O	Read/write switchover signal
94	AB29	PCCRD$\overline{WR}$/PTE[7]	I/O	PCMCIA read/write switchover signal
95	AC28	Vss	Power*2	Power supply (0 V)
96	AC29	$\overline{CS0}$	O	Chip select 0
97	AD28	Vcc	Power*2	Power supply (3.3 V)
98	AD29	$\overline{CS2}$/PTK[0]	I/O	Chip select 2 / I/O port
99	AE28	$\overline{CS3}$/PTK[1]	I/O	Chip select 3 / I/O port
100	AE29	$\overline{CS4}$/PTK[2]	I/O	Chip select 4 / I/O port
101	AF28	$\overline{CS5}$/$\overline{CE1A}$/PTK[3]	I/O	Chip select 5 / PCMCIA CE1A / I/O port
102	AF29	$\overline{CS6}$/$\overline{CE1B}$	O	Chip select 6 / PCMCIA CE1B

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
103	AG28	$\overline{\text{CE2A}}$/PTE[4]	I/O	PCMCIA CE2A / I/O port
104	AG29	$\overline{\text{CE2B}}$/PTE[5]	I/O	PCMCIA CE2B / I/O port
105	AH28	NC/PTK[5]	I/O	I/O port
106	AJ28	$\overline{\text{RAS}}$/PTJ[0]	I/O	Area 3 RAS / I/O port
107	AH27	$\overline{\text{RAS2}}$/PTJ[1]	I/O	Area 2 RAS / I/O port
108	AJ27	$\overline{\text{CASLL}}$/PTJ[2]	I/O	D7–D0 selection CAS / I/O port
109	AH26	Vss	Power*2	Power supply (0 V)
110	AJ26	$\overline{\text{CASLH}}$/PTJ[3]	I/O	D15–D8 selection CAS / I/O port
111	AH25	Vcc	Power*2	Power supply (3.3 V)
112	AJ25	$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$/PTJ[4]	I/O	D23–D16 selection CAS / D7–D0 selection CAS in area 2 / I/O port
113	AH24	$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$/PTJ[5]	I/O	D31–D24 selection CAS / D15–D8 selection CAS in area 2 / I/O port
114	AJ24	$\overline{\text{DACK0}}$/PTD[5]	I/O	DMA transfer strobe 0 / I/O port
115	AH23	$\overline{\text{DACK1}}$/PTD[7]	I/O	DMA transfer strobe 1 / I/O port
116	AJ23	$\overline{\text{PCCREG}}$/PTE[6]	I/O	PCMCIA REG pin / I/O port
117	AH22	$\overline{\text{PCC0DRV}}$/PTE[3]	I/O	PCMCIA0 buffer control pin / I/O port
118	AJ22	PCC0RESET/PTE[2]	I/O	PCMCIA0 reset output / I/O port
119	AH21	$\overline{\text{PCC1DRV}}$/PTE[1]	I/O	PCMCIA1 buffer control pin / I/O port
120	AJ21	PCC1RESET/PTE[0]	I/O	PCMCIA1 reset output / I/O port
121	AH20	$\overline{\text{BACK}}$	O	Bus acknowledge
122	AJ20	$\overline{\text{BREQ}}$	I	Bus request
123	AH19	$\overline{\text{WAIT}}$	I	Hardware wait request
124	AJ19	$\overline{\text{PCC0WAIT}}$/PTH[6]	I	PCMCIA0 hardware wait request / input port
125	AH18	$\overline{\text{PCC1WAIT}}$/PTH[5]	I	PCMCIA1 hardware wait request / input port
126	AJ18	PCC0WP/PTG[7]	I	PCMCIA0 WP pin / input port
127	AH17	PCC0READY/PTG[6]	I	PCMCIA0 BUSY/READY pin / input port
128	AJ17	PCC0BVD1/PTG[5]	I	PCMCIA0 BVD1 pin / input port
129	AH16	PCC0BVD2/PTG[4]	I	PCMCIA0 BVD2 pin / input port

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
130	AJ16	$\overline{\text{PCC0CD1}}$/PTG[3]	I	PCMCIA0 CD1 pin / input port
131	AH15	$\overline{\text{PCC0CD2}}$/PTG[2]	I	PCMCIA0 CD2 pin / input port
132	AJ15	Vss	Power*2	Power supply (0 V)
133	AH14	$\overline{\text{PCC0VS1}}$/PTG[1]	I	PCMCIA0 VS1 pin / input port
134	AJ14	Vcc	Power*2	Power supply (3.3 V)
135	AH13	$\overline{\text{PCC0VS2}}$/PTG[0]	I	PCMCIA0 VS2 pin / input port
136	AJ13	PCC1WP/PTF[7]	I	PCMCIA1 WP pin / input port
137	AH12	PCC1READY/PTF[6]	I	PCMCIA1 BUSY/READY pin / input port
138	AJ12	PCC1BVD1/PTF[5]	I	PCMCIA1 BVD1 pin / input port
139	AH11	PCC1BVD2/PTF[4]	I	PCMCIA1 BVD2 pin / input port
140	AJ11	$\overline{\text{PCC1CD1}}$/PTF[3]	I	PCMCIA1 CD1 pin / input port
141	AH10	$\overline{\text{PCC1CD2}}$/PTF[2]	I	PCMCIA1 CD2 pin / input port
142	AJ10	$\overline{\text{PCC1VS1}}$/PTF[1]	I	PCMCIA1 VS1 pin / input port
143	AH09	$\overline{\text{PCC1VS2}}$/PTF[0]	I	PCMCIA1 VS2 pin / input port
144	AJ09	MD0	I	Mode pin
145	AH08	Vcc(PLL1)	Power*3	PLL power supply (3.3 V)
146	AJ08	CAP1	—	External capacitance pin (for PLL1)
147	AH07	Vss(PLL1)	Power*3	PLL power supply (0 V)
148	AJ07	Vss(PLL2)	Power*3	PLL power supply (0 V)
149	AH06	CAP2	—	External capacitance pin (for PLL2)
150	AJ06	Vcc(PLL2/3)	Power*3	PLL power supply (3.3 V)
151	AH05	Vss(PLL2)	Power*3	PLL power supply (0 V)
152	AJ05	Vss(PLL3)	Power*3	PLL power supply (0 V)
153	AH04	Vss	Power*2	Power supply (0 V)
154	AJ04	Vcc	Power*2	Power supply (3.3 V)
155	AH03	XTAL	O	Crystal resonator pin
156	AJ03	EXTAL	I	External clock / crystal resonator pin
157	AH02	STATUS0/PTJ[6]	I/O	Processing status / I/O port
158	AH01	STATUS1/PTJ[7]	I/O	Processing status / I/O port

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
159	AG02	TCLK/PTH[7]	I/O	Clock input/output (for TMU/RTC) / I/O port
160	AG01	$\overline{\text{IRQOUT}}$	O	Interrupt / refresh request notification
161	AF02	Vss	Power*2	Power supply (0 V)
162	AF01	CKIO	I/O	System clock input/output
163	AE02	Vcc	Power*2	Power supply (3.3 V)
164	AE01	TxD0/SCPT[0]	O	Serial port 0 data output / output port
165	AD02	SCK0/SCPT[1]	I/O	Serial port 0 clock input/output / I/O port
166	AD01	TxD1/SCPT[2]	O	Serial port 1 data output / output port
167	AC02	SCK1/SCPT[3]	I/O	Serial port 1 clock input/output / I/O port
168	AC01	TxD2/SCPT[4]	O	Serial port 2 data output / output port
169	AB02	SCK2/SCPT[5]	I/O	Serial port 2 clock input/output / I/O port
170	AB01	RTS2/SCPT[6]	I/O	Serial port 2 transmit request / I/O port
171	AA02	RxD0/SCPT[0]	I	Serial port 0 data input/ I/O port
172	AA01	RxD1/SCPT[2]	I	Serial port 1 data input/ I/O port
173	Y02	Vss	Power*2	Power supply (0 V)
174	Y01	RxD2/SCPT[4]	I	Serial port 2 data input/ input port
175	W02	Vcc	Power*2	Power supply (3.3 V)
176	W01	CTS2/IRQ5/SCPT[7]	I	Serial port 2 transmit enable / input port
177	V02	UD[3]/PTC[7]	I/O	LCD data output / I/O port
178	V01	UD[2]/PTC[6]	I/O	LCD data output / I/O port
179	U02	UD[1]/PTC[5]	I/O	LCD data output / I/O port
180	U01	UD[0]/PTC[4]	I/O	LCD data output / I/O port
181	T02	Vss	Power*2	Power supply (0 V)
182	T01	CL1/PTD[3]	I/O	LCD clock output / I/O port
183	R02	Vcc	Power*2	Power supply (0 V)
184	R01	CL2/PTD[2]	I/O	LCD clock output / I/O port
185	P02	LD[3]/PTC[3]	I/O	LCD data output / I/O port

Table 1.2 Pin Functions (cont)

Pin No. (FP-208A)	Pin No. (CSP-216)	Pin Name	I/O	Function
186	P01	LD[2]/PTC[2]	I/O	LCD data output / I/O port
187	N02	LD[1]/PTC[1]	I/O	LCD data output / I/O port
188	N01	LD[0]/PTC[0]	I/O	LCD data output / I/O port
189	M02	FLM/PTD[1]	I/O	LCD control pin / I/O port
190	M01	DON/PTD[0]	I/O	LCD control pin / I/O port
191	L02	$\overline{\text{DREQ0}}$/PTD[4]	I	DMA transfer request 0 / input port
192	L01	$\overline{\text{DREQ1}}$/PTD[6]	I	DMA transfer request 1 / input port
193	K02	$\overline{\text{RESET}}$	I	Reset pin
194	K01	Vcc	Power*2	Power supply (3.3 V)
195	J02	MD3	I	Mode pin
196	J01	MD4	I	Mode pin
197	H02	MD5	I	Mode pin
198	H01	AVss	Power*4	ADC/DAC power supply (0 V)
199	G02	AN[0]/PTL[0]	I	Analog input pin / input port
200	G01	AN[1]/PTL[1]	I	Analog input pin / input port
201	F02	AN[2]/PTL[2]	I	Analog input pin / input port
202	F01	AN[3]/PTL[3]	I	Analog input pin / input port
203	E02	AN[4]/PTL[4]	I	Analog input pin / input port
204	E01	AN[5]/PTL[5]	I	Analog input pin / input port
205	D02	AVcc	Power*4	ADC/DAC power supply (3.3 V)
206	D01	AN[6]/DA[1]/PTL[6]	I/O	Analog input/output pin / input port
207	C02	AN[7]/DA[0]/PTL[7]	I/O	Analog input/output pin / input port
208	C01	AVss	Power*4	ADC/DAC power supply (0 V)

Notes:
1. Power must be supplied regardless of whether or not the RTC is used.
2. Connect all V_{CC}/V_{SS} pins to the system power supply (provide a constant power supply). When only the RTC is operating (in standby mode), power must still be supplied to all V_{CC}/V_{SS} power supply pins, not only to V_{CC} (RTC) and V_{SS} (RTC).
3. Power must be supplied regardless of whether or not the on-chip PLL circuits are used.
4. Power must be supplied regardless of whether or not the on-chip ADC/DAC is used.
5. Pin No. A01, A28, A29, B01, AH29, AJ01, AJ02 and AJ29 of CSP package are NC (No Connect) pins. These must be left open.

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 SH7707 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.

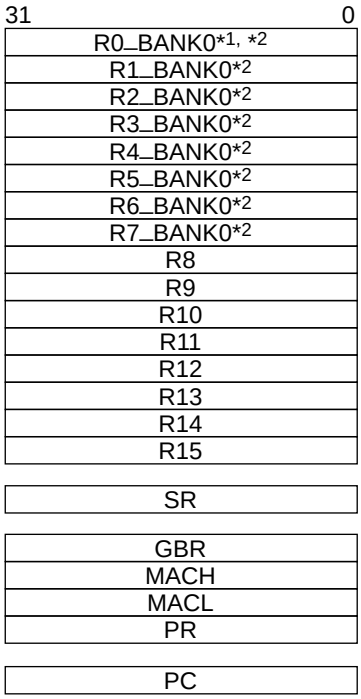
When the RB bit is 0, BANK0 general registers R0_BANK0–R7_BANK0 and nonbanked general registers R8–R15 function as the general register set, with BANK1 general registers R0_BANK1–R7_BANK1 accessed only by the LDC/STC instructions. In user mode, the 16 registers comprising bank 0 general registers R0_BANK0–R7_BANK0 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.

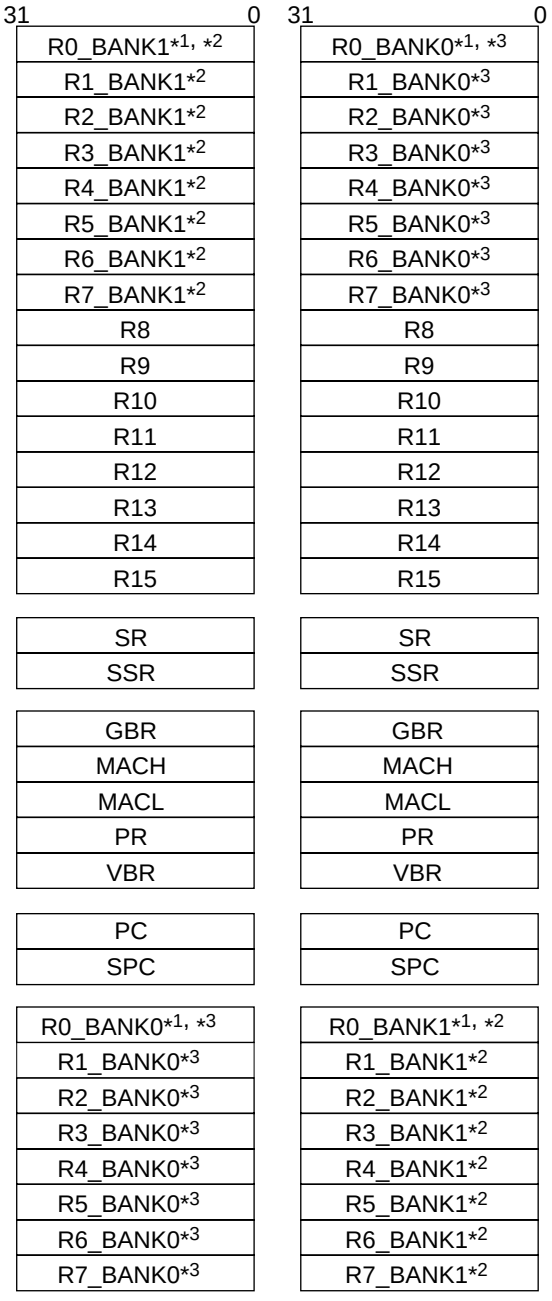
Switching between user mode and privileged mode is controlled by the processor mode bit (MD) in the status register.



User mode register configuration

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 in 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 an LDC/STC instruction.
3. Banked register
   When the RB bit in 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 an 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.

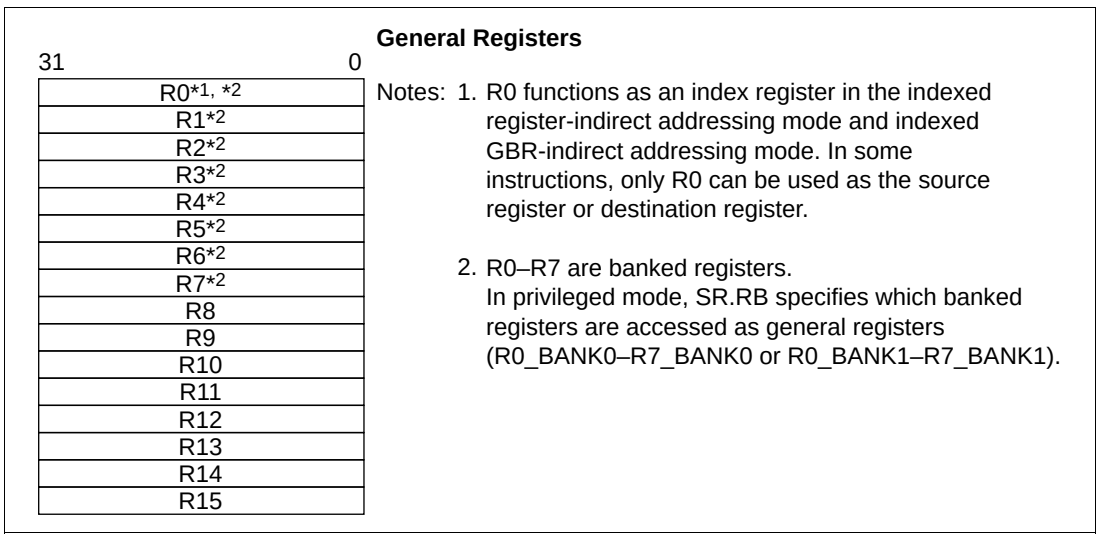


Figure 2.3 General Registers

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 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.

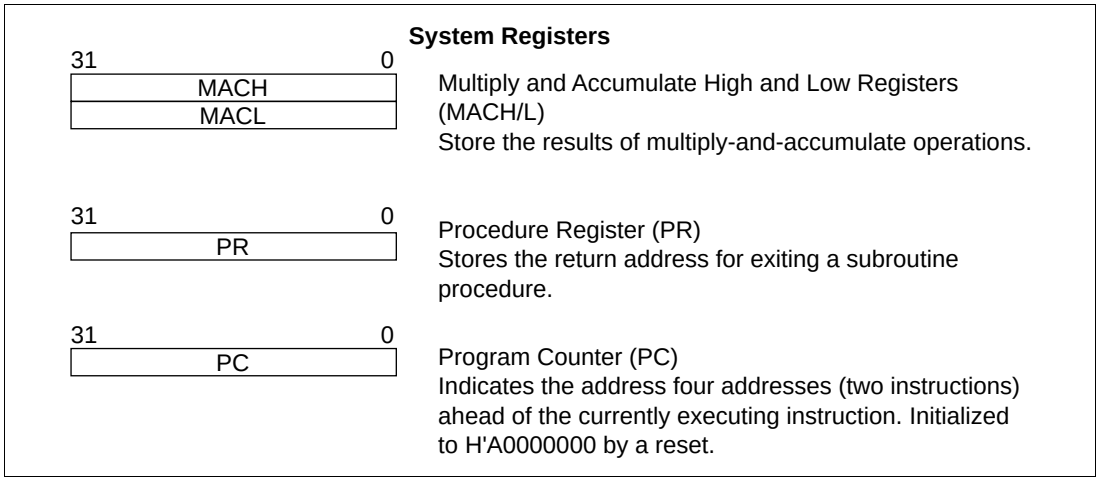
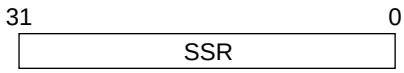


Figure 2.4 System Registers

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:

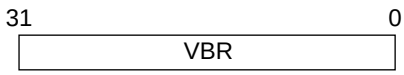
- Status register (SR)
- Saved status register (SSR)
- Saved program counter (SPC)
- Global base register (GBR)
- Vector base register (VBR)



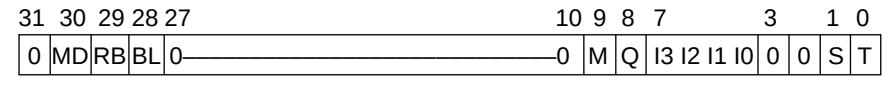
Saved Status Register (SSR)
Stores current SR value at time of exception to indicate processor status in return to instruction stream from exception handler.

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.



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.

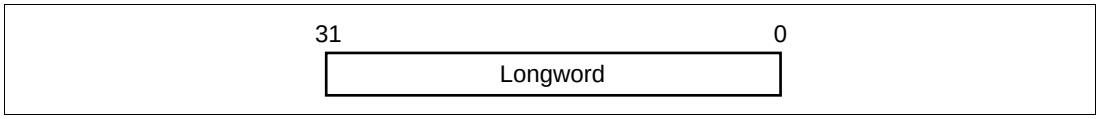


Figure 2.6 Longword

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.

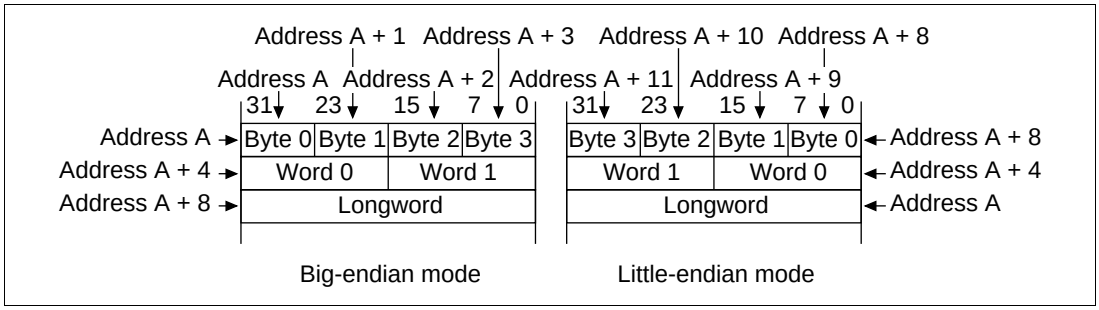


Figure 2.7 Byte, Word, and Longword Alignment

2.3 Instruction Features

2.3.1 Execution Environment

Data Length: The SH7707 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 SH7707 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 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

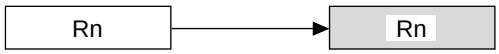
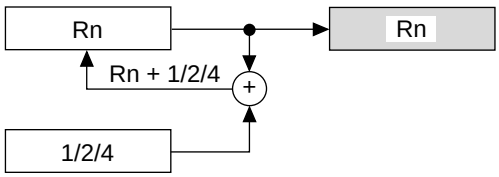
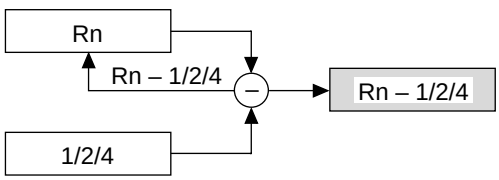
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective 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)

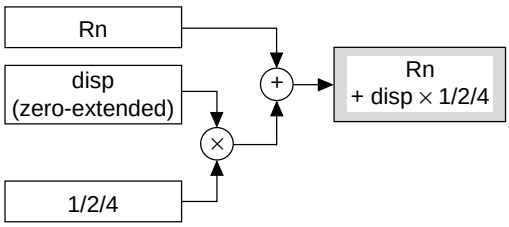
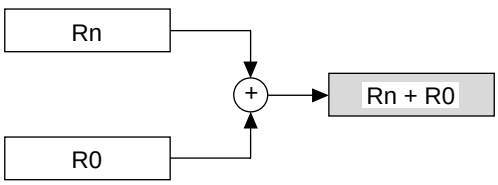
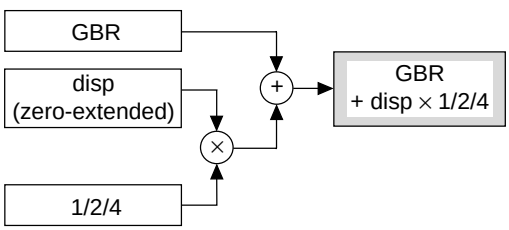
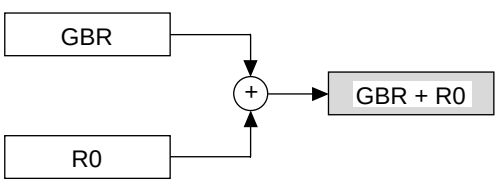
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword),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

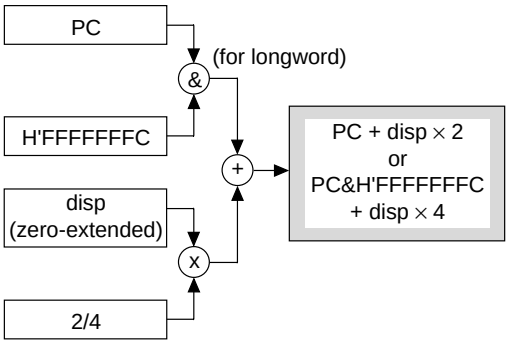
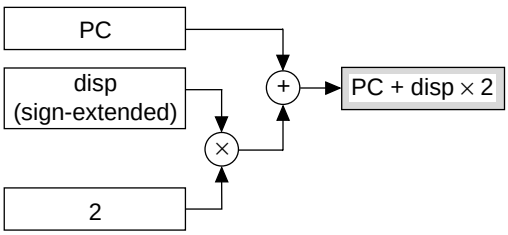
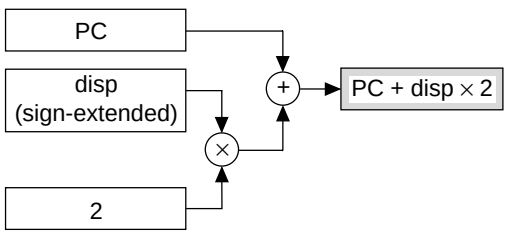
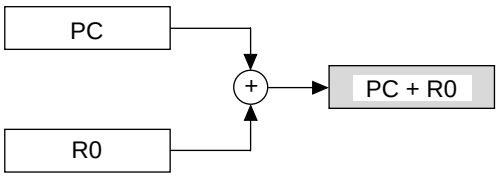
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is register PC contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	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 chip. 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

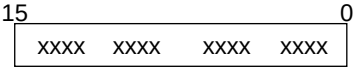
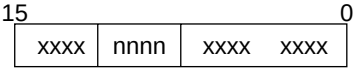
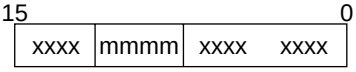
Instruction Format	Source Operand	Destination Operand	Instruction Example
0 format 	—	—	NOP
n format 	—	nnnn: register direct	MOVT Rn
	Control register or system register	nnnn: register direct	STS MACH,Rn
	Control register or system register	nnnn: register indirect with pre-decrement	STC.L SR,@–Rn
m format 	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)

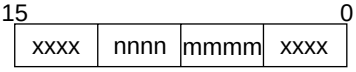
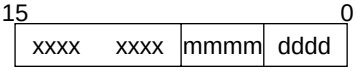
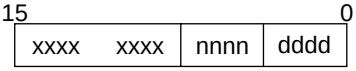
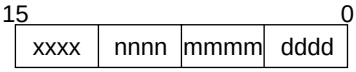
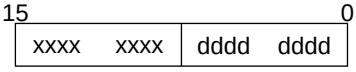
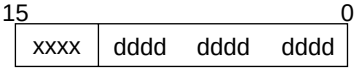
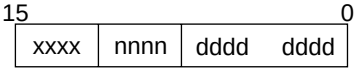
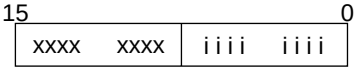
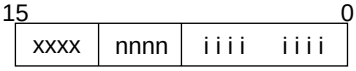
Instruction Format	Source Operand	Destination Operand	Instruction Example
nm format 	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: register indirect with post-increment (multiply-and-accumulate operation) nnnn: * register indirect 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 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 format 	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		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		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		dddddddddddd: PC-relative	—	BRA label (label = disp + PC)
nd8 format		dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i format		iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm, @(R0,GBR)
		iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
		iiiiiiii: immediate	—	TRAPA #imm
ni format		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 SH7707 instruction set includes 68 basic instruction types, as listed in table 2.4.

Table 2.4 Classification of Instructions

Classification	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 (cont)	21	MUL	Double-precision multiplication (32 × 32 bits)	33
		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 setting	
		TST	Logical AND and T bit setting	
		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	S bit clear	
		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	S bit setting	
		SETT	T bit setting	
		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 SH7707 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 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
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 SH7707 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	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	1	—

Table 2.6 Data Transfer Instructions (cont)

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	1	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	—	1	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	—	1	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	1	—
MOV.B	R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1	—
MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1	—
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1	—
MOV.B	@(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1	—
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1	—
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1	—
MOVA	@(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1	—
MOVT	Rn	T → Rn	0000nnnn00101001	—	1	—
SWAP.B	Rm,Rn	Rm → Swap 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 SH7707 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 SH7707 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 SH7707 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 SH7707 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 SH7707 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	√	5	LSB
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	1	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	√	1	—
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	√	1	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	√	1	—
LDC	Rm,R0_BANK	Rm → R0_BANK	0100mmmm10001110	√	1	—
LDC	Rm,R1_BANK	Rm → R1_BANK	0100mmmm10011110	√	1	—
LDC	Rm,R2_BANK	Rm → R2_BANK	0100mmmm10101110	√	1	—
LDC	Rm,R3_BANK	Rm → R3_BANK	0100mmmm10111110	√	1	—
LDC	Rm,R4_BANK	Rm → R4_BANK	0100mmmm11001110	√	1	—
LDC	Rm,R5_BANK	Rm → R5_BANK	0100mmmm11011110	√	1	—
LDC	Rm,R6_BANK	Rm → R6_BANK	0100mmmm11101110	√	1	—
LDC	Rm,R7_BANK	Rm → R7_BANK	0100mmmm11111110	√	1	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	√	7	LSB
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	1	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	√	1	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	√	1	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	√	1	—
LDC.L	@Rm+, R0_BANK	(Rm) → R0_BANK, Rm + 4 → Rm	0100mmmm10000111	√	1	—
LDC.L	@Rm+, R1_BANK	(Rm) → R1_BANK, Rm + 4 → Rm	0100mmmm10010111	√	1	—
LDC.L	@Rm+, R2_BANK	(Rm) → R2_BANK, Rm + 4 → Rm	0100mmmm10100111	√	1	—
LDC.L	@Rm+, R3_BANK	(Rm) → R3_BANK, Rm + 4 → Rm	0100mmmm10110111	√	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	√	1	—
LDC.L	@Rm+, R5_BANK	(Rm) → R5_BANK, Rm + 4 → Rm	0100mmmm11010111	√	1	—
LDC.L	@Rm+, R6_BANK	(Rm) → R6_BANK, Rm + 4 → Rm	0100mmmm11100111	√	1	—
LDC.L	@Rm+, R7_BANK	(Rm) → R7_BANK, Rm + 4 → Rm	0100mmmm11110111	√	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	√	1	—
NOP		No operation	0000000000001001	—	1	—
PREF	@Rm	(Rm) → cache	0000mmmm10000011	—	1	—
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	√	4	—
SETS		1 → S	0000000001011000	—	1	—
SETT		1 → T	0000000000011000	—	1	1
SLEEP		Sleep	0000000000011011	√	4*	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	√	1	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	1	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	√	1	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	√	1	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	√	1	—
STC	R0_BANK,Rn	R0_BANK→ Rn	0000nnnn10000010	√	1	—
STC	R1_BANK,Rn	R1_BANK→ Rn	0000nnnn10010010	√	1	—
STC	R2_BANK,Rn	R2_BANK→ Rn	0000nnnn10100010	√	1	—
STC	R3_BANK,Rn	R3_BANK→ Rn	0000nnnn10110010	√	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	√	1	—
STC	R5_BANK,Rn	R5_BANK→ Rn	0000nnnn11010010	√	1	—
STC	R6_BANK,Rn	R6_BANK→ Rn	0000nnnn11100010	√	1	—
STC	R7_BANK,Rn	R7_BANK→ Rn	0000nnnn11110010	√	1	—
STC.L	SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	√	1	—
STC.L	GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	—	1	—
STC.L	VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	√	1	—
STC.L	SSR,@–Rn	Rn–4 → Rn, SSR → (Rn)	0100nnnn00110011	√	1	—
STC.L	SPC,@–Rn	Rn–4 → Rn, SPC → (Rn)	0100nnnn01000011	√	1	—
STC.L	R0_BANK, @–Rn	Rn–4 → Rn, R0_BANK → (Rn)	0100nnnn10000011	√	2	—
STC.L	R1_BANK, @–Rn	Rn–4 → Rn, R1_BANK → (Rn)	0100nnnn10010011	√	2	—
STC.L	R2_BANK, @–Rn	Rn–4 → Rn, R2_BANK → (Rn)	0100nnnn10100011	√	2	—
STC.L	R3_BANK, @–Rn	Rn–4 → Rn, R3_BANK → (Rn)	0100nnnn10110011	√	2	—
STC.L	R4_BANK, @–Rn	Rn–4 → Rn, R4_BANK → (Rn)	0100nnnn11000011	√	2	—
STC.L	R5_BANK, @–Rn	Rn–4 → Rn, R5_BANK → (Rn)	0100nnnn11010011	√	2	—
STC.L	R6_BANK, @–Rn	Rn–4 → Rn, R6_BANK → (Rn)	0100nnnn11100011	√	2	—
STC.L	R7_BANK, @–Rn	Rn–4 → Rn, R7_BANK → (Rn)	0100nnnn11110011	√	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 below that use a displacement (disp), 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. 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

2.4.2 Instruction Code Map

Table 2.12 shows the instruction code map.

Table 2.12 Instruction Code Map

Instruction Code MSB LSB	Fx: 0000 MD: 00	Fx: 0001 MD: 01	Fx: 0010 MD: 10	Fx: 0011 to 1111 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 MSB LSB	Fx: 0000 MD: 00	Fx: 0001 MD: 01	Fx: 0010 MD: 10	Fx: 0011 to 1111 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 MSB LSB	Fx: 0000 MD: 00	Fx: 0001 MD: 01	Fx: 0010 MD: 10	Fx: 0011 to 1111 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 SH7700 Series Programming Manual for details.

2.5 Processor States and Processor Modes

2.5.1 Processor States

The SH7707 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 $\overline{\text{BREQ}}$ pin is high, or the manual reset state if the $\overline{\text{BREQ}}$ 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 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 routine.

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 handling routine found from the sum of the contents of the vector base address and the vector offset. See section 4, Exception Handling, 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 two modes in the power-down state: sleep mode and 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.

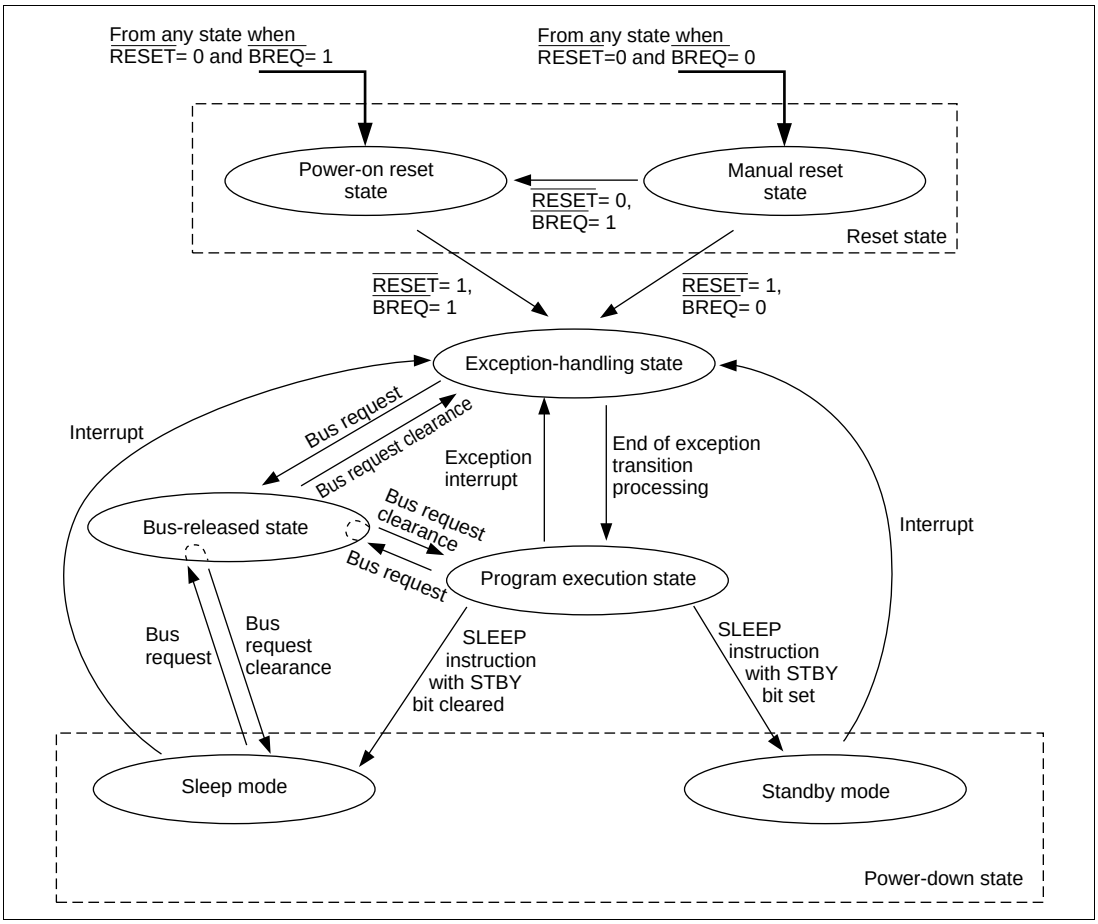


Figure 2.8 Processor State Transitions

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 SH7707 has an on-chip memory management unit (MMU) that implements address translation. The SH7707 features a resident translation look-aside 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 look-aside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails—that is, if an exception is generated—switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

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

In the following text, SH7707 address space in virtual memory is referred to as virtual address space, and address space in physical memory as physical memory space.

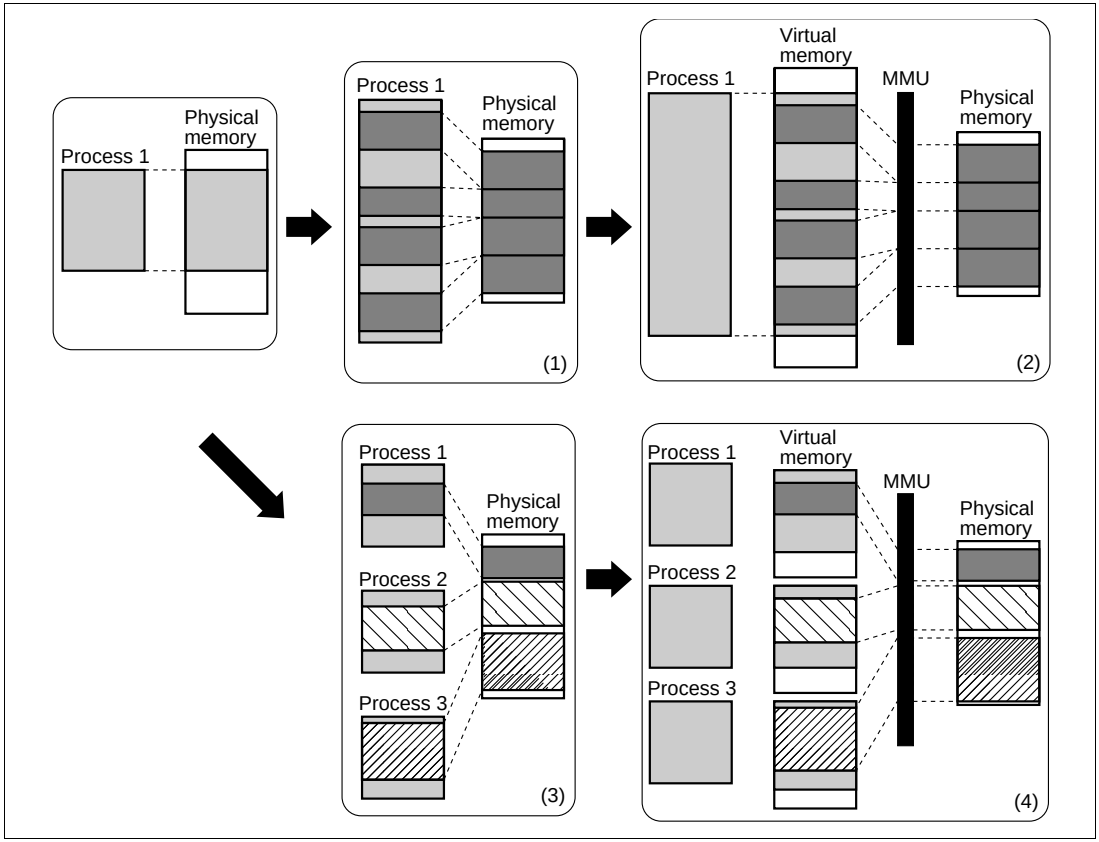


Figure 3.1 MMU Functions

3.1.3 Virtual Address Space

Virtual Address Map: The SH7707 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 MSB (bit 31) to 0 generates the corresponding physical address. Caching is available for P1 area access, 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. Caching is not available for P2 area access.

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

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 will generate an address error. Copy-back or write-through mode can be selected for write accesses by means of a CCR setting.

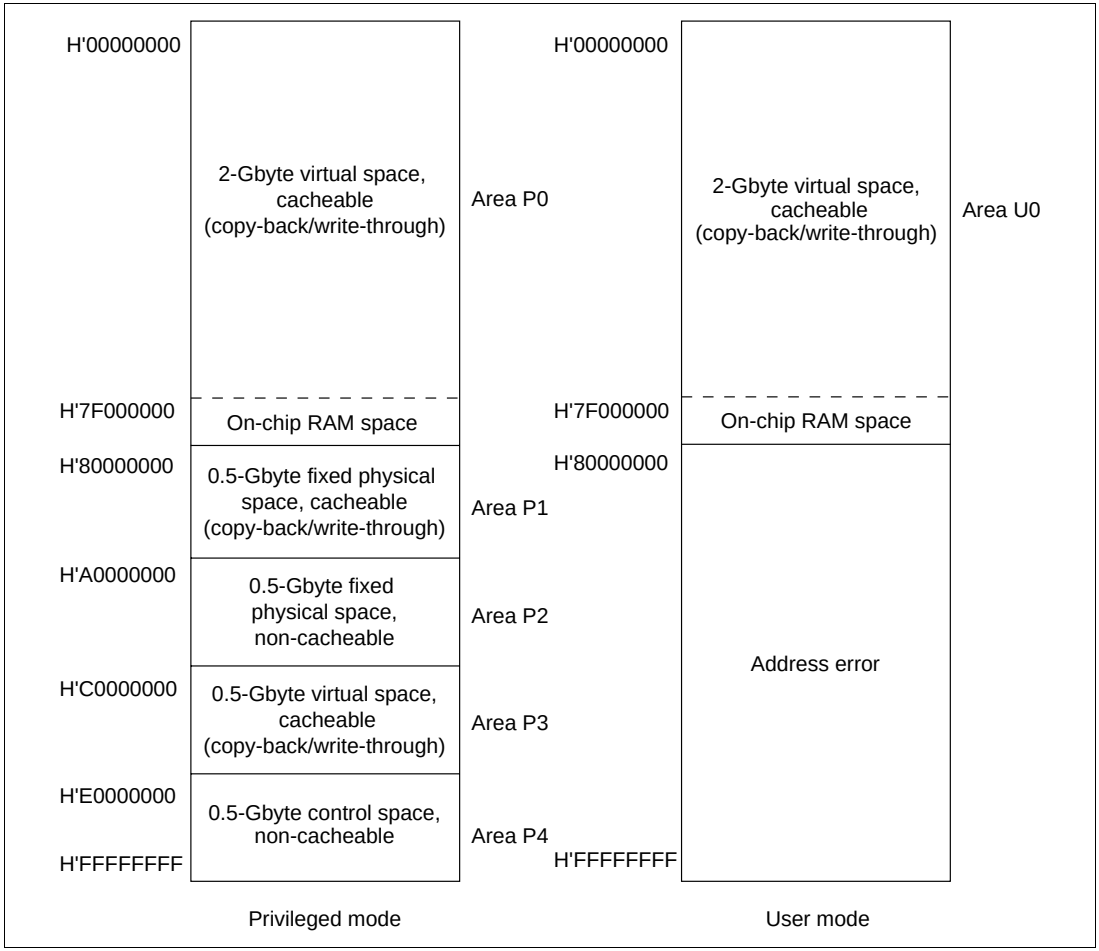


Figure 3.2 Virtual Address Space Mapping

Physical Address Space: The SH7707 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 (BSC), 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 area P1 or P2 occurs, there is no TLB access and the physical address is defined uniquely by the hardware. If it belongs to area P0, P3, or U0, the TLB is searched by virtual address and, if that virtual address is registered in the TLB, the access hits the TLB. The corresponding physical address and the page control information are read from the TLB and the physical address is determined.

If the virtual address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that results in a physical address space of H'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 SH7707 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. According to 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 the ASID. The ASID is 8 bits in length and is held in PTEH within the MMU indicating the current process. With ASIDs, 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 is a key means of protecting 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

Name	Abbreviation	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

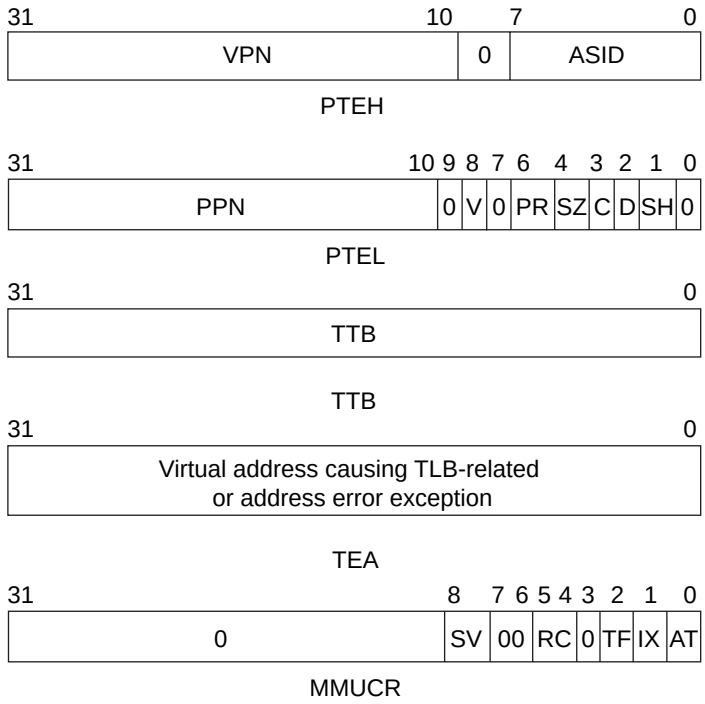
3.2 Register Descriptions

There are five registers for MMU processing. These are all on-chip supporting 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 is 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.

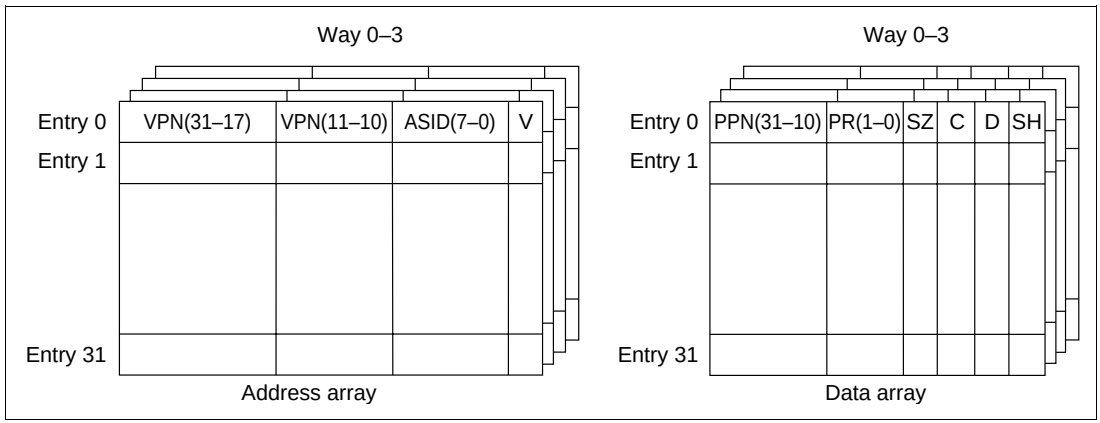
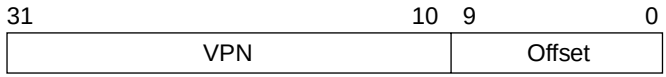
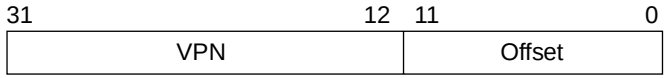


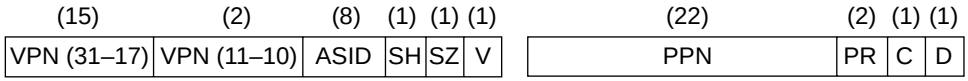
Figure 3.4 Overall Configuration of the TLB



Virtual address (1-kbyte page)



Virtual address (4-kbyte page)



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).
Clear 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 lower 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.

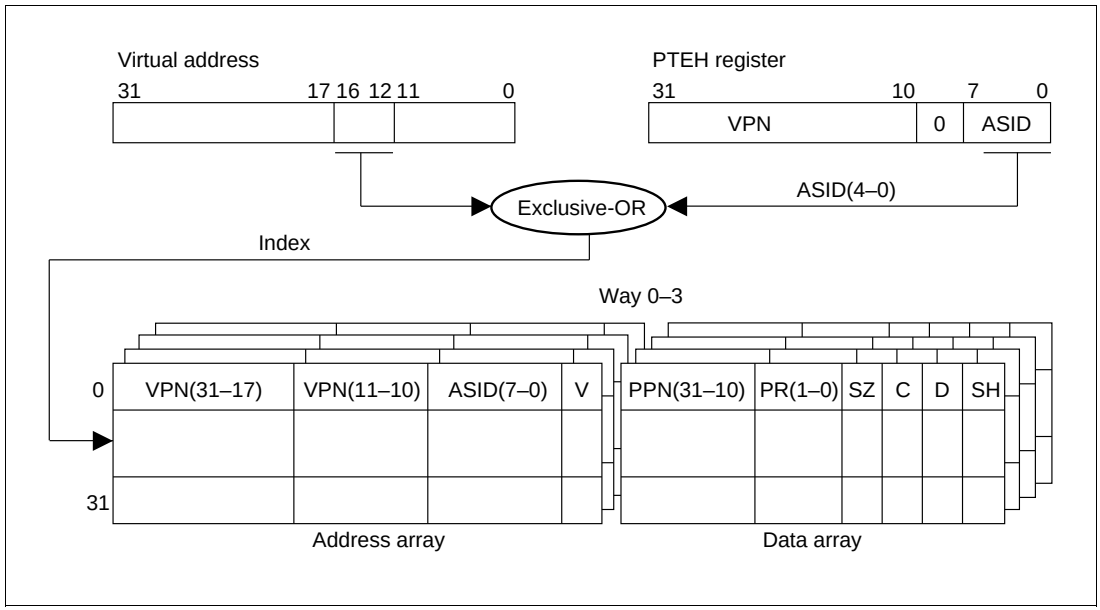


Figure 3.6 TLB Indexing (IX = 1)

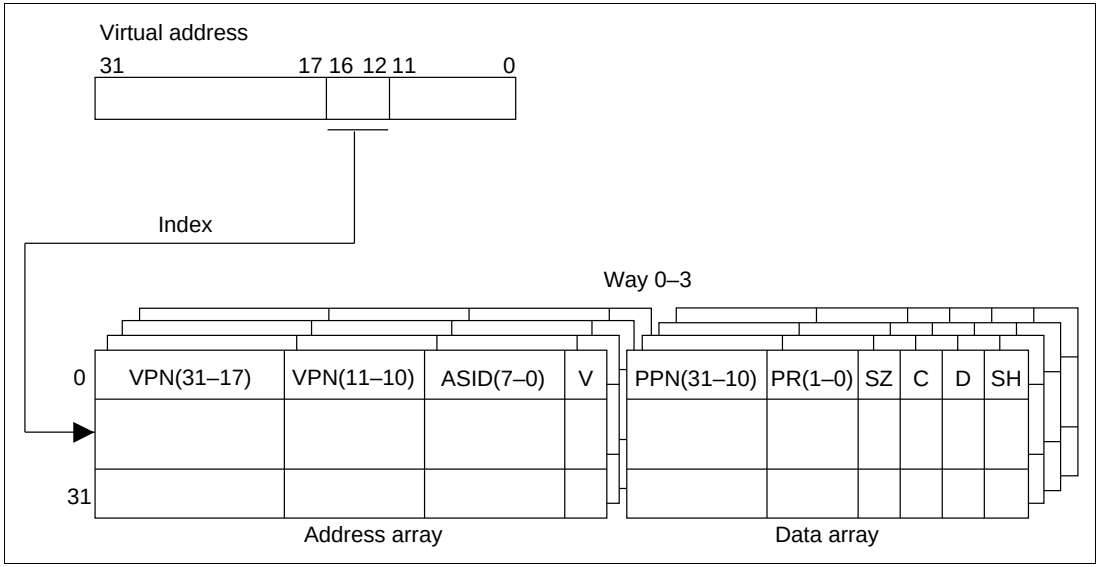


Figure 3.7 TLB Indexing (IX = 0)

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 in 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 PTEH 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 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 set (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 set. The objects of address comparison are shown in figure 3.8.

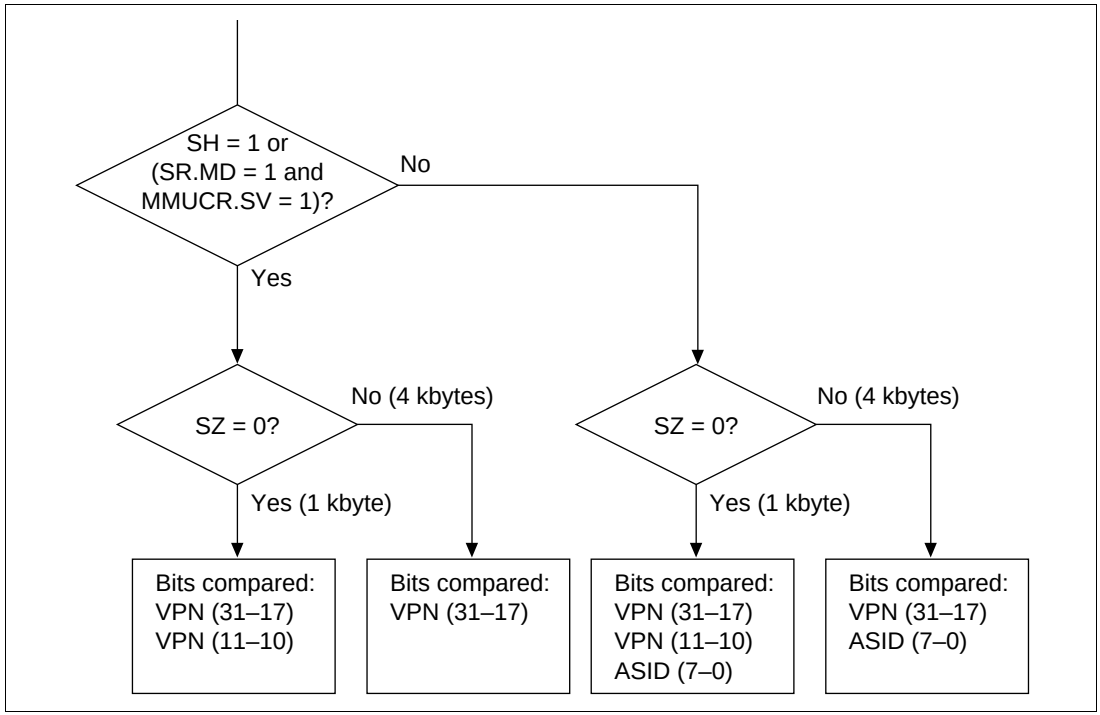


Figure 3.8 Objects of Address Comparison

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. 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. 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-mapped 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 only 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.

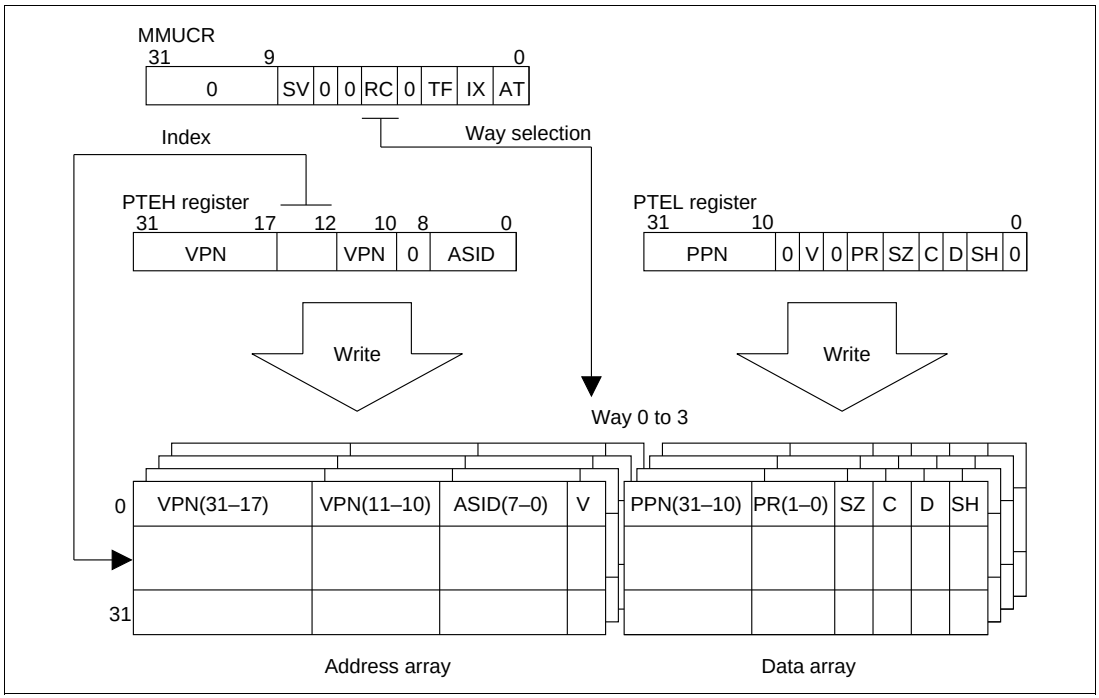


Figure 3.9 Operation of LDTLB Instruction

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 SH7707 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.

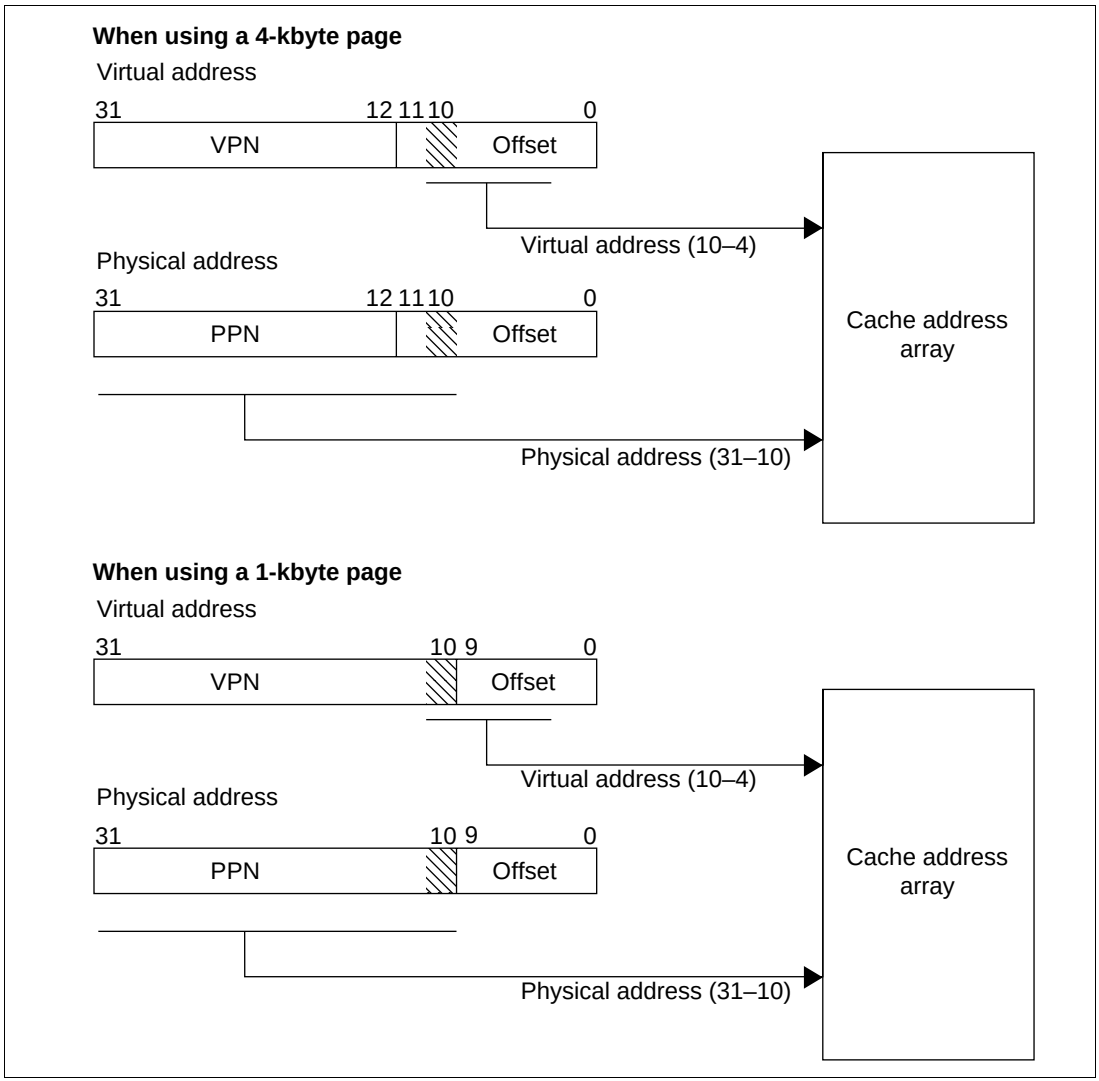


Figure 3.10 Synonym Problem

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 SH7707 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 SH7707 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 among 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 SH7707.

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 SH7707 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 SH7707 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 SH7707 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 SPC. If the exception occurred in a delay slot, the PC value indicating the address of the delayed branch instruction is written to 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 SH7707 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 occurs 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 SH7707 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 SPC. If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to 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 SH7707 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.

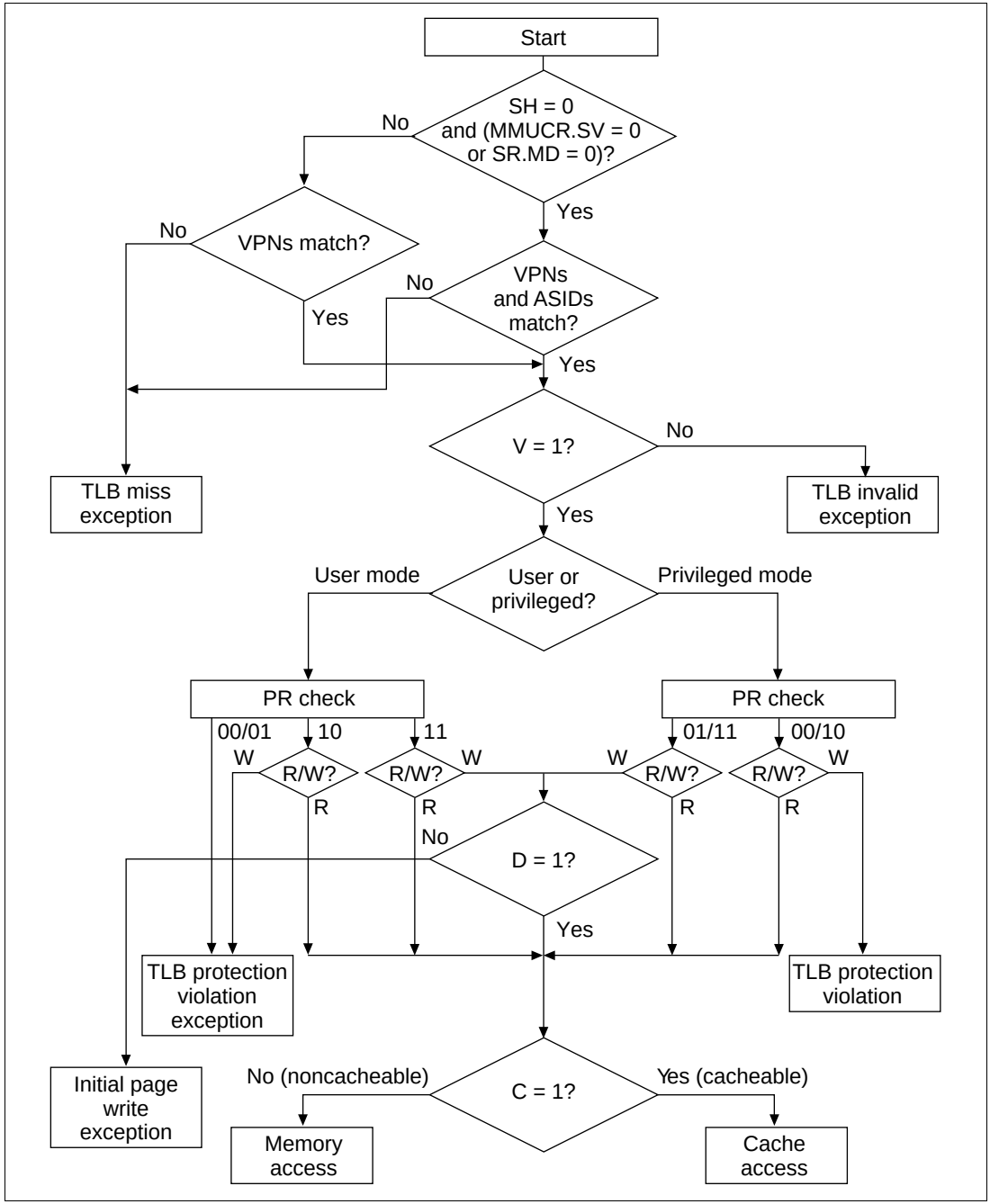


Figure 3.11 MMU Exception Generation Flowchart

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.

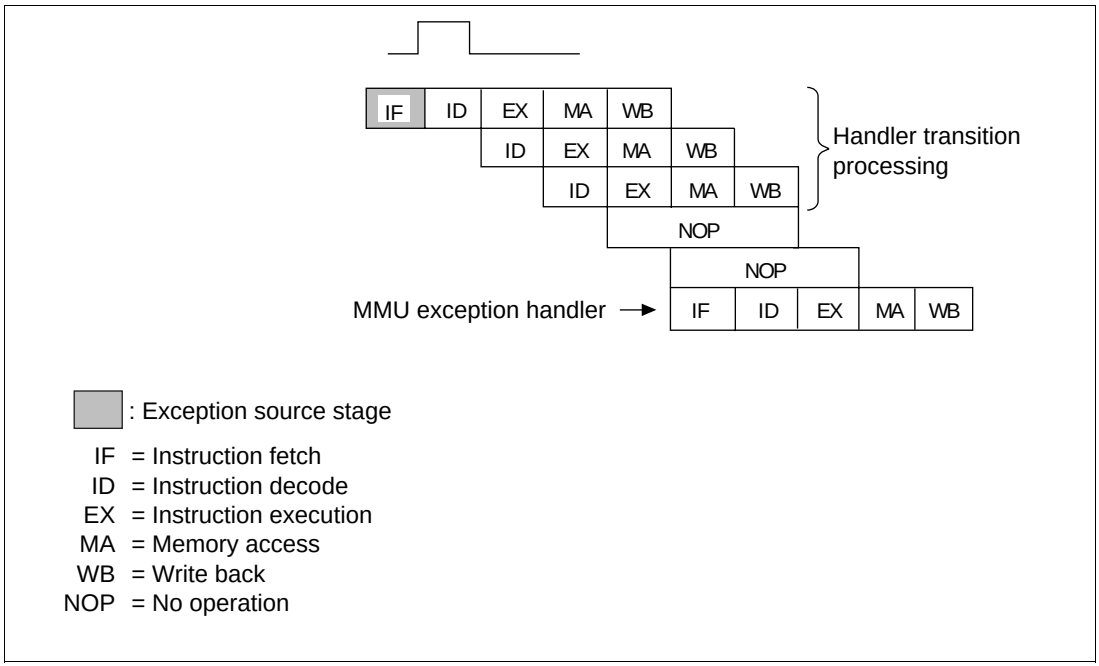


Figure 3.12 MMU Exception Signals in Instruction Fetch

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

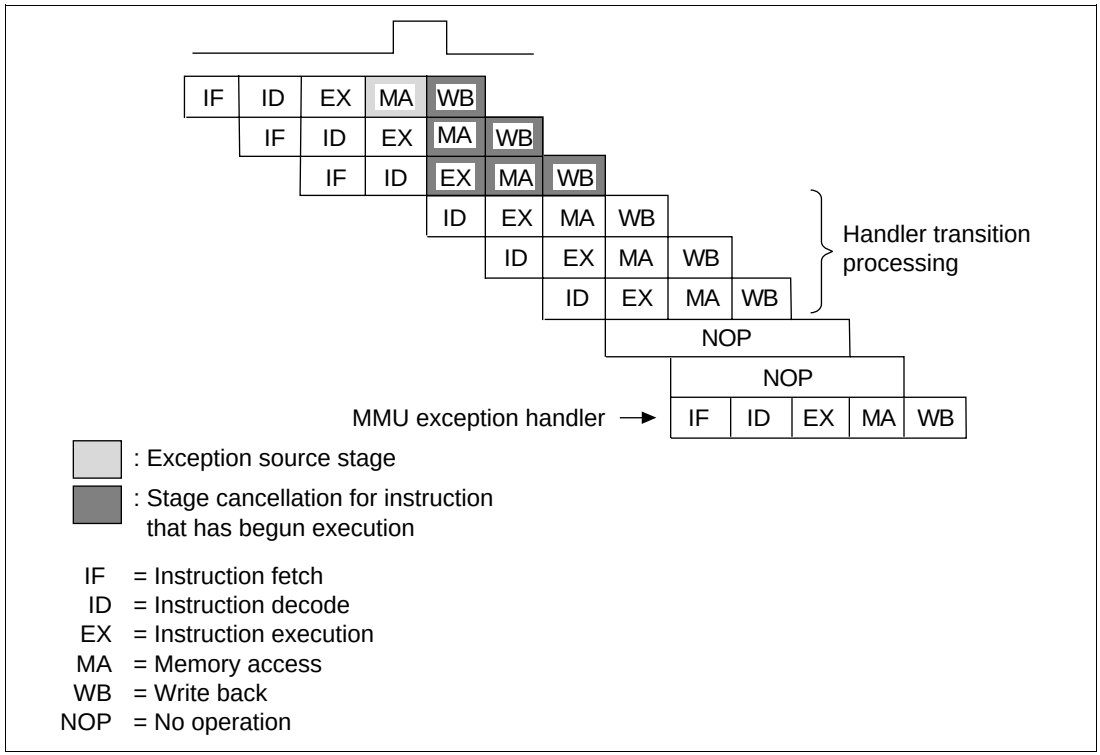


Figure 3.13 MMU Exception Signals in Data Access

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–H'F2FFFFFF. To access the 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

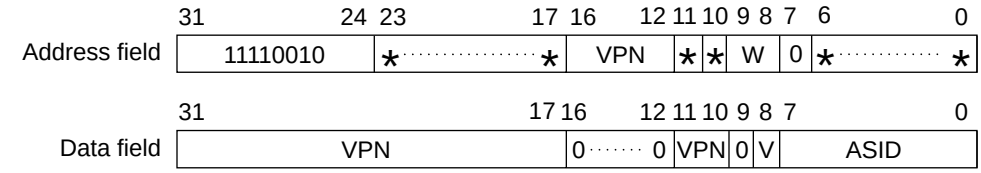
The data array is assigned to H'F3000000–H'F3FFFFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. These are specified in general registers. The address field specifies information for selecting the entry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

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'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.

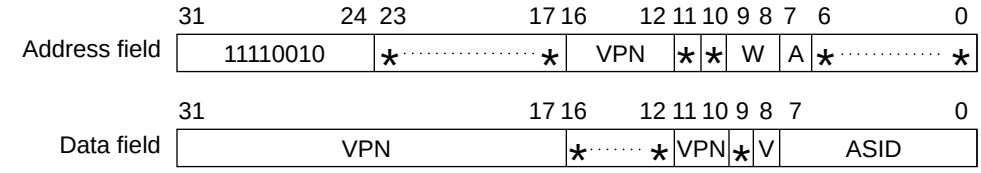
Both reading and writing use the longword of the data array specified by the entry address and way number. The access size of the data array is fixed at longword.

(1) TLB Address Array Access

Read access



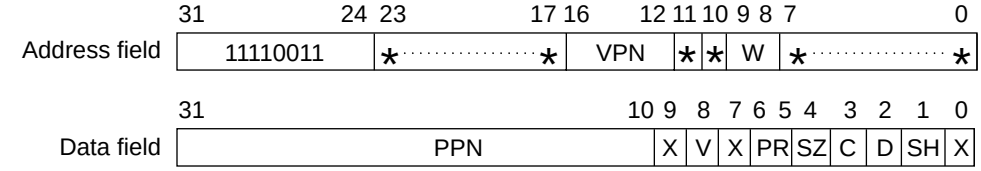
Write access



VPN: Virtual page number
V: Valid bit
A: Association bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
ASID: Address space identifier
*: Don't care bit

(2) TLB Data Array Access

Read/write access



PPN: Physical page number
PR: Protection key field
C: Cacheable bit
SH: Share status bit
VPN: Virtual page number
X: 0 for read, don't care bit for write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
V: Valid bit
SZ: Page size bit
D: Dirty bit
*: Don't care bit

Figure 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 are 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 field 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

Registers 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	Abbr.	R/W	Size	Initial Value	Address
TRAPA exception register	TRA	R/W	Longword	Undefined	H'FFFFFFD0
Exception event register	EXPEVT	R/W	Longword	Power-on reset: H'000 Manual reset: H'020	H'FFFFFFD4
Interrupt event register	INTEVT	R/W	Longword	Undefined	H'FFFFFFD8
Interrupt event register 2	INTEVT2	R	Longword	Undefined	H'04000000*

Note: See Appendix B, Control Registers, for the register address.

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 on completion of the routine, restoring the contents of PC and SR to return to the processor state at the point of interruption and the address where the exception occurred.

A basic exception handling sequence consists of the following operations:

- The contents of PC and SR are saved in 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 SH7707 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, 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 look-aside 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.

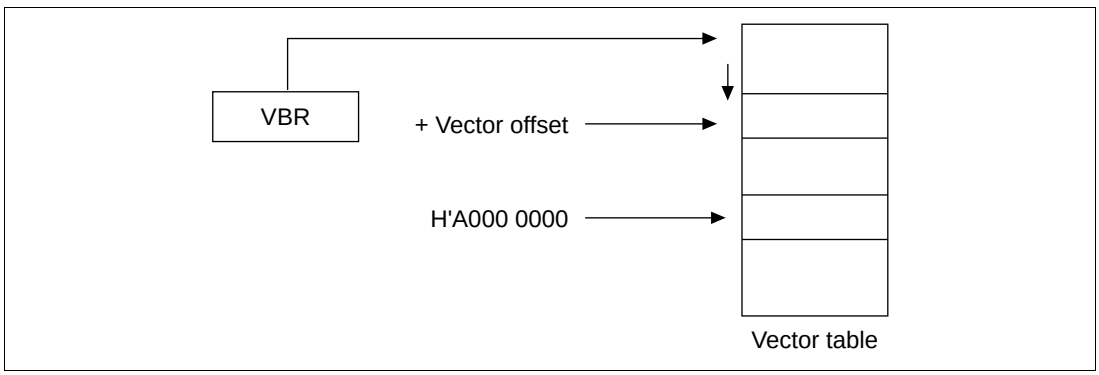


Figure 4.1 Vector Table

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
		Supporting 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 breakpoint traps. 1 is a break point before instruction execution and 11 is a breakpoint after instruction execution. For an operand breakpoint, use 11.
3. Use software to specify relative priorities of external hardware interrupts and supporting 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 cannot 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.

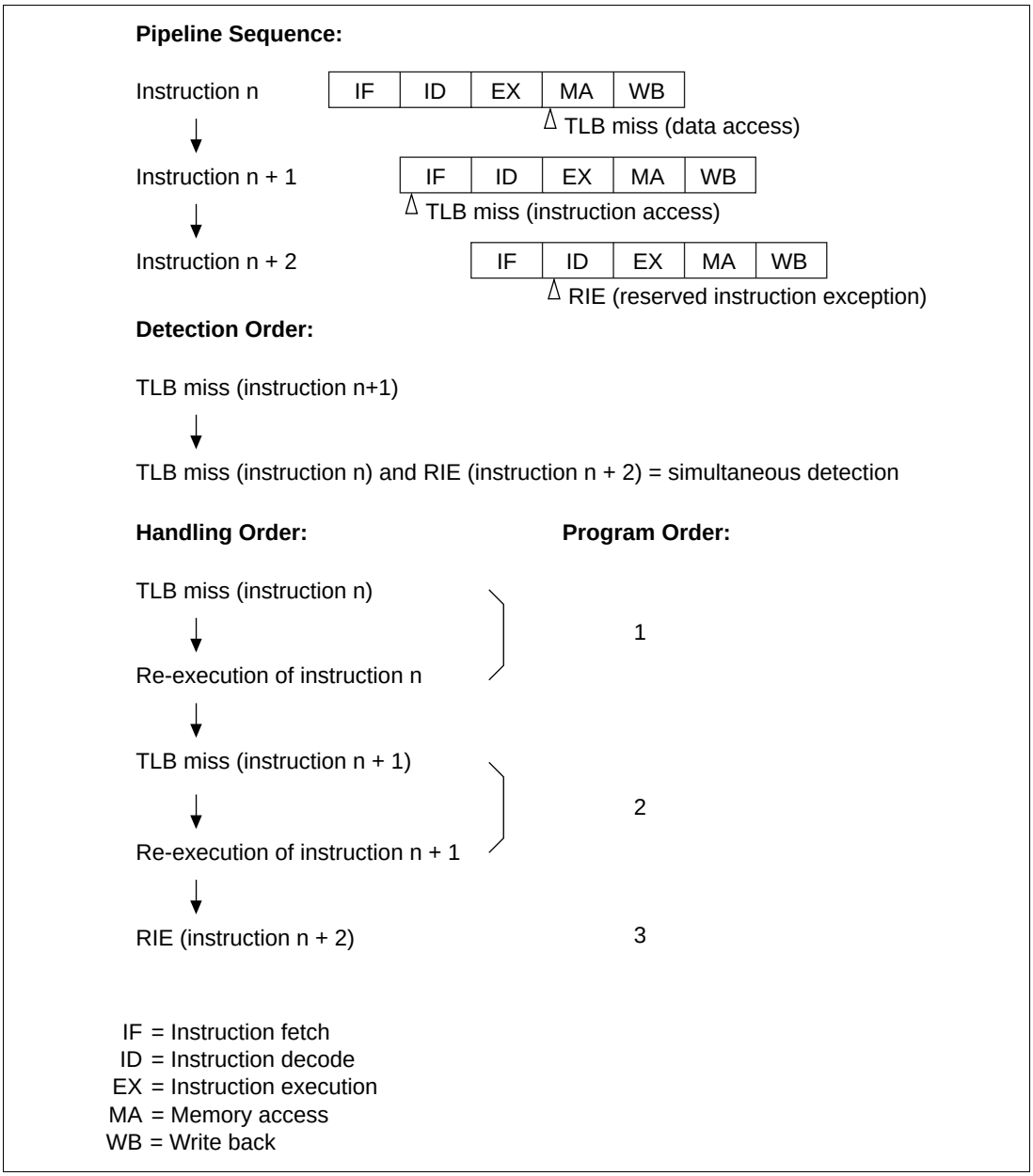


Figure 4.2 Example of Acceptance Order of General Exceptions

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, INTEVT2 register (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:	
	$\overline{IRL3}$–$\overline{IRL0}$ = 0000	H'200
	$\overline{IRL3}$–$\overline{IRL0}$ = 0001	H'220
	$\overline{IRL3}$–$\overline{IRL0}$ = 0010	H'240
	$\overline{IRL3}$–$\overline{IRL0}$ = 0011	H'260
	$\overline{IRL3}$–$\overline{IRL0}$ = 0100	H'280
	$\overline{IRL3}$–$\overline{IRL0}$ = 0101	H'2A0
	$\overline{IRL3}$–$\overline{IRL0}$ = 0110	H'2C0
	$\overline{IRL3}$–$\overline{IRL0}$ = 0111	H'2E0
	$\overline{IRL3}$–$\overline{IRL0}$ = 1000	H'300

Table 4.3 Exception Codes (cont)

Exception Type	Exception Event		Exception Code
General interrupt requests (cont)	External hardware interrupts (cont):		
	$\overline{IRL3}$–$\overline{IRL0}$ = 1001		H'320
	$\overline{IRL3}$–$\overline{IRL0}$ = 1010		H'340
	$\overline{IRL3}$–$\overline{IRL0}$ = 1011		H'360
	$\overline{IRL3}$–$\overline{IRL0}$ = 1100		H'380
	$\overline{IRL3}$–$\overline{IRL0}$ = 1101		H'3A0
	$\overline{IRL3}$–$\overline{IRL0}$ = 1110		H'3C0
	Supporting module interrupts:		
	TMU0	TUNI0	H'400
	TMU1	TUNI1	H'420
	TMU2	TUNI2	H'440
		TICPI2	H'460
	RTC	ATI	H'480
		PRI	H'4A0
		CUI	H'4C0
	SCI	ERI	H'4E0
		RXI	H'500
		TXI	H'520
		TEI	H'540
	WDT	ITI	H'560
	REF	RCMI	H'580
		ROVI	H'5A0
	IRQ	IRQ0	H'600 (H'200–H'3C0*1)
		IRQ1	H'620 (H'200–H'3C0*1)
		IRQ2	H'640 (H'200–H'3C0*1)
		IRQ3	H'660 (H'200–H'3C0*1)
		IRQ4	H'680 (H'200–H'3C0*1)
		IRQ5	H'6A0 (H'200–H'3C0*1)
	PINT	PINT0–PINT7	H'700 (H'200–H'3C0*1)
		PINT8–PINT15	H'720 (H'200–H'3C0*1)

Table 4.3 Exception Codes (cont)

Exception Type	Exception Event		Exception Code
General interrupt requests (cont)	DMAC	DEI0	H'800 (H'200–H'3C0*1)
		DEI1	H'820 (H'200–H'3C0*1)
		DEI2	H'840 (H'200–H'3C0*1)
		DEI3	H'860 (H'200–H'3C0*1)
	IRDA	ERI1	H'880 (H'200–H'3C0*1)
		RX1	H'8A0 (H'200–H'3C0*1)
		BR1	H'8C0 (H'200–H'3C0*1)
		TX1	H'8E0 (H'200–H'3C0*1)
	SCIF	ERI2	H'900 (H'200–H'3C0*1)
		RX2	H'920 (H'200–H'3C0*1)
		BR2	H'940 (H'200–H'3C0*1)
		TX2	H'960 (H'200–H'3C0*1)
	ADC	ADI	H'980 (H'200–H'3C0*1)
	LCDC	LCDI	H'9A0 (H'200–H'3C0*1)
	PCC	PCC0I	H'9C0 (H'200–H'3C0*1)
		PCC1I	H'9E0 (H'200–H'3C0*1)

Notes: 1. The INTEVT register is set with a code corresponding to the interrupt level. See table 6-6 for the codes. The INTEVT2 register is set with a code corresponding 1-to-1 with the interrupt source.
2. 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, SPC and SSR must be saved and the BL bit in SR cleared to 0.

However, when the BLMSK bit in the ICR1 register is 1, NMI interrupts only are accepted regardless of the BL bit setting. If NMI interrupts are used with the BLMSK bit set to 1, the SPC and SSR values set by another external source may be overwritten, preventing return to the original program sequence.

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 PC, and the SSR value in SR, and the return from exception handling is performed by branching to the SPC address.

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

4.3 Register Descriptions

There are four registers related to exception handling. EXPEVT, INTEVT and TRA register can be accessed by specifying the address in privileged mode only. INTEVT2 can be accessed in both privileged mode and user mode.

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 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 interrupt event register 2 (INTEVT2) resides at address H'04000000, and contains a 12-bit exception code. The exception code set in INTEVT2 is that for an interrupt request. The exception code is set automatically by hardware when an exception occurs. INTEVT2 cannot be modified by software.
4. 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, INTEVT2, and TRA registers are shown in figure 4.3.

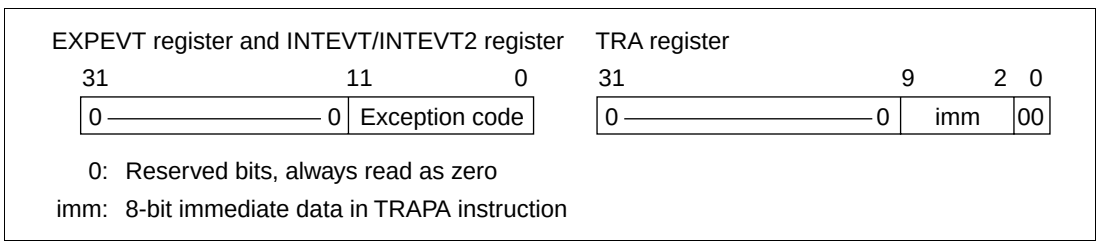


Figure 4.3 Bit Configurations of EXPEVT, INTEVT, INTEVT2, and TRA Registers

4.4 Exception Handler Operation

4.4.1 Reset

The reset sequence is used to power up or restart the SH7707 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 $\overline{\text{BREQ}}$ (bus request) signal is used to distinguish between a power-on reset (high-level input) and manual reset (low-level input). The reset sequence consists of the following operations:

- The MD bit in SR is set to 1 to place the SH7707 in privileged mode.
- The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupts when the BLMSK bit is 1).
- 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 handling request is accepted on completion of the current instruction. The interrupt acceptance sequence consists of the following operations:

- The contents of PC and SR are saved in SPC and SSR, respectively.
- The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupts when the BLMSK bit is 1).
- The MD bit in SR is set to 1 to place the SH7707 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/ INTEVT2 register.
- 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 SH7707 encounters any exception condition other than a reset or interrupt request, it executes the following operations:

- The contents of PC and SR are saved in SPC and SSR, respectively.

- The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupts when the BLMSK bit is 1).
- The MD bit in SR is set to 1 to place the SH7707 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: $\overline{\text{BREQ}}$ pin high and $\overline{\text{RESET}}$ asserted
  - 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 I3–I0 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: $\overline{\text{BREQ}}$ pin low and $\overline{\text{RESET}}$ asserted
  - 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 I3–I0 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

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

  PC and SR of the instruction that generated the exception are saved to 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 is made 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 MMUCR.RC.

  PC and SR of the instruction that generated the exception are saved to 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 is made to PC = VBR + H'0100.
- Initial page write exception
  - — Conditions: A hit occurred in 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.

  PC and SR of the instruction that generated the exception are saved to 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 is made to 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 MMUCR.RC.

  PC and SR of the instruction that generated the exception are saved to 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 is made to PC = VBR + H'0100.
- Address error
  - — Conditions:
    a. Instruction fetch from an odd address (4n + 1, 4n + 3)
    b. Word data access from an address other than a word boundary (4n + 1, 4n + 3)
    c. Longword access from an address other than a longword boundary (4n + 1, 4n + 2, 4n + 3)
    d. Virtual space accessed in user mode in the area H'80000000–H'FFFFFFFF.
  - — Operations: The virtual address (32 bits) that caused the exception is set in TEA. PC and SR of the instruction that generated the exception are saved to 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 is made to PC = VBR + H'0100.
- Unconditional trap
  - — Conditions: TRAPA instruction executed
  - — Operations: The exception is a processing-completion type, so PC of the instruction after the TRAPA instruction is saved to 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 is made 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: PC and SR of the instruction that generated the exception are saved to 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 is made 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 PC in a delay slot is decoded

    Instructions that rewrite PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L, @Rm+, SR
  - 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: PC of the previous delay branch instruction is saved to 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 is made 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, PC of the instruction immediately after the instruction that set the break point is set in SPC. If a pre-execution break occurs, PC of the instruction that set the break point is set in 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 is made to PC = VBR + H'0100. See section 7, User Break Controller, for more information.

4.5.3 Interrupts

- NMI interrupt
  - — Conditions: NMI pin edge detection
  - — Operations: PC and SR after the instruction that receives the interrupt are saved to SPC and SSR, respectively. H'01C0 is set in INTEVT and INTEVT2. The BL, MD, and RB bits in SR are set to 1 and a branch is made to PC = VBR + H'0600. This interrupt is not masked by SR.I3–I0 and is accepted with top priority when the BL bit in SR is 0. When the BL bit is 1, the interrupt is masked. See section 6, Interrupt Controller, for more information.
- IRL interrupts
  - — Conditions: The value of the interrupt mask bits in SR is lower than the $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - — Operations: PC after the instruction that accepts the interrupt is saved to SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ level is set in INTEVT and INTEVT2. The corresponding code is given as H'200 + B'($\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$) × H'20. The BL, MD, and RB bits in SR are set to 1 and a branch is made to VBR + H'0600. The received level is not set in SR.I3–I0. See section 6, Interrupt Controller, for more information.
- IRQ Interrupts
  - — Conditions: The IRQ pin is asserted and SR.I3–I0 is lower than the IRQ priority level, and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - — Operations: PC after the instruction that accepts the interrupt is saved to SPC. SR at the point the interrupt is accepted is saved to SSR. The code corresponding to the interrupt source is set in INTEVT and INTEVT2. The BL, MD, and RB bits of SR are set to 1 and a branch is made to VBR + H'0600. The received level is not set in SR.I3–I0. See section 6, Interrupt Controller, for more information.
- PINT Interrupts
  - — Conditions: The PINT pin is asserted and SR.I3–I0 is lower than the PINT priority level, and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - — Operations: PC after the instruction that accepts the interrupt is saved to SPC. SR at the point the interrupt is accepted is saved to SSR. The code corresponding to the interrupt source is set in INTEVT and INTEVT2. The BL, MD, and RB bits of SR are set to 1 and a branch is made to VBR + H'0600. The received level is not set in SR.I3–I0. See section 6, Interrupt Controller, for more information.

- On-Chip Supporting Module Interrupts
  - Conditions: SR.I3–I0 is lower than the on-chip supporting module (TMU, RTC, SCI, IRDA, SCIF, ADC, DMAC, LCDC, CPG, REF) interrupt level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - Operations: PC after the instruction that accepts the interrupt is saved to SPC. SR at the point the interrupt is accepted is saved to SSR. The code corresponding to the interrupt source is set in INTEVT and INTEVT2. The BL, MD, and RB bits of SR are set to 1 and a branch is made to VBR + H'0600. B'0000 to B'1111 are set in 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 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 SPC in 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 cleared 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 clears 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.
  - 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 for a 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 SPC when an exception occurs is as shown below:
  - Re-executing-type exceptions: PC of the instruction that caused the exception is set in SPC and re-executed after return from exception handling. If the exception occurred in a delay slot, however, PC of the immediately prior delayed branch instruction is set in 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: PC of the instruction after the one that caused the exception is set in 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	Instructions/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 field 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.

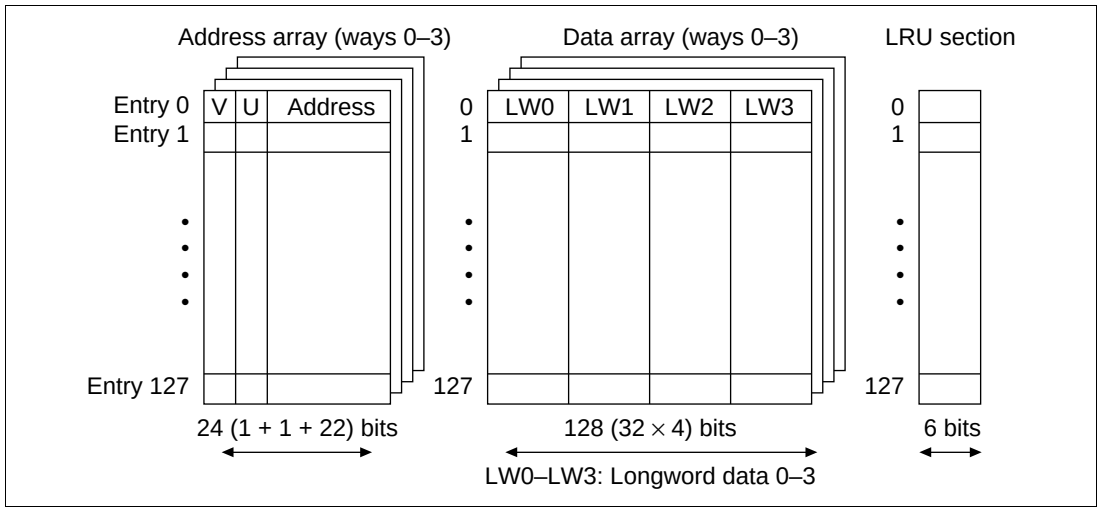


Figure 5.1 Cache Structure

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 SH7707, 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 16-byte instructions 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.

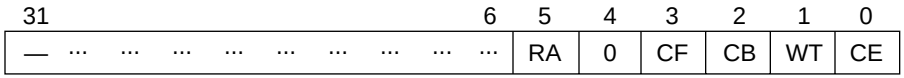
Table 5.3 Register Configuration

Register	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), and a WT bit (which selects either write-through mode or write-back mode). 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.



RA: RAM bit. Indicates the cache operating 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.

—: Reserved bits. Always read 0; writing is ignored.

WT: Write-through bit. Indicates the cache operating mode for areas P0, U0, and P3.
1 = write-through mode, 0 = write-back mode.

CB: P1 area write-back/write-through switch.
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 (virtual) address 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.

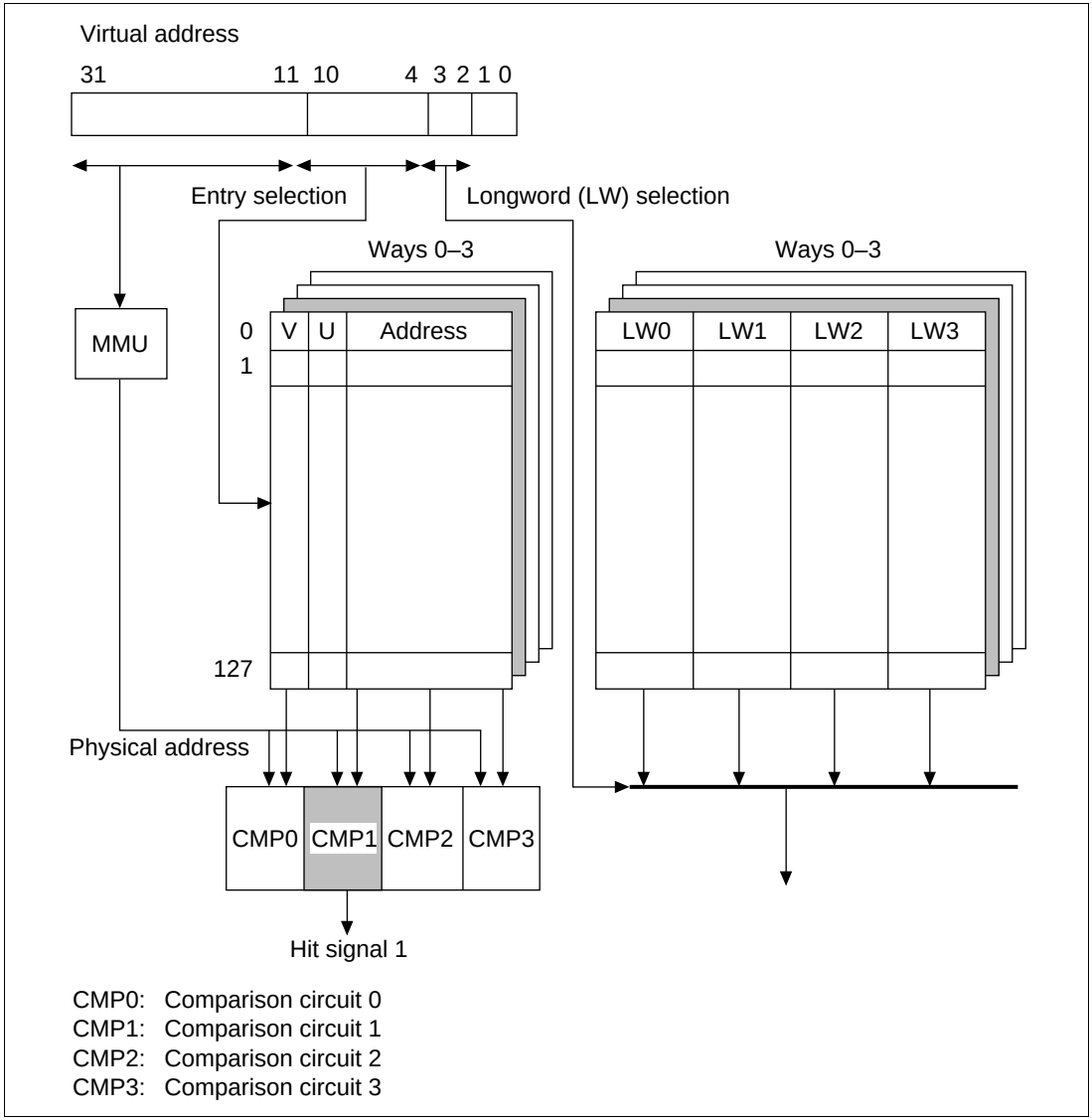


Figure 5.3 Cache Search Scheme (Normal Mode)

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 field 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.

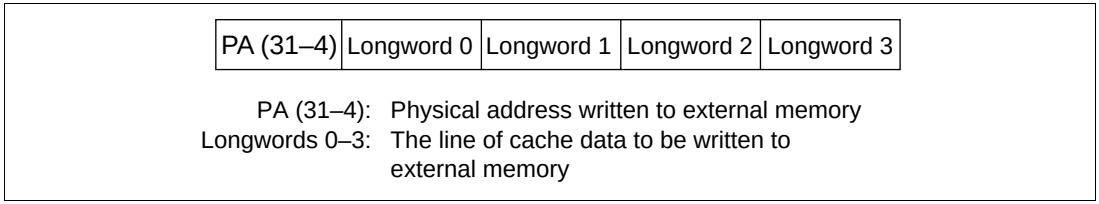


Figure 5.4 Write-Back Buffer Configuration

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 SH7707 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 SH7707, 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–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'7F00000 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 operating 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–H'F0FFFFFF and the data array onto addresses H'F1000000–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–H'F0FFFFFF. To access the 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 SH7707, 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–H'F1FFFFFF. To access the 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

A specific cache entry can be invalidated by writing 0 to the V bit of that entry. 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 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 handle interrupt requests according to the user-set priority.

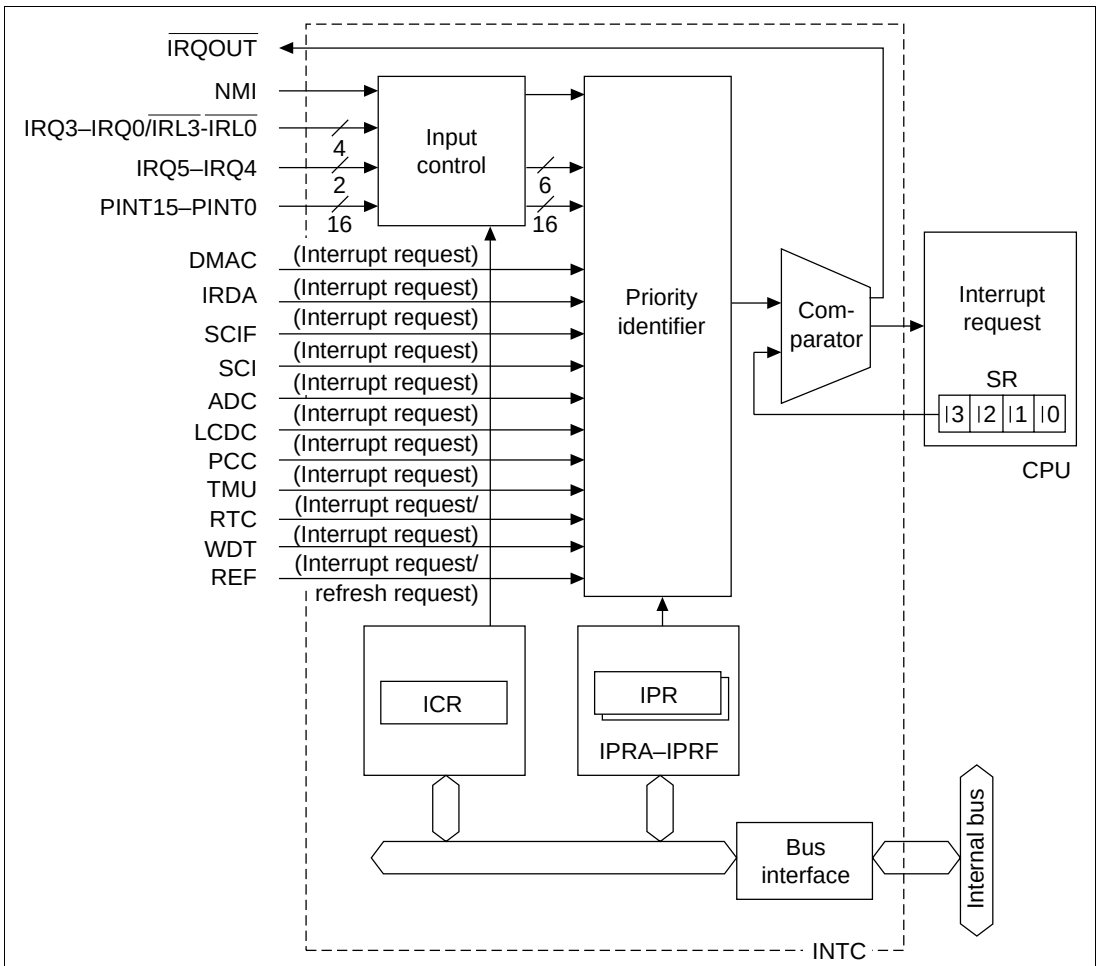
6.1.1 Features

The 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 supporting 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 handling 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 ($\overline{\text{IRQOUT}}$): For example, when the SH7707 has released the bus right, the external bus master can be notified of the fact that an external interrupt, an on-chip supporting module interrupt, or a memory refresh request has occurred, enabling that device to request the bus right.

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the INTC.



TMU: Timer unit
RTC: Real-time clock unit
SCI: Serial communication interface (with smart card interface)
IRDA: Serial communication interface (with IrDA)
SCIF: Serial communication interface (with FIFO)
WDT: Watchdog timer
REF: Memory refresh controller section of the bus state controller
ICR: Interrupt control register
IPRA-IPRF: Registers A-F for setting the interrupt priority 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

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	I	Input of interrupt request signal, not maskable by SR.I(3–0)
Interrupt input pins	IRQ3–IRQ0/ $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ IRQ5–IRQ4	I	Input of interrupt request signals, maskable by SR.I(3–0)
Port interrupt input pins	PINT15–PINT0	I	Port input interrupt request signals, maskable by SR.I(3–0)
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 ten registers listed in table 6.2.

Table 6.2 Register Configuration

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'4000	H'04000010*3	16
Interrupt control register 2	ICR2	R/W	H'0000	H'04000012*3	16
PINT interrupt enable register	PINTER	R/W	H'0000	H'04000014*3	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'04000016*3	16
Interrupt priority level setting register D	IPRD	R/W	H'0000	H'04000018*3	16
Interrupt priority level setting register E	IPRE	R/W	H'0000	H'0400001A*3	16
Interrupt priority level setting register F	IPRF	R/W	H'0000	H'0400001C*3	16
Interrupt request register 0	IRR0	R/W	H'00	H'04000004*3	8
Interrupt request register 1	IRR1	R	H'00	H'04000006*3	8
Interrupt request register 2	IRR2	R	H'00	H'04000008*3	8
Interrupt request register 3	IRR3	R	H'00	H'0400000A*3	8
Interrupt request register 4	IRR4	R	H'00	H'0400000C*3	8

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

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

3. Registers located in the area from H'04000004 to H'040000IC are located in area 1 in physical space. Therefore, when the cache is turned on, either access these registers from the P2 area in virtual space, or make appropriate settings using the MMU to ensure that these registers are not cached.

6.2 Interrupt Sources

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

6.2.1 NMI Interrupt

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

When the BLMSK bit in 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 in the ICR1 register is set to 1).

6.2.2 IRQ Interrupts

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

When IRQ interrupts are used in edge-sense mode, clear the interrupt source by software clearance of the corresponding bit in the IRR0 register.

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).

For edge input interrupt detection, an input pulse width of more than 2 cycles (P clock basis) is necessary.

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

Interrupts IRQ3–IRQ0 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 $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$. The priority level is the level indicated by pins $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$. An $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ 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 IRL interrupt connection. Table 6.3 shows the $\overline{\text{IRL}}$ pins and interrupt levels.

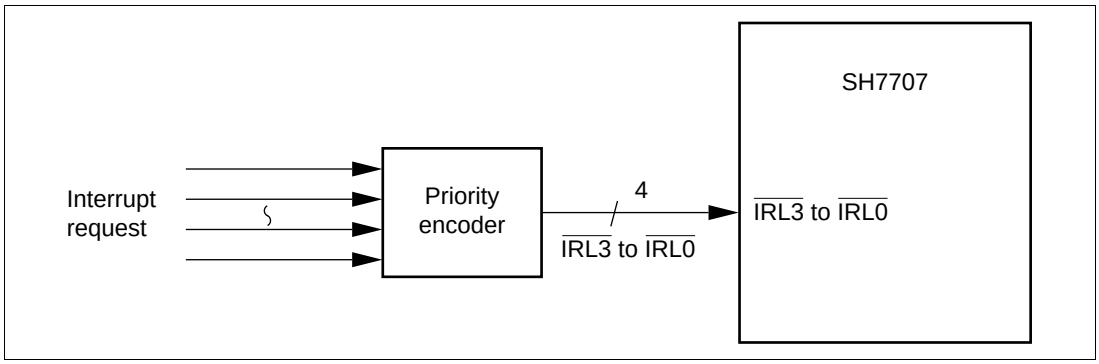


Figure 6.2 Example of IRL Interrupt Connection

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 IRL interrupt handling.

6.2.4 PINT Interrupts

PINT interrupts are input by priority from pins PINT0–PINT15 using level sensing. The priority level can be set in priority setting register D (IPRD) in a range from 0 to 15, in units of PINT0–PINT7 and PINT8–PINT15.

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

Wakeup from standby mode is not possible with PINT interrupts.

6.2.5 On-Chip Supporting Module Interrupts

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

- Timer unit (TMU)
- Real-time clock (RTC)
- Serial communication interface (SCI, IRDA, SCIF)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- PC card controller (PCC)
- Liquid crystal device controller (LCDC)
- Analog-to-digital converter (ADC)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected in the interrupt event registers (INTEVT and INTEVT2). It is easy to identify sources by using the value in the INTEVT or INTEVT2 register as a branch offset (in the exception handling routine).

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

The interrupt mask bits (I3–I0) in the status register are not affected by on-chip supporting module interrupt handling.

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 Handling and Priority

Tables 6.4 and 6.5 show the codes for the interrupt event registers (INTEVT and INTEVT2), and the order of interrupt priority. Each interrupt source is assigned a unique code. The start address of the interrupt handling routine is common to each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as an offset at the start of the interrupt handling routine and branched to in order to identify the interrupt source.

The order of priority of the on-chip supporting modules is set at will within priority levels 0 to 15 using interrupt priority level setting registers A–F (IPRA–IPRF). The order of priority of the on-chip supporting modules is set to zero by a reset.

When the priority for multiple interrupt sources is set to the same level and such interrupts are generated at the same time, they are handled according to the default order shown in tables 6.4 and 6.5.

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*[1] (0x600)	0–15 (0)	IPRC (3–0)	—	↓
	IRQ1	0x200–3C0*[1] (0x620)	0–15 (0)	IPRC (7–4)	—	
	IRQ2	0x200–3C0*[1] (0x640)	0–15 (0)	IPRC (11–8)	—	
	IRQ3	0x200–3C0*[1] (0x660)	0–15 (0)	IPRC (15–12)	—	
	IRQ4	0x200–3C0*[1] (0x680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	0x200–3C0*[1] (0x6A0)	0–15 (0)	IPRD (7-4)	—	
PINT	PINT0	0x200–3C0*[1] (0x700)	0–15 (0)	IPRD (15–12)	—	
	PINT1					
	PINT2					
	PINT3					
	PINT4					
	PINT5					
	PINT6					
	PINT7					
	PINT8	0x200–3C0*[1] (0x720)	0–15 (0)	IPRD (11–8)	—	
	PINT9					
	PINT10					
	PINT11					
	PINT12					
	PINT13					
	PINT14					
	PINT15					
DMAC	DEI0	0x200–3C0*[1] (0x800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	0x200–3C0*[1] (0x820)			↓	
	DEI2	0x200–3C0*[1] (0x840)				
	DEI3	0x200–3C0*[1] (0x860)			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*[1] (0x880)		0–15 (0)	IPRE (11–8)	High	
	RXI1	0x200–3C0*[1] (0x8A0)				↓	
	BRI1	0x200–3C0*[1] (0x8C0)					
	TXI1	0x200–3C0*[1] (0x8E0)				Low	
SCIF	ERI2	0x200–3C0*[1] (0x900)		0–15 (0)	IPRE (7–4)	High	
	RXI2	0x200–3C0*[1] (0x920)				↓	
	BRI2	0x200–3C0*[1] (0x940)					
	TXI2	0x200–3C0*[1] (0x960)				Low	
ADC	ADI	0x200–3C0*[1] (0x980)		0–15 (0)	IPRE (3–0)	—	
LCDC	LCDI	0x200–3C0*[1] (0x9A0)		0–15 (0)	IPRF (11–8)	—	
PCC	PCC0I	0x200–3C0*[1] (0x9C0)		0–15 (0)	IPRF (7–4)	—	
	PCC1I	0x200–3C0*[1] (0x9E0)		0–15 (0)	IPRF (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	
SCI	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: 1. 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	IRL(3:0)=0000	0x200	(0x200)	15	—	—	↓
	IRL(3:0)=0001	0x220	(0x220)	14	—	—	
	IRL(3:0)=0010	0x240	(0x240)	13	—	—	
	IRL(3:0)=0011	0x260	(0x260)	12	—	—	
	IRL(3:0)=0100	0x280	(0x280)	11	—	—	
	IRL(3:0)=0101	0x2A0	(0x2A0)	10	—	—	
	IRL(3:0)=0110	0x2C0	(0x2C0)	9	—	—	
	IRL(3:0)=0111	0x2E0	(0x2E0)	8	—	—	
	IRL(3:0)=1000	0x300	(0x300)	7	—	—	
	IRL(3:0)=1001	0x320	(0x320)	6	—	—	
	IRL(3:0)=1010	0x340	(0x340)	5	—	—	
	IRL(3:0)=1011	0x360	(0x360)	4	—	—	
	IRL(3:0)=1100	0x380	(0x380)	3	—	—	
	IRL(3:0)=1101	0x3A0	(0x3A0)	2	—	—	
	IRL(3:0)=1110	0x3C0	(0x3C0)	1	—	—	
IRQ	IRQ4	0x200–3C0*[1]	(0x680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	0x200–3C0*[1]	(0x6A0)	0–15 (0)	IPRD (7–4)	—	

Table 6.5 Interrupt Exception Vectors and Rankings (IRL Mode) (cont)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
PINT	PINT0	0x200–3C0*[1] (0x700)	0–15 (0)	IPRD (15–12)	—	
	PINT1					
	PINT2					
	PINT3					
	PINT4					
	PINT5					
	PINT6					
	PINT7					
	PINT8	0x200–3C0*[1] (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	

Table 6.5 Interrupt Exception Vectors and Rankings (IRL Mode) (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	
	RXI2	0x200–3C0*	(0x920)			↓	
	BRI2	0x200–3C0*	(0x940)				
	TXI2	0x200–3C0*	(0x960)			Low	
ADC	ADI	0x200–3C0*	(0x980)	0–15 (0)	IPRE (3–0)	—	
LCDC	LCDI	0x200–3C0*	(0x9A0)	0–15 (0)	IPRF (11–8)	—	
PCC	PCC0I	0x200–3C0*	(0x9C0)	0–15 (0)	IPRF (7–4)	—	
	PCC1I	0x200–3C0*	(0x9E0)	0–15 (0)	IPRF (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	
SCI	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 6.6 Interrupt Exception Vectors and Rankings

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

6.3 INTC Registers

6.3.1 Interrupt Priority Registers A to F (IPRA–IPRF)

Interrupt priority registers A to F (IPRA–IPRF) are 16-bit readable/writable registers that set priority levels from 0 to 15 for external interrupts and on-chip supporting module interrupts. These registers are initialized to H'0000 by a power-on reset or manual reset, but are 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 6.7 lists the relationship between the interrupt sources and bits in IPRA–IPRF.

Table 6.7 Interrupt Request Sources and IPRA–IPRF Registers

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	ADC
IPRF	Reserved*	LCDC	PCC0	PCC1

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

As shown in table 6.7, four sets of on-chip supporting modules are assigned to each register. 4-bit groups (bits 15–12, 11–8, 7–4, and 3–0) 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–IPRF to H'0000.

Bits corresponding to an unused interrupt should be cleared to 0.

6.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode of external interrupt input pin NMI and indicates the input signal level at the NMI pin. This register is initialized to H'0000 or H'8000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit:	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/W

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

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: Read-only bits, always read as 0.

6.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ0–IRQ5 individually: rising edge, falling edge, or low level. This register is initialized to H'0000 by a power-on reset or manual reset, but is not initialized in standby mode. Bits 15 and 13 must be cleared. Writing 1 to these bits is prohibited. Bit 12 is a reserved bit. Writing is invalid, but 0 should be written. This bit is always read as 0.

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/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 15-level interrupt pins encoded 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 in 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. Read-only bit, always read as 0.

Bits 11 and 10—IRQ5 Sense Select (IRQ51S, IRQ50S): These bits select whether the interrupt signal to the IRQ5 pin is detected at the rising edge, at the falling edge, or at the 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, IRQ40S): These bits select whether the interrupt signal to the IRQ4 pin is detected at the rising edge, at the falling edge, or at the 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, IRQ30S): These bits select whether the interrupt signal to the IRQ3 pin is detected at the rising edge, at the falling edge, or at the 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, IRQ20S): These bits select whether the interrupt signal to the IRQ2 pin is detected at the rising edge, at the falling edge, or at the 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, IRQ10S): These bits select whether the interrupt signal to the IRQ1 pin is detected at the rising edge, at the falling edge, or at the 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, IRQ00S): These bits select whether the interrupt signal to the IRQ0 pin is detected at the rising edge, at the falling edge, or at the 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)

ICR2 is a 16-bit readable/writable register that sets the detection mode for external interrupt input pins PINT0–PINT15. This register is initialized to H'0000 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
Bit name:	PINT15S	PINT14S	PINT13S	PINT12S	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 to 0—PINT15 to PINT0 Sense Select (PINT15S–PINT0S): These bits select whether interrupt request signals to PINT15–PINT0 are detected at the low level or high level.

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)

PINTER is a 16-bit readable/writable register that enables interrupt requests input to external interrupt input pins PINT0–PINT15. This register is initialized to H'0000 by a 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	PINT12E	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 to 0—PINT15 to PINT0 Interrupt Enable (PINT15E–PINT0E): These bits enable or disable interrupt request input to pins PINT15–PINT0.

Bit 15–0: PINT15E to PINT0E	**Description**	
0	PINT input interrupt requests disabled	(Initial value)
1	PINT input interrupt requests enabled	

When any of pins PINT0–PINT15 are not used as interrupt inputs, the corresponding bits should be cleared to 0.

6.3.6 Interrupt Request Register 0 (IRR0)

IRR0 is an 8-bit register that indicates interrupt requests from external input pins IRQ0–IRQ5 and PINT0–PINT15. This register 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:	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

To clear an IRQnR bit to 0, read the bit while set to 1, then write 0 to the bit.

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

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

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

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

Bit 5—IRQ5 Interrupt Request (IRQ5R): Indicates whether an interrupt request is input to the 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 the IRQ5 pin	(Initial value)
1	An interrupt request is input to the IRQ5 pin	

Bit 4—IRQ4 Interrupt Request (IRQ4R): Indicates whether an interrupt request is input to the 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 the IRQ4 pin	(Initial value)
1	An interrupt request is input to the IRQ4 pin	

Bit 3—IRQ3 Interrupt Request (IRQ3R): Indicates whether an interrupt request is input to the 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 the IRQ3 pin	(Initial value)
1	An interrupt request is input to the IRQ3 pin	

Bit 2—IRQ2 Interrupt Request (IRQ2R): Indicates whether an interrupt request is input to the 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 the IRQ2 pin	(Initial value)
1	An interrupt request is input to the IRQ2 pin	

Bit 1—IRQ1 Interrupt Request (IRQ1R): Indicates whether an interrupt request is input to the 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 the IRQ1 pin	(Initial value)
1	An interrupt request is input to the IRQ1 pin	

Bit 0—IRQ0 Interrupt Request (IRQ0R): Indicates whether an interrupt request is input to the 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 the IRQ0 pin	(Initial value)
1	An interrupt request is input to the IRQ0 pin	

6.3.7 Interrupt Request Register 1 (IRR1)

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 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:	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)

IRR2 is an 8-bit read-only register that indicates whether LCDC, ADC converter, or SCIF interrupt requests are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

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

Bits 7 and 6—Reserved bits: Writing is invalid, but 0 should be written. Always read as 0.

Bit 5—LCDI Interrupt Request (LCDIR): Indicates whether an LCDI (LCDC) interrupt request is generated.

Bit 5: LCDIR	Description	
0	An LCDI interrupt request is not generated	(Initial value)
1	An LCDI interrupt request is generated	

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	

6.3.9 Interrupt Request Register 3 (IRR3)

IRR3 is an 8-bit read-only register that indicates whether PCC channel 0 interrupt requests are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	PC0SWIR	PC0IRIR	PC0SCIR	PC0CDIR	PC0RCIR	PC0BWIR	PC0BDIR
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	R	R	R	R	R	R	R

Bit 7—Reserved bit: Writing is invalid, but 0 should be written. Always read as 0.

Bit 6—PC0SWI Interrupt Request (PC0SWIR): Indicates whether a PC0SWI (PCC) interrupt request is generated.

Bit 6: PC0SWIR	Description	
0	A PC0SWI interrupt request is not generated	(Initial value)
1	A PC0SWI interrupt request is generated	

Bit 5—PC0IRI Interrupt Request (PC0IRIR): Indicates whether a PC0IRI (PCC) interrupt request is generated.

Bit 5: PC0IRIR	Description	
0	A PC0IRI interrupt request is not generated	(Initial value)
1	A PC0IRI interrupt request is generated	

Bit 4—PC0SCI Interrupt Request (PC0SCIR): Indicates whether a PC0SCI (PCC) interrupt request is generated.

Bit 4: PC0SCIR	Description	
0	A PC0SCI interrupt request is not generated	(Initial value)
1	A PC0SCI interrupt request is generated	

Bit 3—PC0CDI Interrupt Request (PC0CDIR): Indicates whether a PC0CDI (PCC) interrupt request is generated.

Bit 3: PC0CDIR	Description	
0	A PC0CDI interrupt request is not generated	(Initial value)
1	A PC0CDI interrupt request is generated	

Bit 2—PC0RCI Interrupt Request (PC0RCIR): Indicates whether a PC0RCI (PCC) interrupt request is generated.

Bit 2: PC0RCIR	Description	
0	A PC0RCI interrupt request is not generated	(Initial value)
1	A PC0RCI interrupt request is generated	

Bit 1—PC0BWI Interrupt Request (PC0BWIR): Indicates whether a PC0BWI (PCC) interrupt request is generated.

Bit 1: PC0BWIR	Description	
0	A PC0BWI interrupt request is not generated	(Initial value)
1	A PC0BWI interrupt request is generated	

Bit 0—PC0BDI Interrupt Request (PC0BDIR): Indicates whether a PC0BDI (PCC) interrupt request is generated.

Bit 0: PC0BDIR	Description	
0	A PC0BDI interrupt request is not generated	(Initial value)
1	A PC0BDI interrupt request is generated	

6.3.10 Interrupt Request Register 4 (IRR4)

IRR4 is an 8-bit read-only register that indicates whether PCC channel 1 interrupt requests are generated. This register is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	PC1SWIR	PC1CDIR	PC1RCIR	PC1BWIR	PC1BDIR
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	R	R	R	R	R

Bits 7 to 5—Reserved bits: Writing is invalid, but 0 should be written. Always read as 0.

Bit 4—PC1SWI Interrupt Request (PC1SWIR): Indicates whether a PC1SWI (PCC) interrupt request is generated.

Bit 4: PC1SWIR	Description	
0	A PC1SWI interrupt request is not generated	(Initial value)
1	A PC1SWI interrupt request is generated	

Bit 3—PC1CDI Interrupt Request (PC1CDIR): Indicates whether a PC1CDI (PCC) interrupt request is generated.

Bit 3: PC1CDIR	Description	
0	A PC1CDI interrupt request is not generated	(Initial value)
1	A PC1CDI interrupt request is generated	

Bit 2—PC1RCI Interrupt Request (PC1RCIR): Indicates whether a PC1RCI (PCC) interrupt request is generated.

Bit 2: PC1RCIR	Description	
0	A PC1RCI interrupt request is not generated	(Initial value)
1	A PC1RCI interrupt request is generated	

Bit 1—PC1BWI Interrupt Request (PC1BWIR): Indicates whether a PC1BWI (PCC) interrupt request is generated.

Bit 1: PC1BWIR	Description	
0	A PC1BWI interrupt request is not generated	(Initial value)
1	A PC1BWI interrupt request is generated	

Bit 0—PC1BDI Interrupt Request (PC1BDIR): Indicates whether a PC1BDI (PCC) interrupt request is generated.

Bit 0: PC1BDIR	Description	
0	A PC1BDI interrupt request is not generated	(Initial value)
1	A PC1BDI interrupt request is generated	

6.4 INTC Operation

6.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 shows 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 and F (IPRA and IPRF). 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 tables 6.4 and 6.5) is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bit (I3–I0) of 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. Detection timing: First, the INTC is synchronized with the peripheral module clock (Pø) and notifies CPU of an interrupt request. 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 handling routine. This jump is not a delay branch. The interrupt handling routine may branch with the INTEVT/INTEVT2 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.

Note: Interrupt mask bits I3–I0 in the status register (SR) are not changed by the acceptance of an interrupt in the SH-3.

$\overline{\text{IRQOUT}}$ outputs a low level until the interrupt request is cleared. In the interrupt handler, the interrupt request source should be cleared by software.

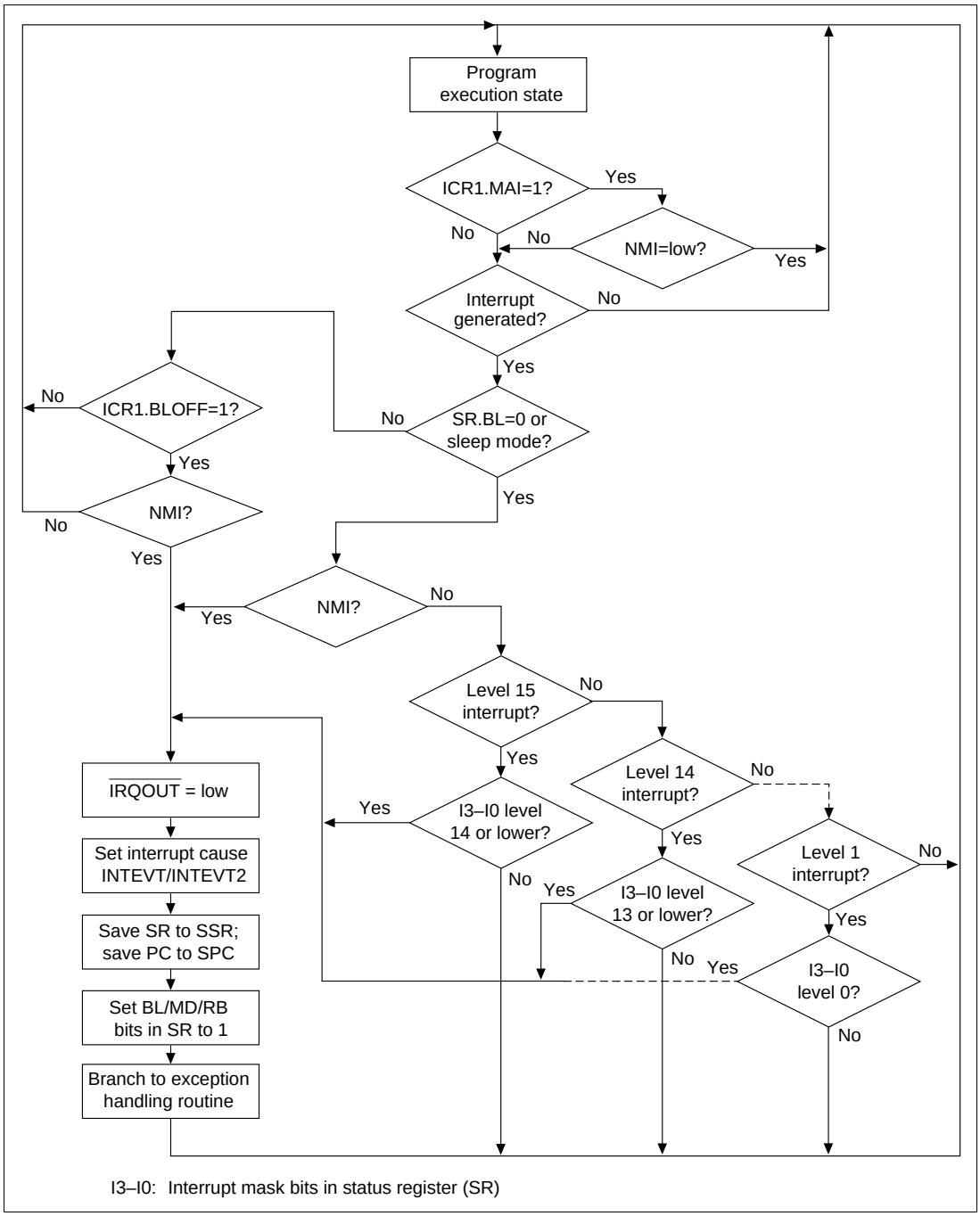


Figure 6.3 Interrupt Operation Flowchart

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 the INTEVT/INTEVT2 register. The code in INTEVT/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 bits I3–I0 in SR.
5. Handle the interrupt.
6. Execute a RTE instruction to exit the handler.

When these steps are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing the BL bit in step 4.

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

Item	Number of States: 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*[4] 0.5 × Icyc + 1 × Bcyc + 2.5 × Pcyc*[5]	0.5 × Icyc + 3.5 × Pcyc	0.5 × Icyc + 1.5 × Pcyc*[6] 0.5 × Icyc + 3 × Pcyc*[7]	
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)

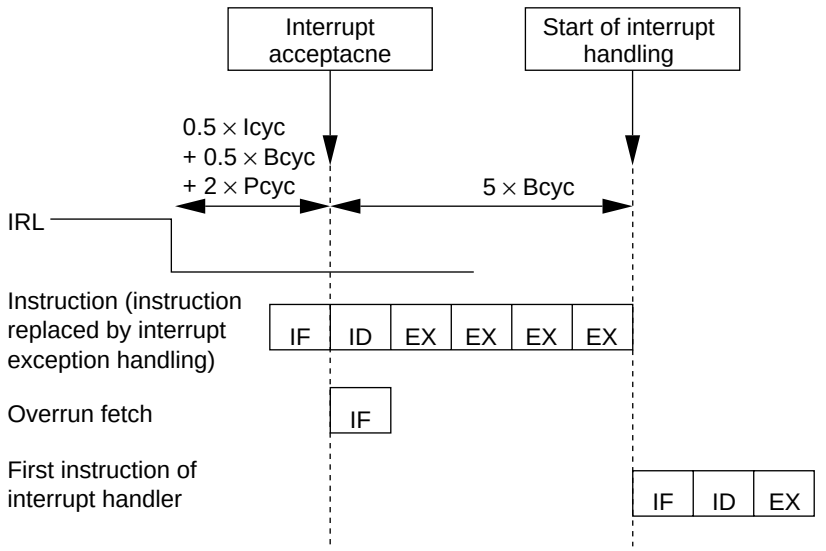
Item		Number of States				Notes
		NMI	IRQ	PINT	Supporting Modules	
Response time	Total	(5.5 + X) × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	(5.5 + X) × Icyc + 1 × Bcyc + 4.5 × Pcyc*4	(5.5 + X) × Icyc + 3.5 × Pcyc	(5.5 + X) × Icyc + 1.5 × Pcyc*6	
			(5.5 + X) × Icyc + 1 × Bcyc + 2.5 × Pcyc*5		(5.5 + X) × Icyc + 3 × Pcyc*7	
	Minimum case*2	7.5	16.5	12.5	8.5*6/11.5*7	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*6 16.5 + S*7	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. IRQ mode
5. IRL mode
6. Modules: TMU, RTC, SCI, WDT, REFC
7. Modules: DMAC, LCDC, ADC, IRDA, SCIF, PCC



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

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 conditions 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 a logical block diagram of the user break controller.

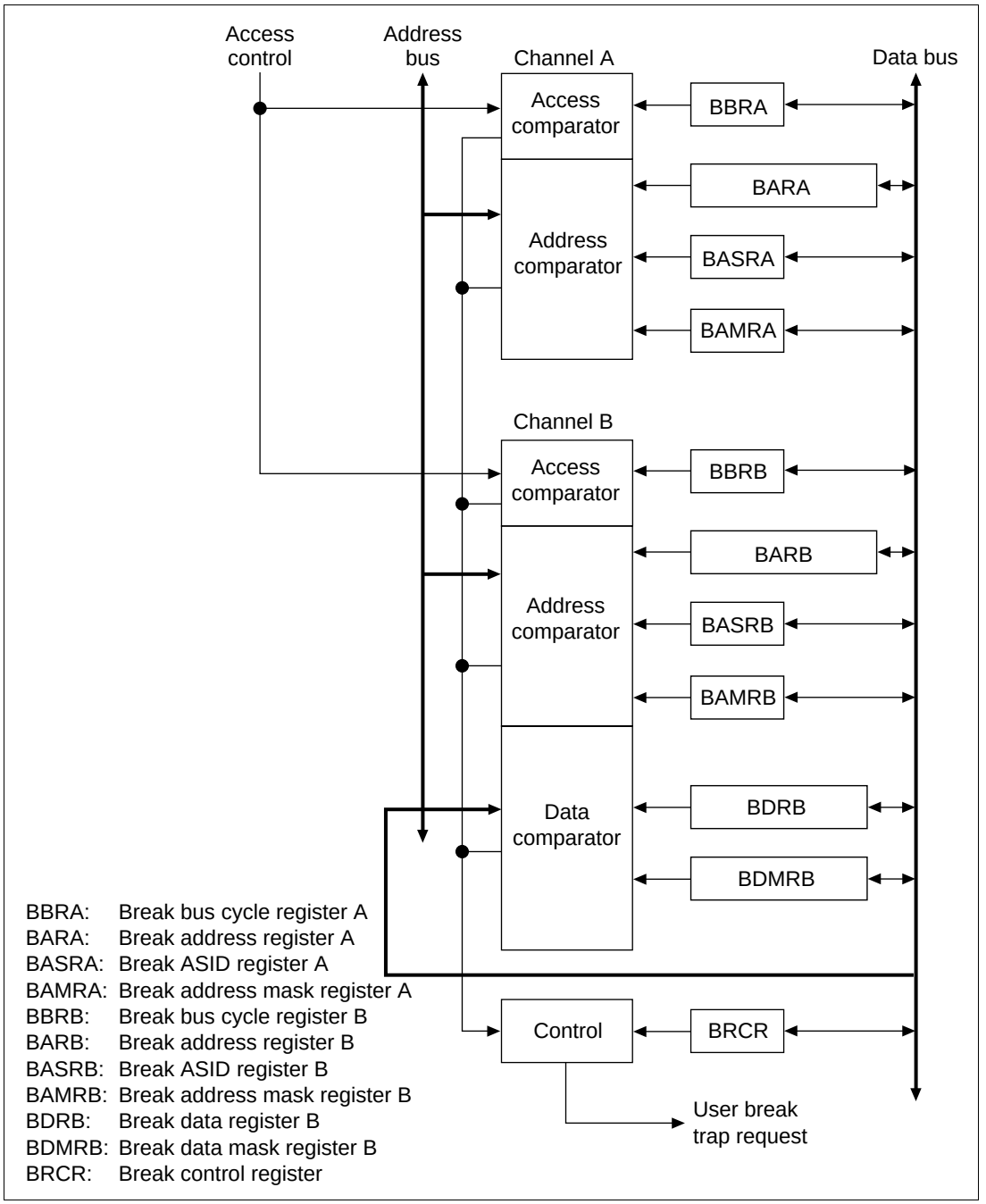


Figure 7.1 Logical Block Diagram of User Break Controller

7.1.3 Register Configuration

Table 7.1 shows the user break controller registers.

Table 7.1 UBC Registers

Channel	Register	Initial Value*[1]	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*[2]	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*[2]	Word	H'FFFFFFA8	R/W
	BDRB	Undefined	Longword	H'FFFFFF90	R/W
	BDMRB	Undefined	Longword	H'FFFFFF94	R/W
Common	BRCR	H'0000*[2]	Word	H'FFFFFF98	R/W

Notes: 1. Value is retained in standby mode.
2. Initialized by a power-on or manual reset.

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, this 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	BA10	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 readable/writable register that stores the virtual address of the channel A break condition. It is not initialized by a manual reset.

Bits 31 to 0—Break Address A31 to 0 (BAA31–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 readable/writable register. It is not initialized by a manual reset.

Bits 7 to 0—Break ASID A7 to 0 (BASA7–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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break address mask register A (BAMRA) is an 8-bit readable/writable 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 manual 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, 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 readable/writable 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, 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 cycle	
1	0	Break on write cycle	
	1	Break on both read and write cycles	

Bits 1 and 0—Operand Size Select A (SZA1, 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 readable/writable register that specifies the data that is the break condition for channel B data breaks. BDRB is not initialized by a manual reset.

BDRB Bits 31 to 0—Break Data B31 to B0 (BDB31–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 readable/writable register that determines which of the bits in the break address set in BDRB are masked. BDMRB is not initialized by a manual reset.

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

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 accesses.

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 readable/writable 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 made, clear the flag 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 made, clear the flag 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	Channel A PC break placed before instruction execution (Initial value)
1	Channel A PC break placed 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	Data bus conditions not included in the channel B conditions (Initial value)
1	Data bus conditions included in the channel B conditions

Note: When the data bus is not included in the break conditions, the IDB1 and IDB0 bits in break bus cycle register B (BBRB) should be set to 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	Channel B PC break placed before instruction execution (Initial value)
1	Channel B PC break placed 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 handling, 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	Channel A and B conditions compared independently (Initial value)
1	Channel A and B conditions compared 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 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 occurs 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 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 SPC in user break interrupt handling 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 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 handled 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. Moreover, a multiplexer, or a bus controller (BSCP) dynamically supplies a clock to reduce power consumption.

8.1.1 Power-Down Modes

The SH7707 has the following three power-down modes, plus a dynamic clocking feature:

1. Sleep mode
2. Standby mode
3. Module standby function (DMAC, LCDC, BSC, TMU, RTC, SCI, DAC, ADC, PCC, IRDA, and SCIF on-chip supporting modules)

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

Table 8.1 Power-Down Modes

Mode	Transition Conditions	State								
		CPG	CPU	CPU Register	On-Chip Memory	On-Chip Supporting Modules	Pins	External Memory	BSC	Canceling Procedure
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts	Held	Held	Runs	Held	Refresh *2	Runs/ Halts *3	1. Interrupt 2. Reset
Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Held	Halts*1	Held	Self-refresh	Halts	1. Interrupt 2. Reset
Module standby	Set MSTP bit to 1 in STBCR	Run	Runs/ Halts*4	Held	Held	Specified module halts	*5	Refresh		1. Set MSTP bit to 0 2. Reset

Notes: 1. The RTC still runs if the START bit is set to 1 in RCR2 (see section 14, Real-Time Clock (RTC)). The TMU still runs when output of the RTC is used as input to its counter (see section 13, Timer (TMU)).

2. The external bus-clock frequency can be reduced to 1/4 during sleep mode by a register setting. The memory refresh rate remains unchanged. This enables the power consumption to be reduced.
3. Only during sleep mode, the supply of the clock to the BSC is halted by a register setting. The clock is supplied only during operation to the BSCP for the DMAC/LCDC, and operation is always enabled.
4. Sleep mode: Halts
   Normal mode: Runs
5. Depends on the on-chip supporting module.
   TMU external pins: Held
   SCI external pins: Reset

8.1.2 Register Configuration

Table 8.2 shows details of the registers used for the power-down modes.

Table 8.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Access Size	Address
Standby control register	STBCR	R/W	H'00	8	H'FFFFFF82
Standby control register 2	STBCR2	R/W	H'00	8	H'FFFFFF88
Standby control register 3	STBCR3	R/W	H'00	8	H'FFFFFF8A

8.1.3 Pin Configuration

Table 8.3 lists the pins used for the power-down modes.

Table 8.3 Pin Configuration

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

Note: In sleep mode, also, the STATUS1 and STATUS2 pins both go low during execution of a refresh cycle.

8.2 Registers

8.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit readable/writable register that sets the power-down mode. STBCR is initialized to H'00 by a power-on reset.

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/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 6–3—Reserved: Read-only bits, always read as 0.

Bit 2—Module Stop 2 (MSTP2): Specifies halting of the clock supply to the timer unit (TMU) 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	
0	TMU runs	(Initial value)
1	Clock supply to TMU halted	

Bit 1—Module Stop 1 (MSTP1): Specifies halting of the clock supply to the real-time clock (RTC) on-chip supporting module. When the MSTP1 bit is set to 1, the supply of the clock to the 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 halted	

Bit 0—Module Stop 0 (MSTP0): Specifies halting of the clock supply to the serial communication interface (SCI) 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 runs	(Initial value)
1	Clock supply to SCI halted	

8.2.2 Standby Control Register 2 (STBCR2)

Standby control register 2 (STBCR2) is an 8-bit readable/writable register that controls individual supporting module operation in sleep mode. STBCR2 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Bit name:	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3
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—Module Stop 10 (MSTP10): Specifies halting of the clock supply to the DMAC on-chip supporting module. When the MSTP10 bit is set to 1, the supply of the clock to the DMAC is halted.

Bit 7: MSTP10	Description	
0	DMAC runs	(Initial value)
1	Clock supply to DMAC halted	

Bit 6—Module Stop 9 (MSTP9): Specifies halting of the clock supply to the DAC on-chip supporting module. When the MSTP9 bit is set to 1, the supply of the clock to the DAC is halted.

Bit 6: MSTP9	Description	
0	DAC runs	(Initial value)
1	Clock supply to DAC halted	

Bit 5—Module Stop 8 (MSTP8): Specifies halting of the clock supply to the ADC on-chip supporting module. When the MSTP8 bit is set to 1, the supply of the clock to the ADC is halted.

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

Bit 4—Module Stop 7 (MSTP7): Specifies the display-off state for the LCDC on-chip supporting module. When the MSTP7 bit is set to 1, the LCDC display turns off.

Bit 4: MSTP7	Description	
0	LCDC runs	(Initial value)
1	LCDC display off	

Bit 3—Module Stop 6 (MSTP6): Specifies halting of display data transfer from memory to the LCDC. When the MSTP6 bit is set to 1, the LCDC halts display data transfer, halts updating of display data, and continuously displays 0 data.

Bit 3: MSTP6	Description	
0	LCDC runs	(Initial value)
1	Display data transfer halted	

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

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

Bit 1—Module Stop 4 (MSTP4): Specifies halting of the clock supply to the serial communication interface with FIFO (SCIF) on-chip supporting module. When the MSTP4 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 serial communication interface with IrDA (IRDA) on-chip supporting 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	

8.2.3 Standby Control Register 3 (STBCR3)

Standby control register 3 (STBCR3) is an 8-bit readable/writable register that specifies bus state controller (BSC) operation in sleep mode and the watchdog timer (WDT) maximum clock division ratio.

Bit:	7	6	5	4	3	2	1	0
Bit name:	MSTP SLP0	—	—	—	—	—	—	CKSS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	—	—	—	—	—	—	R/W

Bit 7—Sleep Mode Module Stop (MSTPSLP0): Specifies halting of the clock supply to the bus state controller (BSC) in sleep mode, when power consumption can be reduced by operating only the bus state controller with lower power (BSCP) since the BSC only performs memory refreshing and data transfer by the LCDC/DMAC. The MSTPSLP0 bit does not affect BSC operation in normal mode. When the MSTPSLP0 bit is set to 1, the supply of the clock to the BSC is halted in sleep mode.

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

Bit 0—Clock Select Special (CKSS): Specifies the clock to be used for the watchdog timer count regardless of the CKS[2:0] bits in WTCSR described in section 9. The P(ϕ) division ratio is defined by CKS[2:0] in the range from 1 to 1/4096 when CKSS is cleared to 0.

Bit 0:CKSS	Description
0	Watchdog timer clock division ratio is specified by CKS[2:0] bits
1	Watchdog timer clock division ratio is fixed at 1/65536 of P(ϕ)*

Note: * Assuming a P(ϕ) frequency of 15MHz, the watchdog timer clock period is 1.12 s. This specification can be used to secure the settling time of 32kHz PLL.

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.

8.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip supporting module) or reset. Interrupts are accepted during sleep mode even when the BL bit in the SR register is 1. Save SPC and SSR on the stack before executing the SLEEP instruction if necessary.

Canceling with an Interrupt: When an NMI, IRQ, IRL or on-chip supporting interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code indicating the interrupt source is set in the INTEVT and INTEVT2 registers.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or manual reset. A code indicating the reset type is set in the EXPEVT register.

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 switches 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 state of registers in standby mode.

Table 8.4 Register States in Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller	—	All registers
On-chip oscillator circuit	—	All registers
Break controller	—	All registers
Bus state controller	—	All registers
PC card controller	—	All registers
Direct memory access controller	—	All registers
Timer unit	TSTR register	Registers other than TSTR
	UNF bit in TCR register, ICPF bit in TCR register	Bits other than those at left
Real-time clock	—	All registers
Serial communication interface	All registers	—
Infrared Data Association interface	All registers	—
Serial communication interface with FIFO	All registers	—
Pin function controller / I/O port	—	All registers
LCD controller	—	All registers
A/D converter	All registers	—
D/A converter	All registers	—

The procedure for switching to standby mode is as follows:

1. Set the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT. Set the WDT's timer counter (WTCNT) to 0 and the CKS2–CKS0 bits in the WTCSR register to appropriate values to obtain the specified oscillation settling time.
2. When PLL circuit 1 is running in clock modes 3–7, set 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, IRQ, or on-chip supporting module) or a reset.

Canceling with an Interrupt: The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRQ[*1], IRL,[*2] or on-chip supporting module (other than interval timer)[*3] interrupt, the clock will be supplied to the entire chip and standby mode canceled after the time set in the WDT's timer control/status register has elapsed. The STATUS1 and STATUS0 pins both go low. Interrupt handling then begins and a code indicating the interrupt source is set in the INTEVT and INTEVT2 registers. After branching to the interrupt handler, clear the STBY bit in the STBCR register. WTCNT will then halt automatically. If this bit is not cleared, WTCNT will continue to count up, and on reaching H'80, will cause a transition to standby mode. This function prevents data corruption due to runaway when the power supply is unstable, etc. Interrupts are accepted during standby mode even when the BL bit in the SR register is 1. Save SPC and SSR on the stack before SLEEP instruction execution if necessary. Immediately after an interrupt is detected, the phase of the CKIO pin clock output may be unstable, until standby mode is canceled. (The canceling condition is that the interrupt request level (IRQ, IRL, or on-chip supporting module) is higher than the mask level in bits I3-I0 in the SR register.)

Notes: 1. IRQ0–IRQ3 can be used. IRQ4 and IRQ5 cannot be used. When the RTC is being used, its use is permitted.
2. When the RTC is being used, standby mode can be canceled using $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$.
3. Standby mode can be canceled with an RTC or TMU (only when running on the RTC clock) interrupt.

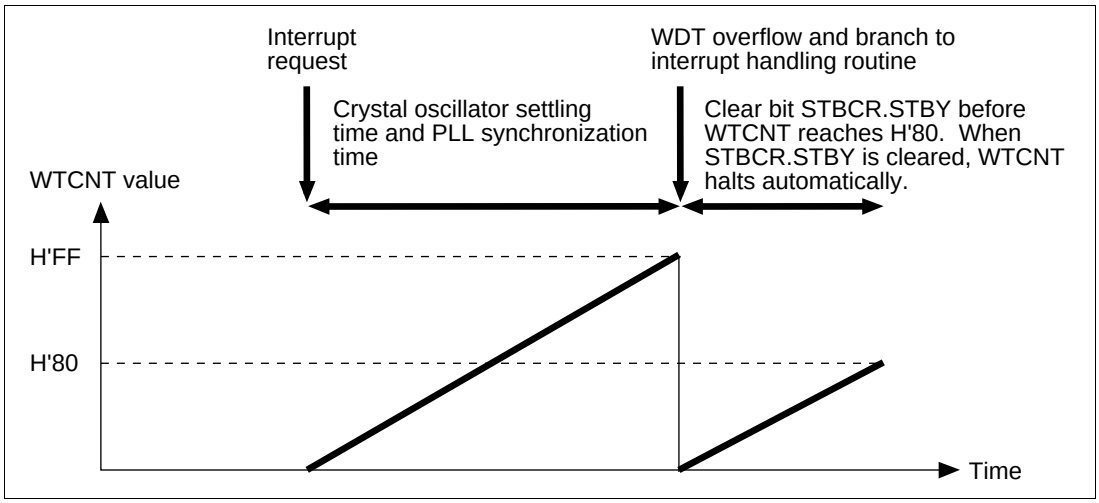


Figure 8.1 Canceling Standby Mode with STBCR.STBY

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 goes low and the STATUS0 pin output goes 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 or IRQ 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 within the chip, the STATUS1–STATUS0 pins both go low, and operation resumes from interrupt handling.

8.5 Module Standby Function

8.5.1 Transition to Module Standby Function

Setting standby control register bits MSTP10–MSTP0 to 1 halts the clock supply to the corresponding on-chip supporting modules. This function can be used to reduce power consumption in normal mode and sleep mode. The module standby function retains the status of the external pins of the on-chip supporting modules prior to the halt. With a few exceptions, all registers hold their values.

Bit	Value	Description
MSTP10	0	DMAC runs
	1	Supply of clock to DMAC halted
MSTP9	0	DAC runs
	1	Supply of clock to DAC halted
MSTP8	0	ADC runs
	1	Supply of clock to ADC halted
MSTP7	0	LCDC runs
	1	LCDC display off
MSTP6	0	LCDC runs
	1	LCDC data transfer halted
MSTP5	0	PCC runs
	1	Supply of clock to PCC halted
MSTP4	0	SCIF runs
	1	Supply of clock to SCIF halted. Registers initialized*[1]
MSTP3	0	IRDA runs
	1	Supply of clock to IRDA halted. Registers initialized*[1]
MSTP2	0	TMU runs
	1	Supply of clock to TMU halted. Some register bits initialized*
MSTP1	0	RTC runs
	1	Supply of clock to RTC halted. Register access prohibited*[3]
MSTP0	0	SCI operates
	1	Supply of clock to SCI halted. Registers initialized*[1]

Notes: 1. The registers initialized are the same as in standby mode (table 8.4).
2. Some register bits are initialized, according to register settings.
3. The counter runs.

8.5.2 Canceling the Module Standby Function

The module standby function is canceled by clearing the MSTP10–MSTP0 bits to 0, by a power-on reset, or by a manual reset.

8.6 Timing of STATUS Pin Changes

The timing of STATUS pin changes is shown in figures 8.1 through 8.9.

The meaning of STATUS is as follows:

- Reset: HH (Both STATUS1 and STATUS0 are high.)
- Sleep: HL (STATUS1 is high and STATUS0 is low.)
- Standby: LH (STATUS1 is low and STATUS0 is high.)
- Normal: LL (Both STATUS1 and STATUS0 are 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 and 1):

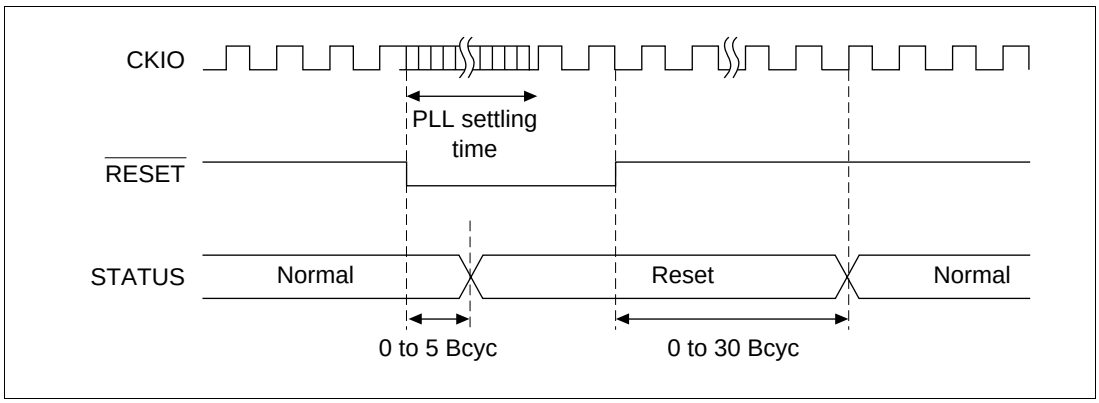


Figure 8.2 Power-On Reset STATUS Output (Clock Modes 0 and 1)

Power-On Reset (Clock Modes 2–7):

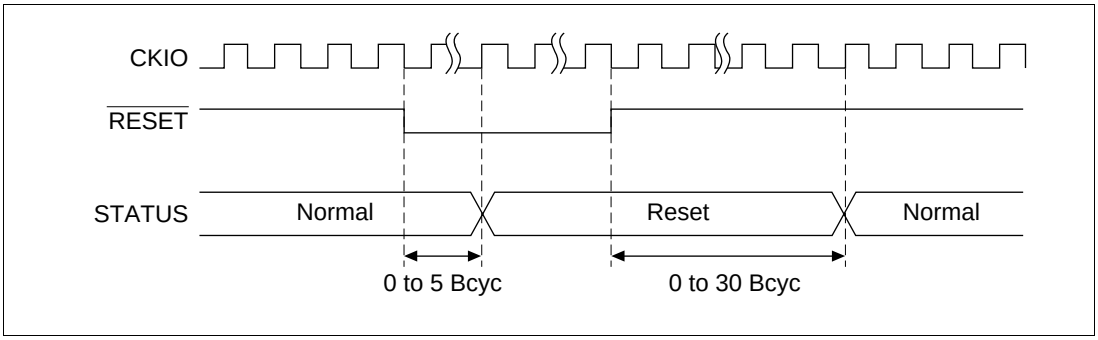
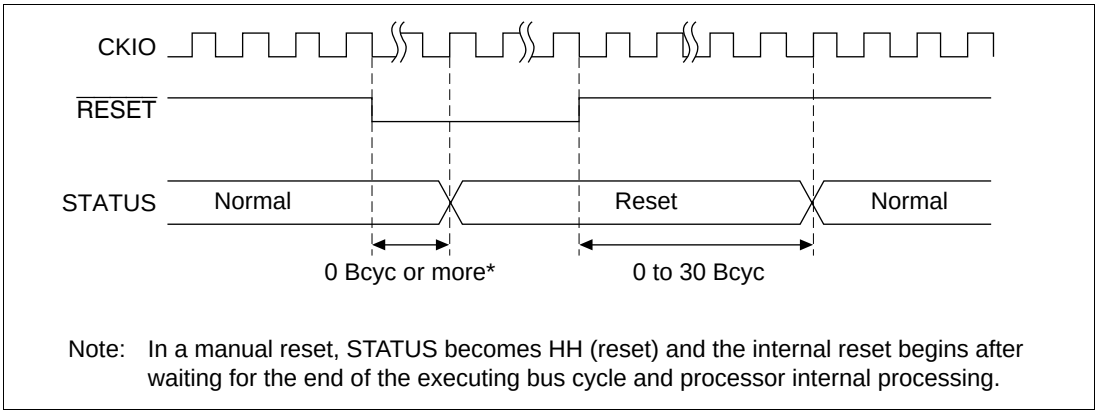


Figure 8.3 Power-On Reset STATUS Output (Clock Modes 2–7)

Manual Reset:



Note: In a manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting for the end of the executing bus cycle and processor internal processing.

Figure 8.4 Manual Reset STATUS Output

8.6.2 Timing for Canceling Standby Mode

Standby to Interrupt:

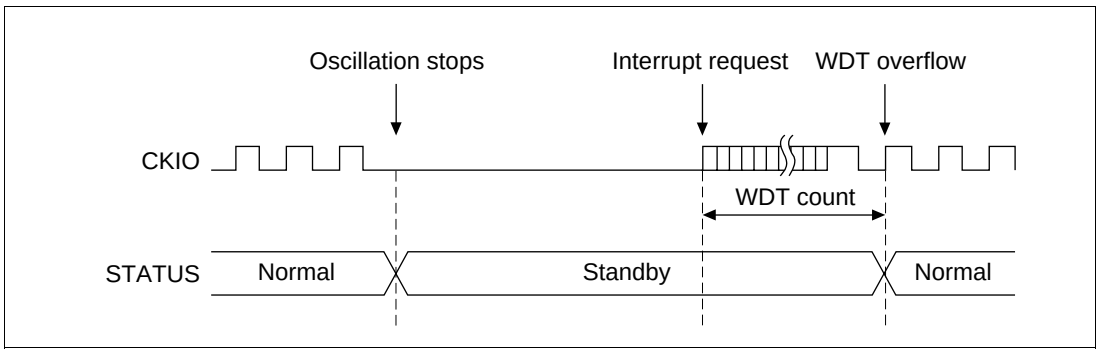


Figure 8.5 Standby to Interrupt STATUS Output (1)
–Normal Case–

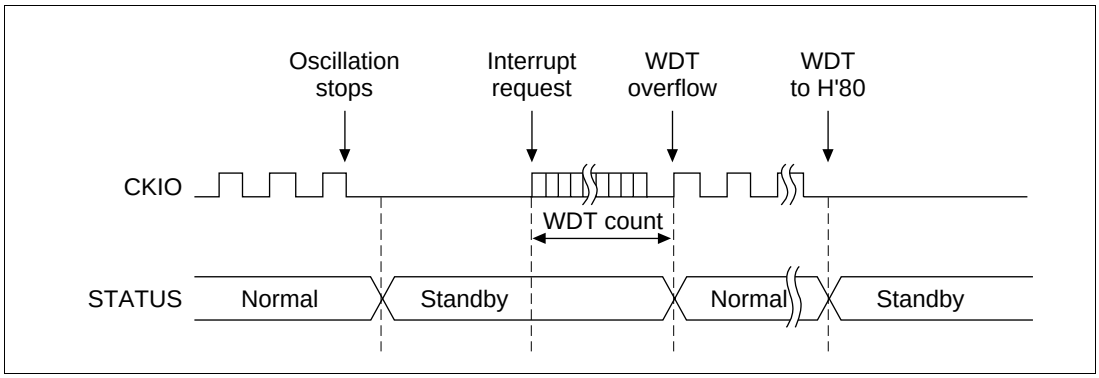
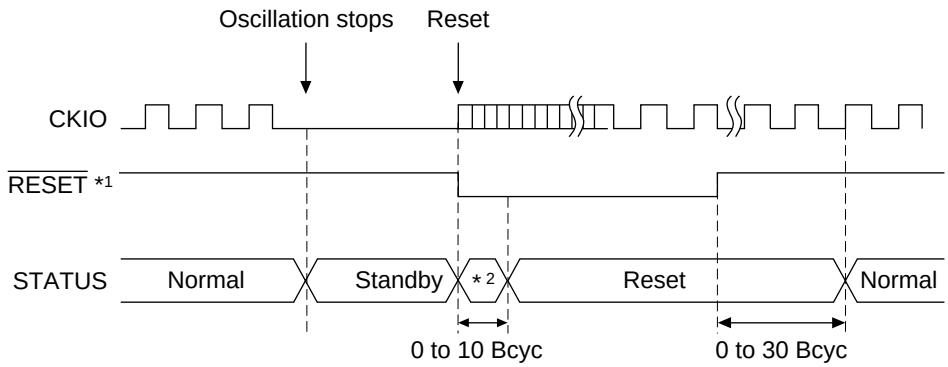


Figure 8.6 Standby to Interrupt STATUS Output (2)
–Return to Standby Mode–

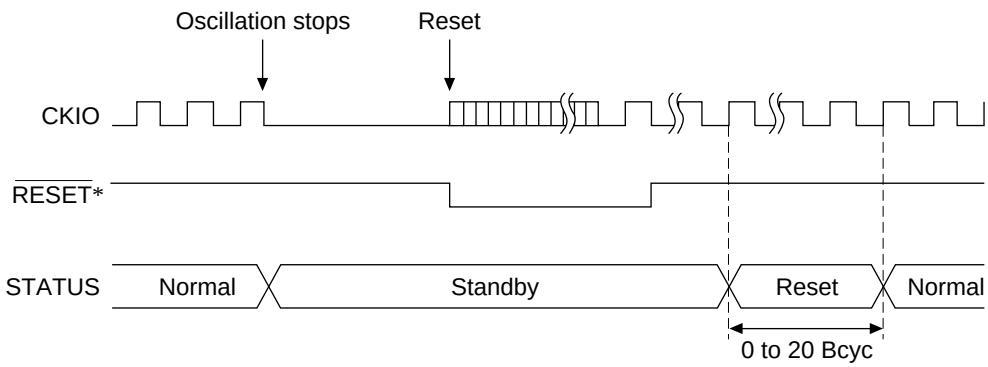
Standby to Power-On Reset:



Notes: 1. When standby mode is cleared with a power-on reset, the WDT does not count. Keep $\overline{\text{RESET}}$ low during the PLL's oscillation settling time.
2. Undefined

Figure 8.7 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 $\overline{\text{RESET}}$ low during the PLL's oscillation settling time.

Figure 8.8 Standby to Manual Reset STATUS Output

8.6.3 Timing for Canceling Sleep Mode

Sleep to Interrupt:

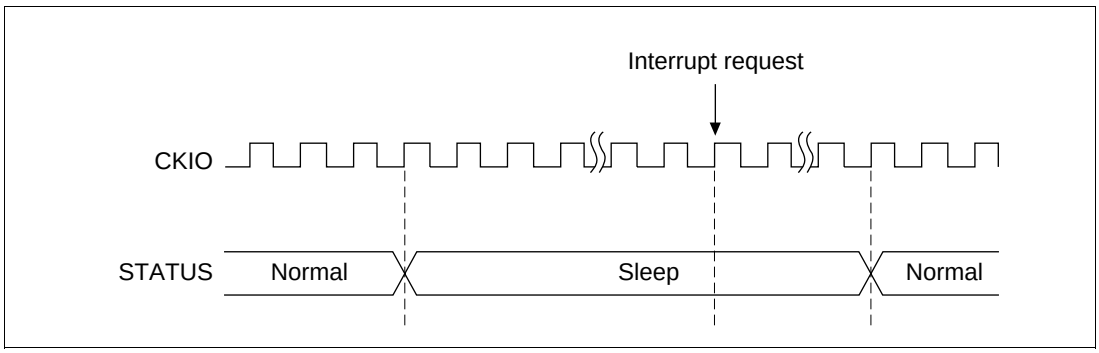
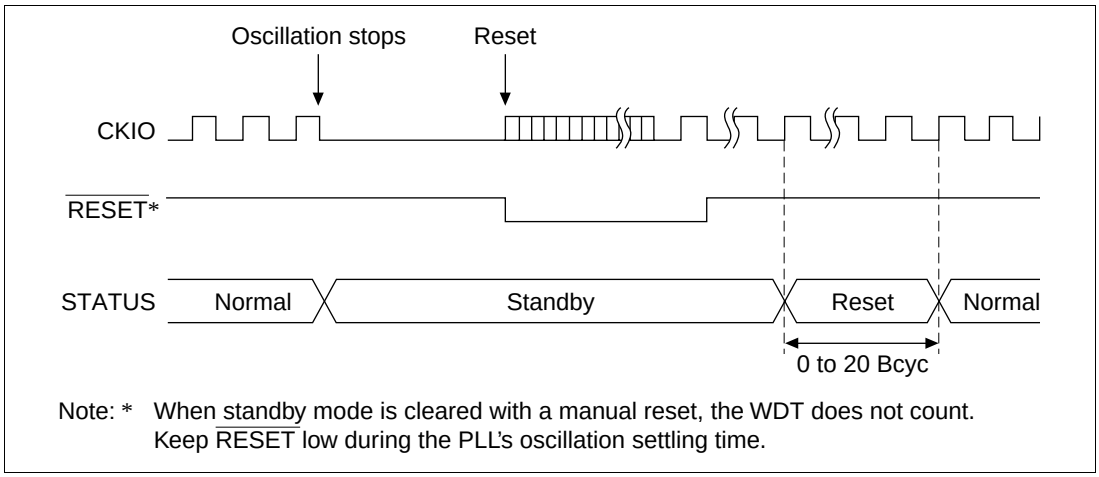


Figure 8.9 Sleep to Interrupt STATUS Output

Sleep to Power-On Reset:



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

Figure 8.10 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset:

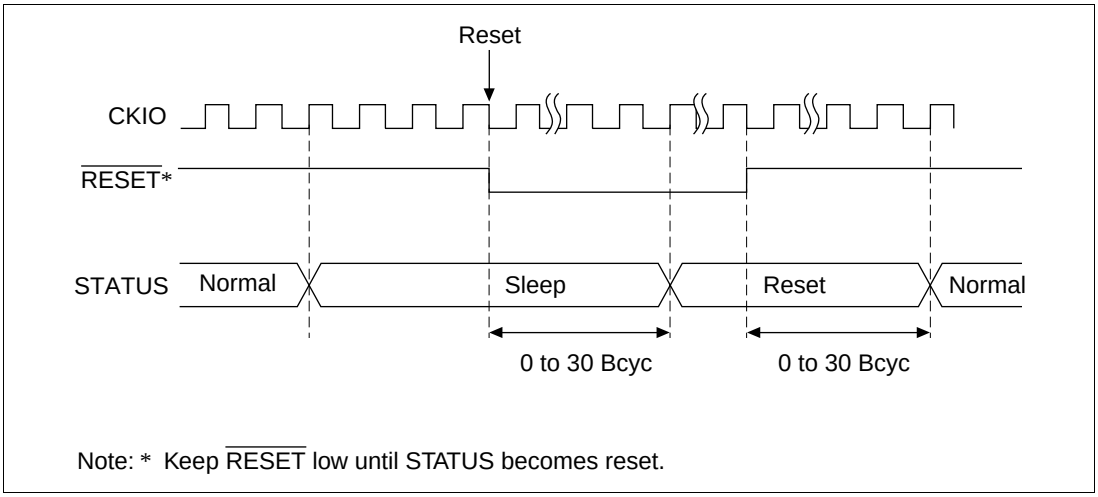


Figure 8.11 Sleep to Manual Reset STATUS Output

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:

- Eight clock modes: Select from eight clock modes for different frequency ranges, power consumptions, direct crystal input, external clock input, and a clock multiplied from 32.768 kHz for the RTC .
- Three clocks generated independently: An internal clock (I ϕ) for the CPU, cache, and TLB; a peripheral clock (P ϕ) for the on-chip supporting modules; and a bus clock (CKIO) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- PLL on/off: 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: An internal reset occurs after counter overflow. Select between power-on reset and manual reset.
- Generates interrupts in interval timer mode: An internal timer interrupt occurs after counter overflow.
- Select from nine counter input clocks. Nine clocks ($\times$1 to $\times$1/4096 and $\times$1/65536) can be obtained by dividing the peripheral clock.

9.2 Overview of CPG

9.2.1 CPG Block Diagram

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

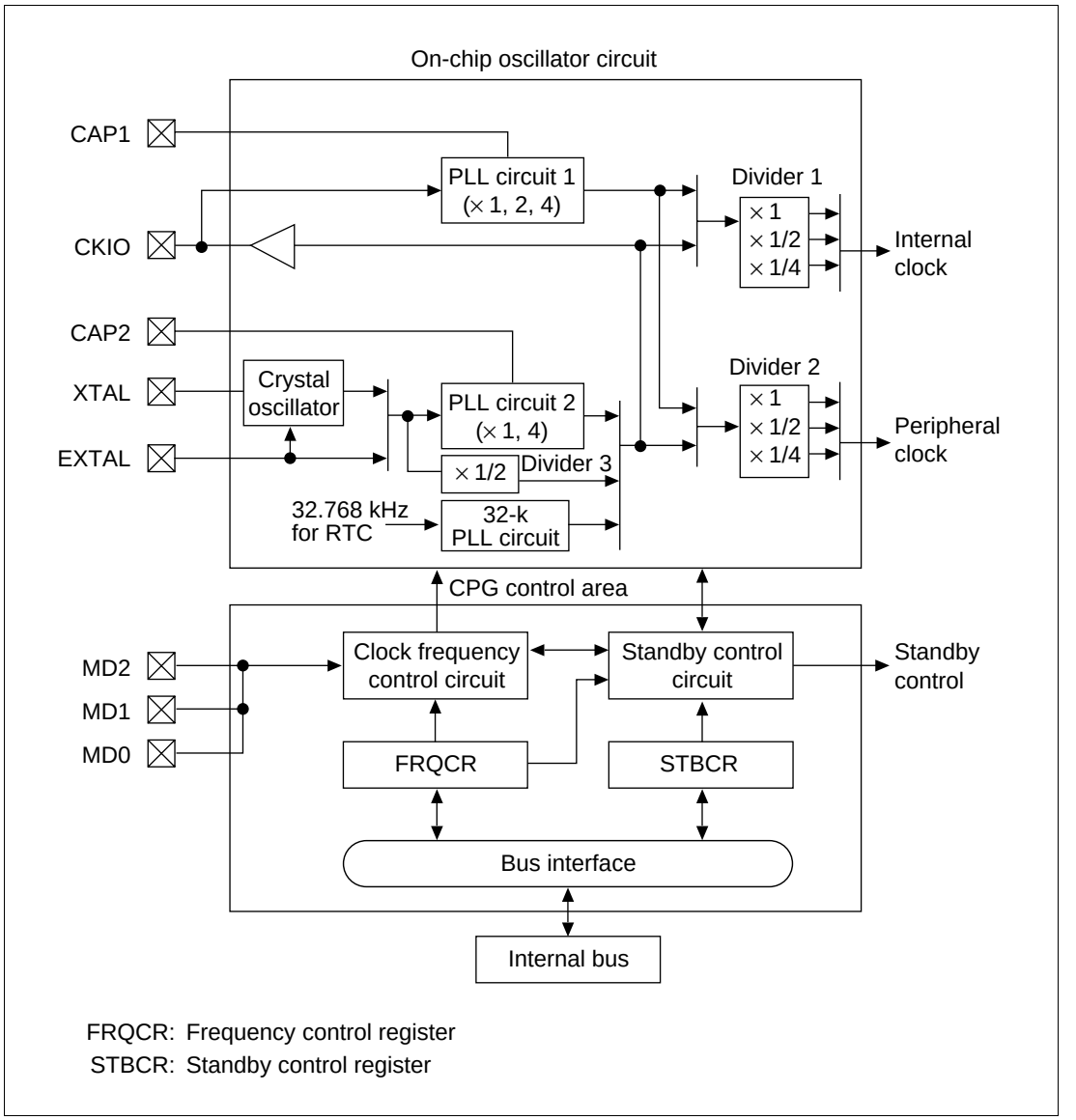


Figure 9.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

1. PLL Circuit 1: PLL circuit 1 doubles, 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 on 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 modes. The clock operation mode is set by 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. 32-K PLL Circuit: This circuit multiplies the 32.768-kHz clock in the crystal oscillation circuit for the RTC to generate 14.7456 MHz and 11.0756 MHz. The clock consists of the 32.768-kHz crystal oscillator which is connected to pins EXTAL2 and XTAL2. This circuit can be used in clock modes 6 and 7.
5. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal clock (I ϕ). The operating frequency can be 1, 1/2, or 1/4 times the output frequency of PLL circuit 1, as long as it not lower than the clock frequency of the CKIO pin. The dividing ratio is set in the frequency control register.
6. Divider 2: Divider 2 generates a clock at the operating frequency used by the peripheral clock (P ϕ). The operating frequency can be 1, 1/2, or 1/4 times the output frequency of PLL circuit 1 or the clock frequency of the CKIO pin, as long as it does not exceed the clock frequency of the CKIO pin. The dividing ratio is set in the frequency control register.
7. Divider 3: Divider 3 makes the duty of the clock waveform 50% by reducing the input clock frequency to 1/2, when clock input is supplied from the EXTAL pin to the internal device without using PLL2 or 32-K PLL.
8. Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD pins and the frequency control register.
9. Standby Control Circuit: The standby control circuit controls the status of the on-chip oscillator circuit and other modules during clock switching and sleep/standby modes.
10. 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 dividing ratio of the internal clock and the peripheral clock.
11. 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

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 terminal	CKIO	I/O	Inputs or outputs 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 summarizes the CPG register.

Table 9.2 Register Configuration

Register Name	Abbreviation	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 lists the relationship between the mode control pin (MD2–MD0) combinations and the clock operating modes. Table 9.4 lists the usable frequency ranges in the clock operating modes.

Table 9.3 Clock Operating Modes

Mode	Pin Value			Clock I/O		PLL2 ON/OFF	32-K PLL	Divider 3	PLL1 ON/OFF	Divider 1 Input	Divider 2 Input	CKIO Frequency
	MD2	MD1	MD0	Source	Output							
0	0	0	0	EXTAL	CKIO	ON Multiplier ratio: 1	OFF	OFF	ON	Output of PLL1	Output of PLL1	(EXTAL)
1	0	0	1	Crystal oscillator	CKIO	ON Multiplier ratio: 4	OFF	OFF	ON	Output of PLL1	Output of PLL1	(Crystal) x 4
2	0	1	0	CKIO	—	OFF	OFF	OFF	ON	Output of PLL1	Output of PLL1	(CKIO)
3	0	1	1	Crystal oscillator	CKIO	ON Multiplier ratio: 1	OFF	OFF	OFF (Initial value)	Output of PLL2	Output of PLL2	(Crystal)
									ON	Output of PLL1		
4	1	0	0	EXTAL	CKIO	ON Multiplier ratio: 1	OFF	OFF	OFF (Initial value)	Output of PLL2	Output of PLL2	(EXTAL)
									ON	Output of PLL1		
5	1	0	1	EXTAL	CKIO	OFF	OFF	ON (1/2)	OFF (Initial value)	Output of divider 3	Output of divider 3	(EXTAL) × 1/2
									ON	Output of PLL1		
6	1	1	0	32.768-kHz crystal oscillator for RTC	CKIO	OFF	ON	OFF	OFF (Initial value)	Output of 32-k PLL	Output of 32-k PLL	14.7456 MHz
									ON	Output of PLL1		
7	1	1	1	32.768-kHz crystal oscillator for RTC	CKIO	OFF	ON	OFF	OFF (Initial value)	Output of 32-k PLL	Output of 32-k PLL	11.0756 MHz
									ON	Output of PLL1		

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7707. PLL circuit 1 is constantly on, and there are no frequency range restrictions compared to mode 4. An input clock frequency of 10 MHz to 30 MHz can be used, and the CKIO frequency range is 10 MHz to 30 MHz.

Mode 1: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside the SH7707, allowing a low crystal frequency to be used. A crystal oscillation frequency of 5 MHz to 7.5 MHz can be used, and the CKIO frequency range is 20 MHz to 30 MHz.

Mode 2: 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 SH7707. In modes 0, 1, 3 to 7, the system clock is generated from the output of the SH7707'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.

Mode 3: The on-chip crystal oscillator operates, with its output supplied inside the SH7707 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 10 MHz to 15 MHz can be used, and the CKIO frequency range is 10 MHz to 15 MHz.

Mode 4: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7707. 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 10 MHz to 15 MHz can be used, and the CKIO frequency range is 10 MHz to 15 MHz.

Mode 5: An external clock is input from the EXTAL pin and undergoes waveform shaping with its frequency scaled by 1/2 by frequency divider 3 before being supplied inside the SH7707. The external clock frequency must be twice that of the desired CKIO frequency. Since PLL circuit 2 is not used, the chip power consumption can be kept lower than in mode 3. When PLL circuit 1 is not used, an input clock frequency of 2 MHz to 30 MHz can be used, and the CKIO frequency range is 1 MHz to 15 MHz. When PLL circuit 1 is used, an input clock frequency of 20 MHz to 30 MHz can be used, and the CKIO frequency range is 10 MHz to 15 MHz.

Mode 6: A 32.768 kHz clock for the RTC is input and multiplied to 14.7456 MHz by PLL circuit 3 before being supplied inside the SH7707. 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. As only a single 32.768 kHz crystal for the RTC is used in this mode, costs are reduced. However, note that a long PLL circuit 3 oscillation settling time is required in this mode.

Mode 7: A 32.768 kHz clock for the RTC is input and multiplied to 11.0756 MHz by PLL circuit 3 before being supplied inside the SH7707. 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. As only a single 32.768 kHz crystal for the RTC is used in this mode, costs are reduced. However, note that a long PLL circuit 3 oscillation settling time is required in this mode.

Table 9.4 Range of Usable Frequencies for Each Clock Operating Mode

Mode	FRQCR	PLL1	PLL2	Clock Ratio* (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
0	0102	ON (×1)	ON (×1)	1:1:1/4	10 MHz to 30 MHz	10 MHz to 30 MHz
	0101	ON (×1)	ON (×1)	1:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0100	ON (×1)	ON (×1)	1:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0112	ON (×2)	ON (×1)	2:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0111	ON (×2)	ON (×1)	2:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0115	ON (×2)	ON (×1)	1:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0116	ON (×2)	ON (×1)	1:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0122	ON (×4)	ON (×1)	4:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	0126	ON (×4)	ON (×1)	2:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	012a	ON (×4)	ON (×1)	1:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
1	0102	ON (×1)	ON (×4)	4:4:1	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0101	ON (×1)	ON (×4)	4:4:2	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0100	ON (×1)	ON (×4)	4:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0112	ON (×2)	ON (×4)	8:4:2	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	0111	ON (×2)	ON (×4)	8:4:4	5 MHz to 7.5 MHz	20 MHz to 30 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 7.5 MHz	20 MHz to 30 MHz
2	0102	ON (×1)	OFF	1:1:1/4	10 MHz to 30 MHz	10 MHz to 30 MHz
	0101	ON (×1)	OFF	1:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0100	ON (×1)	OFF	1:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0112	ON (×2)	OFF	2:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0111	ON (×2)	OFF	2:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0115	ON (×2)	OFF	1:1:1	10 MHz to 30 MHz	10 MHz to 30 MHz
	0116	ON (×2)	OFF	1:1:1/2	10 MHz to 30 MHz	10 MHz to 30 MHz
	0122	ON (×4)	OFF	4:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	0126	ON (×4)	OFF	2:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	012a	ON (×4)	OFF	1:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz

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

Mode	FRQCR	PLL1	PLL2	Clock Ratio* (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
3 or 4	0102	OFF	ON (×1)	1:1:1/4	10 MHz to 15 MHz	10 MHz to 15 MHz
	0101	OFF	ON (×1)	1:1:1/2	10 MHz to 15 MHz	10 MHz to 15 MHz
	0100	OFF	ON (×1)	1:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01d1	ON (×2)	ON (×1)	2:1:1/2	10 MHz to 15 MHz	10 MHz to 15 MHz
	01d0	ON (×2)	ON (×1)	2:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01d4	ON (×2)	ON (×1)	1:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01d5	ON (×2)	ON (×1)	1:1:1/2	10 MHz to 15 MHz	10 MHz to 15 MHz
	01d6	ON (×2)	ON (×1)	1:1:1/4	10 MHz to 15 MHz	10 MHz to 15 MHz
	01e0	ON (×4)	ON (×1)	4:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01e4	ON (×4)	ON (×1)	2:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01e5	ON (×4)	ON (×1)	2:1:1/2	10 MHz to 15 MHz	10 MHz to 15 MHz
	01e8	ON (×4)	ON (×1)	1:1:1	10 MHz to 15 MHz	10 MHz to 15 MHz
	01e9	ON (×4)	ON (×1)	1:1:1/2	10 MHz to 15 MHz	10 MHz to 15 MHz
	01ea	ON (×4)	ON (×1)	1:1:1/4	10 MHz to 15 MHz	10 MHz to 15 MHz
5	0102	OFF	OFF	1/2:1/2:1/8	2 MHz to 30 MHz	1 MHz to 15 MHz
	0101	OFF	OFF	1/2:1/2:1/4	2 MHz to 30 MHz	1 MHz to 15 MHz
	0100	OFF	OFF	1/2:1/2:1/2	2 MHz to 30 MHz	1 MHz to 15 MHz
	01d1	ON (×2)	OFF	1:1/2:1/4	20 MHz to 30 MHz	10 MHz to 15 MHz
	01d0	ON (×2)	OFF	1:1/2:1/2	20 MHz to 30 MHz	10 MHz to 15 MHz
	01d4	ON (×2)	OFF	1/2:1/2:1/2	20 MHz to 30 MHz	10 MHz to 15 MHz
	01d5	ON (×2)	OFF	1/2:1/2:1/4	20 MHz to 30 MHz	10 MHz to 15 MHz
	01d6	ON (×2)	OFF	1/2:1/2:1/8	20 MHz to 30 MHz	10 MHz to 15 MHz
	01e0	ON (×4)	OFF	2:1/2:1/2	20 MHz to 30 MHz	10 MHz to 15 MHz
	01e4	ON (×4)	OFF	1:1/2:1/2	20 MHz to 30 MHz	10 MHz to 15 MHz
	01e5	ON (×4)	OFF	1:1/2:1/4	20 MHz to 30 MHz	10 MHz to 15 MHz
	01e8	ON (×4)	OFF	1/2:1/2:1/2	20 MHz to 30 MHz	10 MHz to 15 MHz
	01e9	ON (×4)	OFF	1/2:1/2:1/4	20 MHz to 30 Mhz	10 MHz to 15 MHz
	01ea	ON (×4)	OFF	1/2:1/2:1/8	20 MHz to 30 MHz	10 MHz to 15 MHz

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

Mode	FRQCR	PLL1	PLL2	Clock Ratio* (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
6	0102	OFF	OFF	1:1:1/4	14.7456 MHz	14.7456 MHz
	0101	OFF	OFF	1:1:1/2	14.7456 MHz	14.7456 MHz
	0100	OFF	OFF	1:1:1	14.7456 MHz	14.7456 MHz
	01d1	ON (×2)	OFF	2:1:1/2	14.7456 MHz	14.7456 MHz
	01d0	ON (×2)	OFF	2:1:1	14.7456 MHz	14.7456 MHz
	01d4	ON (×2)	OFF	1:1:1	14.7456 MHz	14.7456 MHz
	01d5	ON (×2)	OFF	1:1:1/2	14.7456 MHz	14.7456 MHz
	01d6	ON (×2)	OFF	1:1:1/4	14.7456 MHz	14.7456 MHz
	01e0	ON (×4)	OFF	4:1:1	14.7456 MHz	14.7456 MHz
	01e4	ON (×4)	OFF	2:1:1	14.7456 MHz	14.7456 MHz
	01e5	ON (×4)	OFF	2:1:1/2	14.7456 MHz	14.7456 MHz
	01e8	ON (×4)	OFF	1:1:1	14.7456 MHz	14.7456 MHz
	01e9	ON (×4)	OFF	1:1:1/2	14.7456 MHz	14.7456 MHz
	01ea	ON (×4)	OFF	1:1:1/4	14.7456 MHz	14.7456 MHz
7	0102	OFF	OFF	1:1:1/4	11.0756 MHz	11.0756 MHz
	0101	OFF	OFF	1:1:1/2	11.0756 MHz	11.0756 MHz
	0100	OFF	OFF	1:1:1	11.0756 MHz	11.0756 MHz
	01d1	ON (×2)	OFF	2:1:1/2	11.0756 MHz	11.0756 MHz
	01d0	ON (×2)	OFF	2:1:1	11.0756 MHz	11.0756 MHz
	01d4	ON (×2)	OFF	1:1:1	11.0756 MHz	11.0756 MHz
	01d5	ON (×2)	OFF	1:1:1/2	11.0756 MHz	11.0756 MHz
	01d6	ON (×2)	OFF	1:1:1/4	11.0756 MHz	11.0756 MHz
	01e0	ON (×4)	OFF	4:1:1	11.0756 MHz	11.0756 MHz
	01e4	ON (×4)	OFF	2:1:1	11.0756 MHz	11.0756 MHz
	01e5	ON (×4)	OFF	2:1:1/2	11.0756 MHz	11.0756 MHz
	01e8	ON (×4)	OFF	1:1:1	11.0756 MHz	11.0756 MHz
	01e9	ON (×4)	OFF	1:1:1/2	11.0756 MHz	11.0756 MHz
	01ea	ON (×4)	OFF	1:1:1/4	11.0756 MHz	11.0756 MHz

Note: * Input clock is 1.

Notes: 1. When clock operating modes 3–7 are used:
- The on/off 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 entering 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.
- The 32-k PLL circuit when PLL circuit 1 is off and the 32-k PLL circuit is on.

3. The input of divider 2 becomes the output of:
- PLL circuit 1 when the clock operating mode is 0–2.
- PLL circuit 2 when the clock operating mode is 3 or 4.
- Divider 3 when the clock operating mode is 5.
- The 32-k PLL circuit when the clock operating mode is 6 or 7.

4. The frequency of the internal clock (I ϕ) 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 the 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 (P ϕ) 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.
- The product of the frequency of the CKIO pin and the division ratio of divider 2 when the clock operating mode is 3–7.
- The peripheral clock frequency (P ϕ) should not be set higher than the frequency of the CKIO pin. Do not set the peripheral clock frequency higher than 30 MHz. Set the peripheral clock higher than or equal to 1/4 of the internal clock (I ϕ).

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 60 MHz.
7. $\times 1$, $\times 2$, or $\times 4$ can be used as the multiplication ratio of PLL circuit 1. $\times 1$, $\times 1/2$, or $\times 1/4$ can be selected as the division ratio of dividers 1 and 2. Set the rate in the frequency control register.

9.4 Register Descriptions

9.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit readable/writable register used to specify whether the clock output is produced from the CKIO pin, on/off control of PLL circuit 1, PLL standby, the frequency multiplying 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 FRQCR. The FRQCR register is initialized to H'0102 by a power-on reset. This register holds its values in a manual reset and in standby mode.

FRQCR:

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

Bit 9—Division Ratio of External Bus Clock in Sleep Mode (SLPFRQ): SLPFRQ specifies the division ratio of the external bus clock in sleep mode in clock modes 3–7. The clock frequency output from the CKIO pin is not changed. Since the flash memory cycle is temporarily retained by the bus state controller, RTSCR register and other settings need not be changed.

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

Bit 8—Clock Enable (CKOEN): CKOEN is 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 SH7707 will operate internally at the frequency before the level was fixed. In case of clock operating mode 2, the CKIO pin becomes an input pin irrespective of the value of this bit.

Bit 8: CKOEN	Description	
0	CKIO pin level is fixed	
1	Clock is output from the CKIO pin	(Initial value)

Bit 7—PLL Circuit Enable (PLLEN): PLLEN specifies the on/off status of PLL circuit 1. This bit becomes valid in clock operating modes 3–7. PLL circuit 1 is turned on when the clock operating mode is 0–2 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): PSTBY specifies PLL standby. When PLL standby is active, PLL circuit 1 will be in standby at the frequency specified by the STC bit. This function is valid in clock operating modes 3–7.

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

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

Bit 5: STC1	Bit 4: STC0	Description	
0	0	× 1	(Initial value)
0	1	× 2	
1	0	× 4	
1	1	Setting prohibited (Do not set)	

Note: Do not set the output frequency of PLL circuit 1 higher than 60 MHz.

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

Bit 3: IFC1	Bit 2: IFC0	Description	
0	0	× 1	(Initial value)
0	1	× 1/2	
1	0	× 1/4	
1	1	Setting prohibited (Do not set)	

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

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

Bit 1: PFC1	Bit 0: PFC0	Description	
0	0	× 1	
0	1	× 1/2	
1	0	× 1/4	(Initial value)
1	1	Setting prohibited (Do not set)	

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

9.5 Changing the Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication ratio of PLL circuit 1 or by changing the division ratios of dividers 1 and 2. All of these are controlled by software through the frequency control register. The methods are described below. In modes 3–7, 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 Ratio

The PLL settling time is required to change the multiplication ratio of PLL circuit 1. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication ratio of PLL circuit 1 is 1.
2. Set a value for 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: WDT count clock division ratio
   WTCNT counter: Initial counter value
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in the IFC1–IFC0 bits and PFC1–PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. In clock modes 0–2 , the internal and peripheral clocks both stop. In clock modes 3–7, 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 on 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 ratio is changed simultaneously.

1. In the initial state, IFC1–IFC0 = 00 and PFC1–PFC0 = 10.
2. Set the IFC1, IFC0, PFC1, and PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication ratio of PLL circuit 1. Note that the processor will not operate normally if the wrong value is set.
3. The clock is immediately supplied using the new division ratio.

9.6 PLL Standby Function

9.6.1 Overview of the PLL Standby Function

When operating in clock modes 3–7, the internal clock can be controlled by turning the PLL circuit 1 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 PLL circuit 1, 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 ratio of PLL circuit 1 is set in the STC1–STC0 bits and PSTBY is set to 1, PLL circuit 1 begins oscillating at the specified multiplication ratio. 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 settled, 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 IFC1–IFC0 and PFC1–PFC0. For several cycles before and after the clock switches, the internal clock will be stopped, but the peripheral clock and CKIO 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 or divider 3. At this time, the division ratio can be changed by changing the settings of IFC1–IFC0 and PFC1–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 of 1.

Notes: 1. There are some restrictions on the PLL standby state (PSTBY = 1, PLLEN = 0) as follows: In principle, the settings of the frequency control register's CKOEN, STC1–STC0, IFC1–IFC0, and PFC1–PFC0 bits 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. Software determines the oscillation settling time. When PLLEN is set to 1 before oscillation has settled, malfunctions may be caused by an unstable clock.

3. In clock modes 3–7, the SH7707 cannot enter standby mode while PLL circuit 1 is turned on. Always clear PSTBY and PLLEN to 0 to stop PLL circuit 1 before entering 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 for information on setting the WDT.

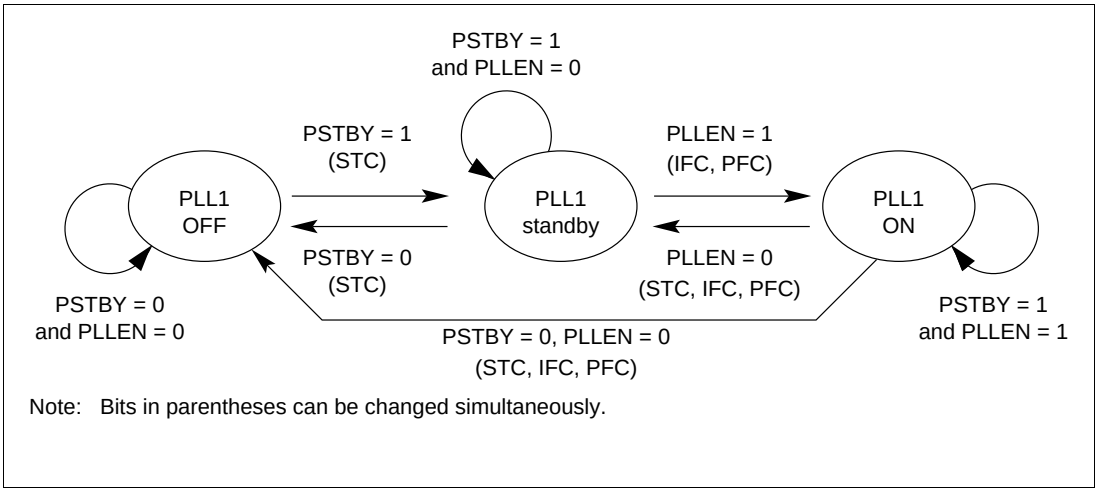


Figure 9.2 State Transitions for the PLL Standby Function

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. The use of the CKOEN bit depends on the clock mode.

9.7.1 Clock Modes 0–1

The CKIO pin level cannot be fixed. Always set the CKOEN bit in FRQCR to 1 (clock output).

9.7.2 Clock Modes 3–7

The CKIO output changes as soon as the CKOEN bit is changed. When the WDT is started by simultaneously changing the multiplication ratio of PLL circuit 1 or switching PLL circuit 1 on or off, the WDT starts running after the CKIO output switches and the internal clock then changes.

9.8 Overview of WDT

9.8.1 Block Diagram of WDT

Figure 9.3 shows a block diagram of the WDT.

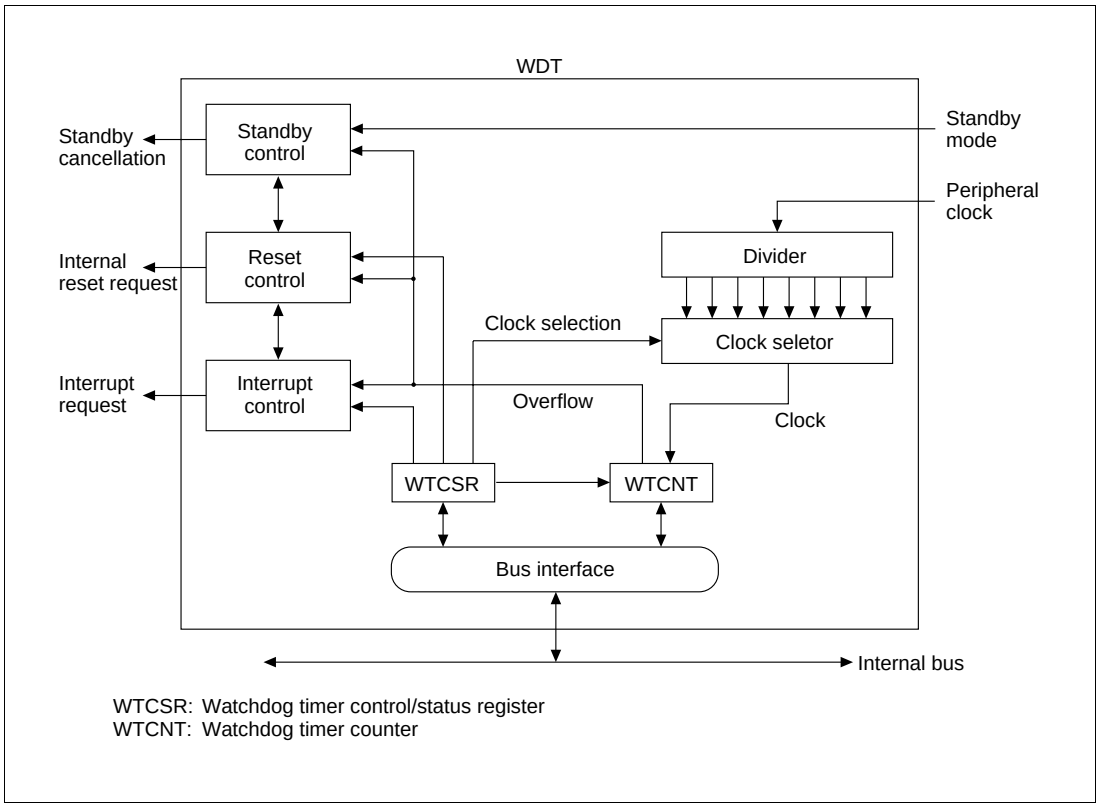


Figure 9.3 Block Diagram of WDT

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 details of these registers.

Table 9.5 Register Configuration

Name	Abbreviation	R/W	Size	Initial Value	Address
Watchdog timer counter	WTCNT	R/W*	R: 8, W: 16*	H'00	H'FFFFFF84
Watchdog timer control/status register	WTCSR	R/W*	R: 8, W: 16*	H'00	H'FFFFFF86

Note: * Write using word access. Write H'5A or H'A5 in the top byte, respectively. Byte or long-word access. Read using byte access.

9.9 WDT Registers

9.9.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit readable/writable counter that increments on the selected clock. Overflow generates a reset in watchdog timer mode or an interrupt in interval timer mode. The address of WTCNT is H'FFFFFF84. The WTCNT counter is initialized to H'00 only by a power-on reset via the RESET pin. Write to the WTCNT counter using word access, with H'5A as the top byte. Read WTCNT using byte access.

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 readable/writable register composed of bits to select the count clock, bits to select the timer mode, and overflow flags. The address of WTCSR is H'FFFFFF86. The WTCSR register is initialized to H'00 only by a power-on reset via the RESET pin. When a WDT overflow causes an internal reset, WTCSR holds its values. When used to count the clock settling time for canceling a standby mode, it holds its values after counter overflow. Write to WTCSR using word access, with H'A5 as the top byte. Read WTCSR using byte access.

Bit:	7	6	5	4	3	2	1	0
	TME	WT/$\overline{IT}$	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 for clock frequency change.

Bit 7: TME	Description	
0	Timer disabled: Starts counting and WTCNT value is held	(Initial value)
1	Timer enabled	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to use the WDT as a watchdog timer or an interval timer.

Bit 6: WT/$\overline{IT}$	Description	
0	Used as interval timer	(Initial value)
1	Used as watchdog timer	

Note: Modifying the WT/$\overline{IT}$ bit during WDT operation may cause incorrect incrementing.

Bit 5—Reset Select (RSTS): Selects the type of reset when 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	

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 to 0 (CKS2–CKS0): These bits select the clock to be used for the WTCNT count from one of the nine types obtainable by dividing the peripheral clock (Pø). The overflow period in the table is the value when the peripheral clock (Pø) is 15 MHz.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	CKSS	Clock Division Ratio	Overflow Period (for 15 MHz)
0	0	0	0	1	17 µs
		1	0	1/4	68 µs
	1	0	0	1/16	273 µs
		1	0	1/32	546 µs
1	0	0	0	1/64	1.09 ms
		1	0	1/256	4.36 ms
	1	0	0	1/1024	17.46 ms
		1	0	1/4096	69.91 ms
Don’t care	Don’t care	Don’t care	1	1/65536	1.12 s

Note: Modifying the CKS[2:0] bits during WDT operation may cause incorrect incrementing. Be sure to stop the WDT before modifying these bits.

The above ratios are specified only when the CKSS bit in the STBCR3 register is cleared. Otherwise, the ratio is fixed at 1/65536 regardless of the CKS[2:0] bits.

9.9.3 Notes on Register Access

The method of writing to the watchdog timer counter (WTCNT) and the watchdog timer control/status register (WTCSR) differs from that for other registers to prevent them from being easily modified.

A word transfer instruction must be used when writing to WTCNT or WTCSR. A byte transfer instruction or longword transfer instruction cannot be used.

As shown in figure 9.4, when writing to WTCNT, the upper byte must be H'5A, and the lower byte must be the write data. When writing to WTCSTR, the upper byte must be H'A5, and the lower byte must be the write data.

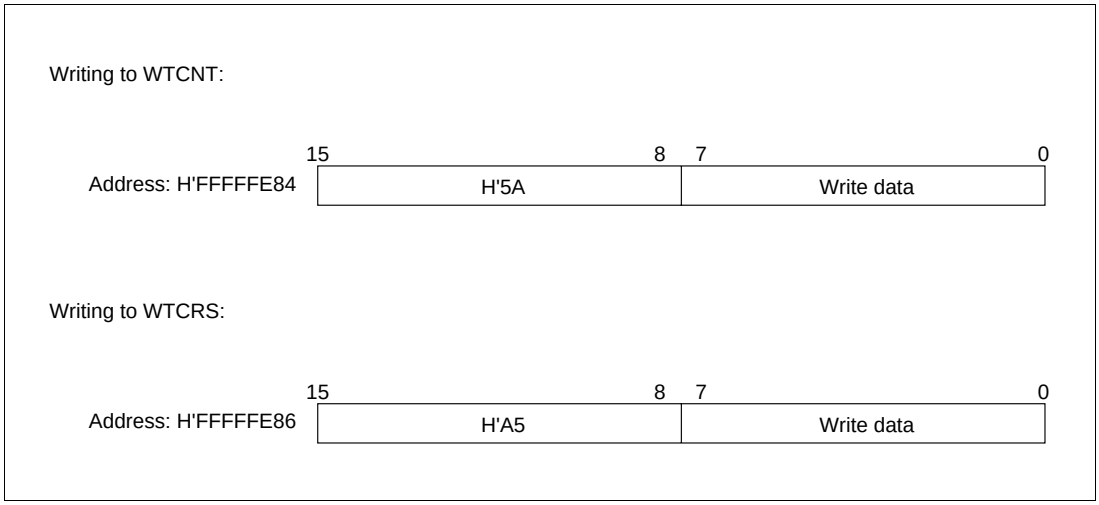


Figure 9.4 Writing to WTCNT and WTCSR

9.10 Using the WDT

9.10.1 Canceling Standby

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 a reset is used for canceling, so keep the RESET pin low until the clock settles.)

1. Before entering 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 to be used in bits CKS2–CKS0 in WTCSR, and the initial counter value in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Enter standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting an edge change in the NMI signal or detecting an interrupt.
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. The WTCNT counter continues to count even after standby mode is canceled. The count is stopped by clearing the STBY bit in the STBCR register to 0. If the WTCNT counter reaches H'80 without being stopped, standby mode will be returned to automatically. Therefore, the STBY bit in the STBCR register must be cleared by the interrupt handler.

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 bits CKS2–CKS0 in WTCSR, and the initial value 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 a value of H'00-H'01. The stop value depends on the clock ratio.

9.10.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{IT}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, set the type of count clock in bits CKS2–CKS0, and set the initial counter value 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

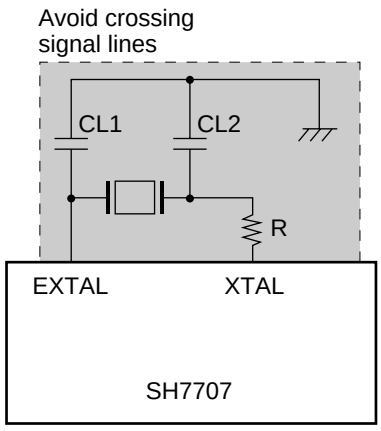
When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at fixed intervals.

1. Clear the WT/$\overline{IT}$ bit in the WTCSR register to 0, set the type of count clock in bits CKS2–CKS0, and set the initial counter value 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 the INTC. The counter then resumes counting.

9.11 Notes on Board Design

9.11.1 When Using a Crystal Oscillator

The crystal oscillator and capacitors must be placed as close to the EXTAL and XTAL pins as possible. Do not cross other signal lines with the lines of these pins since the oscillation may be incorrect due to induction.



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

Figure 9.5 Precautions when Using a Crystal Oscillator

9.11.2 When Using the PLL

The capacitors for stabilizing oscillators C1 and C2 must be placed close to the CAP1 and CAP2 pins, respectively. Oscillation stabilization capacitors C1 and C2 must be grounded to VSS (PLL1) and VSS (PLL2), respectively. Vcc (PLL) and the Vss (PLL) must be powered separately from other Vcc and Vss pins, and bypass capacitors CPB and CB must be inserted near the pin. In clock mode 2, the EXTAL pin must be connected to Vcc or Vss, and the XTAL pin must be left open.

Table 9.6 Capacitor Values for Reference

Capacitor Value	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
C1 = 470 pF	o	o	o	o	o	o	o	o
C2 = 470 pF	o	o	x	o	o	x	x	x

Note: o: Required, x: Not required

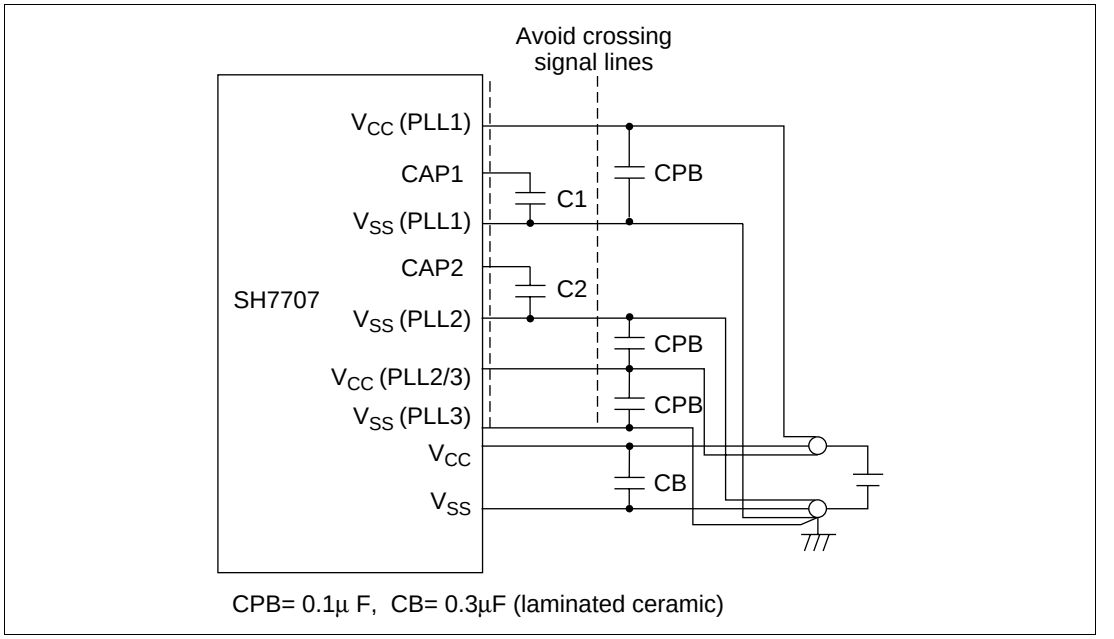


Figure 9.6 Precaution when Using the PLL

Section 10 Bus State Controllers (BSC, BSCP)

10.1 Overview

The SH7707 has two bus state controllers (hereafter referred to as BSC and BSCP) for the CPU/cache memory and for the DMAC/LCDC, respectively. 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 the SH7707 to be connected directly to DRAM, SRAM, ROM, and other memory storage devices without external circuitry. 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 of 64 Mbytes for each of the six areas, 0 and 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 by 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 consecutive memory accesses to different areas or writes directly following reads in 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
- ROM burst interface
  - Insertion of wait states controllable by software
  - Register setting control of burst transfers

- PCMCIA direct-connection interface
  - — Insertion of wait states controllable by software
  - — Bus sizing function for I/O bus width
- Output of clock signals to the BSC is disabled in sleep mode to reduce power consumption

The BSCP has the following features:

- Physical address space is divided into six areas
  - — A maximum of 64 Mbytes for each of the six areas, 0 and 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 by 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 consecutive memory accesses to different areas or writes directly following reads in the same area
- Direct interface to DRAM
  - — Multiplexes row/column addresses according to DRAM capacity
  - — Supports burst operation (high-speed page mode, EDO mode)
  - — Supports short-pitch access operation
  - — Supports CAS-before-RAS refresh and self-refresh
  - — Controls timing of DRAM direct-connection control signals according to register settings
- ROM burst interface
  - — Insertion of wait states controllable by software
  - — Register setting control of burst transfers
- PCMCIA direct-connection interface
  - — Insertion of wait states controllable by software
  - — Bus sizing function for I/O bus width
- Refresh function
  - — Refresh cycles are automatically maintained in 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 a self-refresh operation using low-power-consumption DRAM

- The refresh counter can be used as an interval timer
  - — Outputs an interrupt request signal using the compare-match function
  - — Outputs an interrupt request signal when the refresh counter overflows
- Automatically disables the output of clock signals other than to the refresh counter, except during execution of external bus cycles

10.1.2 Block Diagram

Figures 10.1 (a) and (b) show functional block diagrams of the bus state controllers.

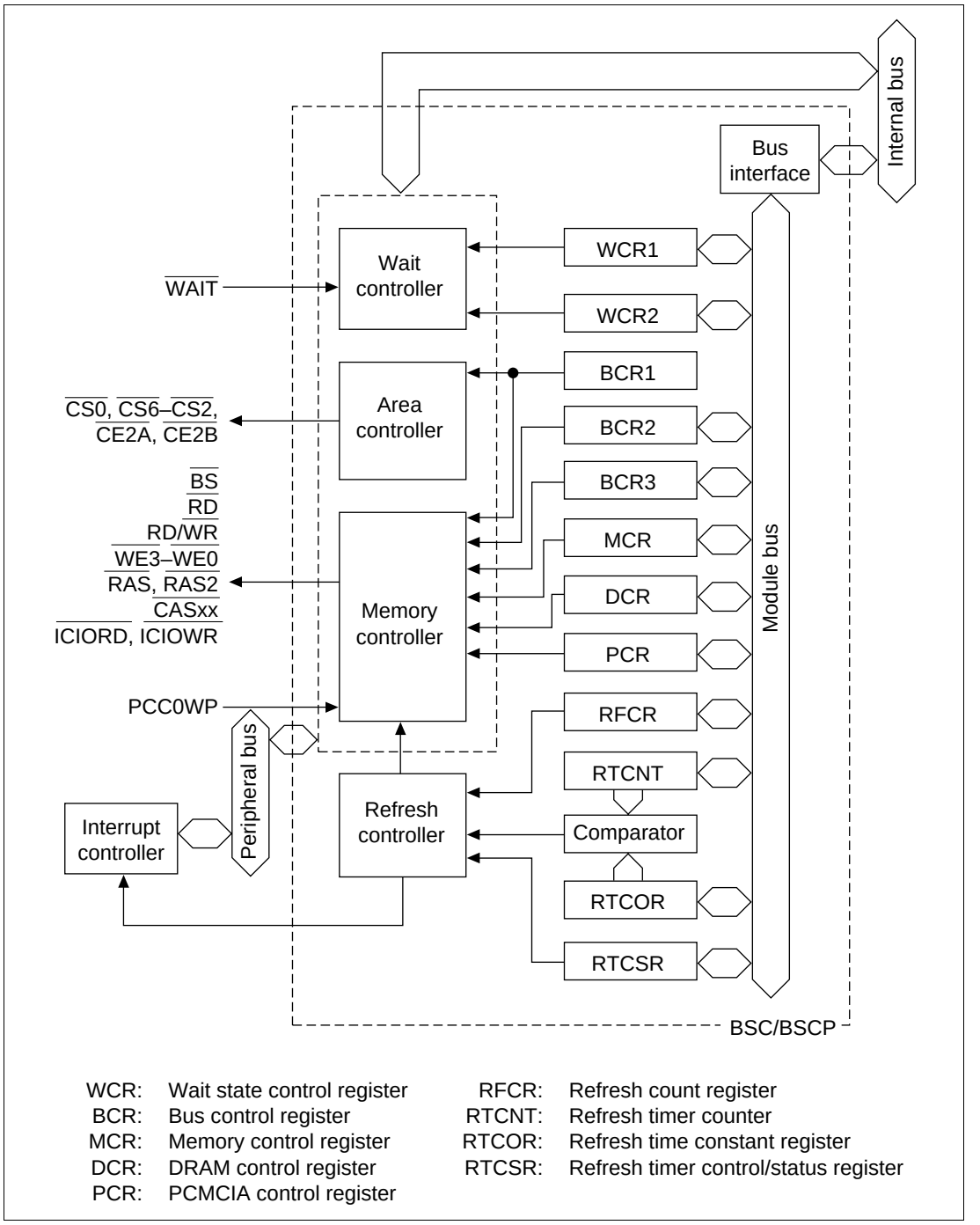


Figure 10.1 BSC/BSCP Functional Block Diagram (a)

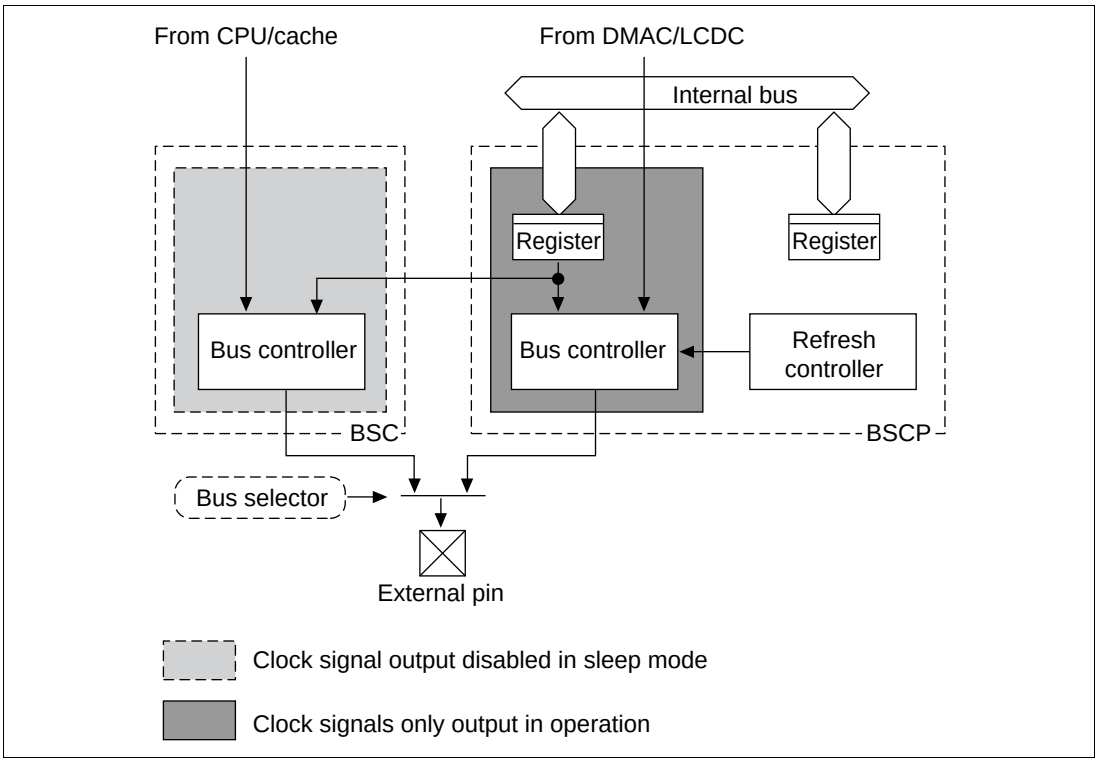


Figure 10.1 BSC/BSCP Functional Block Diagram (b)

10.1.3 Pin Configuration

Table 10.1 lists the BSC/BSCP pin configuration.

Table 10.1 Pin Configuration (Preliminary)

Pin Name	Signal	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 is used, data I/O
Bus cycle start	$\overline{BS}$	O	Indicates start of bus cycle. During burst transfers, asserted every data cycle.
Chip select 0, 2–4	$\overline{CS0}$, $\overline{CS2}$–$\overline{CS4}$	O	Chip select signals to indicate area being accessed.
Chip select 5, 6	$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	O	Chip select signals to indicate area being accessed. $\overline{CS5}$/$\overline{CE1A}$ and $\overline{CS6}$/$\overline{CE1B}$ can also be used as $\overline{CE1A}$ and $\overline{CE1B}$ of PCMCIA.
PCMCIA card select	$\overline{CE2A}$, $\overline{CE2B}$	O	Card enable signals for PCMCIA
Read/write	RD/$\overline{WR}$	O	Data bus direction indicator signal. DRAM/PCMCIA write indicator signal.
Row address strobe	$\overline{RAS}$	O	When DRAM is used, $\overline{RAS}$ signal for area 3.
Row address strobe	$\overline{RAS2}$	O	When DRAM is used in area 2, $\overline{RAS2}$ signal for area 2.
Column address strobe	$\overline{CASLL}$	O	When DRAM is used, $\overline{CASLL}$ signal for D7–D0.
Column address strobe LH	$\overline{CASLH}$	O	When DRAM is used, $\overline{CASLH}$ signal for D15–D8
Column address strobe HL	$\overline{CASHL}$/$\overline{CAS2L}$	O	When DRAM is used, $\overline{CASHL}$ signal for D23–D16. When area 2 DRAM is being used, $\overline{CAS2L}$ signal for D7–D0.
Column address strobe HH	$\overline{CASHH}$/$\overline{CAS2H}$	O	When DRAM is used, $\overline{CASHH}$ signal for D31–D24. When area 2 DRAM is being used, $\overline{CAS2H}$ signal for D15–D8.

Table 10.1 Pin Configuration (Preliminary) (cont)

Pin Name	Signal	I/O	Description
Data enable 0	$\overline{WE0}$	O	When memory is used, selects D7–D0 write strobe signal.
Data enable 1	$\overline{WE1}$/$\overline{WE}$	O	When memory and PCMCIA are used, selects D15–D8 write strobe signal. When PCMCIA is used, strobe signal that indicates the write cycle.
Data enable 2	$\overline{WE2}$/$\overline{ICIORD}$	O	When memory and PCMCIA are used, selects D23–D16 write strobe signal. When PCMCIA is used, strobe signal indicating I/O read.
Data enable 3	$\overline{WE3}$/$\overline{ICIOWR}$	O	When memory and PCMCIA are used, selects D31–D24 write strobe signal. When PCMCIA is used, strobe signal indicating I/O write.
Read	$\overline{RD}$	O	Strobe signal indicating read cycle
Wait	$\overline{WAIT}$	I	Wait state request signal
Bus release request	$\overline{BREQ}$	I	Bus release request signal
Bus release acknowledge	$\overline{BACK}$	O	Bus release acknowledge signal

10.1.4 Register Configuration

The BSC and BSCP have 12 registers (table 10.2). These registers control direct connection interfaces to memory, wait states, refreshes, and PCMCIA devices.

Table 10.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Bus Width
Bus control register 1	BCR1	R/W	*	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

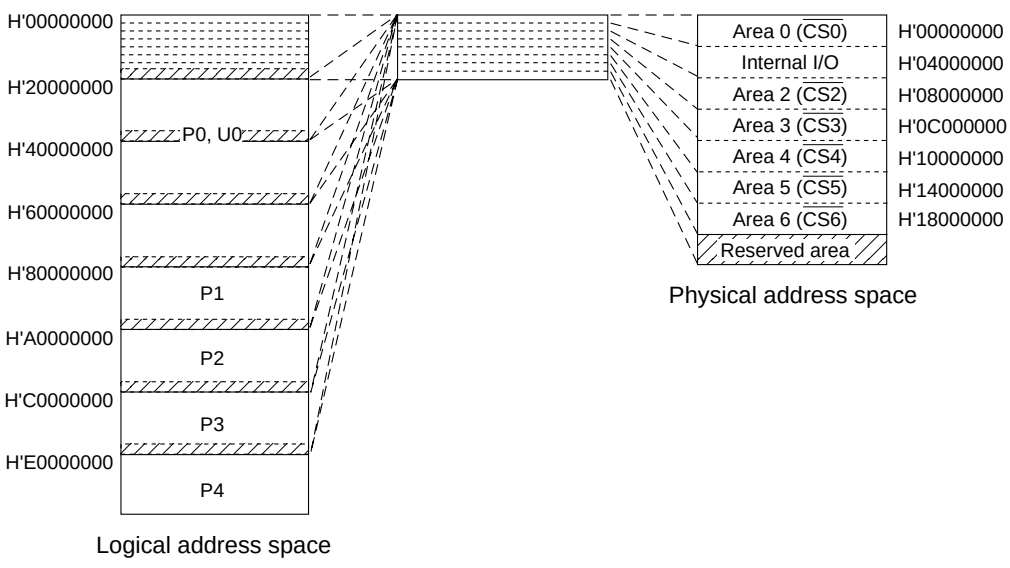
Note: The initial value depends on the level of the MD5 pin that sets the endian mode: H'0000 when low (big-endian mode), H'0800 when high (little-endian mode).

10.1.5 Area Overview

Space Allocation: In the SH7707 architecture, logical spaces have 32-bit address spaces and physical spaces have 29-bit address spaces. The logical space is divided into five areas according to 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 10.3, the SH7707 can be connected directly to seven memory/PCMCIA interface areas, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$–$\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$) for each of these areas. $\overline{CS0}$ is asserted during area 0 access; $\overline{CS6}$ is asserted during area 6 access. When DRAM is connected to area 2 or 3, signals such as $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, are also asserted. When the PCMCIA interface is selected in area 5 or 6, in addition to $\overline{CS5}$/$\overline{CS6}$, $\overline{CE2A}$/$\overline{CE2B}$ is asserted for the corresponding byte accessed.



Note: For logical address spaces P0 and P3, when the address conversion 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 Correspondence between 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, burst ROM	64 Mbytes	8, 16, 32*1
	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	64 Mbytes	8, 16, 32*6
	H'04000000 + H'20000000 × n to H'07FFFFFF + H'20000000 × n		Shadow	n: 1–6
2	H'08000000 to H'0BFFFFFF	Ordinary memory, DRAM	64 Mbytes	8, 16, 32*2, *3
	H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n		Shadow	n: 1–6
3	H'0C000000 to H'0FFFFFFF	Ordinary memory, DRAM	64 Mbytes	8, 16, 32*2, *4
	H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n		Shadow	n: 1–6
4	H'10000000 to H'13FFFFFF	Ordinary memory	64 Mbytes	8, 16, 32*2
	H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n		Shadow	n: 1–6
5*5	H'14000000 to H'15FFFFFF	Ordinary memory, PCMCIA, burst ROM	32 Mbytes	8, 16, 32 *2, *5
	H'16000000 to H'17FFFFFF	Ordinary memory, 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, PCMCIA, burst ROM	32 Mbytes	8, 16, 32 *2, *5
	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		n: 0–7 *7

Notes:
1. Use external pin to specify memory bus width.
2. Use register to specify memory bus width.
3. With DRAM interfaces, bus width must be 16.
4. With DRAM interfaces, bus width must be 16 or 32.
5. With PCMCIA interface, bus width must be 8 or 16.
6. Depending on register size.
7. Do not access the reserved area. If the reserved area is accessed, correct operation cannot be guaranteed.

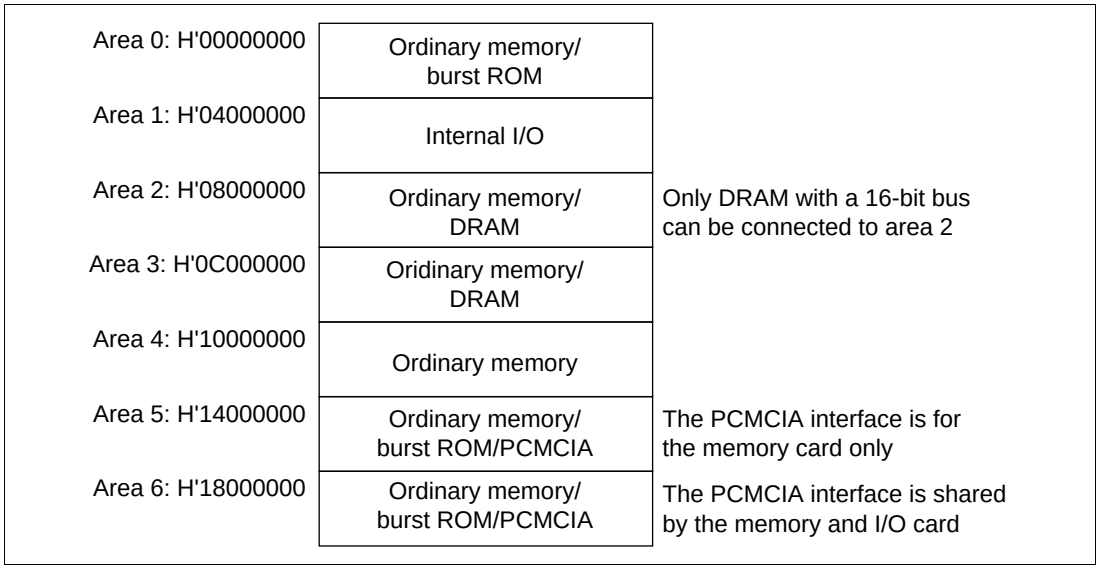


Figure 10.3 Physical Space Allocation

Memory Size: The memory size in the SH7707 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) in a power-on reset. The correspondence between the external pins (MD4 and MD3) and memory size is shown in the table below.

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 is used.

When area 2 is used as a DRAM area, set the bus width of areas 2 and 3 to word.

- When using the DRAM interface, select a bus width of 16 or 32 bits.
- When using area 2 as a DRAM area, set a bus width of 16 bits for areas 2 and 3.
- For the DRAM interface bus width setting, ensure that the bus control register 2 (BCR2) and individual memory control register (MCR) set values match.

When areas 5 and 6 are used as PCMCIA interfaces, set the bus width to byte or word. When using the port A or port B function, set each of the bus widths to byte or word for all areas. For more information, see section 19.3.1, Port A Control Register (PACR), section 19.3.2, Port B Control Register (PBCR), section 20.2, Port A, and section 20.3, Port B.

Shadow Space: Areas 0 and 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 range of area 7 addresses is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n–H'1FFFFFFF + H'20000000 × n (n = 0–7) corresponding to the area 7 shadow space is reserved, and must not be used.

10.1.6 PCMCIA Support

The SH7707 supports PCMCIA standard interface specifications in physical space areas 5 and 6.

The interface supported by the BSC, BSCP, and PCC (PC card controller) is basically the IC memory card interface and I/O card interface defined by PCMCIA Rev. 2.1.

Physical space area 5 supports the IC memory card interface only; area 6 supports both the IC memory card interface and the I/O card interface.

Table 10.4 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 for I/O bus width* The PCMCIA interface can be accessed from the address-conversion area and non-address conversion area.

Note: Dynamic bus sizing for the I/O bus width is supported only in little-endian mode.

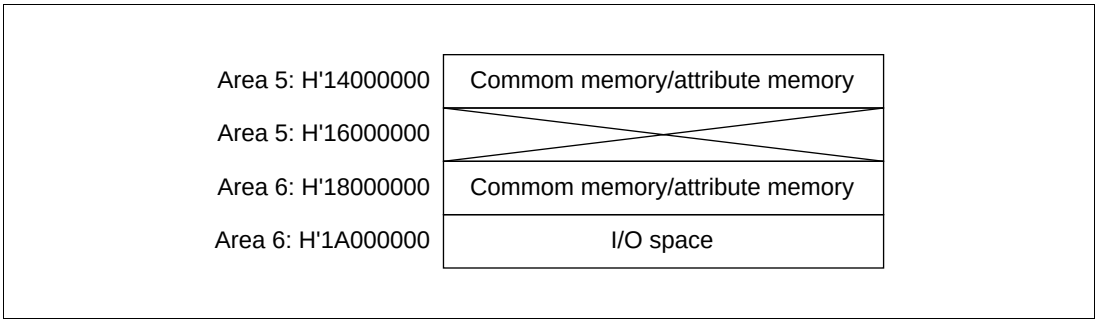


Figure 10.4 PCMCIA Space Allocation

Table 10.5 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	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	–CE1	I	Card enable	–CE1	I	Card enable	$\overline{\text{CE1A}}$ or $\overline{\text{CE1B}}$
8	A10	I	Address	A10	I	Address	A10
9	–OE	I	Output enable	–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	–WE/–PGM	I	Write enable	–WE/–PGM	I	Write enable	$\overline{\text{WE}}$
16	+RDY/–BSY	O	Ready/busy	–IREQ	O	Interrupt request	PCC0READY or PCC1READY
17	VCC		Power supply	VCC		Power supply	—
18	VPP1		Programming power supply	VPP1		Programming and peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3

Table 10.5 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Pin
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	–IOIS16	O	16-bit I/O port	PCC0WP, PCC1WP
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	–CD1	O	Card detection	–CD1	O	Card detection	$\overline{\text{PCC0CD1}}$ or $\overline{\text{PCC1CD1}}$
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	–CE2	I	Card enable	–CE2	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	–VS1	I	Voltage sense	–VS1	I	Voltage sense	$\overline{\text{PCC0VS1}}$ or $\overline{\text{PCC1VS1}}$
44	RFU		Reserved	–IORD	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	–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
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—
52	VPP2		Programming power supply	VPP2		Programming and peripheral power supply	—

Table 10.5 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	**Signal Name**	**I/O**	**Function**	**Signal Name**	**I/O**	**Function**	**Pin**
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	–VS2	I	Voltage sense	–VS2	I	Voltage sense	$\overline{\text{PCC0VS2}}$ or $\overline{\text{PCC1VS2}}$
58	+RESET	I	Reset	+RESET	I	Reset	PCC0RESET or PCC1RESET
59	–WAIT	O	Wait request	–WAIT	O	Wait request	$\overline{\text{PCC0WAIT}}$ or $\overline{\text{PCC1WAIT}}$
60	RFU		Reserved	–INPACK	O	Input acknowledge	—
61	–REG	I	Attribute memory space select	–REG	I	Attribute memory space select	$\overline{\text{PCCREG}}$
62	BVD2	O	Battery voltage detection	–SPKR	O	Digital sound signal	PCC0BVD2 or PCC1BVD2
63	BVD1	O	Battery voltage detection	–STSCHG	O	Card status change	PCC0BVD1 or PCC1BVD1
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	–CD2	O	Card detection	–CD2	O	Card detection	$\overline{\text{PCC0CD2}}$ or $\overline{\text{PCC1CD2}}$
68	GND		Ground	GND		Ground	—

10.2 BSC and BSCP Registers

10.2.1 Bus Control Register 1 (BCR1)

BCR1 can be used for the BSC and BSCP. Bus control register 1 (BCR1) is a 16-bit readable/writable 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 in 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:				HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
Initial value:	—	—	—	—	0/1*	0	0	0
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 (MD5) designating endian upon a power-on reset.

Bits 15 to 13—Reserved: Read-only bits, always read as 0.

Bit 12—High-Z Control (HIZCNT): Specifies the status of the RAS and the CAS signals in standby and on bus release.

Bit 12: HIZCNT	Description
0	The RAS and the CAS signals are high-impedance status (High-Z) at standby and on bus release (Initial value)
1	The RAS and the CAS signals are driven in standby and on bus release

Bit 11—Endian Flag (ENDIAN): Samples the value of the external pin (MD5) 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 SH7707 is set as big-endian
1	(On reset) Endian setting external pin (MD5) is high. Indicates the SH7707 is set as little-endian

Bits 10 and 9—Area 0 Burst ROM Control (A0BST1, A0BST0): These bits specify whether to use burst ROM in physical space area 0. When burst ROM is used, they set the number of burst transfers.

Bit 10: A0BST1	Bit 9: A0BST0	Description
0	0	Area 0 accessed as ordinary memory (Initial value)
	1	Area 0 accessed as burst ROM (4 consecutive accesses). Can be used when bus width is 8, 16, or 32
1	0	Area 0 accessed as burst ROM (8 consecutive accesses). Can be used when bus width is 8 or 16
	1	Area 0 accessed as burst ROM (16 consecutive accesses). Can be used only when bus width is 8

Bits 8 and 7—Area 5 Burst Enable (A5BST1, A5BST0): These bits specify whether to use burst ROM and PCMCIA burst mode in physical space area 5. When burst ROM and PCMCIA burst mode are used, they set the number of burst transfers.

Bit 8: A5BST1	Bit 7: A5BST0	Description
0	0	Area 5 accessed as ordinary memory (Initial value)
	1	Burst access to area 5 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32
1	0	Burst access to area 5 (8 consecutive accesses). Can be used when bus width is 8 or 16
	1	Burst access to area 5 (16 consecutive accesses). Can be used only when bus width is 8

Bits 6 and 5—Area 6 Burst Enable (A6BST1–A6BST0): These bits specify whether to use burst ROM and PCMCIA burst mode in physical space area 6. When burst ROM and PCMCIA burst mode are used, they set the number of burst transfers.

Bit 6: A6BST1	Bit 5: A6BST0	Description
0	0	Area 6 accessed as ordinary memory (Initial value)
	1	Burst access to area 6 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32
1	0	Burst access to area 6 (8 consecutive accesses). Can be used when bus width is 8 or 16
	1	Burst access to area 6 (16 consecutive accesses). Can be used only when bus width is 8

Bits 4 to 2—Area 2, Area 3 Memory Type (DRAMTP2, DRAMTP1, DRAMTP0): These bits 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.

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Areas 2 and 3 are normal memory
		1	Reserved (Do not set)
	1	0	Reserved (Do not set)
		1	Reserved (Do not set)
1	0	0	Area 2: normal memory; area 3: DRAM
		1	Areas 2 and 3 are DRAM *
	1	0	Reserved (Do not set)
		1	Reserved (Do not set)

Note: When selecting these bits, set the area 2 and 3 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	Physical space area 5 accessed as normal memory
1	Physical space area 5 accessed 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	Physical space area 6 accessed as normal memory
1	Physical space area 6 accessed as PCMCIA space

10.2.2 Bus Control Register 2 (BCR2)

BCR2 can be used for the BSC and BSCP. Bus control register 2 (BCR2) is a 16-bit readable/writable 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 in 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, 2, 1, and 0—Reserved: Read-only bits, always read as 0.

Bits 2n + 1, 2n—Area n (2–6) Bus Size Specification (AnSZ1, AnSZ0): These bits specify the bus size of physical space area n (n = 2 to 6).

Bit 2n + 1: AnSZ1	Bit 2n: AnSZ0	Description
0	0	Reserved (Do not set)
	1	Byte (8-bit) size
1	0	Word (16-bit) size
	1	Longword (32-bit) size

For the DRAM interface bus width setting, ensure that the bus control register 2 (BCR2) and individual memory control register (MCR) set values match.

10.2.3 Bus Control Register 3 (BCR3)

Bus control register 3 (BCR3) is a 16-bit readable/writable 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. BCR3 is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or in standby mode. Bits EXTEND, TPC31-30, RCD31-30, TRAS31-30, TPC21-20, RCD21-20, and TRAS21-20 are written to during initialization after a power-on reset and are not modified again.

Bit:	15	14	13	12	11	10	9	8
Bit name:	EXTEND		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 Access Mode Extended Control (EXTEND): Specifies the short pitch access mode to the DRAM (areas 2 and 3) for the BSCP (LCDC/DMAC bus state controller). The EXTEND bit does not affect the BSC (CPU/cache bus state 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 subsequent data. In addition, by setting the EXTEND bit to 1, other bits of the BCR3 register (TPC3(1-0), RCD3(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 BCR3 is set to 1, this TPC setting is used. 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	Function
TPC31	TPC30	Normal
0	0	1 cycle (Initial value)
0	1	2 cycles
1	0	(Reserved)
1	1	(Reserved)

When the same area is accessed immediately after a self-refresh, a time equivalent to the TPC value must be inserted under user control.

Bits 11 and 10—RAS-CAS Delay (RCD31, RCD30): When the EXTEND bit in BCR3 is set to 1, the BCSP uses the RCD3 (1-0) bits in place of the RCD (1-0) bits in 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 BCR3 is set to 1, this TRAS3(1-0) setting is used. 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 cycles	(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 BCR3 is set to 1, this TPC2(1-0) setting is used. 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.

Set the same value in BCR3.TPC3(1-0), TPC2(1-0), MCR.TPC(1-0), and DCR.TPC(1-0).

Bit 5	Bit 4	Function	
TPC21	**TPC20**	**Normal**	
0	0	1 cycle	(Initial value)
0	1	2 cycles	
1	0	(Reserved)	
1	1	(Reserved)	

When the same area is accessed immediately after a self-refresh, a time equivalent to the TPC value must be inserted under user control.

Bits 3 and 2—RAS-CAS Delay (RCD21, RCD20): When the EXTEND bit in BCR3 is set to 1, the BSCP uses the RCD2 (1-0) bits in place of the RCD (1-0) bits in DCR. The RCD bits set the RAS-CAS assert delay time 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 BCR3 is set to 1, this TRAS2(1-0) setting is used. When the DRAM interface is selected as the connected memory of 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 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 as 0. 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 readable/writable 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 SH7707 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 in 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: Read-only bits, always read as 0.

Bits 2n + 1, 2n—Area n (6–2, 0) Intercycle Idle Specification (AnIW1, AnIW0): These bits specify the number of idles inserted between bus cycles when switching between physical space area n (6–2, 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 readable/writable 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 in 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 to 13—Area 6 Wait Control (A6W2, A6W1, A6W0): These bits specify the number of wait states inserted into physical space area 6. They also specify the burst pitch for burst transfer.

			Description			
Bit 15: A6W2	**Bit 14: A6W1**	**Bit 13: A6W0**	**Basic Number of Wait States**	**Basic External Wait**	**Burst Pitch**	**Burst External Wait**
0	0	0	0	Disabled	2	Enabled
		1	1	Enabled	2	Enabled
	1	0	2	Enabled	3	Enabled
		1	3	Enabled	4	Enabled
1	0	0	4	Enabled	4	Enabled
		1	6	Enabled	6	Enabled
	1	0	8	Enabled	8	Enabled
		1	10	Enabled	10	Enabled (Initial value)

Bits 12 to 10—Area 5 Wait Control (A5W2, A5W1, A5W0): These bits specify the number of wait states inserted into physical space area 5. They also specify the burst pitch for burst transfer.

			Description			
Bit 12: A5W2	**Bit 11: A5W1**	**Bit 10: A5W0**	**Basic Number of Wait States**	**Basic External Wait**	**Burst Pitch**	**Burst External Wait**
0	0	0	0	Disabled	2	Enabled
		1	1	Enabled	2	Enabled
	1	0	2	Enabled	3	Enabled
		1	3	Enabled	4	Enabled
1	0	0	4	Enabled	4	Enabled
		1	6	Enabled	6	Enabled
	1	0	8	Enabled	8	Enabled
		1	10	Enabled	10	Enabled (Initial value)

Bits 9 to 7—Area 4 Wait Control (A4W2, A4W1, A4W0): These bits specify the number of wait states inserted into physical space area 4.

			Description	
Bit 9: A4W2	**Bit 8: A4W1**	**Bit 7: A4W0**	**Basic Number of Wait States**	**External Wait Input**
0	0	0	0	Disabled
		1	1	Enabled
	1	0	2	Enabled
		1	3	Enabled
1	0	0	4	Enabled
		1	6	Enabled
	1	0	8	Enabled
		1	10	Enabled (Initial value)

Bits 6 and 5—Area 3 Wait Control (A3W1, A3W0): These bits specify the number of wait states inserted into physical space area 3. When the DRAM is used for area 3 and the EXTEND bit in BCR3 is set, the BSCP ignores bits 6 and 5.

		Normal Memory		
Bit 6: A3W1	**Bit 5: A3W0**	**Basic Number of Wait States**	**External Wait Input**	
0	0	0	Disabled	
	1	1	Enabled	
1	0	2	Enabled	
	1	3	Enabled	(Initial value)

Bit 6: A3W1	**Bit 5: A3W0**	**DRAM: CAS Assert Period**	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

Bits 4 and 3—Area 2 Wait Control (A2W1, A2W0): These bits specify the number of wait states inserted into physical space area 2. When the DRAM is used for area 2 and the EXTEND bit in BCR3 is set, the BSCP ignores bits 4 and 3.

		Normal Memory		
Bit 4: A2W0	**Bit 3: A2W0**	**Basic Number of Wait States**	**External Wait Input**	
0	0	0	Disabled	
	1	1	Enabled	
1	0	2	Enabled	
	1	3	Enabled	(Initial value)

Bit 4: A2W0	**Bit 3: A2W0**	**DRAM: CAS Assert Period**	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

Bits 2 to 0—Area 0 Wait Control (A0W2, A0W1, A0W0): These bits specify the number of wait states inserted into physical space area 0. They also specify the burst pitch for burst transfer.

			Description			
Bit 2: A0W2	**Bit 1: A0W1**	**Bit 0: A0W0**	**Basic Number of Wait States**	**Basic External Wait**	**Burst Pitch**	**Burst External Wait**
0	0	0	0	Disabled	2	Enabled
		1	1	Enabled	2	Enabled
	1	0	2	Enabled	3	Enabled
		1	3	Enabled	4	Enabled
1	0	0	4	Enabled	4	Enabled
		1	6	Enabled	6	Enabled
	1	0	8	Enabled	8	Enabled
		1	10	Enabled	10	Enabled (Initial value)

10.2.6 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit readable/writable register shared between the BSC and BSCP that specifies RAS and CAS timing and burst control for DRAM (area 3 only), specifies address multiplexing, and controls refreshing. This enables direct connection of DRAM, without external circuits.

MCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or in standby mode. Bits TPC1–TPC0, RCD1–RCD0, TRAS1–TRAS0, BE, SZ, AMX1–AMX0, and EDOMODE are written to in 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, 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			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

Bit:	7	6	5	4	3	2	1	0
Bit name:		BE	SZ	AMX1	AMX0	RFSH	RMODE	EDO MODE
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

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.

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)

When the same area is accessed immediately after a self-refresh, a time equivalent to the TPC value must be inserted under user control.

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 the EXTEND bit in 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—Reserved: Read-only bits.

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.

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: Read-only bit, always read as 0.

Bit 6—Burst Enable (BE): Specifies whether burst access is performed on DRAM.

Bit 6: BE	Description	
0	Burst disabled	(Initial value)
1	When DRAM interface is selected, high-speed page mode access is performed	

Bit 5—Memory Data Size (SZ): Specifies the memory data bus-width size for DRAM. Set the same value as A3SZ (1-0) in the BCR2 register.

Bit 5: SZ	Description	
0	Word (16-bit)	(Initial value)
1	Longword (32-bit)	

Bits 4 and 3—Address Multiplex (AMX1, AMX0): The AMX bits specify address multiplexing for DRAM.

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.)	

Bit 2—Refresh Control (RFSH): The RFSH bit determines whether or not DRAM refresh operations are 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): 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 is performed on DRAM, 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 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.

When areas 2 and 3 are both designated as DRAM and self-refreshing is used, the following restriction applies.

In an operation in which one or both areas are self-refreshed during CAS-before-RAS refreshing of both areas 2 and 3, temporarily turn off CAS-before-RAS for both areas 2 and 3, set self-refreshing for the relevant area, and then set CAS-before-RAS refreshing for the other area.

The above restriction does not apply when only area 3 is designated as DRAM.

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): 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.

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 readable/writable 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 refreshing. When DRAM is connected to area 2, the bus width is fixed at 16 bits. In such cases, set the area 3 bus width to 16 bits as well. Other areas should be 8 bits or 16 bits. DCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or in standby mode. When using DRAM, do not access area 2 until this register is initialized.

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/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/W	—	R/W	R/W	R/W	R/W	—

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)

When the same area is accessed immediately after a self-refresh, a time equivalent to the TPC value must be inserted under user control.

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): Specifies whether to conduct burst access to the DRAM connected to area 2.

Bit 6: BE	Description	
0	Burst disabled	(Initial value)
1	High-speed page mode access performed	

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): Determines whether or not DRAM refresh operations are performed on the DRAM connected to area 2.

Bit 2: RFSH	Description	
0	No refresh	(Initial value)
1	Refresh	

Bit 1—Refresh Mode (RMODE): Selects the refresh mode for the DRAM connected to area 2.

When areas 2 and 3 are both designated as DRAM and self-refreshing is used, the following restriction applies.

In an operation in which one or both areas are self-refreshed during CAS-before-RAS refreshing of both areas 2 and 3, temporarily turn off CAS-before-RAS for both areas 2 and 3, set self-refreshing for the relevant area, and then set CAS-before-RAS refreshing for the other area.

The above restriction does not apply when only area 3 is designated as DRAM.

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: Read-only bits, always read as 0.

10.2.8 PCMCIA Control Register (PCR)

The PCMCIA control register (PCR) is a 16-bit readable/writable 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 and 6. The OE and 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 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:	—	—	—	—	—	—	—	—

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 to 8—Reserved: Read-only bits, always read as 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 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTSCR) is a 16-bit readable/writable 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 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:	—	—	—	—	—	—	—	—

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 to 8—Reserved: Read-only bits, always read as 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 Clearing 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 Setting condition: RTCNT = RTCOR

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables the interrupt request caused when the CMF status flag in RTCSR is set to 1.

Bit 6: CMIE	Description
0	Interrupt request caused by CMF is disabled (Initial value)
1	Interrupt request caused by CMF is enabled

Bits 5 to 3—Clock Select Bits (CKS2–CKS0): These bits select the clock input to RTCNT. The source clock is the external bus clock (CKIO). 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.

Bit 5: CKS2	Bit 4: CKS1	Bit 3: CKS0	Description	
			Normal external bus clock	1/4-bus clock
0	0	0	Clock input disabled	Clock input disabled (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): 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 Clearing conditions: When 0 is written to OVF (Initial value)
1	RFCR has exceeded the count limit value set in LMTS Setting conditions: When the RFCR value has exceeded the count limit value set in LMTS*

Note: Contents do not change when 1 is written to OVF.

Bit 1—Refresh Count Overflow Interrupt Enable (OVIE): Enables or disables the interrupt requested caused when the OVF status flag in RTCSR is set to 1.

Bit 1: OVIE	Description
0	Interrupt request caused by OVF is disabled (Initial value)
1	Interrupt request caused by OVF is enabled

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 in RFCR exceeds 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

10.2.10 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit readable/writable register used as an 8-bit counter that counts up on input clock pulses. The clock select bits (CKS2–CKS0) in RTCSR select the input clock. When RTCNT matches RTCOR, the CMF bit in RTCSR is set and RTCNT is cleared. RTCNT is initialized to H'00 by a power-on reset, but it continues incrementing after a manual reset. It is not initialized in standby mode, but retains 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

10.2.11 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit readable/writable register. The values of RTCOR and RTCNT (lower 8 bits) are constantly compared. When the values match, the compare match flag (CMF) in RTCSR is set and RTCNT is cleared to 0. When the refresh bit (RFSH) in 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 in standby mode, but retains 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

10.2.12 Refresh Count Register (RFCR)

The refresh count register (RFCR) is a 16-bit readable/writable register used as a 10-bit counter that increments every time RTCOR and RTCNT match. When RFCR exceeds the count limit value set in the LMTS bit in RTCSR, the OVF bit in RTCSR is set and RFCR is cleared. RFCR is initialized to H'0000 by a power-on reset. It is not initialized by a manual reset or in standby mode, but retains 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

10.2.13 Notes 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). Use the following procedures to perform reads and writes:

1. When writing to RFCR, RTCSR, RTCNT, and RTCOR, use only word transfer instructions. Byte transfer instructions cannot be used.

   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, use 16-bit access. 0 is read out from undefined bits.

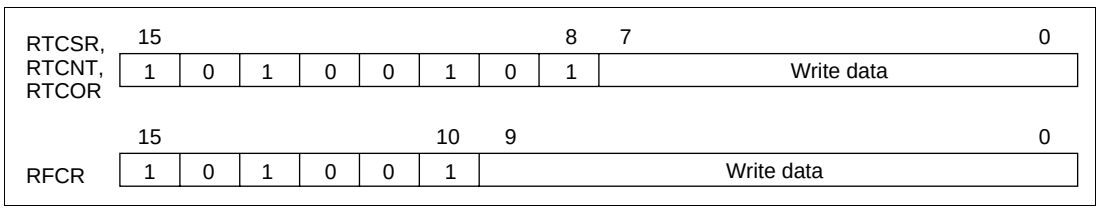


Figure 10.5 Writing to RFCR, RTCSR, RTCNT, and RTCOR

10.3 BSC/BSCP Operation

10.3.1 Endian/Access Size and Data Alignment

The SH7707 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. Switching between the two is designated by an external pin (MD5) at the time of a power-on reset. After a power-on reset, big endian is selected when MD5 is low, and little endian when MD5 is high.

Three data bus widths are available for ordinary memory (byte, word and longword), and two data bus widths (word and longword) for DRAM. For the PCMCIA interface, byte or word can be selected. This means that data alignment is carried out 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, four read operations are required. In the SH7707, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 10.6 through 10.11 show the relationship between endian, device data width, and access unit.

Table 10.6 32-Bit External Device/Big-Endian Access and Data Alignment

$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation
Asserted	—	—	—	Data 7–0	—	—	—	Byte access at 0
—	Asserted	—	—	—	Data 7–0	—	—	Byte access at 1
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 2
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 3
Asserted	Asserted	—	—	Data 15–8	Data 7–0	—	—	Word access at 0
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 2
Asserted	Asserted	Asserted	Asserted	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Longword access at 0

Table 10.7 16-Bit External Device/Big-Endian Access and Data Alignment

$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 0	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 1	
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 2	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 3	
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 0	
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 2	
—	—	Asserted	Asserted	—	—	Data 31–24	Data 23–16	Longword access at 0	1st time at 0
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0		2nd time at 2

Table 10.8 8-Bit External Device/Big-Endian Access and Data Alignment

$\overline{WE3}$, $\overline{CASHH}$	$\overline{WE2}$, $\overline{CASHL}$	$\overline{WE1}$, $\overline{CASLH}$	$\overline{WE0}$, $\overline{CASLL}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 0	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 1	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 2	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 3	
—	—	—	Asserted	—	—	—	Data 15–8	Word access at 0	1st time at 0
—	—	—	Asserted	—	—	—	Data 7–0		2nd time at 1
—	—	—	Asserted	—	—	—	Data 15–8	Word access at 2	1st time at 2
—	—	—	Asserted	—	—	—	Data 7–0		2nd time at 3
—	—	—	Asserted	—	—	—	Data 31–24	Longword access at 0	1st time at 0
—	—	—	Asserted	—	—	—	Data 23–16		2nd time at 1
—	—	—	Asserted	—	—	—	Data 15–8		3rd time at 2
—	—	—	Asserted	—	—	—	Data 7–0		4th time at 3

Table 10.9 32-Bit External Device/Little-Endian Access and Data Alignment

$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 0
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 1
—	Asserted	—	—	—	Data 7–0	—	—	Byte access at 2
Asserted	—	—	—	Data 7–0	—	—	—	Byte access at 3
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 0
Asserted	Asserted	—	—	Data 15–8	Data 7–0	—	—	Word access at 2
Asserted	Asserted	Asserted	Asserted	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Longword access at 0

Table 10.10 16-Bit External Device/Little-Endian Access and Data Alignment

$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 0	
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 1	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 2	
—	—	Asserted	—	—	—	Data 7–0	—	Byte access at 3	
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 0	
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Word access at 2	
—	—	Asserted	Asserted	—	—	Data 15–8	Data 7–0	Longword access at 0	1st time at 0
—	—	Asserted	Asserted	—	—	Data 31–24	Data 23–16		2nd time at 2

Table 10.11 8-Bit External Device/Little-Endian Access and Data Alignment

$\overline{WE3}$, $\overline{CASHH}$	$\overline{WE2}$, $\overline{CASHL}$	$\overline{WE1}$, $\overline{CASLH}$	$\overline{WE0}$, $\overline{CASLL}$	D31–D24	D23–D16	D15–D8	D7–D0	Operation	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 0	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 1	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 2	
—	—	—	Asserted	—	—	—	Data 7–0	Byte access at 3	
—	—	—	Asserted	—	—	—	Data 7–0	Word access at 0	1st time at 0
—	—	—	Asserted	—	—	—	Data 15–8		2nd time at 1
—	—	—	Asserted	—	—	—	Data 7–0	Word access at 2	1st time at 2
—	—	—	Asserted	—	—	—	Data 15–8		2nd time at 3
—	—	—	Asserted	—	—	—	Data 7–0	Longword access at 0	1st time at 0
—	—	—	Asserted	—	—	—	Data 15–8		2nd time at 1
—	—	—	Asserted	—	—	—	Data 23–16		3rd time at 2
—	—	—	Asserted	—	—	—	Data 31–24		4th time at 3

10.3.2 Description of Areas

Area 0: Area 0 physical address bits A28–A26 are 000. Address bits 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 (MD3, MD4). When the Area 0 space is accessed, the $\overline{CS0}$ signal is asserted. The $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$–$\overline{WE3}$ 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 in WCR2. Also, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 10 according to the number of waits.

Area 1: Area 1 physical address bits A28–A26 are 001. Address bits 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 on-chip supporting modules. External memory cannot be connected.

Area 2: Area 2 physical address bits A28–A26 are 010. Address bits 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, can be connected to this space. Byte, word, or longword can be selected as the bus width using the A2SZ1–A2SZ0 bits in BCR2 for ordinary memory. When DRAM is connected to area 2, the bus width is fixed at 16 bits. The bus width for area 3 also needs to be 16 bits, while all other areas must be either 8 bits or 16 bits.

When the area 2 space is accessed, the $\overline{\text{CS2}}$ signal is asserted. When ordinary memory is connected, the $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are also asserted. The number of bus cycles is selected between 0 and 3 wait cycles using the A2W1–A2W0 bits in WCR2. Only when ordinary memory is connected, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

When DRAM is connected, the $\overline{\text{RAS2}}$ signal, $\overline{\text{CAS2H}}$ signal, $\overline{\text{CAS2L}}$ signal, and RD/$\overline{\text{WR}}$ signal are all asserted and addresses multiplexed. Control of $\overline{\text{RAS2}}$, $\overline{\text{CAS}}$, data timing, and address multiplexing is set with DCR.

Area 3: Area 3 physical address bits A28–A26 are 011. Address bits 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, can be connected to this space. Byte, word or longword can be selected as the bus width using the A3SZ1–A3SZ0 bits in BCR2 for ordinary memory. When DRAM is connected, a bus width of 16 or 32 bits can be selected. In this case, ensure that the SZ bit setting in MCR matches the setting of bits A3SZ1–A3SZ0 in BCR2.

When area 3 space is accessed, $\overline{\text{CS3}}$ is asserted.

When ordinary memory is connected, the $\overline{\text{RD}}$ signal that can be used as $\overline{\text{OE}}$ and the $\overline{\text{WE0}}$–$\overline{\text{WE3}}$ signals for write control are asserted. The number of bus cycles is selected between 0 and 3 wait cycles using the A3W1–A3W0 bits in WCR2. Only when ordinary memory is connected, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

When DRAM is connected, the $\overline{RAS}$ signal, $\overline{CASHH}$ signal, $\overline{CASHL}$ signal, $\overline{CASLH}$ signal, $\overline{CASLL}$ signal, and RD/$\overline{WR}$ signal are all asserted and addresses multiplexed. For all of these, control of RAS, CAS, data timing, and address multiplexing is set with MCR.

Area 4: Area 4 physical address bits A28–A26 are 100. Address bits 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 in BCR2. When the area 4 space is accessed, the $\overline{CS4}$ signal is asserted. The $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$–$\overline{WE3}$ 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 in WCR2. Also, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$).

Area 5: Area 5 physical address bits A28–A26 are 101. Address bits 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 is 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 in BCR2. For the PCMCIA interface, byte or word can be selected as the bus width using the A5SZ1–A5SZ0 bits in BCR2.

When the area 5 space is accessed and ordinary memory is connected, the $\overline{CS5}$ signal is asserted. The $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$–$\overline{WE3}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{CE1A}$ signal, $\overline{CE2A}$ signal, $\overline{OE}$ signal, and $\overline{WE1}$ signal are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A5W2–A5W0 bits in WCR2. Also, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 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 to 3.5 using bits A5TED1–A5TED0 and A5TEH1–A5TEH0 in the PCR register.

Area 6: Area 6 physical address bits A28–A26 are 110. Address bits 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 in BCR2. For the PCMCIA interface, byte or word can be selected as the bus width using the A6SZ1–A6SZ0 bits in BCR2.

When the area 6 space is accessed and ordinary memory is connected, the $\overline{CS6}$ signal is asserted. The $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$–$\overline{WE3}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{CE1B}$ signal, $\overline{CE2B}$ signal, $\overline{OE}$ signal, and $\overline{WE}$, $\overline{ICIORD}$, and $\overline{ICIOWR}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A6W2–A6W0 bits in WCR2. Also, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 10 according to the number of waits. The setup and hold times of address $\overline{CS6}$ for the read/write strobe signals can be set in the range 0.5 to 3.5 using bits A6TED1–A6TED0 and A6TEH1–A6TEH0 in the PCR register.

10.3.3 Basic Interface

Basic Timing: The basic interface of the SH7707 uses strobe signal output in consideration of the fact that mainly SRAM 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 $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{CSn}$ 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 $\overline{WE}$ 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 copy-back follows the set bus width and transfers a total of 16 bytes consecutively. 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.

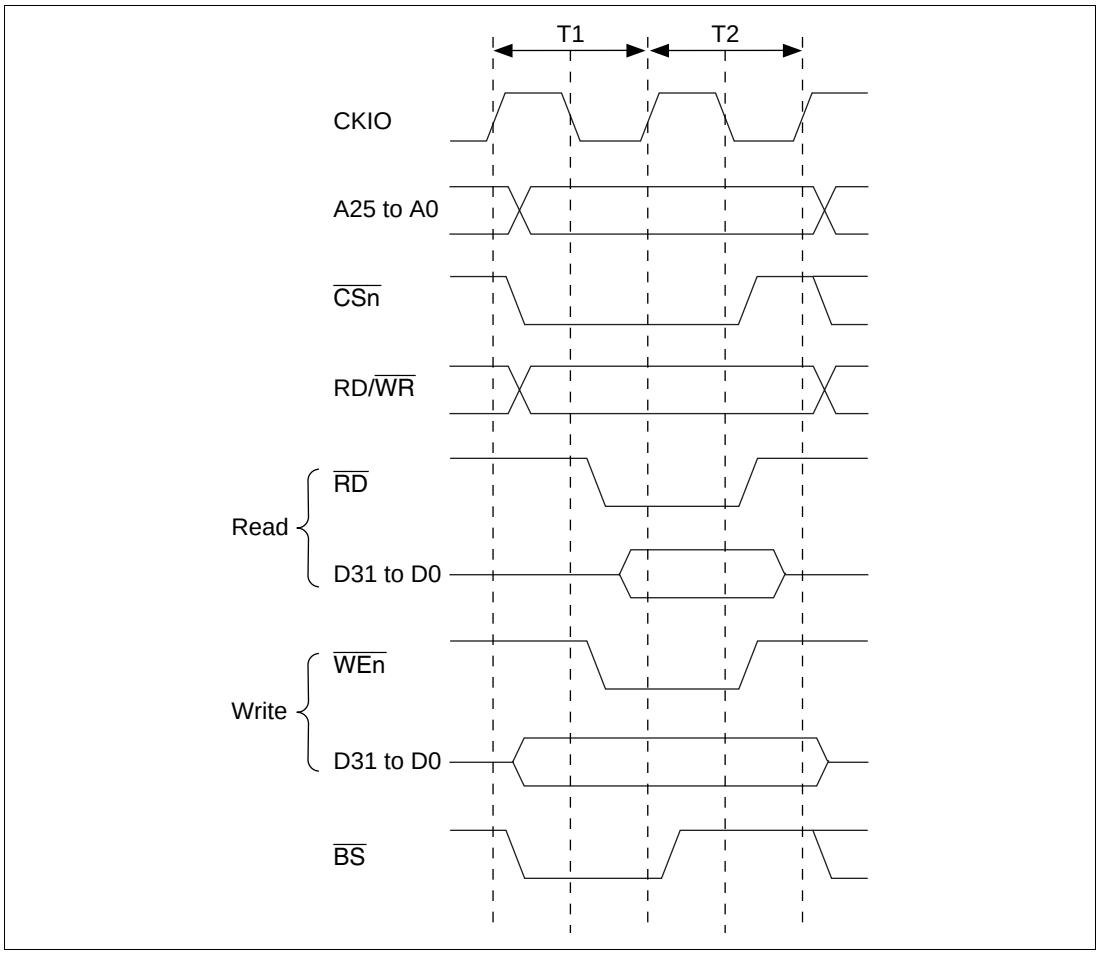


Figure 10.6 Basic Timing of Basic Interface

Figures 10.7, 10.8, and 10.9 show examples of connection to 32-, 16-, and 8-bit data width SRAM.

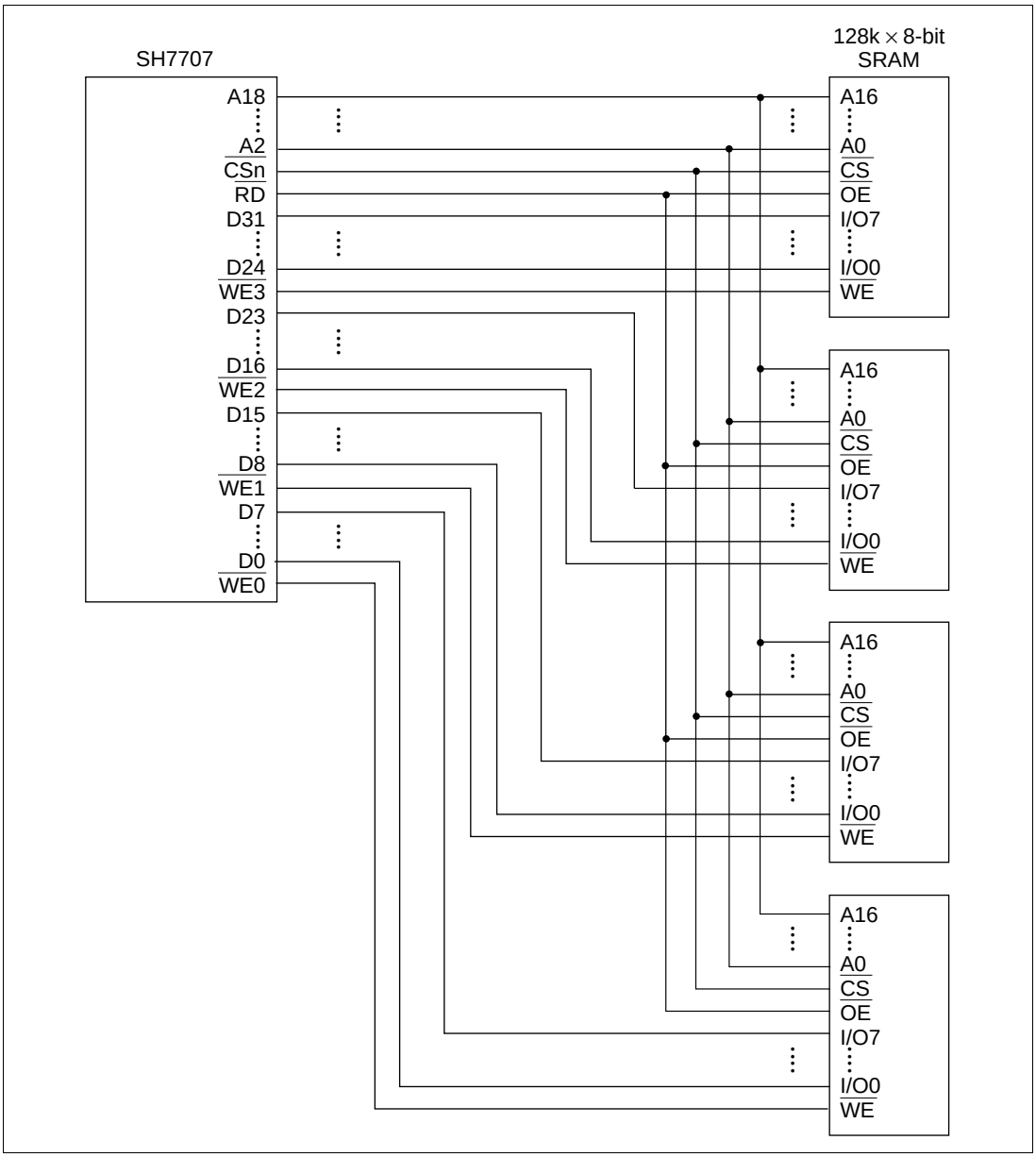


Figure 10.7 Example of 32-Bit Data Width SRAM Connection

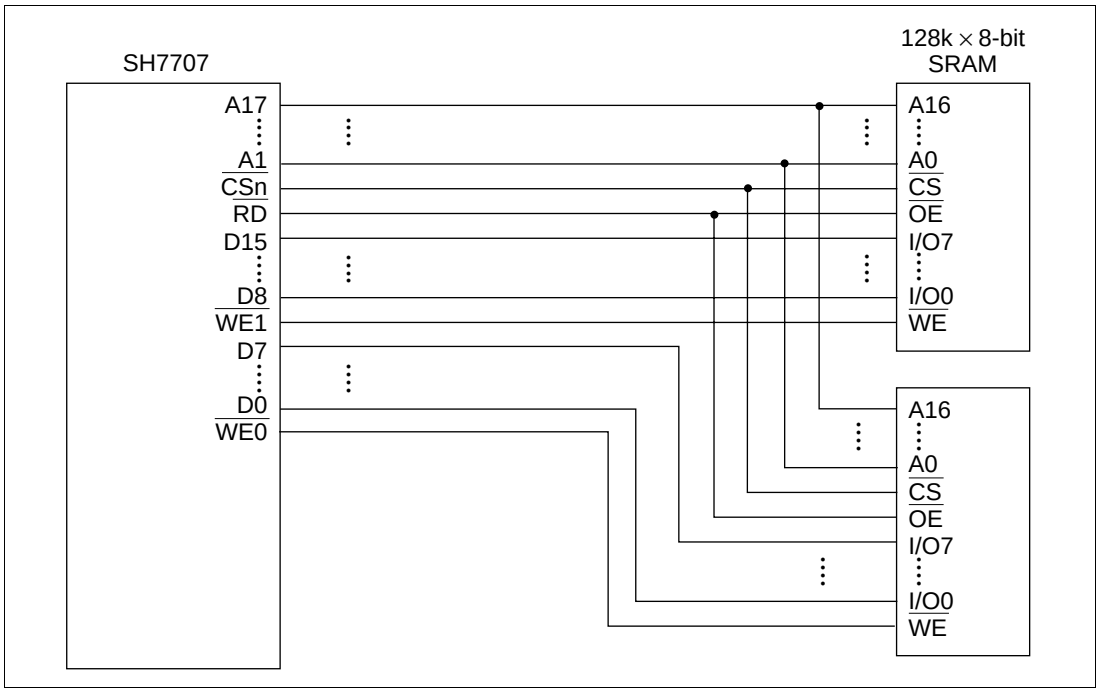


Figure 10.8 Example of 16-Bit Data Width SRAM Connection

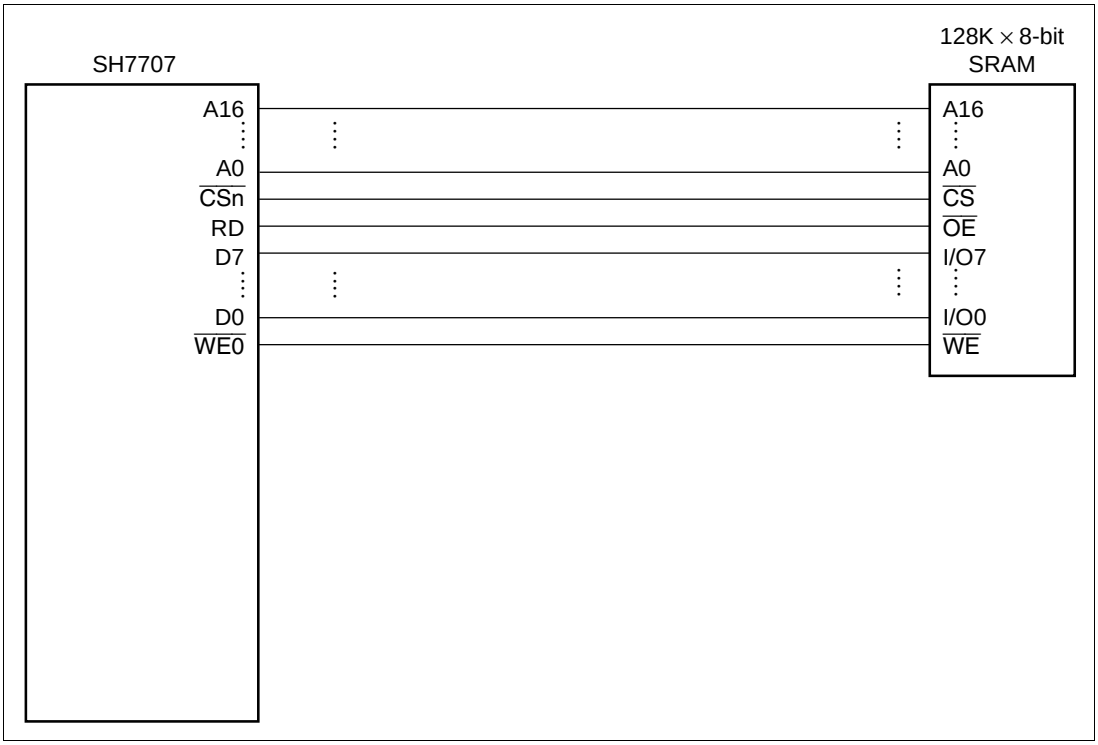


Figure 10.9 Example of 8-Bit Data Width SRAM Connection

Wait State Control: Wait state insertion on the basic interface can be controlled by 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 10.2.5, 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.

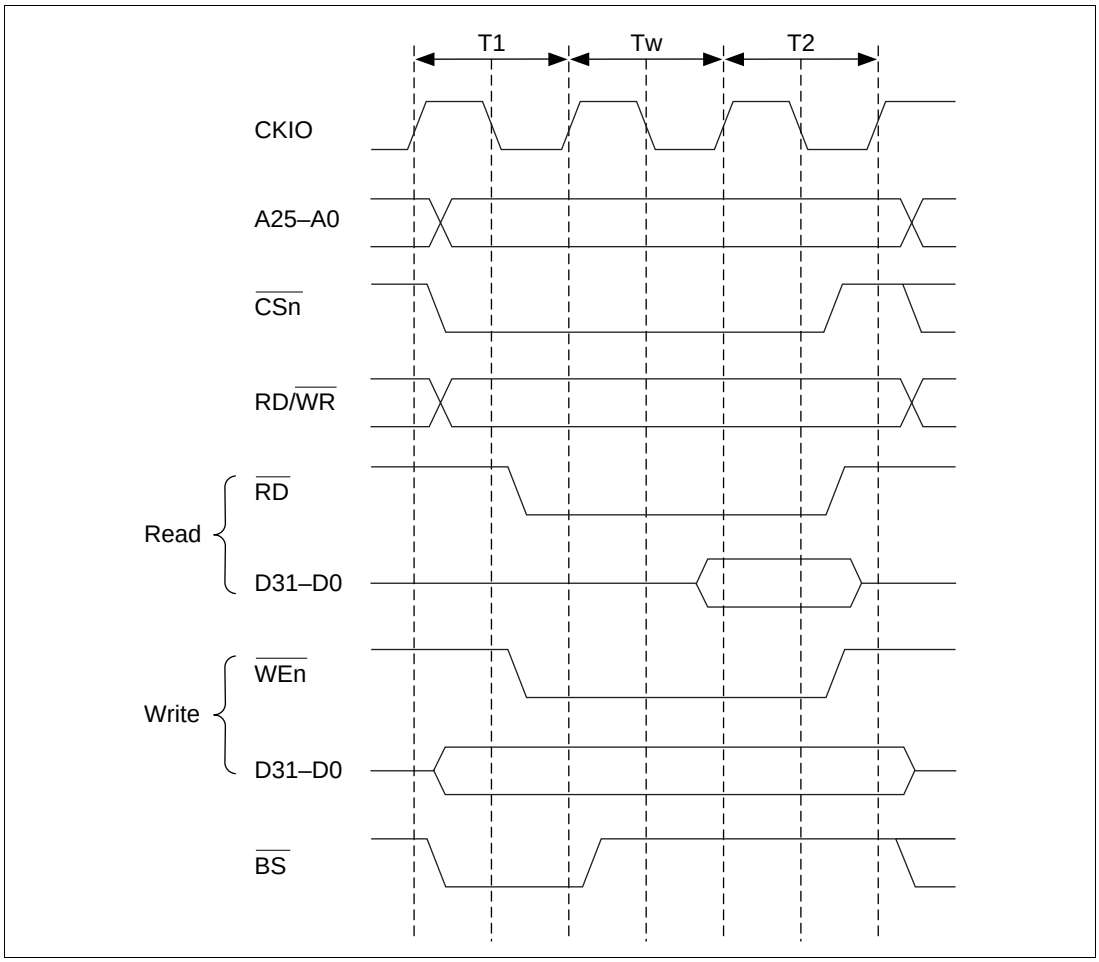


Figure 10.10 Basic Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by WCR2, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ 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 $\overline{\text{WAIT}}$ signal has no effect if asserted in the T1 cycle or the first Tw cycle. The $\overline{\text{WAIT}}$ signal is sampled on the rising edge of the clock.

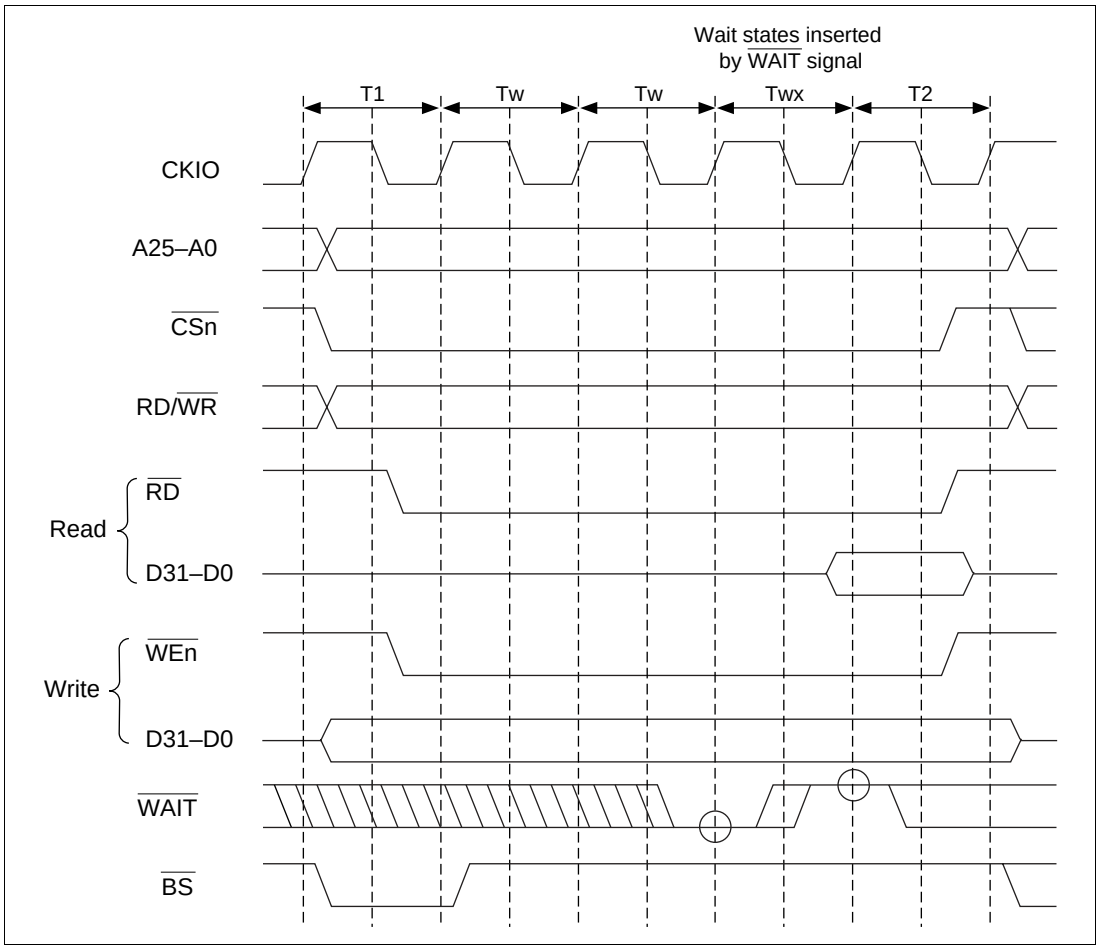


Figure 10.11 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal)

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 SH7707 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 both area 2 and area 3 when bits DRAMTP2-0 are set to 101.

2-CAS 16-bit DRAMs can be connected, since $\overline{\text{CAS}}$ is used to control byte access.

Signals used for connection when DRAM is connected to area 3 are $\overline{\text{RAS}}$, $\overline{\text{CASHH}}$, $\overline{\text{CASHL}}$, $\overline{\text{CASLH}}$, $\overline{\text{CASLL}}$, and RD/$\overline{\text{WR}}$. $\overline{\text{CASHH}}$ and $\overline{\text{CASHL}}$ 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 $\overline{\text{RAS2}}$, $\overline{\text{CAS2H}}$, $\overline{\text{CAS2L}}$, and RD/$\overline{\text{WR}}$, and those for area 3 DRAM connection are $\overline{\text{RAS}}$, $\overline{\text{CASLH}}$, $\overline{\text{CASLL}}$, and RD/$\overline{\text{WR}}$.

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.

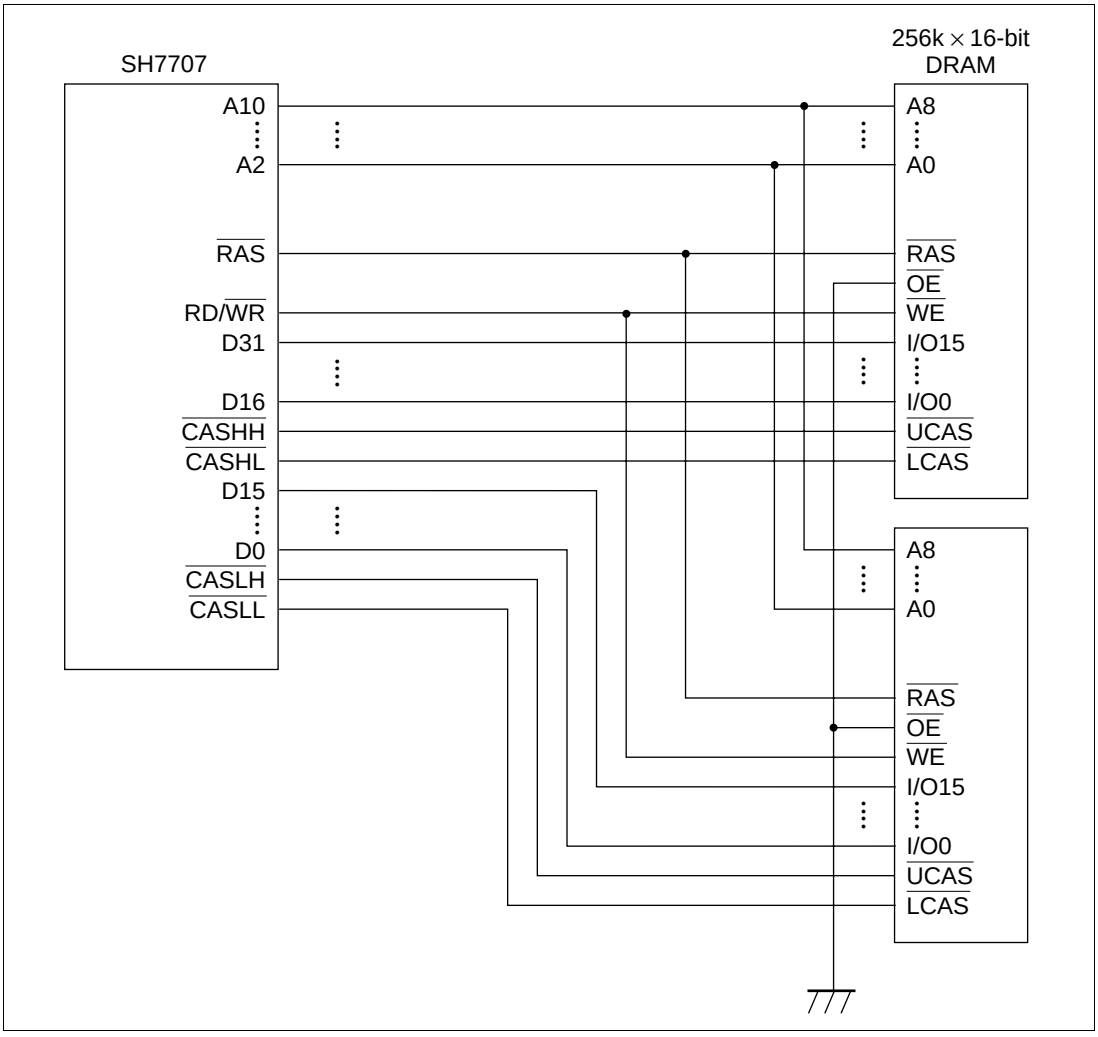


Figure 10.12 Example of DRAM Connection (32-Bit Data Width)

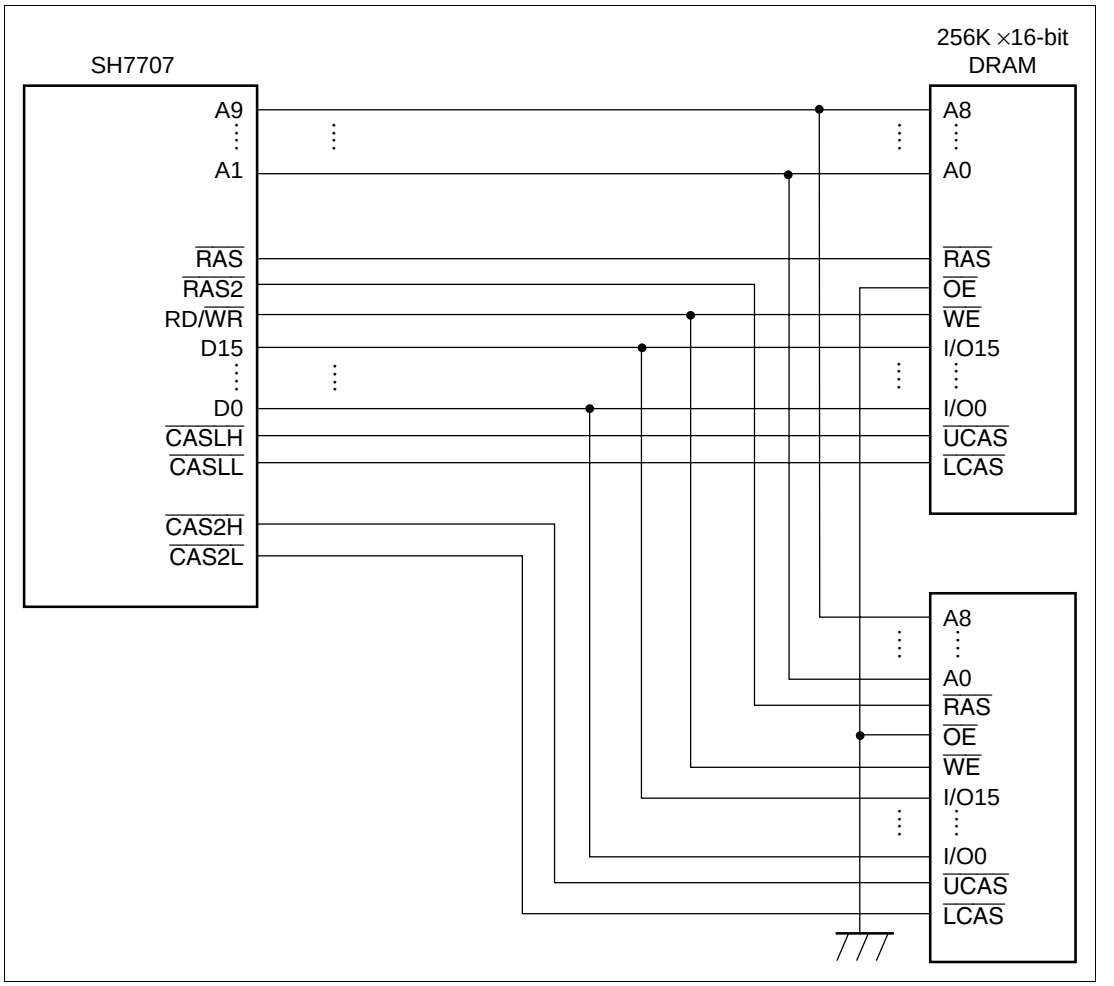


Figure 10.13 Example of DRAM Connection (16-Bit Data Width)

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 SH7707 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 table 10.12. The address output pins subject to address multiplexing are A15–A1. Pins A25–A16 carry the original address.

Table 10.12 Relationship between AMX1-0 and Address Multiplexing

Setting		Number of Column		External Address Pins	
AMX1	AMX0	Address Bits	Output Timing	A1–A14	A15
0	0	8 bits	Column address	A1–A14	A15
			Row address	A9–A22	A23
0	1	9 bits	Column address	A1–A14	A15
			Row address	A10–A23	A24
1	0	10 bits	Column address	A1–A14	A15
			Row address	A11–A24	A25
1	1	11 bits	Column address	A1–A14	A15
			Row address	A12–A25	A15

Basic Timing: The basic timing for DRAM access is 3 cycles (RCD (1–0) = 00, AnW (1–0) = 00). 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.

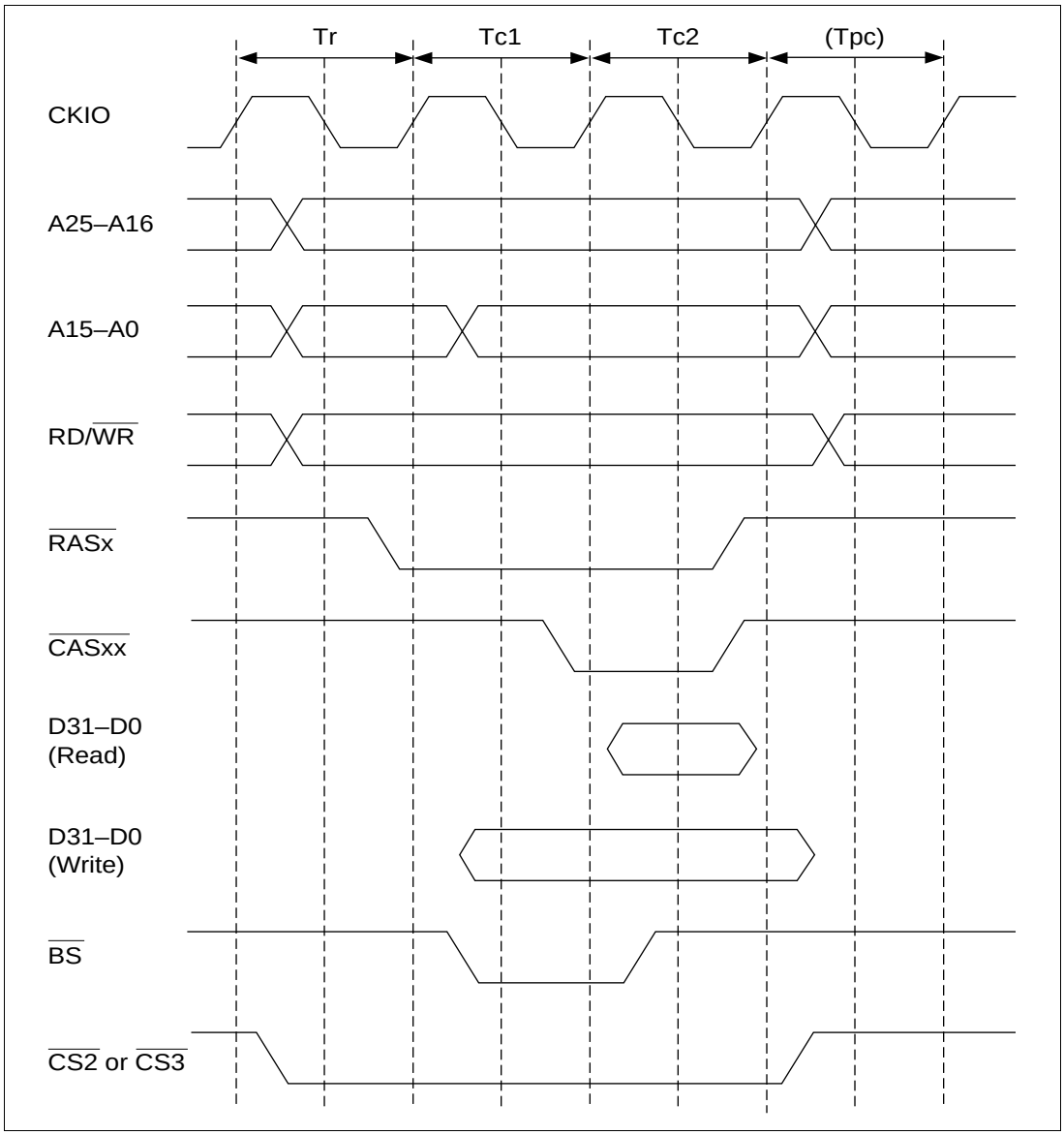


Figure 10.14 Basic Timing for DRAM Access

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, DCR and BCR3. 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, DCR and BCR3, 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, DCR and BCR3. 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 A2W (1-0) or A3W (1-0) in WCR2.

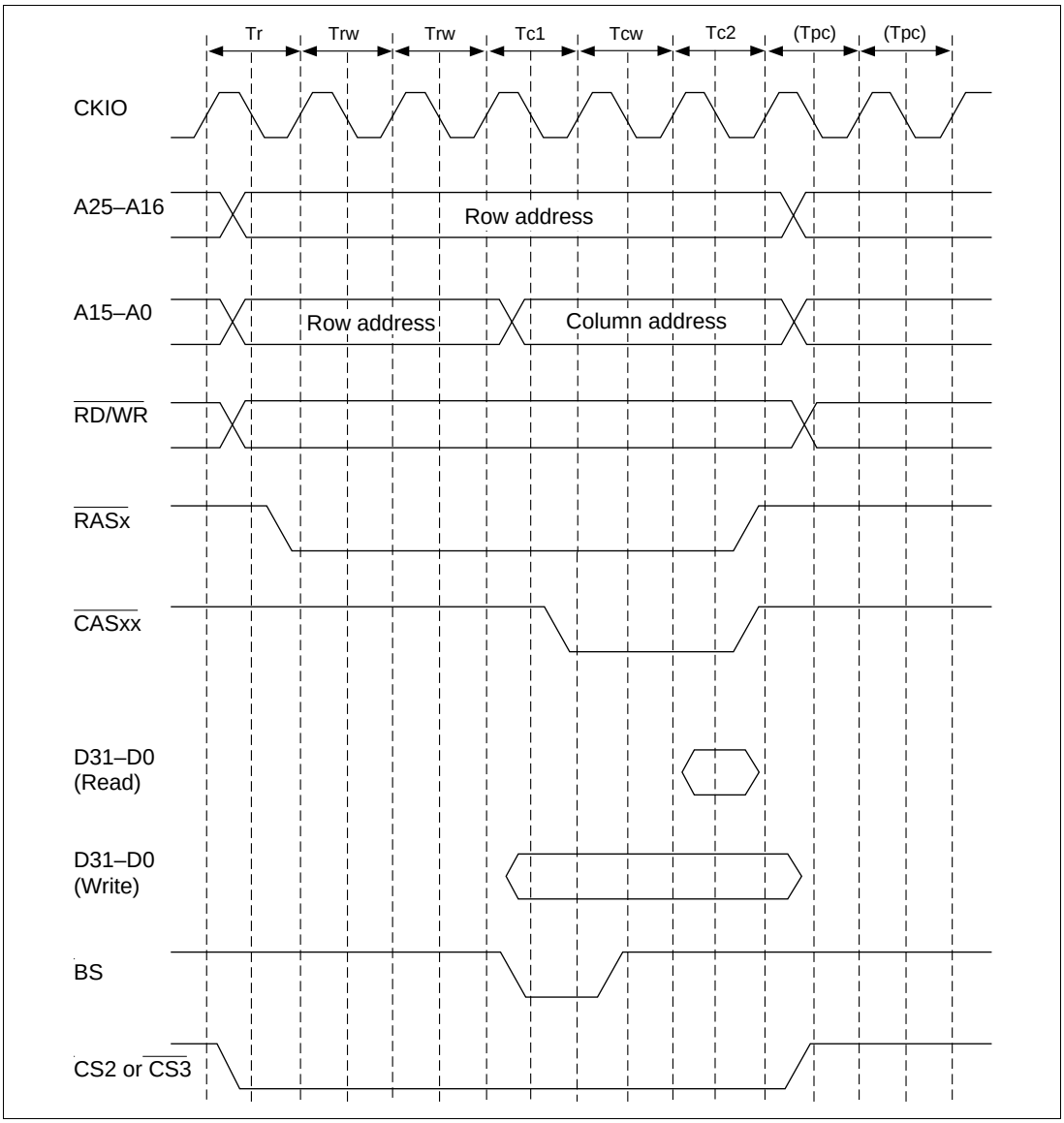


Figure 10.15 DRAM Wait State Timing

Short Pitch Access: The bus state controller (only BSCP) of the SH7707 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.

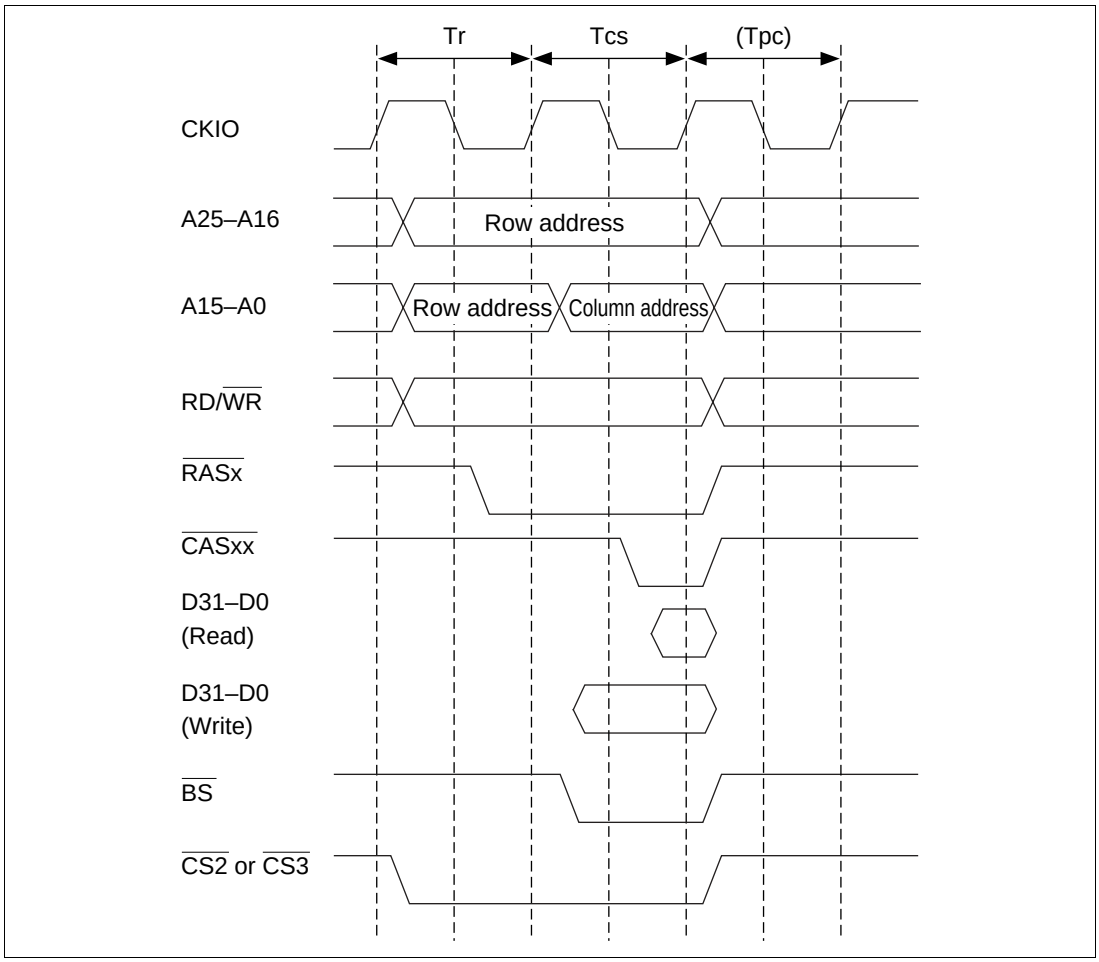


Figure 10.16 DRAM Short-Pitch Access Timing

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 in first-to-last order for 16-byte boundary data.

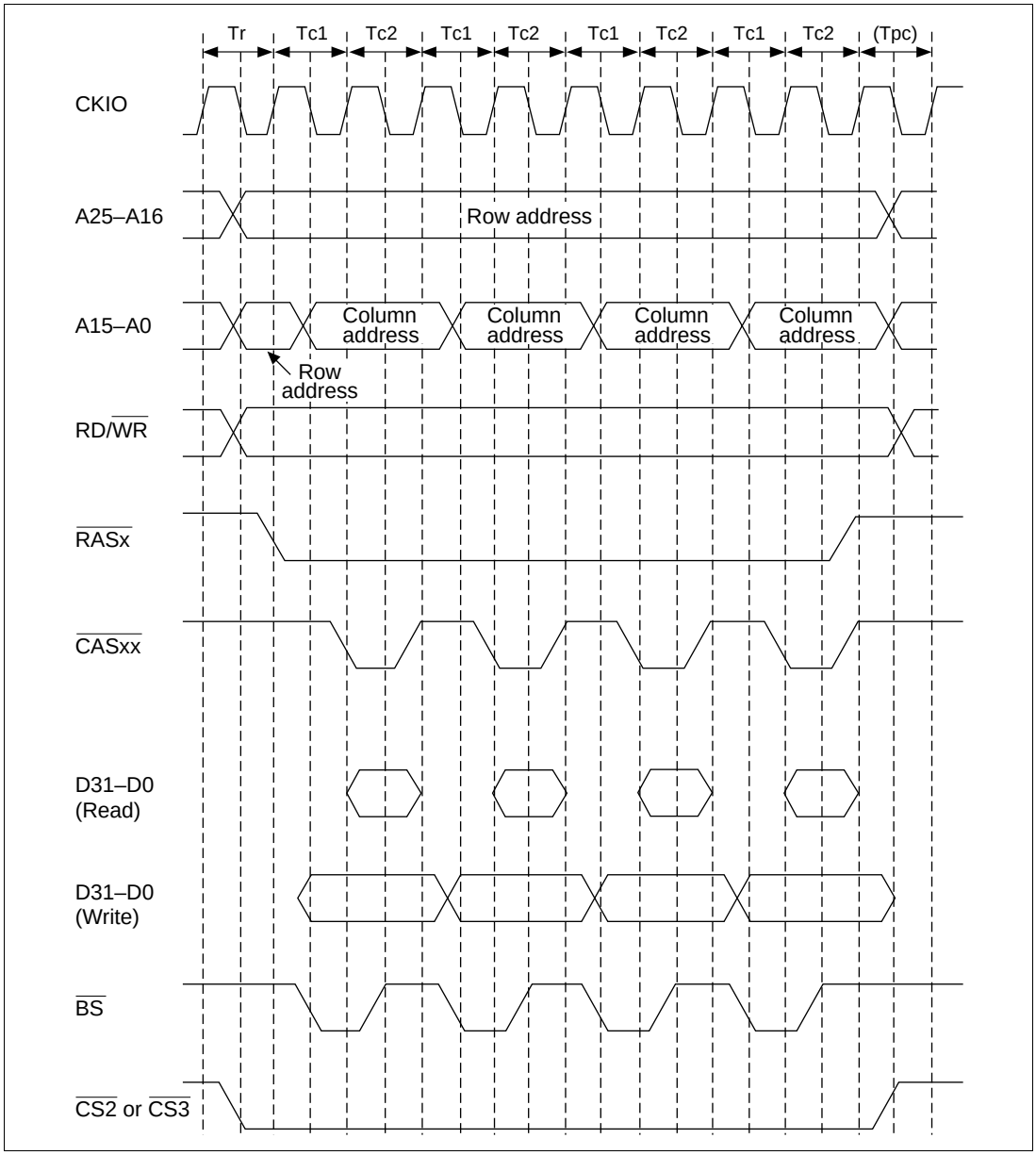


Figure 10.17 DRAM Burst Access Timing

Short-Pitch Burst Access: The bus state controller (only BSCP) of the SH7707 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.

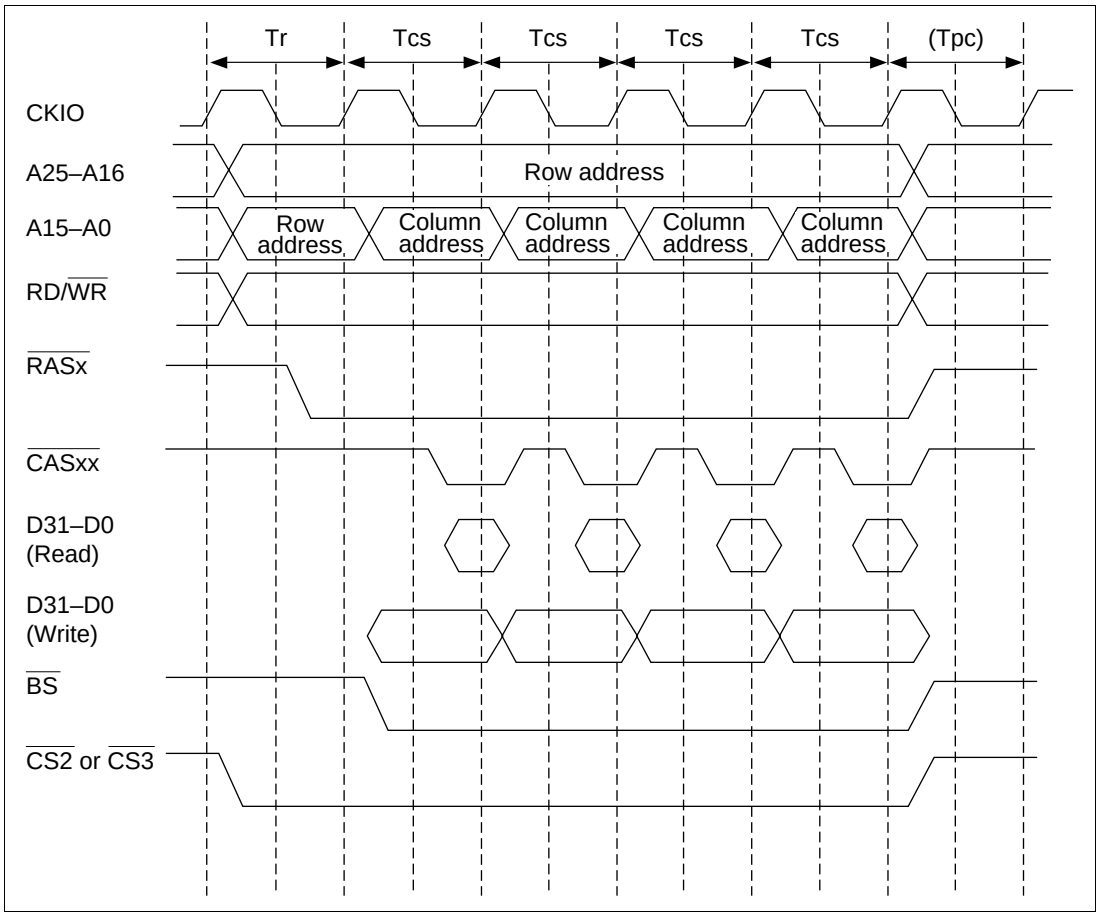


Figure 10.18 DRAM Short-Pitch Burst Access Timing

EDO Mode: With DRAM, in addition to the mode in which data is output to the data bus only while the $\overline{\text{CAS}}$ signal is asserted in a data read cycle, an EDO (extended data out) mode is also provided in which, once the $\overline{\text{CAS}}$ signal is asserted while the $\overline{\text{RAS}}$ signal is asserted, even if the $\overline{\text{CAS}}$ signal is negated, data is output to the data bus until the $\overline{\text{CAS}}$ signal is next asserted. In the SH7707, 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 figures 10.19 (a) and 10.19 (b), and burst access in figures 10.20 (a) and 10.20 (b).

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 $\overline{CAS}$ 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.

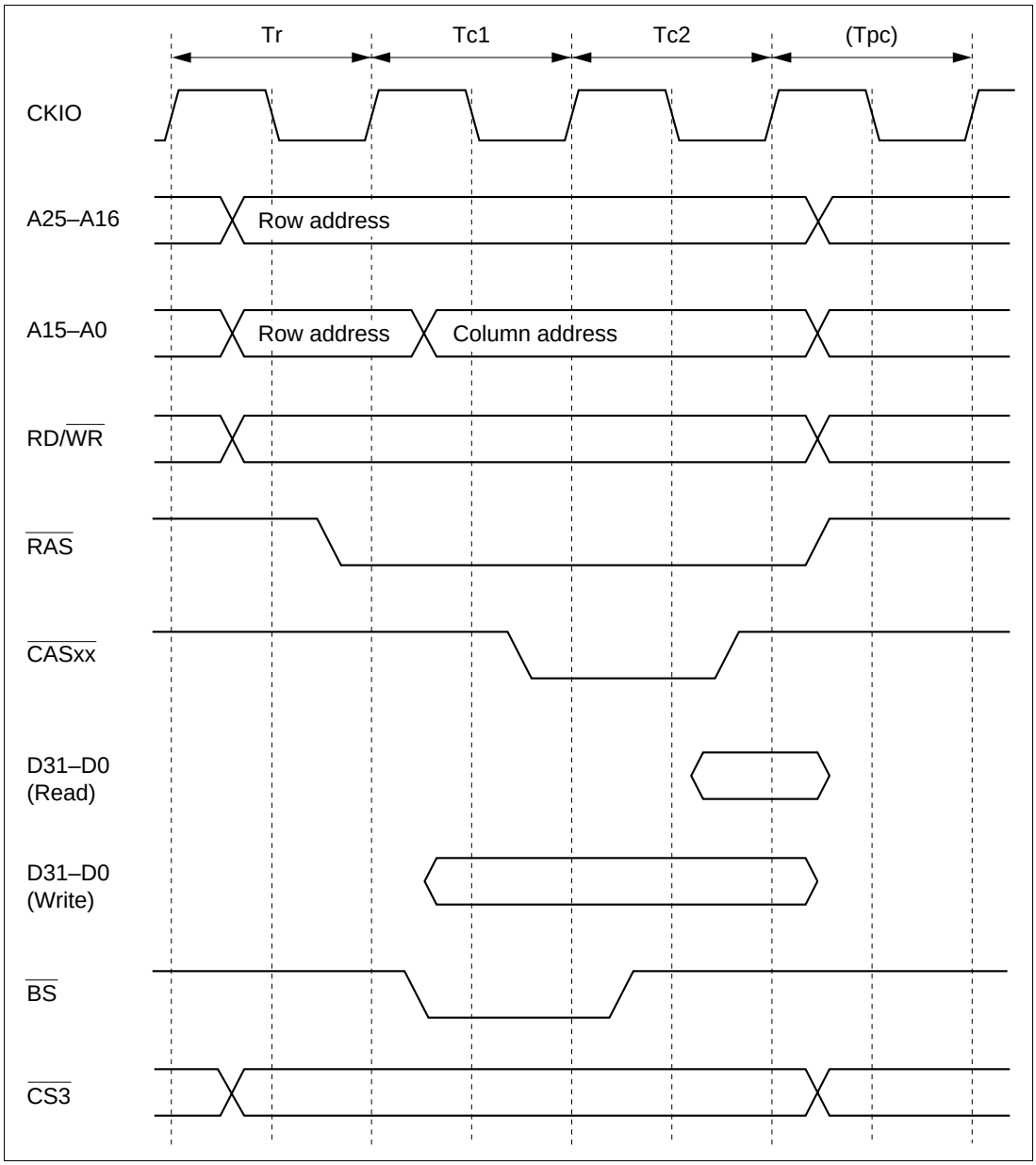


Figure 10.19 (a) Normal Access Timing in DRAM EDO Mode

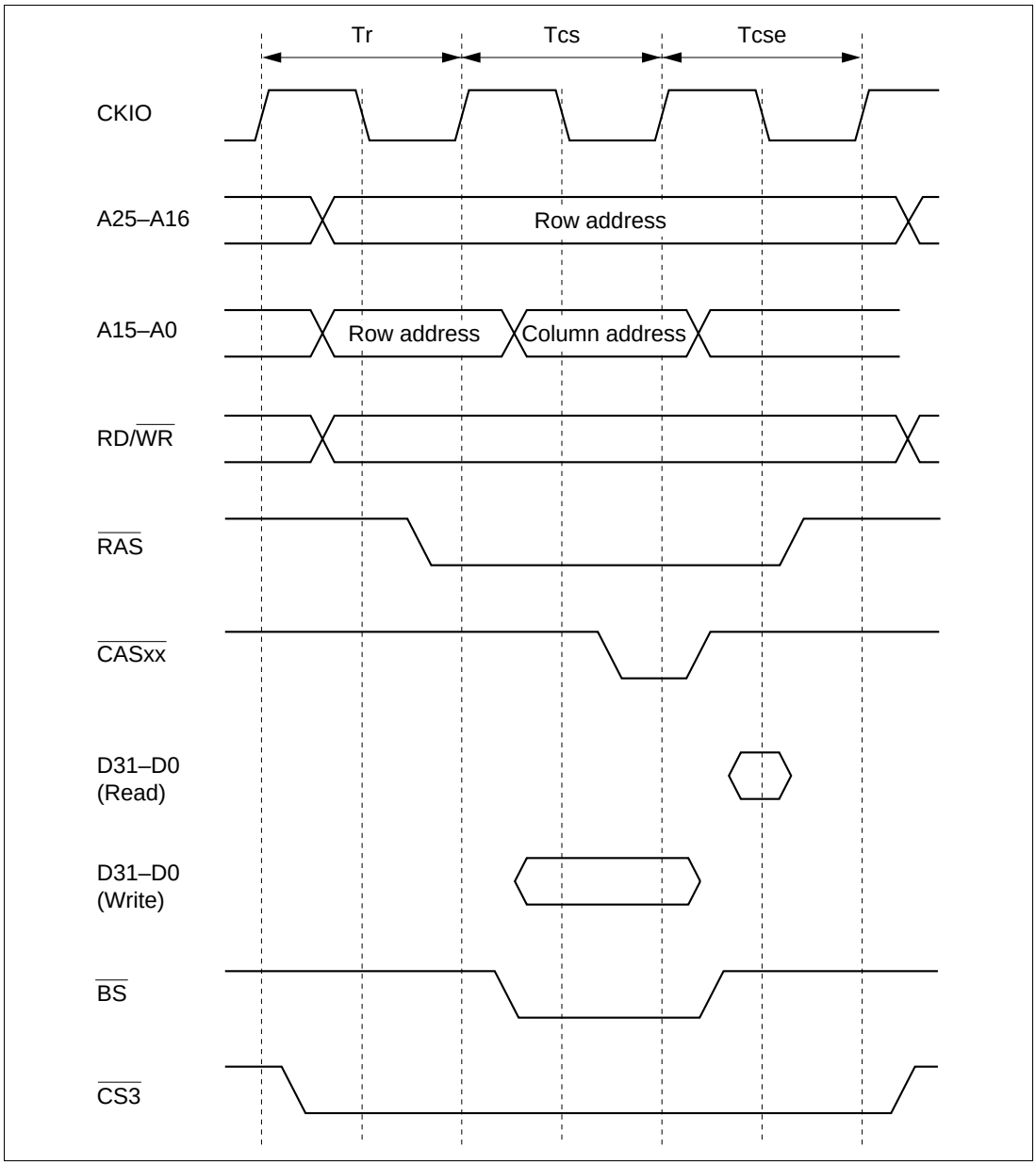


Figure 10.19 (b) Short-Pitch Access Timing in DRAM ED0 Mode

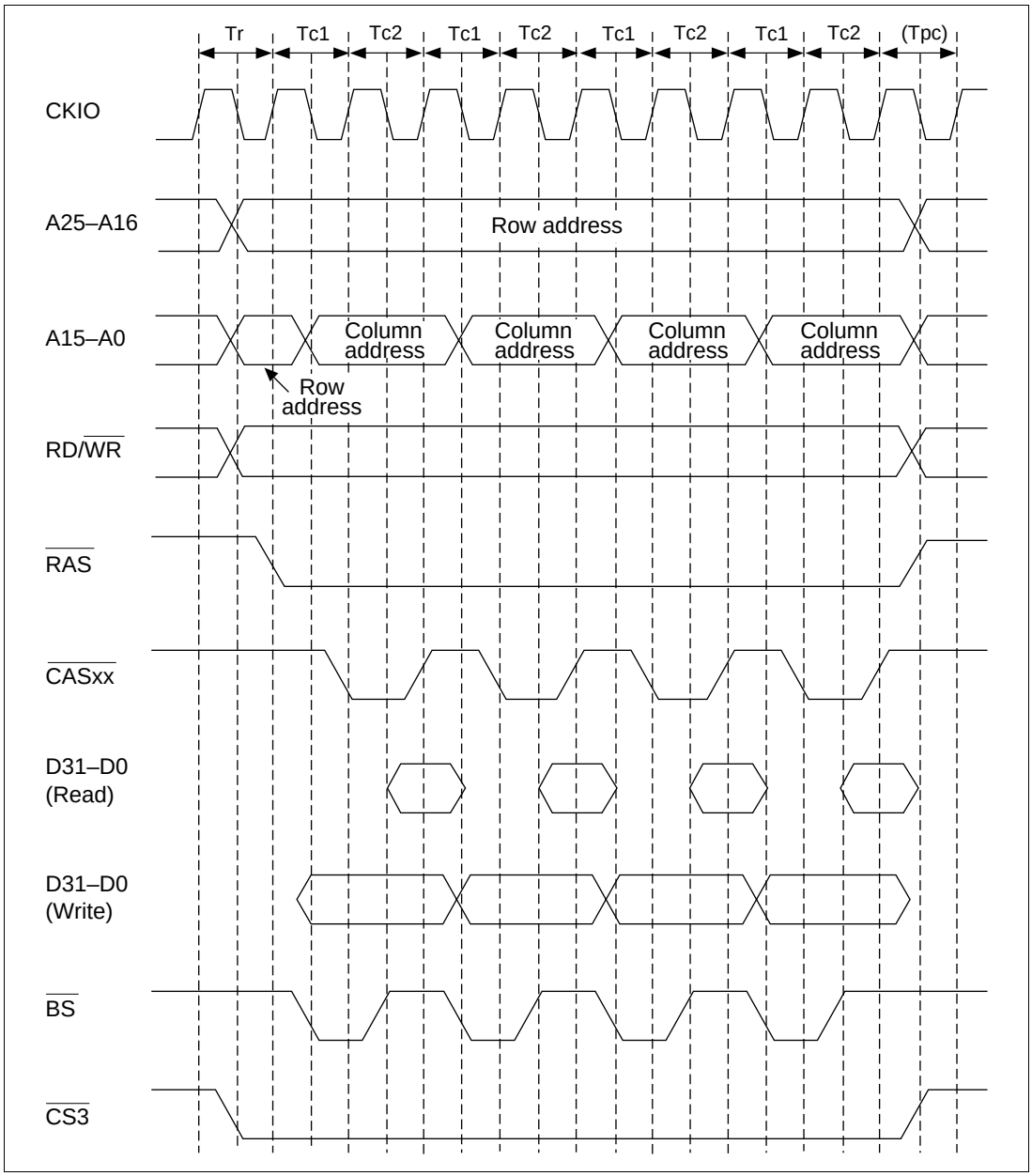


Figure 10.20 (a) Burst Access Timing in DRAM EDO Mode

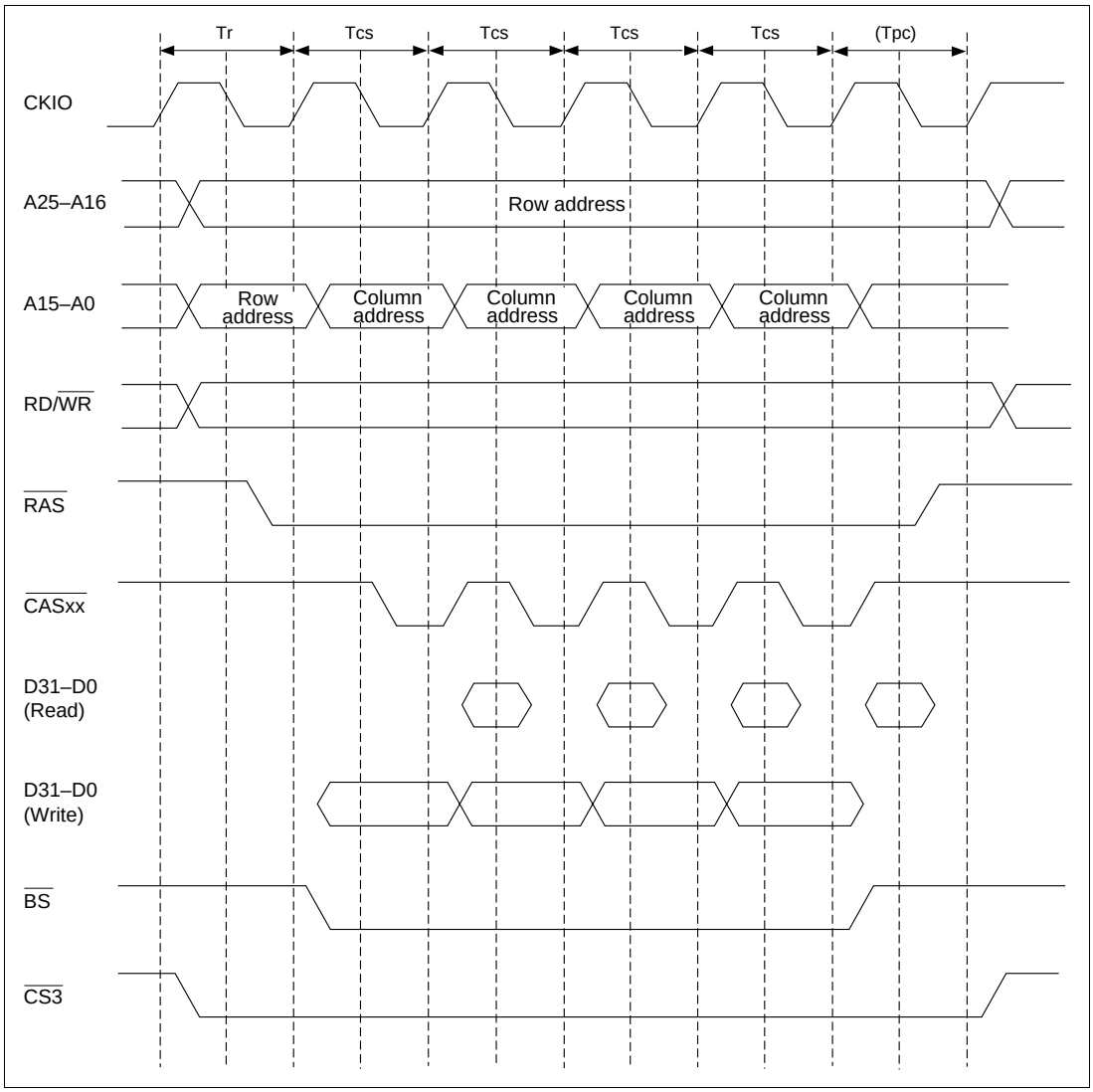


Figure 10.20 (b) DRAM Short-Pitch Burst Access Timing in DRAM ED0 Mode

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 $\overline{\text{IRQOUT}}$ pin goes low. If the SH7707's external bus can be used, CAS-before-RAS refreshing is performed, and if there is no other interrupt request the $\overline{\text{IRQOUT}}$ 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.

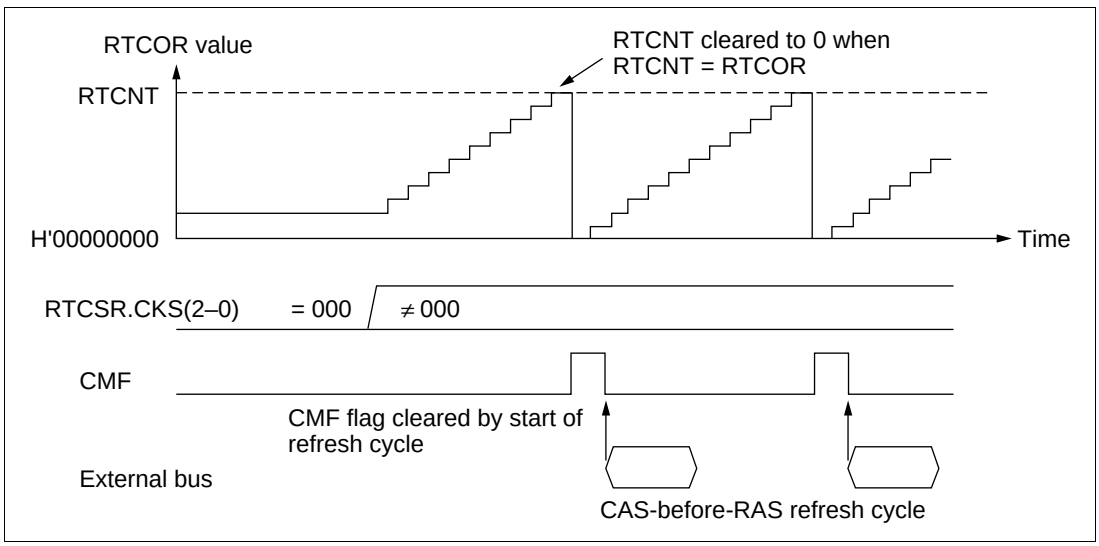


Figure 10.21 CAS-Before-RAS Refresh Operation

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.

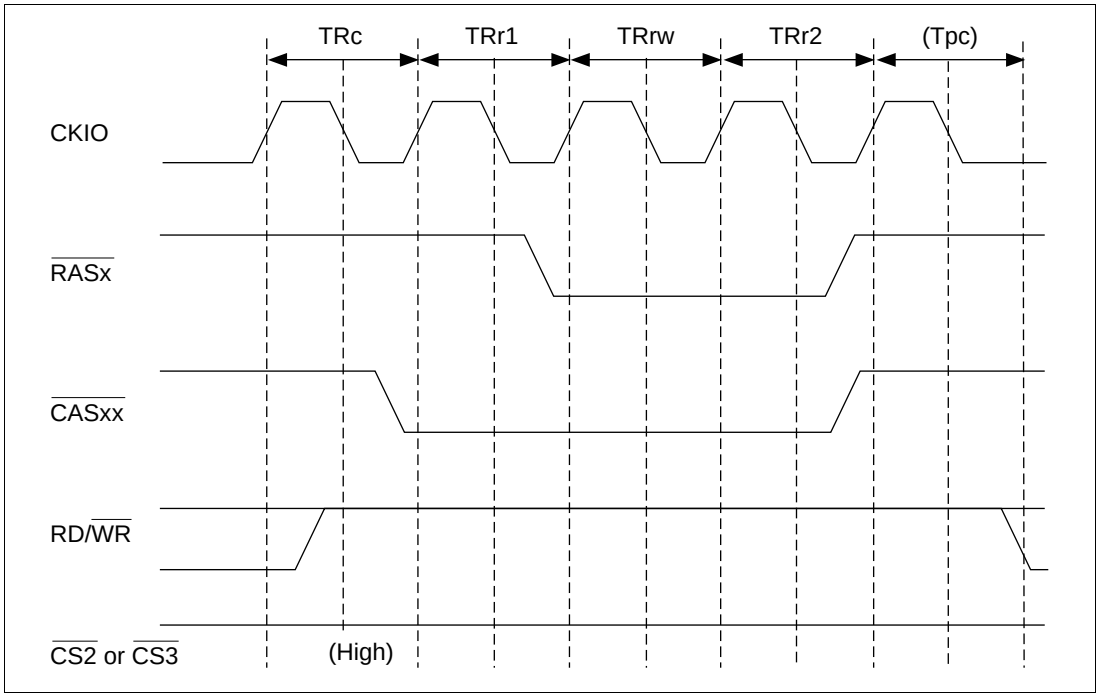


Figure 10.22 DRAM CAS-Before-RAS Refresh Cycle Timing

The self-refreshing supported by the SH7707 is shown in figure 10.23.

After the self-refresh is cleared, the refresh controller immediately generates a refresh request. When the same area is accessed immediately after a self-refresh, a time equivalent to the TPC value must be inserted under user control.

DRAMs include low-power products (L versions) with a long refresh cycle time (for example, the L version has a refresh cycle rate 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 in 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 event of a manual reset.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in the event 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 $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ 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.

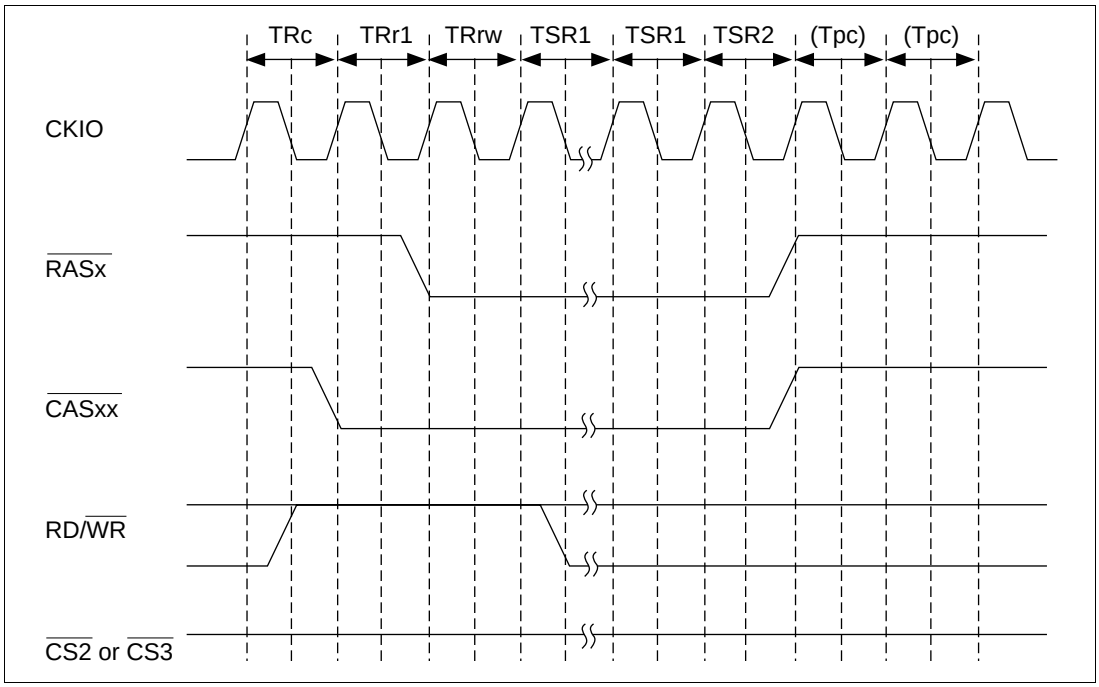


Figure 10.23 DRAM Self-Refresh Cycle Timing

Power-On Sequence: Regarding use of DRAM after powering on, a wait time (at least 100 µs or 200 µs) during which no access can be performed should 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 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 10.24. Two wait cycles are set. Basically, access is performed in the same way as for normal space, but when the first cycle ends the $\overline{CS0}$ 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.

$\overline{WAIT}$ 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 10.25.

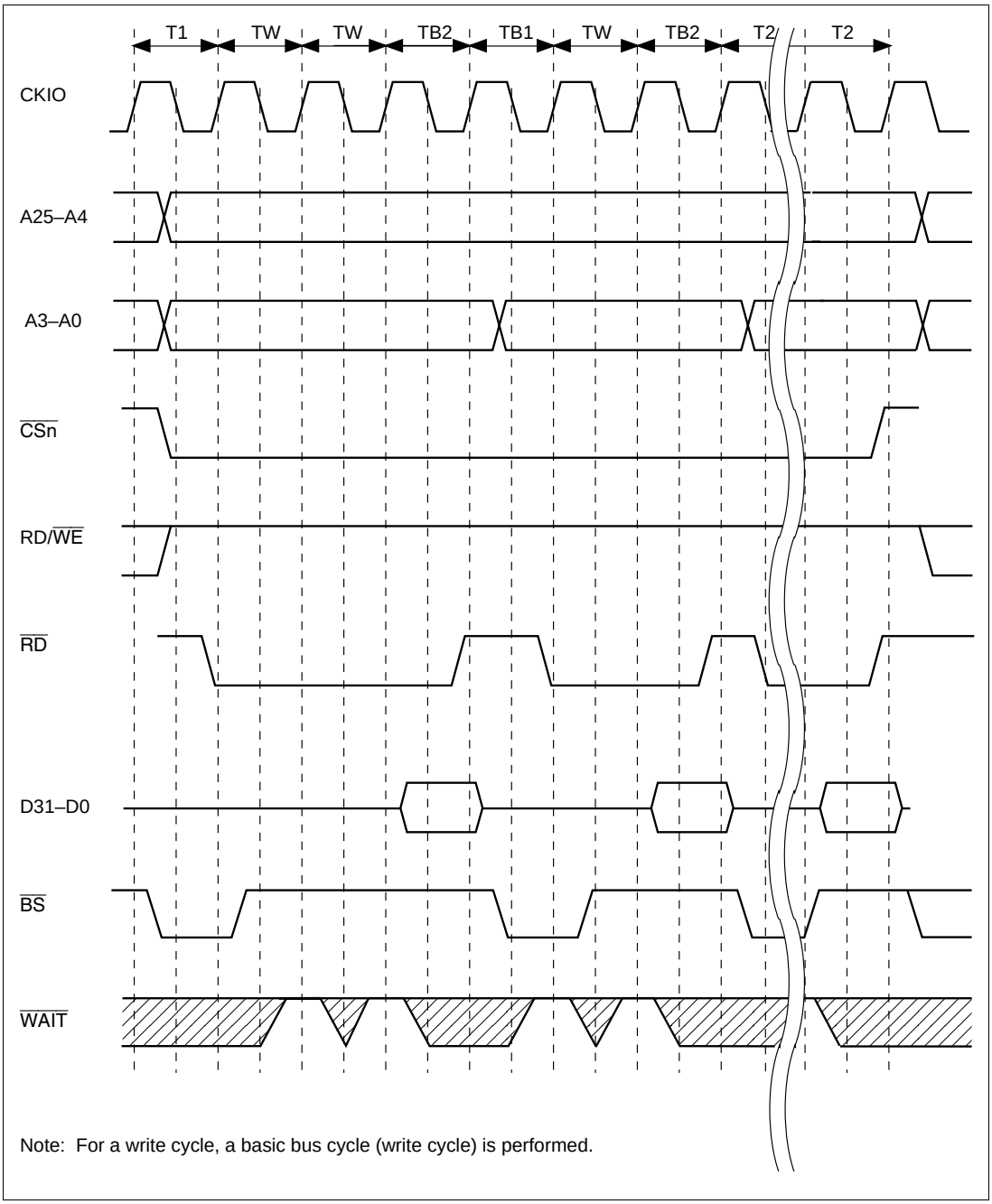
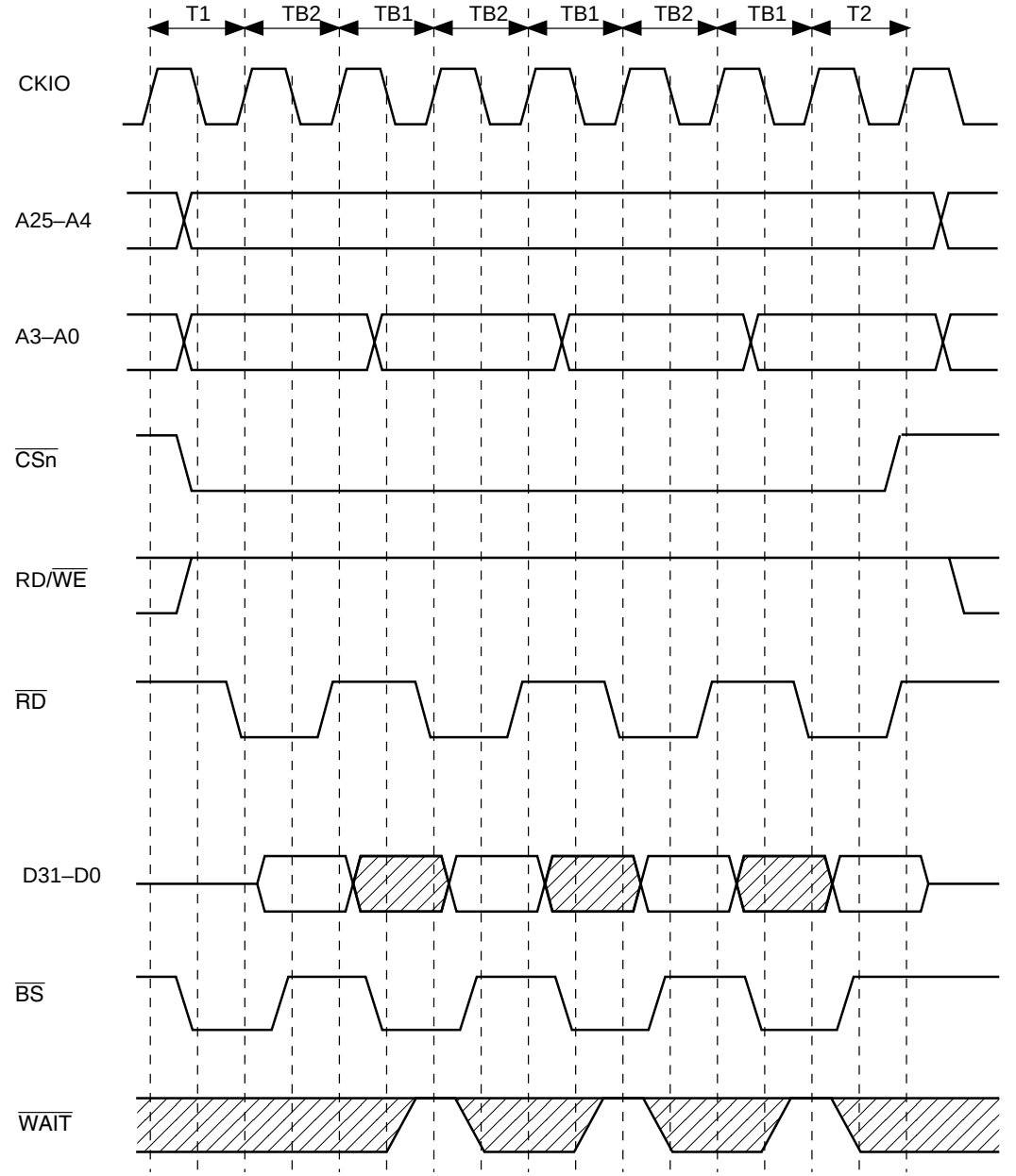


Figure 10.24 Burst ROM Wait Access Timing



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 10.25 Burst ROM Basic Access Timing

10.3.6 PCMCIA Interface

In the SH7707, 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 PCMCIA Rev. 2.1 (JEIDA Ver. 4.2). 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 PCMCIA Rev. 2.1.

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 10.26 shows an example of PC card connection to the SH7707. To enable active insertion of the PC cards (i.e. insertion or removal while system power is being supplied), a 3-state buffer must be connected between the SH7707’s bus interface and the PC cards. For more details on interface to the PC card, refer to section 11, PC Card Controller (PCC).

As operation in big-endian mode is not explicitly stipulated in the PCMCIA Rev. 2.1 (JEIDA Ver. 4.2), the PCMCIA interface for the SH7707 in big-endian mode is stipulated independently.

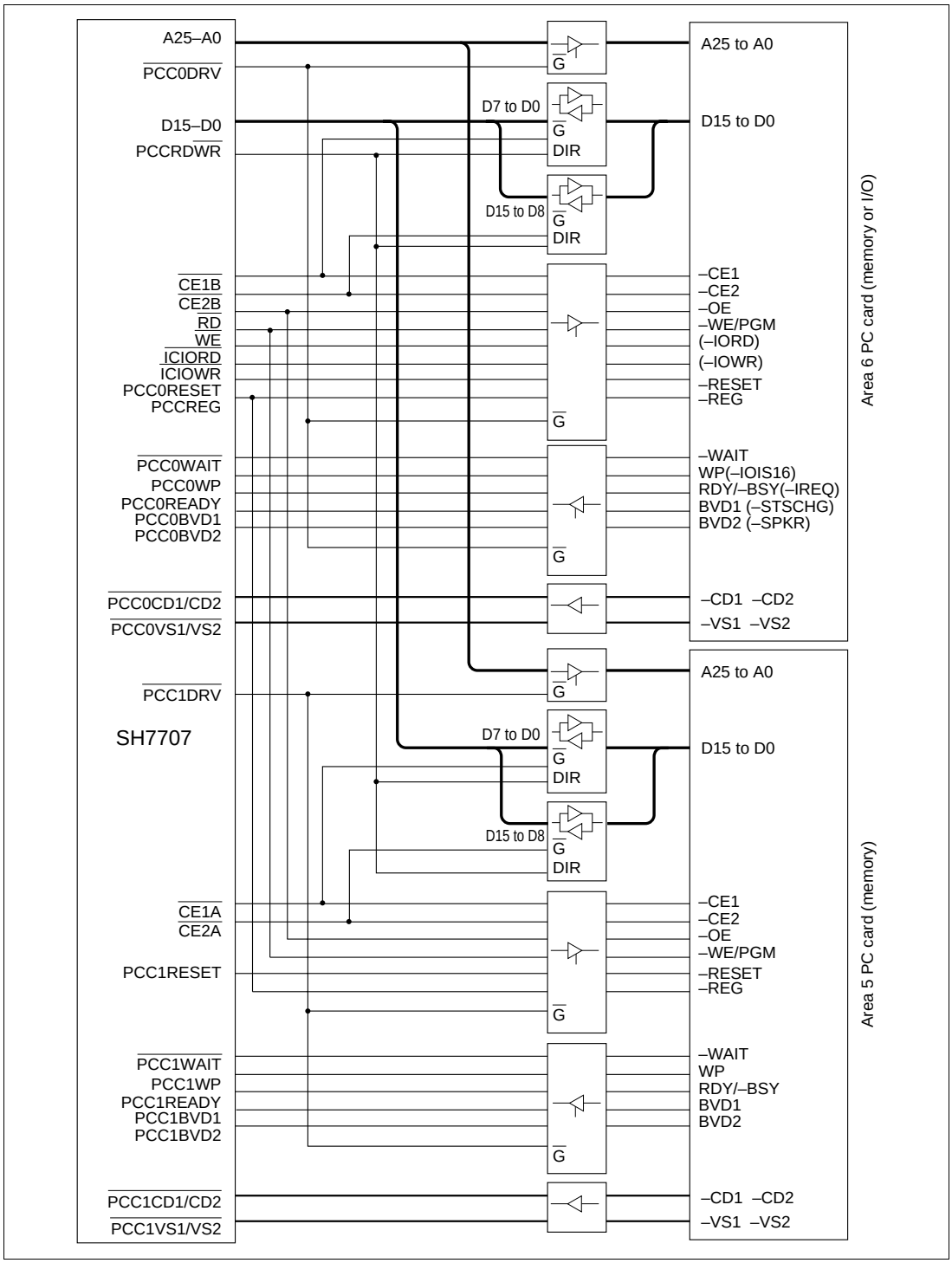


Figure 10.26 Example of PCMCIA Interface

Memory Card Interface Basic Timing: Figure 10.27 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 interface accesses when the common memory 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 signals ($\overline{CS5}$, $\overline{CE2A}$, $\overline{CS6}$, $\overline{CE2B}$), and write data (D15-D0) in a write cycle, become insufficient with respect to $\overline{RD}$ and $\overline{WR}$ (the $\overline{WE1}$ pin in the SH7707). In the SH7707, provision is made 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 $\overline{WAIT}$ pin can be inserted in the same way as for the basic interface. Figure 10.28 shows the PCMCIA memory bus wait timing.

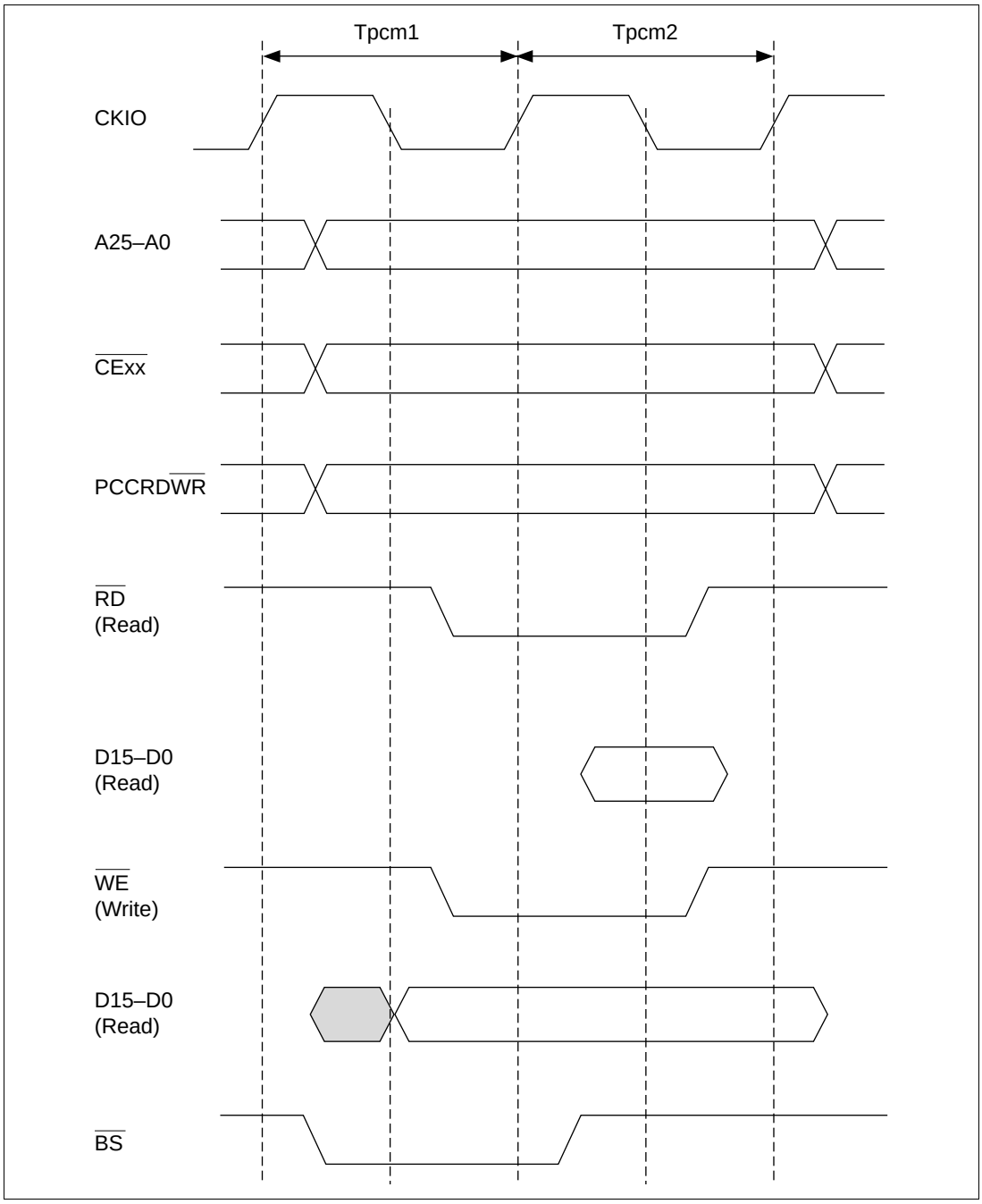


Figure 10.27 Basic Timing for PCMCIA Memory Card Interface

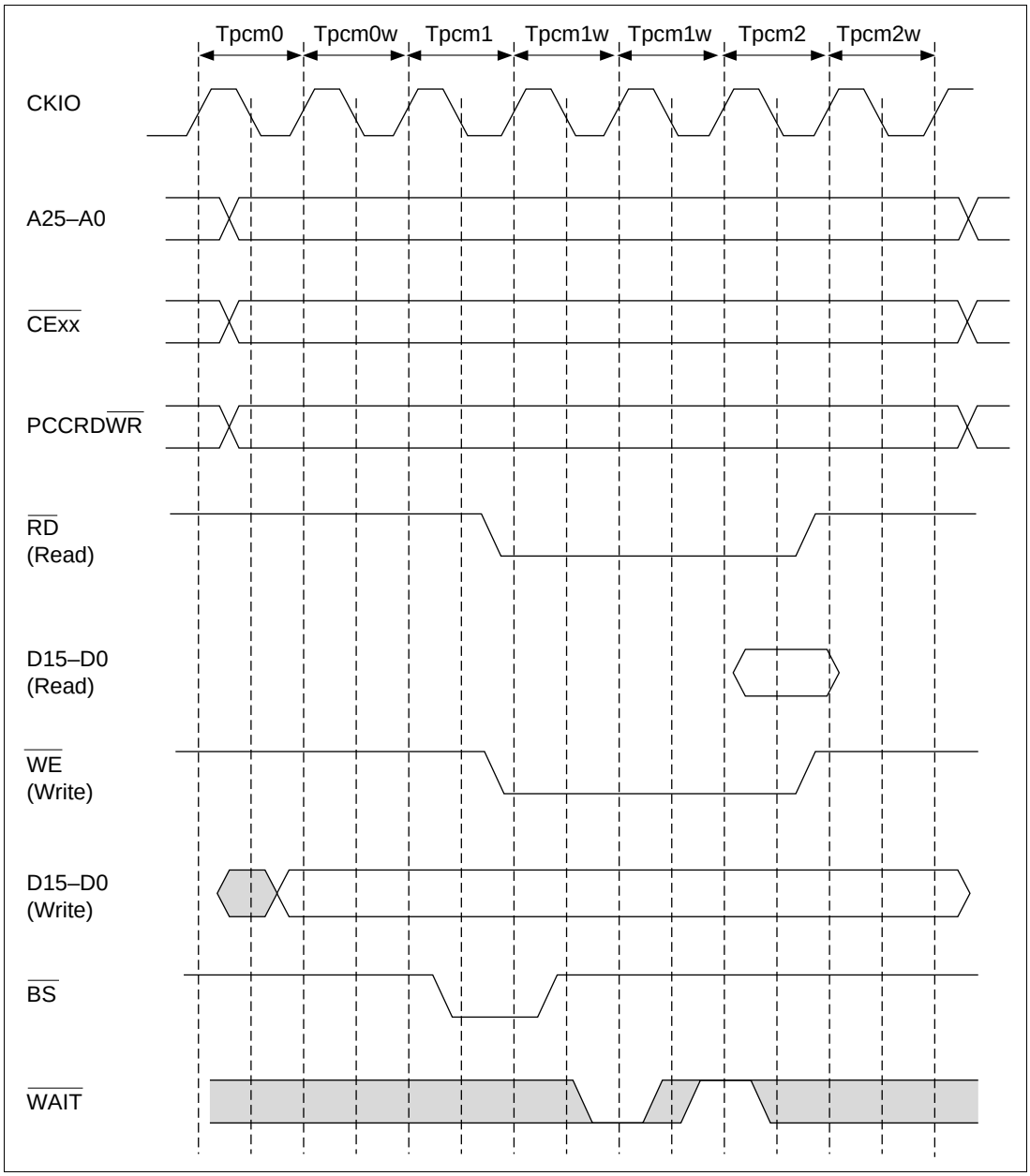


Figure 10.28 Wait Timing for PCMCIA Memory Card Interface

I/O Card Interface Timing: Figures 10.29 and 10.30 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 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 PCC0WP pin. When a 16-bit bus width is set for area 6, if the PCC0WP 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 10.31 shows the basic timing for dynamic bus sizing.

In big-endian mode, the PCC0WP signal is not supported, and is ignored.

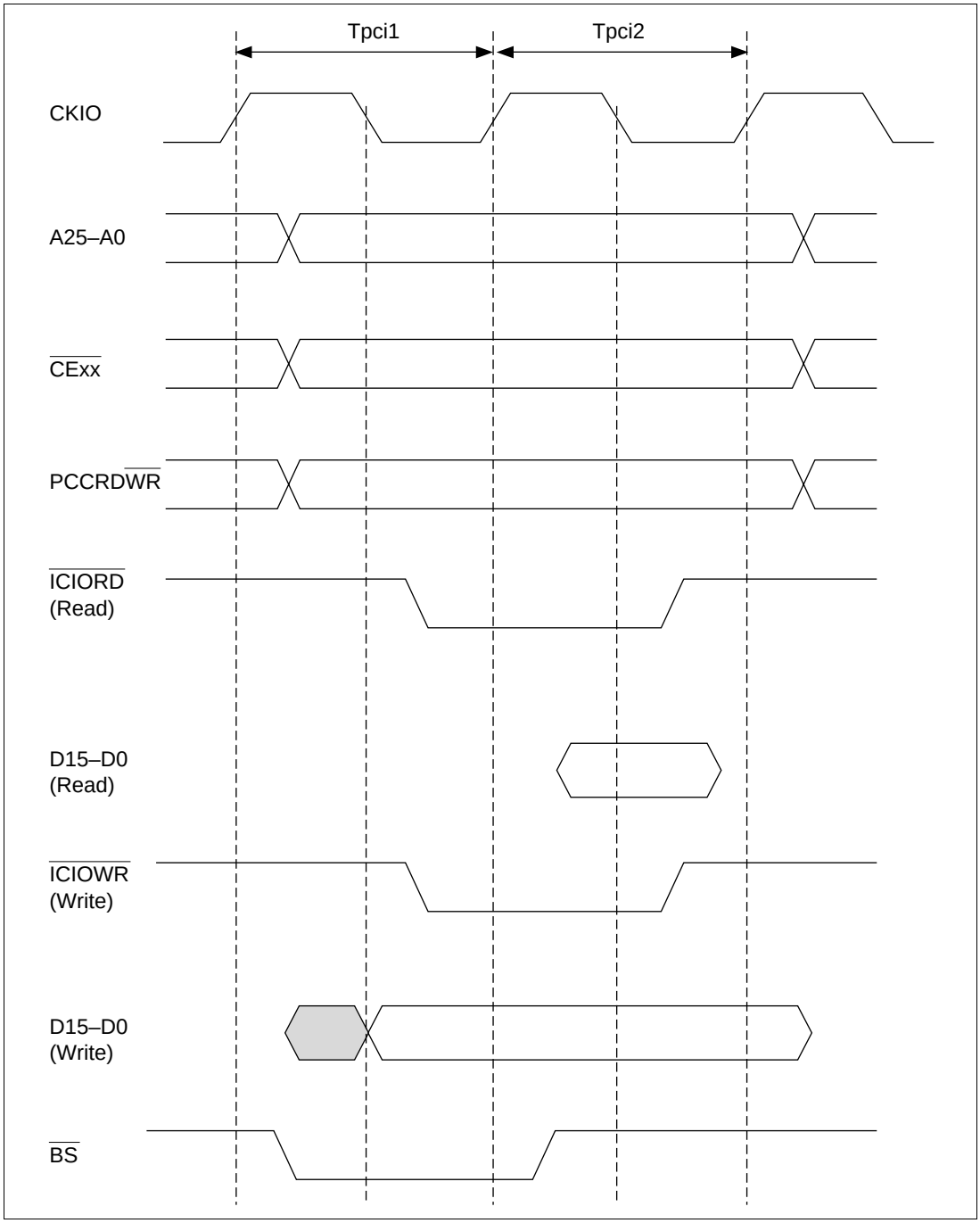


Figure 10.29 Basic Timing for PCMCIA I/O Card Interface

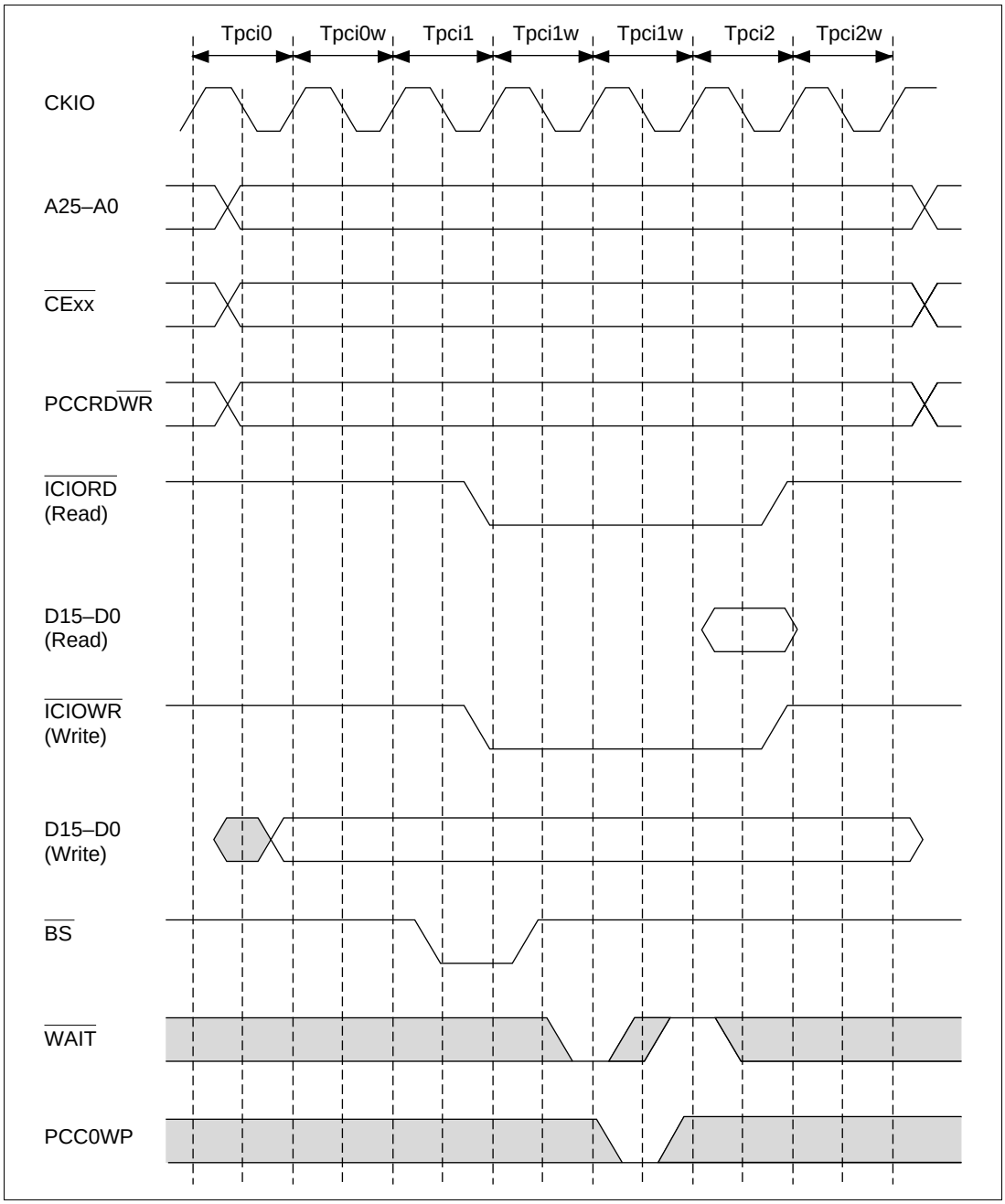


Figure 10.30 Wait Timing for PCMCIA I/O Card Interface

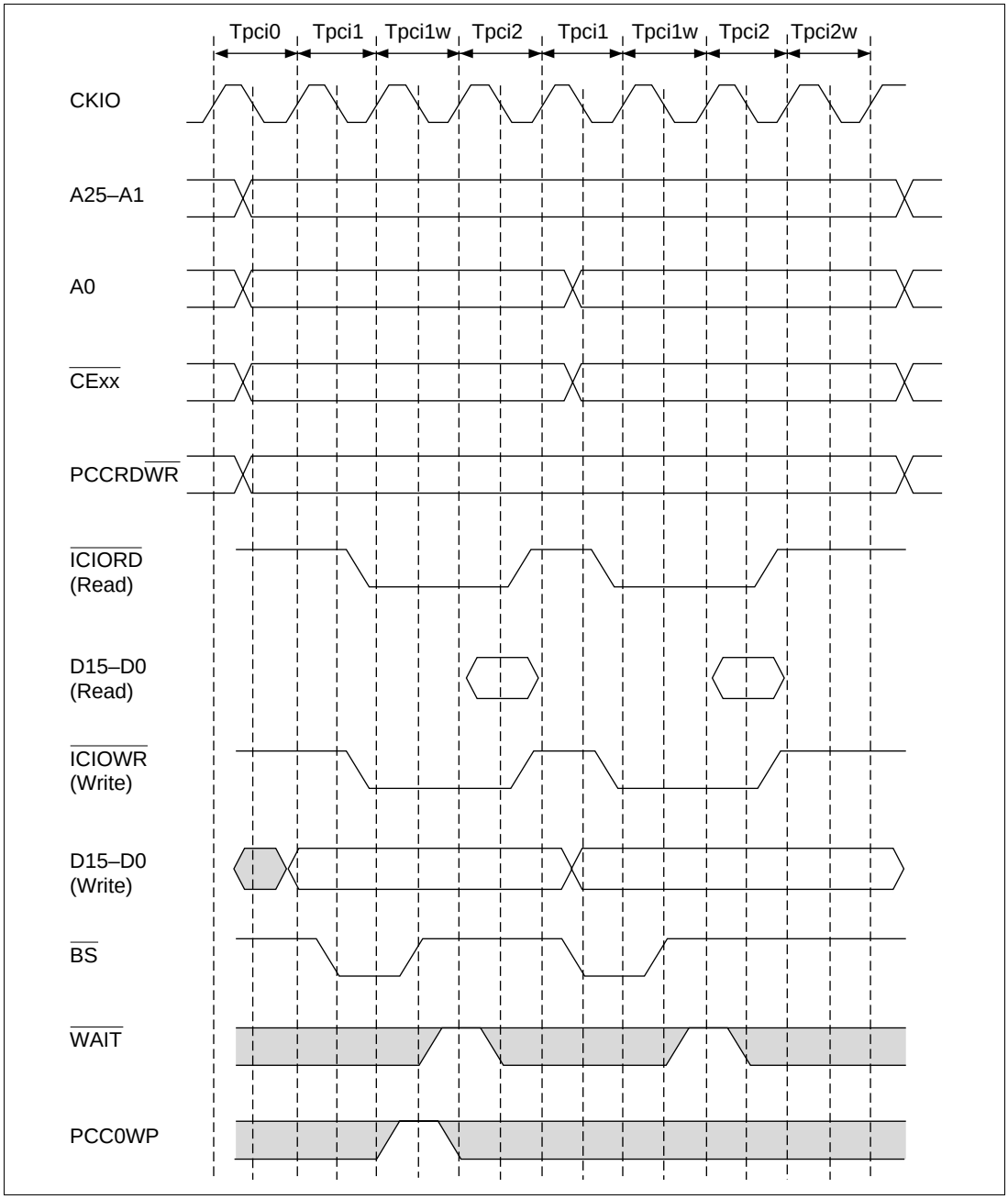


Figure 10.31 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

10.3.7 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 the data in the next access, and so resulting 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, and if there is a possibility of a bus collision when the next access is started, inserts a wait cycle before the access cycle to prevent 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 SH7707. When the SH7707 performs consecutive write cycles, the data transfer direction is fixed (from the SH7707 to other memory) and there is no problem. With read accesses to the same area, also, in principle data is output from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and AnIW0 (n = 0,2-6) 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 SH7707 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 two cycles are inserted for arbitration purposes.

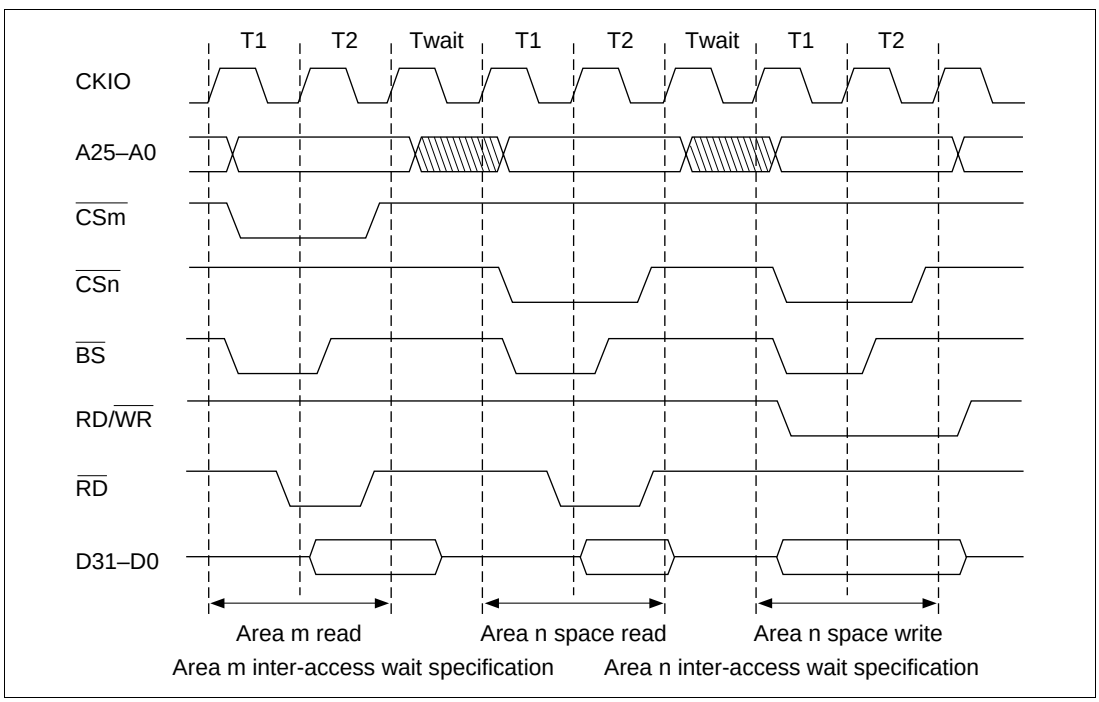


Figure 10.32 Waits between Access Cycles

10.3.8 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is received from an external device, the bus is released after the bus cycle being executed is completed and a bus grant signal ($\overline{\text{BACK}}$) is output. The bus is not released during burst transfers for cache fills. At the negation of $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ is negated and bus use is restarted. See Appendix B, Pin States, for the pin states when the bus is released.

The SH7707 sometimes needs to retrieve a bus it has released. For example, when memory generates a refresh request or an interrupt request internally, the SH7707 must perform the appropriate processing. The SH7707 has a bus request signal ($\overline{\text{IRQOUT}}$) for this purpose. When it must retrieve the bus, it asserts the $\overline{\text{IRQOUT}}$ signal. Devices asserting an external bus release request receive the assertion of the $\overline{\text{IRQOUT}}$ signal and negate the $\overline{\text{BREQ}}$ signal to release the bus. The SH7707 retrieves the bus and carries out its processing.

Section 11 PC Card Controller (PCC)

11.1 Overview

The PC card controller (PCC) controls the external buffer, interrupts, and exclusive ports of the PC card interface to be connected to the SH7707. Using the PCC enables two slots of PC cards that conform to the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard to be easily connected to the SH7707.

11.1.1 Features

The PCC has the following features:

- Two PC card interface slots can be simultaneously controlled.
  - As a PC card interface to be connected to physical area 6, an IC memory card interface and an I/O card interface are supported.
  - As a PC card interface to be connected to physical area 5, an IC memory card interface is supported.
  - Outputs control signals for the external buffer.
  - Supports a preemptive operating system by switching attribute memory, common memory, and I/O space by using addresses.
  - Provides a segment bit (an address bit for the PC card) for common memory, enabling access to a 64-MB space (fully conforming to PCMCIA specifications).

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the PC card controller.

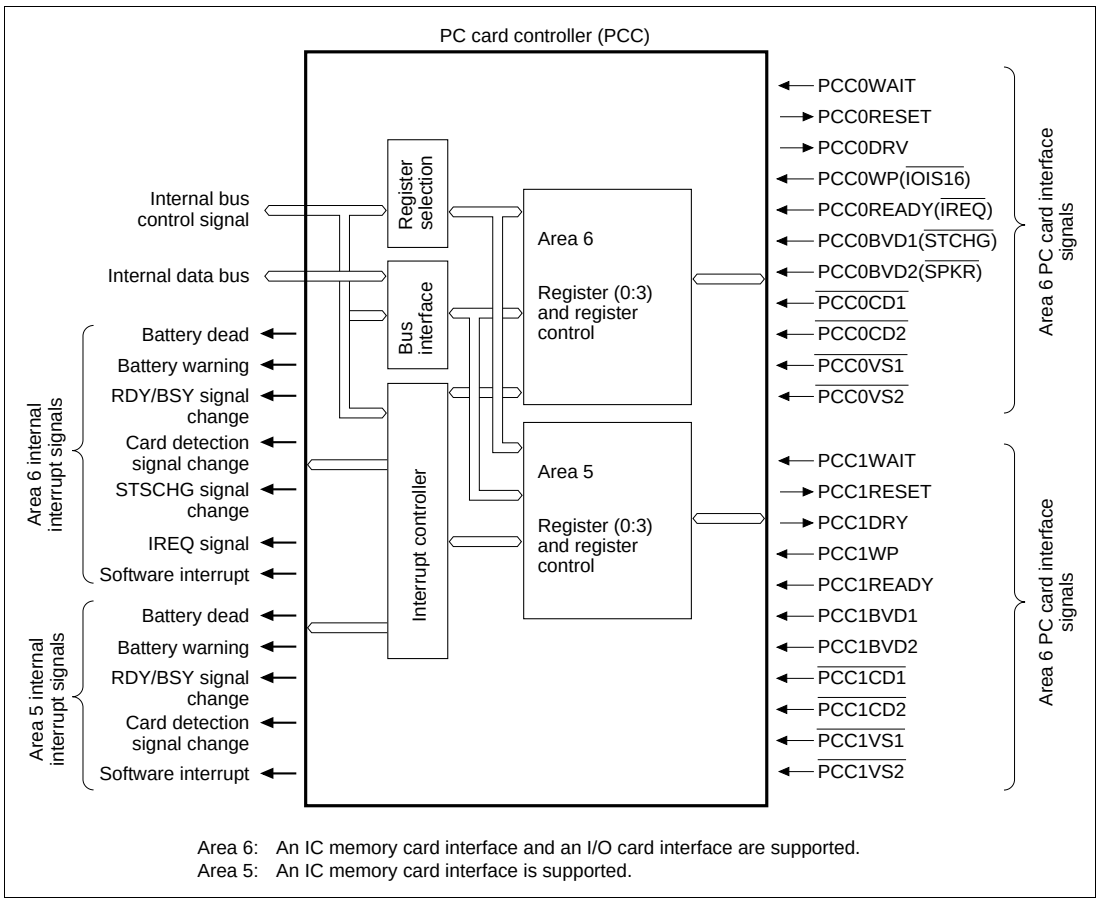


Figure 11.1 PC Card Controller Block Diagram

11.1.3 Register Configuration

Table 11.1 lists the PC card controller registers.

Table 11.1 PC Card Controller Registers

Physical Area	Register Name	Symbol	Read/ Write	Initial Value	Address	Access Size
Physical area 6 (PCC0)	Area 6 interface status register	PCC0ISR	R	*	H'040000E0	8 bits
	Area 6 general control register	PCC0GCR	R/W	H'00	H'040000E2	8 bits
	Area 6 card status change register	PCC0CSCR	R/W	H'00	H'040000E4	8 bits
	Area 6 card status change interrupt enable register	PCC0CSCIER	R/W	H'00	H'040000E6	8 bits
Physical area 5 (PCC1)	Area 5 interface status register	PCC1ISR	R	*	H'040000F0	8 bits
	Area 5 general control register	PCC1GCR	R/W	H'00	H'040000F2	8 bits
	Area 5 card status change register	PCC1CSCR	R/W	H'00	H'040000F4	8 bits
	Area 5 card status change interrupt enable register	PCC1CSCIER	R/W	H'00	H'040000F6	8 bits

Note: Depends on the PC card status.

11.1.4 PCMCIA Support

The SH7707 supports an interface based on PCMCIA specifications for physical areas 5 and 6. Interfaces supported are the IC memory card interface and I/O card interface defined in the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard. Only the IC memory card interface is supported in physical area 5. Both the IC memory card interface and I/O card interface are supported in physical area 6.

Table 11.2 Features of the PCMCIA Interface

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 for I/O bus width* The PCMCIA interface can be accessed from the address-conversion region and non-address-conversion region.

Note: Dynamic bus sizing for the I/O bus width is supported only in little-endian mode.

The SH7707 can directly access 32- and 16-MB physical areas in a 64-MB memory space and an I/O space of the PC card (continuous 32/16-MB area mode). The SH7707 provides a segment bit (an address bit for the PC card) in the general control register for areas 5 and 6 to support a common memory space with full PCMCIA specifications (64 MB).

Continuous 32-MB Area Mode: Setting 0 (initial value) in bit 3 (PxMMOD) of the general control register enables the continuous 32-MB area mode. In this mode, the attribute memory space and I/O memory space are 32 MB and the common memory space is 64 MB. In the common memory space, set 1 in bit 2 (PxPA25) of the general control register to access an address of more than 32 MB. By this operation, 1 is output to pin A25, enabling an address space of more than 32 MB to be accessed. When an address of 32 MB or less is accessed, no setting is required (initial value: 0). This bit does not affect access to attribute memory space or I/O memory space.

Figure 11.2 shows the relationship between the memory space of the SH7707 and the memory and I/O spaces of the PC card in the continuous 32-MB area mode. Although memory space and I/O space are supported in area 6, I/O space is not supported in area 5.

In area 5 or 6, set 1 in bit 0 (PxREG) of the general control register to access the common memory space of the PC card, and set 0 in bit 0 to access the attribute memory space (initial value: 0). By

this operation, the set value is output to pin PCCREG, enabling any space to be accessed. When the I/O space is accessed in area 6, the output of pin PCCREG is always 0 regardless of the value of bit 0 (PxREG).

See the register descriptions in section 11.2 for details of register settings.

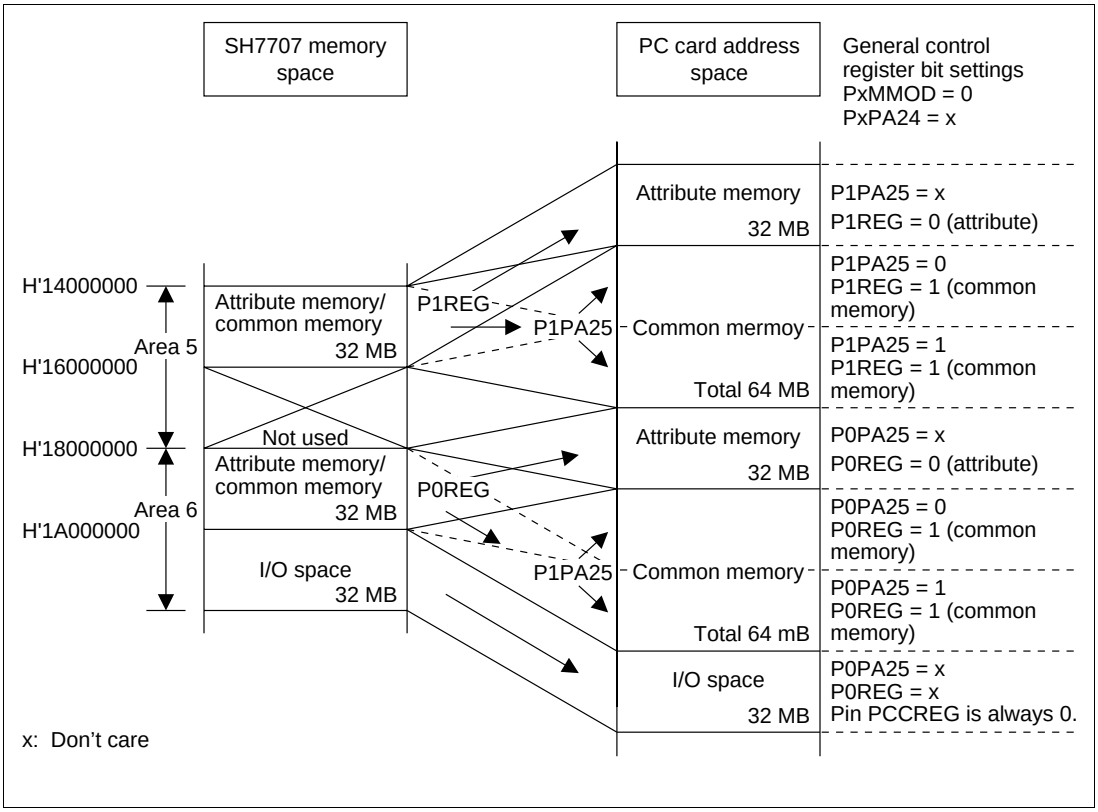


Figure 11.2 Continuous 32-MB Area Mode

Continuous 16-MB Area Mode: Setting 1 in bit 3 (PxMMOD) of the general control register enables the continuous 16-MB area mode. In this mode, the attribute memory space and I/O memory space are 16 MB, and the common memory space is 64 MB. In the common memory space, set the PC card address in bit 2 (PxPA25) and bit 1 (PxPA24) of the general control register to access an address of more than 16 MB. By this operation, values are output to pins A25 and A24, enabling an address space of more than 16 MB to be accessed (initial value: 0 for PxPA25 and PxPA24). When an address of 16 MB or less is accessed, no settings are required. This bit does not affect access to attribute memory space or I/O memory space.

Figures 11.3 and 11.4 show the relationship between the memory space of the SH7707 and the memory and I/O spaces of the PC card in the continuous 16-MB area mode. Although memory space and I/O space are supported in area 6, I/O space is not supported in area 5.

The attribute memory space, common memory space, and I/O space of the PC card are provided as 16-MB physical spaces in this mode. Therefore, the SH7707 automatically controls pin PCCREG (the value of bit 0 (PxREG) in the general control register is ignored). In area 5, the output of pin PCCREG is 0 when the attribute memory space is accessed, and 1 when the common memory space is accessed. In area 6, the output of pin PCCREG is 0 when the attribute memory space or I/O space is accessed, and 1 when the common memory space is accessed.

See the register descriptions in section 11.2 for details of register settings.

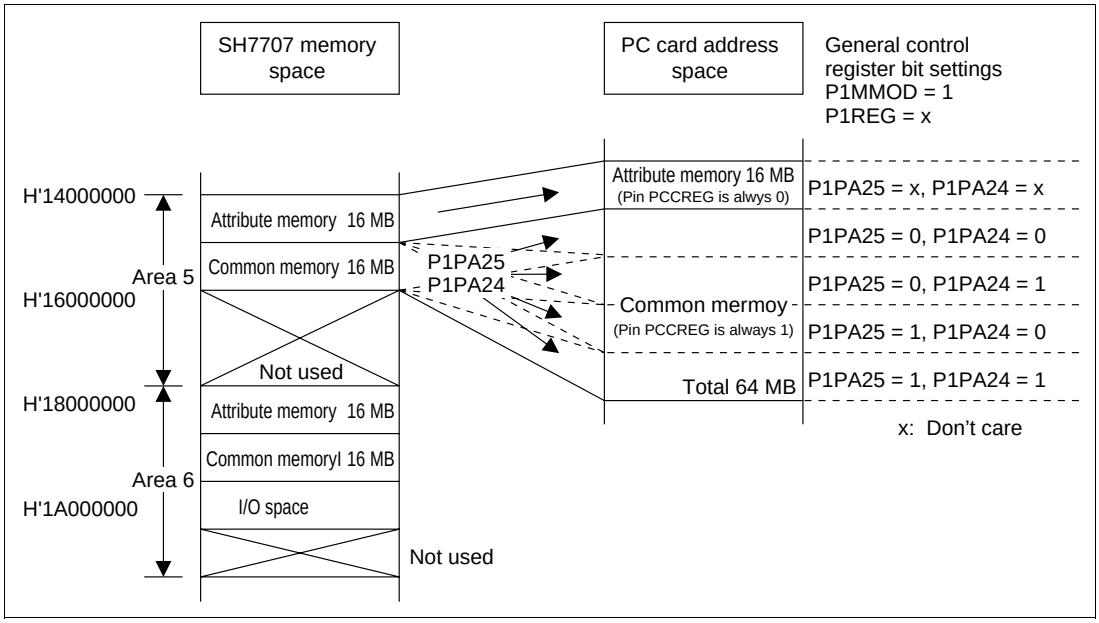


Figure 11.3 Continuous 16-MB Area Mode (Area 5)

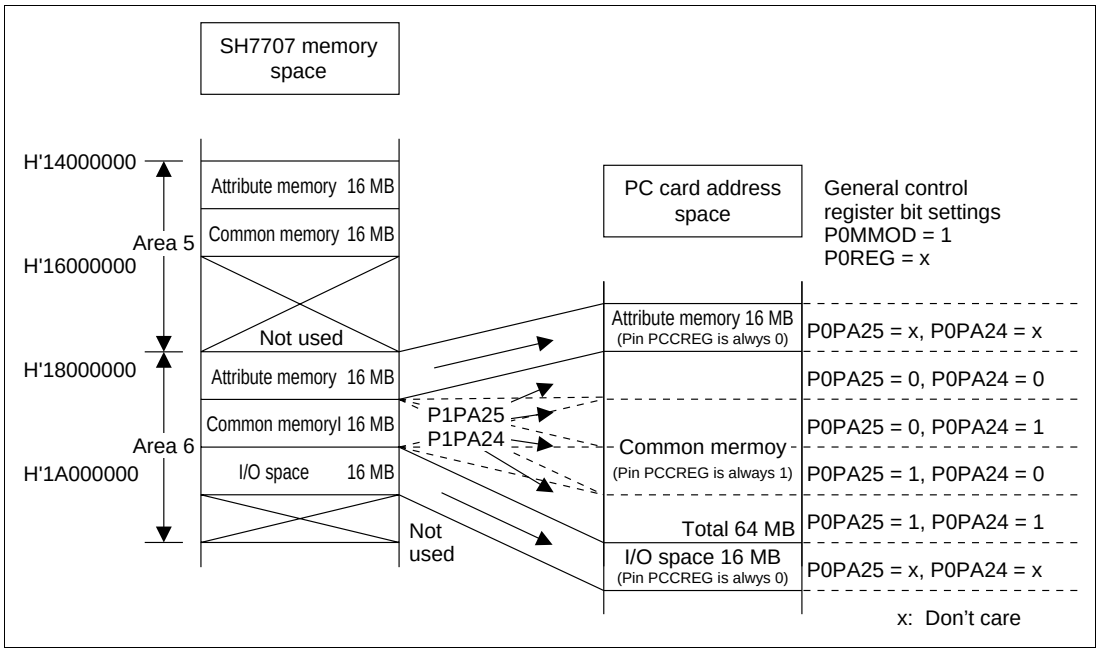


Figure 11.4 Continuous 16-MB Area Mode (Area 6)

11.2 Register Descriptions

11.2.1 Area 6 Interface Status Register (PCC0ISR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0RDY/ IREQ	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2 SPKR	P0BVD1 STSCHG
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: Depends on the PC card status.

The area 6 interface status register (PCC0ISR) is an 8-bit read-only register which is used to read the status of the PC card connected to area 6. The initial value of PCC0ISR depends on the PC card status.

Bit 7—PCC0 Ready (P0RDY/IREQ): The value of pin RDY/BSY of the PC card connected to area 6 is read when the IC memory card interface is connected. The value of pin IREQ of the PC card connected to area 6 is read when the I/O card interface is connected. This bit cannot be written to.

Bit 7: P0RDY/IREQ	Description
0	Indicates that the value of pin RDY/BSY is 0 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of pin IREQ is 0 when the PC card connected to area 6 is the I/O card interface type
1	Indicates that the value of pin RDY/BSY is 1 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of pin IREQ is 1 when the PC card connected to area 6 is the I/O card interface type

Bit 6—PCC0 Write Protect (P0MWP): The value of pin WP of the PC card connected to area 6 is read when the IC memory card interface is connected. 0 is read when the I/O card interface is connected. This bit cannot be written to.

Bit 6: P0MWP	Description
0	Indicates that the value of pin WP is 0 when the PC card connected to area 6 uses the IC memory card interface. The value of bit 6 is always 0 when the PC card connected to area 6 is the I/O card interface type
1	Indicates that the value of pin WP is 1 when the PC card connected to area 6 is the IC memory card interface type

Bit 5—PCC0 Voltage Sense 2 (P0VS2): The value of pin VS2 of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 5: P0VS2	Description
0	The value of pin VS2 of the PC card connected to area 6 is 0
1	The value of pin VS2 of the PC card connected to area 6 is 1

Bit 4—PCC0 Voltage Sense 1 (P0VS1): The value of pin VS1 of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 4: P0VS1	Description
0	The value of pin VS1 of the PC card connected to area 6 is 0
1	The value of pin VS1 of the PC card connected to area 6 is 1

Bit 3—PCC0 Card Detect 2 (P0CD2): The value of pin CD2 of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 3: P0CD2	Description
0	The value of pin CD2 of the PC card connected to area 6 is 0
1	The value of pin CD2 of the PC card connected to area 6 is 1

Bit 2—PCC0 Card Detect 1 (P0CD1): The value of pin CD1 of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 2: P0CD1	Description
0	The value of pin CD1 of the PC card connected to area 6 is 0
1	The value of pin CD1 of the PC card connected to area 6 is 1

Bits 1 and 0—PCC0 Battery Voltage Detect 2 and 1 (P0BVD2, P0BVD1): The values of pins BVD2 and BVD1 of the PC card connected to area 6 are read when the IC memory card interface is connected. The values of pins SPKR and STSCHG of the PC card connected to area 6 are read when the I/O card interface is connected. These bits cannot be written to.

(IC Memory Interface)

Bit 1: P0BVD2	Bit 0: P0BVD1	Description
1	1	The battery voltage of the PC card connected to area 6 is normal (Battery Good)
0	1	The battery must be changed although data is guaranteed for the PC card connected to area 6 (Battery Warning)
1	0	The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead)
0	0	The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead)

(I/O Card Interface)

Bit 1: SPKR	Description
0	The value of pin STSCHG of the PC card connected to area 6 is 0
1	The value of pin STSCHG of the PC card connected to area 6 is 1

Bit 0: STSCHG	Description
0	The value of pin SPKR of the PC card connected to area 6 is 0
1	The value of pin SPKR of the PC card connected to area 6 is 1

11.2.2 Area 6 General Control Register (PCC0GCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0DRVE	P0PCCR	P0PCCT	—	P0MMOD	P0PA25	P0PA24	P0REG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

The area 6 general control register (PCC0GCR) is an 8-bit readable/writable register which controls the external buffer, resets, address pins A25 and A24, and pin REG, and sets the PC card type for the PC card connected to area 6. PCC0GCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Buffer Control (P0DRVE): Controls the external buffer for the PC card connected to area 6.

Bit 7: P0DRVE	Description
0	High-level setting for control pin PCC0DRV of the external buffer for the PC card connected to area 6 (Initial value)
1	Low-level setting for control pin PCC0DRV of the external buffer for the PC card connected to area 6

Bit 6—PCC0 Card Reset (P0PCCR): Controls resets for the PC card connected to area 6.

Bit 6: P0PCCR	Description
0	Low-level setting for reset pin PCC0RESET for the PC card connected to area 6 (Initial value)
1	High-level setting for reset pin PCC0RESET for the PC card connected to area 6

Bit 5—PCC0 Card Type (P0PCCT): Specifies the type of the PC card connected to area 6. Cleared to 0 when the PC card is the IC memory card interface type; set to 1 when the PC card is the I/O card interface type.

Bit 5: P0PCCT	Description
0	The PC card connected to area 6 is handled as the IC memory card interface type (Initial value)
1	The PC card connected to area 6 is handled as the I/O card interface type

Bit 4—Reserved: Always reads 0. The write value should always be 0.

Bit 3—PCC0 Mode (P0MMOD): Controls pins PCCREG and A24 for the PC card connected to area 6. Specifies either A24 of the address to be accessed or bit P0REG for outputting to pin PCCREG. When the common memory space is accessed, specifies either A24 of the address to be accessed or bit P0PA24 for outputting to pin A24. By this operation, continuous 32 or 16 Mbytes can be selected for the address area of the common memory space of the PC card.

Bit 3: P0MMOD	Description
0	Bit P0REG is output to pin PCCREG, and A24 of address to be accessed is output to pin A24 (continuous 32-MB area mode) (Initial value)
1	A24 of address to be accessed is output to pin PCCREG. When the common memory space is accessed, P0PA24 is output to pin A24 (continuous 16-MB area mode)

Bit 2—PC Card Address (P0PA25): Controls pin A25 for the PC card connected to area 6. When the common memory space is accessed for the PC card connected to area 6, this bit is output to pin A25. When the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 2: P0PA25	Description
0	When the common memory space is accessed for the PC card connected to area 6, 0 is output to pin A25 (Initial value)
1	When the common memory space is accessed for the PC card connected to area 6, 1 is output to pin A25

Bit 1—PC Card Address (P0PA24): Controls pin A24 for the PC card connected to area 6. When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, this bit is output to pin A24. When bit P0MMOD is 0 or the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 1: P0PA24	Description
0	When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 0 is output to pin A24 (Initial value)
1	When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 1 is output to pin A24

Bit 0—PCC0REG Space Indication (P0REG): Controls pin PCCREG for the PC card connected to area 6. When bit P0MMOD is 0, this bit is output to pin PCCREG for the PC card connected to area 6. When bit P0MMOD is 1 or the I/O card interface is accessed, this bit is meaningless.

Bit 0: P0REG	Description
0	When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 0 is output to pin PCCREG (Initial value)
1	When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 1 is output to pin PCCREG

11.2.3 Area 6 Card Status Change Register (PCC0CSCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0SCDI	—	IREQ	SC	P0CDC	P0RC	P0BW	P0BD
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

The area 6 card status change register (PCC0CSCR) is an 8-bit readable/writable register. PCC0CSCR bits are set to 1 by interrupt sources of the PC card connected to area 6 (only bit 7 can be set to 1 as required). PCC0CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Software Card Detect Change Interrupt (P0SCDI): A software card detect change interrupt can be generated by writing 1 to this bit. When this bit is set to 1, the same interrupt as the card detect change interrupt (bit 3 set status) occurs if bit 3 (card detection enable) in the area 6 card status change interrupt enable register (PCC0CSCIER) is set to 1. If bit 3 is cleared to 0, no interrupt occurs.

Bit 7: P0SCDI	Description
0	No software card detection interrupt occurs for the PC card connected to area 6 (Initial value)
1	Software card detection interrupt occurs for the PC card connected to area 6

Bit 6—Reserved: Always reads 0. The write value should always be 0.

Bit 5—IREQ Request (IREQ): Indicates the interrupt request for the IREQ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. The IREQ bit is set to 1 when an interrupt request signal in pulse mode or level mode is input to the IREQ pin. The mode is selected by bits 5 and 6 (IREQE interrupt enable bits) in the area 6 card status change interrupt enable register (PCC0CSCIER). This bit can be cleared to 0 only in pulse mode. Write 0 to bit 5 to clear the bit to 0. This bit is not changed if 1 is written. In level mode, bit 5 is a read-only bit which reflects the IREQ pin state (if the IREQ pin is low, 1 is read). This bit always reads 0 on the IC memory card interface.

Bit 5: IREQ	Description
0	No interrupt request on the IREQ pin of the PC card when the PC card is on the I/O card interface (Initial value)
1	An interrupt request on the IREQ pin of the PC card has occurred when the PC card is on the I/O card interface

Bit 4—PCC0 Status Change (SC): Indicates a change in the value of the STSCHG pin of the PC card when the PC card connected to area 6 is the I/O card interface type. When the STSCHG pin is changed from 1 to 0, the SC bit is set to 1. When STSCHG is not changed, the SC bit remains at 0. Write 0 to bit 4 when this bit is set to 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the IC memory card interface.

Bit 4: SC	Description
0	STSCHG of the PC card is not changed when the PC card is on the I/O card interface (Initial value)
1	STSCHG of the PC card is changed from 1 to 0 when the PC card is on the I/O card interface

Bit 3—PCC0 Card Detect Change (P0CDC): Indicates a change in the value of the CD1 and CD2 pins in the PC card connected to area 6. When the CD1 and CD2 values are changed, the P0CDC bit is set to 1. When the values are not changed, the P0CDC bit remains at 0. Write 0 to bit 3 in order to clear this bit to 0. This bit is not changed if 1 is written.

Bit 3: P0CDC	Description
0	CD1 and CD2 in the PC card are not changed (Initial value)
1	CD1 and CD2 in the PC card are changed

Bit 2—PCC0 Ready Change (P0RC): Indicates a change in the value of the RDY/BSY pin of the PC card when the PC card connected to area 6 is the IC memory card interface type. When the RDY/BSY pin is changed from 0 to 1, the P0RC bit is set to 1. When the RDY/BSY pin is not changed, the P0RC bit remains at 0. Write 0 to bit 2 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 2: P0RC	Description
0	RDY/BSY in the PC card is not changed when the PC card is on the IC memory card interface (Initial value)
1	RDY/BSY in the PC card is changed from 0 to 1 when the PC card is on the IC memory card interface

Bit 1—PCC0 Battery Warning (P0BW): Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which “the battery must be changed although the data is guaranteed” when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 0 and 1, respectively, the P0BW bit is set to 1; in other cases, the P0BW bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 1: P0BW	Description
0	BVD2 and BVD1 of the PC card are not in the battery warning state when the PC card is in the IC memory card interface (Initial value)
1	BVD2 and BVD1 of the PC card are in the battery warning state and “the battery must be changed although the data is guaranteed” when the PC card is on the IC memory card interface

Bit 0—PCC0 Battery Dead (P0BD): Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which “the battery must be changed since the data is not guaranteed” when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 1 and 0 or 0 and 0, the P0BD bit is set to 1; in other cases, the P0BD bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 0 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 0: P0BD	Description
0	BVD2 and BVD1 of the PC card are not in the state in which “the battery must be changed since the data is not guaranteed” when the PC card is on the IC memory card interface (Initial value)
1	BVD2 and BVD1 of the PC card are in the state in which “the battery must be changed since the data is not guaranteed” when the PC card is on the IC memory card interface

11.2.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0CRE	IREQE1	IREQE0	SCE	P0CDE	P0RE	P0BWE	P0BDE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The area 6 card status change interrupt enable register (PCC0CSCIER) is an 8-bit readable/writable register. PCC0CSCIER enables or disables interrupt sources for the PC card connected to area 6. When a PCC0CSCIER is set to 1, the corresponding interrupt is enabled, and when the bit is cleared to 0, the interrupt is disabled. PCC0CSCIER is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Card Reset Enable (P0CRE): When this bit is set to 1, and when the CD1 and CD2 pins detect that a PC card is connected to area 6, the area 6 general control register (PCC0GCR) is initialized.

Bit 7: P0CRE	Description
0	The area 6 general control register (PCC0GCR) is not initialized even if a PC card is detected in area 6 (Initial value)
1	The area 6 general control register (PCC0GCR) is initialized when a PC card is detected connected to area 6

Bits 6 and 5—PCC0IREQ Request Enable (IREQE1, IREQE0): These bits enable or disable IREQ pin interrupt requests and select the interrupt mode when the PC card connected to area 6 is the I/O card interface type. Note that bit 5 (IREQ) in the status change register (PCC0CSCR) is cleared if the values in bits 6 and 5 in this register are changed. These bits have no meaning on the IC memory card interface.

Bit 6: IREQE1	Bit 5: IREQE0	Description
0	0	IREQ requests are not accepted for the PC card connected to area 6. Bit 5 in the status change register (PCC0CSCR) functions as a read-only bit that indicates the inverse of the IREQ pin signal. (Initial value)
0	1	The level-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In level mode, an interrupt occurs when level 0 of the signal input from the IREQ pin is detected.
1	0	The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the IREQ pin is detected.
1	1	The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the IREQ pin is detected.

Bit 4—PCC0 Status Change Enable (P0SCE): When the PC card connected to area 6 is on the I/O card interface, bit 4 enables or disables the interrupt request when the value of the $\overline{\text{BVD1}}$ pin (STSCHG) is changed. This bit has no meaning in the IC memory card interface.

Bit 4: P0SCE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the value of the $\overline{\text{BVD1}}$ pin (STSCHG) (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the value of the $\overline{\text{BVD1}}$ pin (STSCHG) is changed from 1 to 0

Bit 3—PCC0 Card Detect Change Enable (P0CDE): Bit 3 enables or disables the interrupt when the values of the CD1 and CD2 pins are changed.

Bit 3: P0CDE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the values of the CD1 and CD2 pins (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the values of the CD1 and CD2 pins are changed

Bit 2—PCC0 Ready Change Enable (P0RE): When the PC card connected to area 6 is on the IC memory card interface, bit 2 enables or disables the interrupt when the value of the RDY/BSY pin is changed. This bit has no meaning on the I/O card interface.

Bit 2: P0RE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the value of the RDY/BSY pin (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the value of the RDY/BSY pin is changed from 0 to 1

Bit 1—PCC0 Battery Warning Enable (P0BWE): When the PC card connected to area 6 is on the IC memory card interface, bit 1 enables or disables the interrupt when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”. This bit has no meaning on the I/O card interface.

Bit 1: P0BWE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed” (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”

Bit 0—PCC0 Battery Dead Enable (P0BDE): When the PC card connected to area 6 is on the IC memory card interface, bit 0 enables or disables the interrupt when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed”. This bit has no meaning on the I/O card interface.

Bit 0: P0BDE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed” (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed”

11.2.5 Area 5 Interface Status Register (PCC1ISR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P1RDY	P1MWP	P1VS2	P1VS1	P1CD2	P1CD1	P1BVD2	P1BVD1
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: Always reflects the PC card status.

The area 5 interface status register (PCC1ISR) is an 8-bit read-only register which is used to read the status of the PC card connected to area 5. PCC1ISR always reflects the PC card status.

Bit 7—PCC1 Ready (P1RDY): The value of the RDY/BSY pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 7: P1RDY	Description
0	The value of pin RDY/BSY of the PC card connected to area 5 is 0
1	The value of pin RDY/BSY of the PC card connected to area 5 is 1

Bit 6—PCC1 Write Protect (P1MWP): The value of the WP pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 6: P1MWP	Description
0	The value of WP pin of the PC card connected to area 5 is 0
1	The value of WP pin of the PC card connected to area 5 is 1

Bit 5—PCC1 Voltage Sense 2 (P1VS2): The value of the VS2 pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 5: P1VS2	Description
0	The value of pin VS2 of the PC card connected to area 5 is 0
1	The value of pin VS2 of the PC card connected to area 5 is 1

Bit 4—PCC1 Voltage Sense 1 (P1VS1): The value of the VS1 pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 4: P1VS1	Description
0	The value of pin VS1 of the PC card connected to area 5 is 0
1	The value of pin VS1 of the PC card connected to area 5 is 1

Bit 3—PCC1 Card Detect 2 (P1CD2): The value of the CD2 pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 3: P1CD2	Description
0	The value of pin CD2 of the PC card connected to area 5 is 0
1	The value of pin CD2 of the PC card connected to area 5 is 1

Bit 2—PCC1 Card Detect 1 (P1CD1): The value of the CD1 pin of the PC card connected to area 5 is read. This bit cannot be written to.

Bit 2: P1CD1	Description
0	The value of pin CD1 of the PC card connected to area 5 is 0
1	The value of pin CD1 of the PC card connected to area 5 is 1

Bits 1 and 0—PCC1 Battery Coltage Detect 2 and 1 (P1BVD2, P1BVD1): Read the values of the BVD2 and BVD1 pins of the PC card connected to area 5. This bit cannot be written to.

Bit 1: P1BVD2	Bit 0: P1BVD1	Description
0	0	The battery voltage for the PC card connected to area 5 is abnormal and the data is not guaranteed (Battery Dead)
0	1	Indicates that the battery must be changed although the data is guaranteed for the PC card connected to area 5 (Battery Warning)
1	0	The battery voltage for the PC card connected to area 5 is abnormal and the data is not guaranteed (Battery Dead)
1	1	The battery voltage for the PC card connected to area 5 is normal (Battery Good)

11.2.6 Area 5 General Control Register (PCC1GCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P1DRVE	P1PCCR	—	—	P1MMOD	P1PA25	P1PA24	P1REG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	—	—	R/W	R/W	R/W	R/W

The area 5 general control register (PCC1GCR) is an 8-bit readable/writable register which controls the external buffer, resets, address pins A25 and A24, and pin REG, and sets the PC card type for the PC card connected to area 5. PCC1GCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC1 Buffer Control (P1DRVE): Controls the external buffer for the PC card connected to area 5.

Bit 7: P1DRVE	Description
0	High-level setting for control pin PCC1DRV of the external buffer for the PC card connected to area 5 (Initial value)
1	Low-level setting for control pin PCC1DRV of the external buffer for the PC card connected to area 5

Bit 6—PCC1 Card Reset (P1PCCR): Controls resets for the PC card connected to area 5.

Bit 6: P1PCCR	Description
0	Low-level setting for reset pin PCC1RESET for the PC card connected to area 5 (Initial value)
1	High-level setting for reset pin PCC1RESET for the PC card connected to area 5

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

Bit 3—PCC1 Mode (P1MMOD): Controls pins PCCREG and A24 for the PC card connected to area 5. Specifies either A24 of the address to be accessed or bit P1REG for outputting to pin PCCREG. When the common memory space is accessed, specifies either A24 of the address to be accessed or bit P1PA24 for outputting to pin A24. By this operation, continuous 32 or 16 Mbytes can be selected for the address area of the common memory space of the PC card.

Bit 3: P1MMOD	Description
0	Bit P1REG is output to pin PCCREG, and A24 of address to be accessed is output to pin A24 (continuous 32-MB area mode) (Initial value)
1	A24 of address to be accessed is output to pin PCCREG. When the common memory space is accessed, P1PA24 is output to pin A24 (continuous 16-MB area mode)

Bit 2—PC Card Address (P1PA25): Controls pin A25 for the PC card connected to area 5. When the common memory space is accessed for the PC card connected to area 5, this bit is output to pin A25. When the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 2: P1PA25	Description
0	When the common memory space is accessed for the PC card connected to area 5, 0 is output to pin A25 (Initial value)
1	When the common memory space is accessed for the PC card connected to area 5, 1 is output to pin A25

Bit 1—PC Card Address (P1PA24): Controls pin A24 for the PC card connected to area 5. When bit P1MMOD is 1 and the common memory space is accessed for the PC card connected to area 5, this bit is output to pin A24. When bit P1MMOD is 0 or the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 1: P1PA24	Description
0	When bit P1MMOD is 1 and the common memory space is accessed for the PC card connected to area 5, 0 is output to pin A24 (Initial value)
1	When bit P1MMOD is 1 and the common memory space is accessed for the PC card connected to area 5, 1 is output to pin A24

Bit 0—PCC1REG Space Indication (P1REG): Controls pin PCCREG for the PC card connected to area 5. When bit P1MMOD is 0, this bit is output to pin PCCREG for the PC card connected to area 5. When bit P1MMOD is 1 or the I/O card interface is accessed, this bit is meaningless.

Bit 0: P1REG	Description
0	When bit P1MMOD is 0 and the PC card connected to area 5 is accessed, 0 is output to pin PCCREG (Initial value)
1	When bit P1MMOD is 0 and the PC card connected to area 5 is accessed, 1 is output to pin PCCREG

11.2.7 Area 5 Card Status Change Register (PCC1CSCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P1SCDI	—	—	—	P1CDC	P1RC	P1BW	P1BD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	—	—	—	R/W	R/W	R/W	R/W

The area 5 card status change register (PCC1CSCR) is an 8-bit readable/writable register. PCC1CSCR bits are set to 1 by interrupt sources of the PC card connected to area 5 (only bit 7 can be set to 1 as required). PCC1CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC1 Software Card Detect Change Interrupt (P1SCDI): A software card detect change interrupt can be generated by writing 1 to this bit. When this bit is set to 1, the same interrupt as the card detect change interrupt (bit 3 set status) occurs if bit 3 (card detection enable) in area 5 card status change interrupt enable register (PCC1CSCIER) is set to 1. If bit 3 is cleared to 0, no interrupt occurs.

Bit 7: P1SCDI	Description
0	No software card detection interrupt occurs for the PC card connected to area 5 (Initial value)
1	Software card detection interrupt occurs for the PC card connected to area 5

Bits 6 to 4—Reserved: These bits always read 0. The write value should always be 0.

Bit 3—PCC1 Card Detect Change (P1CDC): Indicates a change in the value of the CD1 and CD2 pins in the PC card connected to area 5. When the CD1 and CD2 values are changed, the P1CDC bit is set to 1. When the values are not changed, the P1CDC bit remains at 0. Write 0 to bit 3 in order to clear this bit to 0. This bit is not changed if 1 is written.

Bit 3: P1CDC	Description	
0	CD1 and CD2 in the PC card are not changed	(Initial value)
1	CD1 and CD2 in the PC card are changed	

Bit 2—PCC1 Ready Change (P1RC): Indicates a change in the value for the RDY/BSY pin when the PC card connected to area 5 is in the IC memory card interface type. When the RDY/BSY pin is changed from 0 to 1, the P1RC bit is set to 1. When the RDY/BSY pin is not changed, the P1RC bit remains at 0. Write 0 to bit 2 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 2: P1RC	Description	
0	RDY/BSY in the PC card is not changed	(Initial value)
1	RDY/BSY in the PC card is changed from 0 to 1	

Bit 1—PCC1 Battery Warning (P1BW): Indicates whether the BVD2 and BVD1 pins in the PC card connected to area 5 are in the state in which “the battery must be changed although the data is guaranteed”. When the BVD2 and BVD1 pins are 0 and 1, respectively, the P1BW bit is set to 1; in other cases, the P1BW bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 1 in order to clear this bit to 0. This bit is not changed if 1 is written.

Bit 1: P1BW	Description
0	BVD1 and BVD2 in the PC card are not in the battery warning state (Initial value)
1	BVD1 and BVD2 in the PC card are in the battery warning state and “the battery must be changed although the data is guaranteed”

Bit 0—PCC1 Battery Dead (P1BD): Indicates whether the BVD2 and BVD1 pins in the PC card are in the state in which “the battery must be changed since the data is not guaranteed”. When the BVD2 and BVD1 pins are 1 and 0 or 0 and 0, the P1BD bit is set to 1; in other cases, the P1BD bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 0 in order to clear this bit to 0. This bit is not changed if 1 is written.

Bit 0: P1BD	Description
0	BVD2 and BVD1 in the PC card are not in the state in which “the battery must be changed since the data is not guaranteed” (Initial value)
1	BVD2 and BVD1 in the PC card are in the state in which “the battery must be changed since the data is not guaranteed”

11.2.8 Area 6 Card Status Change Interrupt Enable Register (PCC1CSCIER)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P1CRE	—	—	—	P1CDE	P1RE	P1BWE	P1BDE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	—	—	—	R/W	R/W	R/W	R/W

The area 5 card status change interrupt enable register (PCC1CSCIER) is an 8-bit readable/writable register. PCC1CSCIER enables or disables interrupt sources for the PC card connected to area 5. When a PCC1CSCIER bit is set to 1, the corresponding interrupt is enabled, and when it is cleared to 0, the interrupt is disabled. PCC1CSCIER is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC1 Card Reset Enable (P1CRE): When this bit is set to 1, and when the CD1 and CD2 pins detect that a PC card is connected to area 5, the area 5 general control register (PCC1GCR) is initialized.

Bit 7: P1CRE	Description
0	The area 5 general control register (PCC1GCR) is not initialized even if a PC card is detected in area 5 (Initial value)
1	The area 5 general control register (PCC1GCR) is initialized when a PC card is detected connected to area 5

Bits 6 to 4—Reserved: These bits always read 0. The write value should always be 0.

Bit 3—PCC1 Card Detect Change Enable (P1CDE): Enables or disables the interrupt when the values of the CD1 and CD2 pins are changed.

Bit 3: P1CDE	Description
0	No interrupt occurs for the PC card connected to area 5 regardless of the values of the CD1 and CD2 pins (Initial value)
1	An interrupt occurs for the PC card connected to area 5 when the values of the CD1 and CD2 pins are changed

Bit 2—PCC1 Ready Change Enable (P1RE): When the PC card connected to area 5 is on the IC memory card interface, bit 2 enables or disables the interrupt when the value of the RDY/BSY pin is changed.

Bit 2: P1RE	Description
0	No interrupt occurs for the PC card connected to area 5 regardless of the value of the RDY/BSY pin (Initial value)
1	An interrupt occurs for the PC card connected to area 5 when the value of the RDY/BSY pin is changed from 0 to 1

Bit 1—PCC1 Battery Warning Enable (P1BWE): When the PC card connected to area 5 is on the IC memory card interface, bit 1 enables or disables the interrupt when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”.

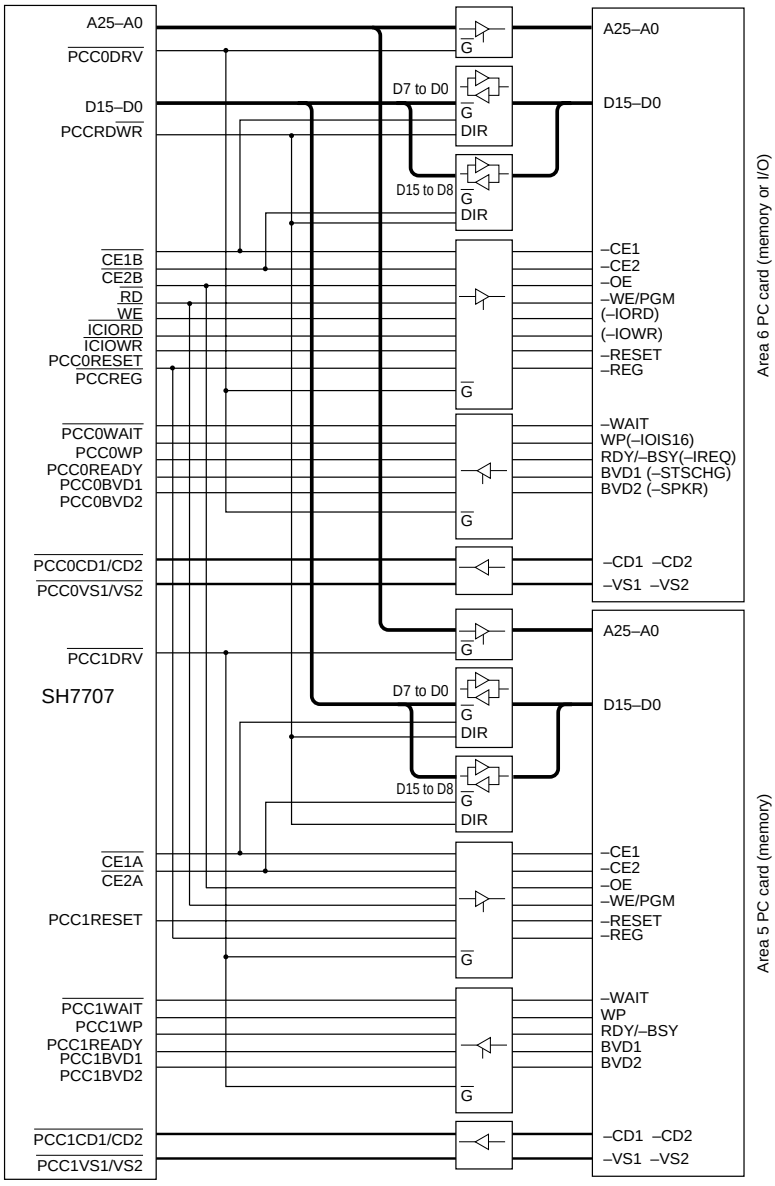
Bit 1: P1BWE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”' (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed although the data is guaranteed”

Bit 0—PCC1 Battery Dead Enable (P1BDE): Enables or disables the interrupt when the BVD2 and BVD1 pins in the PC card connected to area 5 are in the state in which “the battery must be changed since the data is not guaranteed”.

Bit 0: P1BDE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed” (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which “the battery must be changed since the data is not guaranteed”

11.3 Operation

11.3.1 PC card Connection Specifications (Interface Diagram, Pin Correspondence)



Note: For the method of connecting pull-up resistors, etc., refer to the PCMCIA Rev. 2.1 / JEIDA Ver. 4.2 specifications.

Figure 11.5 SH7707 Interface

Table 11.3 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	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	–CE1	I	Card enable	–CE1	I	Card enable	$\overline{\text{CE1A}}$ or $\overline{\text{CE1B}}$
8	A10	I	Address	A10	I	Address	A10
9	–OE	I	Output enable	–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	–WE/–PGM	I	Write enable	–WE/–PGM	I	Write enable	$\overline{\text{WE}}$
16	+RDY/–BSY	O	Ready/busy	–IREQ	O	Interrupt request	PCC0READY or PCC1READY
17	VCC		Power supply	VCC		Power supply	—
18	VPP1		Programming power supply	VPP1		Programming and peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3

Table 11.3 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Pin
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	–IOIS16	O	16-bit I/O port	PCC0WP, PCC1WP
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	–CD1	O	Card detection	–CD1	O	Card detection	$\overline{\text{PCC0CD1}}$ or $\overline{\text{PCC1CD1}}$
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	–CE2	I	Card enable	–CE2	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	–VS1	I	Voltage sense	–VS1	I	Voltage sense	$\overline{\text{PCC0VS1}}$ or $\overline{\text{PCC1VS1}}$
44	RFU		Reserved	–IORD	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	–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
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—
52	VPP2		Programming power supply	VPP2		Programming and peripheral power supply	—

Table 11.3 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7707 Corresponding
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Pin
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	–VS2	I	Voltage sense	–VS2	I	Voltage sense	$\overline{\text{PCC0VS2}}$ or $\overline{\text{PCC1VS2}}$
58	+RESET	I	Reset	+RESET	I	Reset	PCC0RESET or PCC1RESET
59	–WAIT	O	Wait request	–WAIT	O	Wait request	$\overline{\text{PCC0WAIT}}$ or $\overline{\text{PCC1WAIT}}$
60	RFU		Reserved	–INPACK	O	Input acknowledge	—
61	–REG	I	Attribute memory space select	–REG	I	Attribute memory space select	$\overline{\text{PCCREG}}$
62	BVD2	O	Battery voltage detection	–SPKR	O	Digital sound signal	PCC0BVD2 or PCC1BVD2
63	BVD1	O	Battery voltage detection	–STSCHG	O	Card status change	PCC0BVD1 or PCC1BVD1
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	–CD2	O	Card detection	–CD2	O	Card detection	$\overline{\text{PCC0CD2}}$ or $\overline{\text{PCC1CD2}}$
68	GND		Ground	GND		Ground	—

11.3.2 PC Card Interface Timing

(1) Memory card interface timing

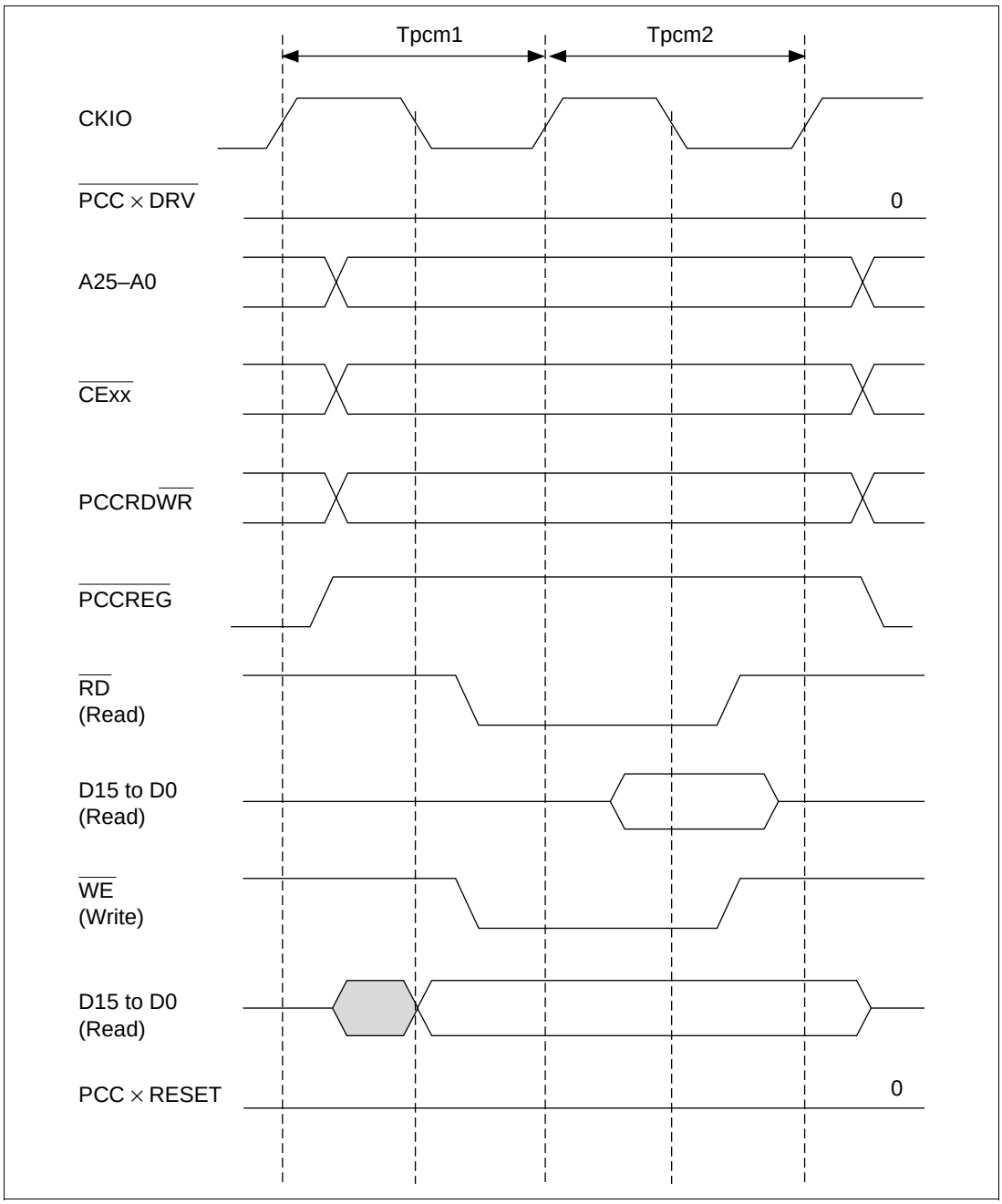


Figure 11.6 PCMCIA Memory Card Interface Basic Timing

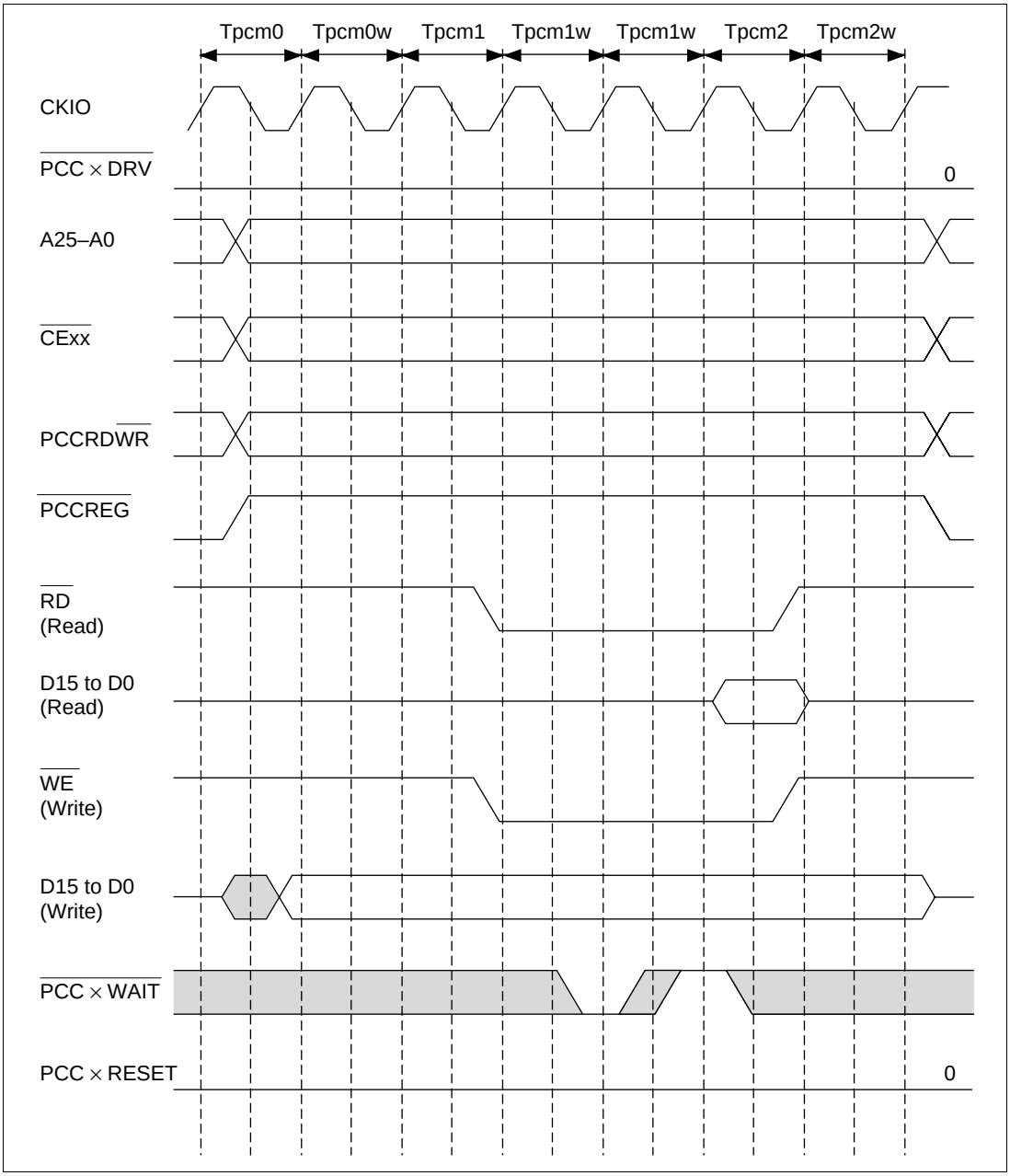


Figure 11.7 PCMCIA Memory Card Interface Wait Timing

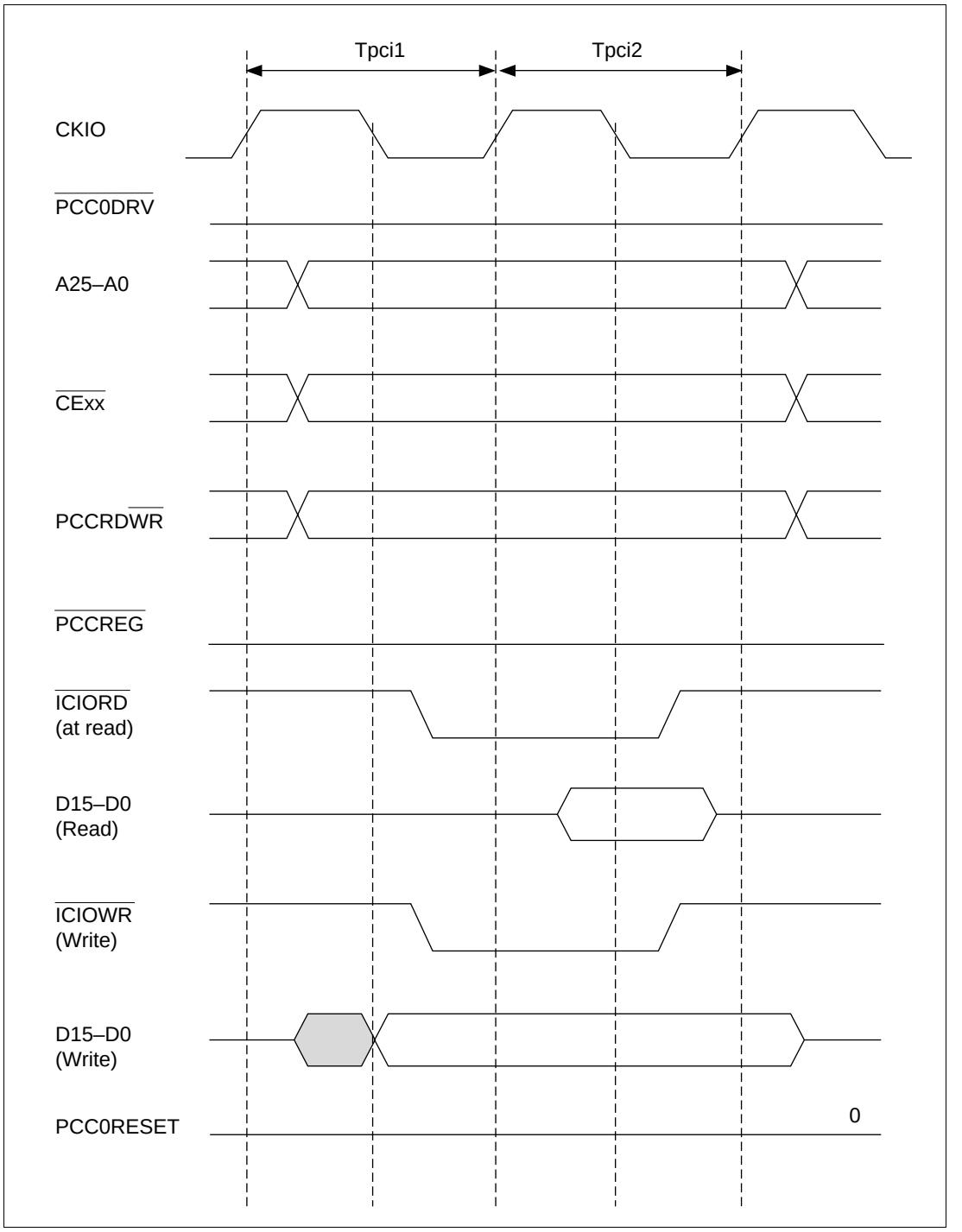


Figure 11.8 PCMCIA I/O Card Interface Basic Timing

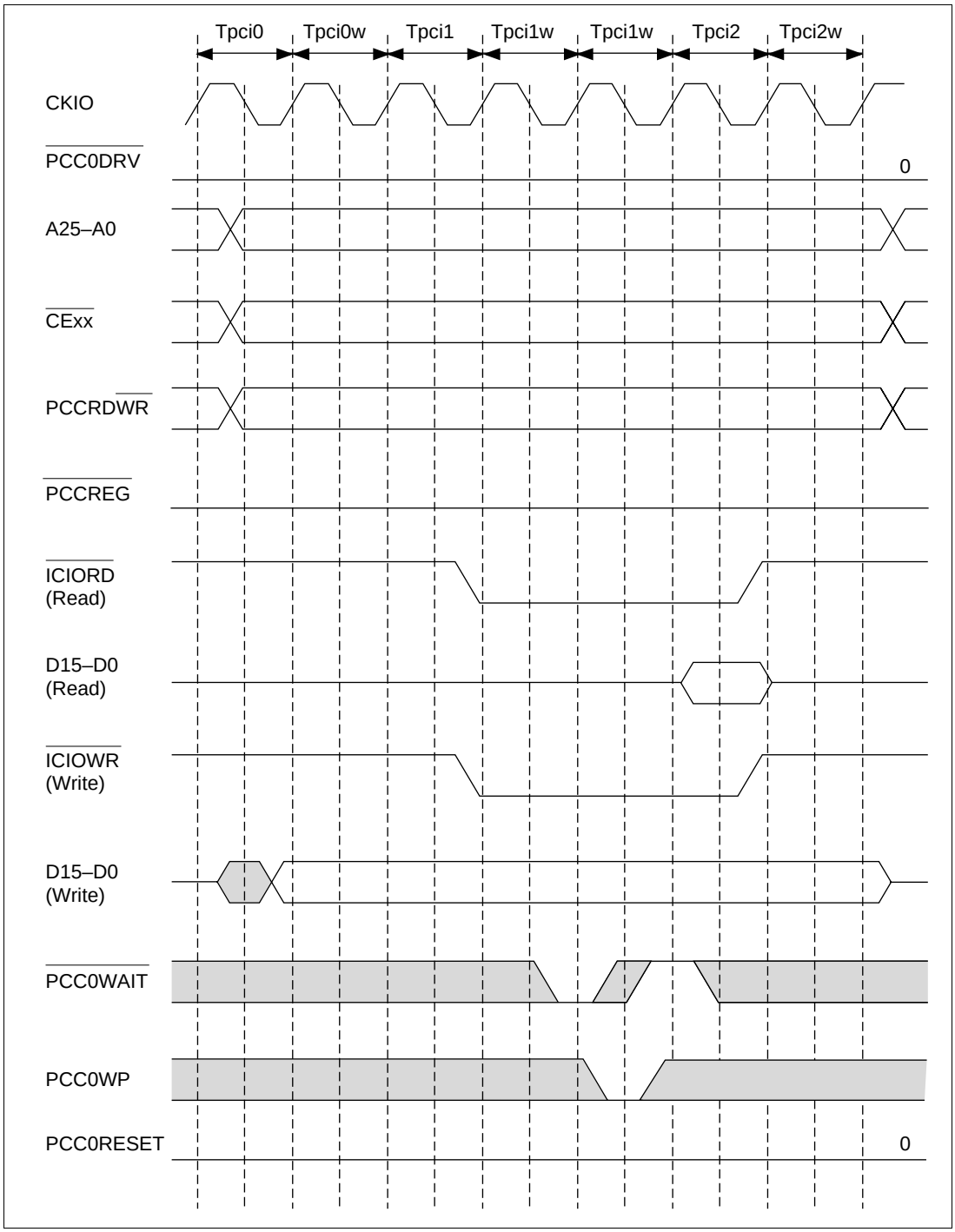


Figure 11.9 PCMCIA I/O Card Interface Wait Timing

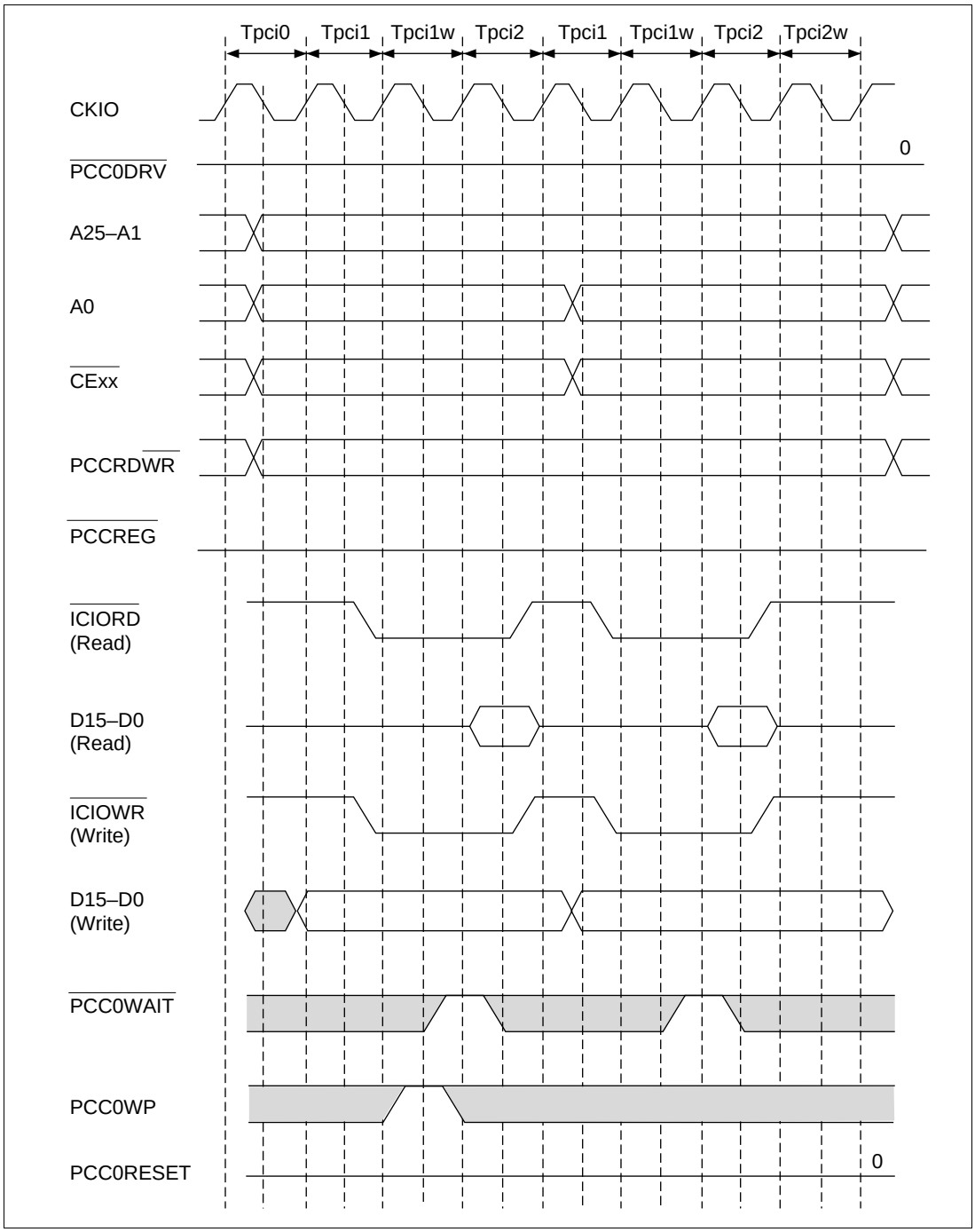


Figure 11.10 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

Refer to Section 10, Bus State Controllers (BSC, BSCP), and Section 24, Electrical Characteristics, for more details.

11.3.3 Usage Notes

External Bus Frequency Limit when Using PC Card: When using a PC card, the PC card control register (PCR) and wait state control registers (WCR1, WCR2) must be set in accordance with the SH7707's external bus frequency. Note, however, that a PC card cannot be connected directly to the SH7707 via a buffer if the following frequencies are exceeded.

PC Card	Maximum SH7707 External Bus Frequency
3.3 V PC card	23 MHz
5 V PC card	41 MHz

This limit is the value when the maximum value is set for all parameters.
PCR register TED: 3.5-cycle delay
PCR register TEH: 3.5-cycle delay
WCR2 register WAIT: 10 waits
WCR1 register idle states: insertion of 3 idle cycles

Reason:

According to the PC card standard, the attribute memory access time is specified as 600 ns (3.3 V)/300 ns (5 V). Therefore, when the SH7707 accesses attribute memory, the bus cycle must be coordinated with the PC card interface timing. In the SH7707, the timing can be adjusted by setting the TED and TEH values in the PCR register, and the number of waits and number of idle states in the WCR1 and WCR2 registers, allowing a PC card to be used within the above frequency ranges.

The common memory access time and I/O access time (based on the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals) are also similarly specified (see table below), and a PC card must be used within the above ranges in order to satisfy all these specifications.

PC Card Space	Access Time (5 V Operation)	Access Time (3.3 V Operation)
Attribute memory	300 ns	600 ns
Common memory	250 ns	600 ns
I/O space ($\overline{\text{IORD}}$/$\overline{\text{IOWR}}$ pulse width)	165 ns	165 ns

Pin Function Control and Card Type Switching: When setting pin function controller pin functions to dedicated PC card use ("other function"), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made. However, this restriction does not apply to the card detection pins (CD1, CD2).

When changing the card type bit (P0PCCT) in the area 6 general control register (PCC0GCR), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made.

Reason:

When PC card controller settings are modified, the functions of PC card pins that generate various interrupts change, with the result that unnecessary interrupts may be generated.

Section 12 Direct Memory Access Controller (DMAC)

12.1 Overview

The SH7707 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 $\overline{\text{DACK}}$ (transfer request acknowledge signal), external memory, memory-mapped external devices, and on-chip supporting modules (IRDA, SCIF, A/D converter, D/A converter, PCC, and I/O ports). Using the DMAC reduces the burden on the CPU and increases overall operating efficiency.

12.1.1 Features

The DMAC has the following features.

- Four channels
- Physical address space
- 8-, 16-, or 32-bit data transfer unit
- 16 M (16,777,216) 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 a direct address transfer mode and an indirect address transfer mode.

    Direct address transfer mode: The values specified in the DMAC registers indicate the transfer source and transfer destination. Two 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 in direct address transfer mode. This function is only valid for channel 3.
- Channel functions: Transfer modes that can be specified differ from channel to 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 every four 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 every four DMA transfers. This function is only valid for 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 supporting modules such as the serial communication interfaces (IRDA, SCIF), A/D converter (A/D), and 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 the end of the specified number of transfers.

12.1. 2 Block Diagram

Figure 12.1 shows a block diagram of the DMAC.

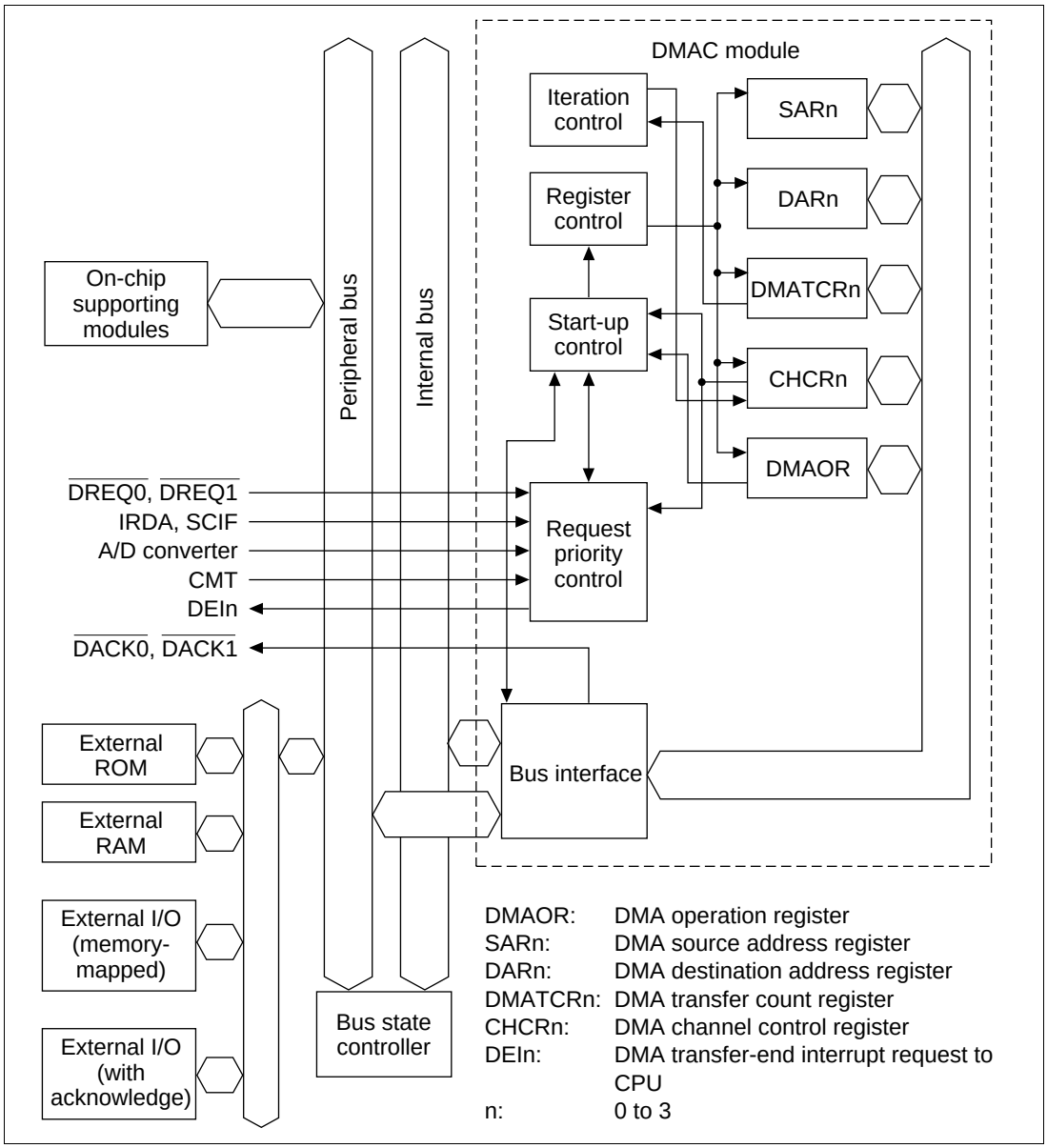


Figure 12.1 DMAC Block Diagram

12.1.3 Pin Configuration

Table 12.1 shows the DMAC pins.

Table 12.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	I	DMA transfer request input from external device to channel 0
	DREQ acknowledge	$\overline{\text{DACK0}}$	O	Strobe output to an external I/O upon DMA transfer request 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	$\overline{\text{DACK1}}$	O	Strobe output to an external I/O upon DMA transfer request from external device to channel 1

12.1.4 Register Configuration

Table 12.2 summarizes the DMAC registers. The DMAC has a total of 17 registers. Each channel has four control registers. One other control register is shared by all channels.

Table 12.2 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'4000020	32 bits	16, 32
	DMA destination address register 0	DAR0	R/W	Undefined	H'4000024	32 bits	16, 32
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'4000028	24 bits	16, 32
	DMA channel control register 0	CHCR0	R/(W)*1	H'00000000	H'400002C	32 bits	8, 16, 32
1	DMA source address register 1	SAR1	R/W	Undefined	H'4000030	32 bits	16, 32
	DMA destination address register 1	DAR1	R/W	Undefined	H'4000034	32 bits	16, 32
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'4000038	24 bits	16, 32
	DMA channel control register 1	CHCR1	R/(W)*1	H'00000000	H'400003C	32 bits	8, 16, 32
2	DMA source address register 2	SAR2	R/W	Undefined	H'4000040	32 bits	16, 32
	DMA destination address register 2	DAR2	R/W	Undefined	H'4000044	32 bits	16, 32
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'4000048	24 bits	16, 32
	DMA channel control register 2	CHCR2	R/(W)*1	H'00000000	H'400004C	32 bits	8, 16, 32
3	DMA source address register 3	SAR3	R/W	Undefined	H'4000050	32 bits	16, 32
	DMA destination address register 3	DAR3	R/W	Undefined	H'4000054	32 bits	16, 32
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'4000058	24 bits	16, 32
	DMA channel control register 3	CHCR3	R/(W)*1	H'00000000	H'400005C	32 bits	8, 16, 32
Shared	DMA operation register	DMAOR	R/(W)*1	H'0000	H'4000060	16 bits	8, 16, 32

Notes: 1. Only 0s can be written to bit 1 of CHCR0–CHCR3, and bits 1 and 2 of DMAOR to clear the flags after 1 is read.
2. If 16-bit access is performed on SAR0–SAR3, DAR0–DAR3, or CHCR0–CHCR3, the value in the 16 bits that were not accessed is retained.
3. DMATCR has 24 bits. Therefore, writting 1 to upper bits 24–31 is invalid; 0 is always returned if these bits are read.

12.2 Register Descriptions

12.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3)

DMA source address registers 0–3 (SAR0–SAR3) are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in 16 bits or in 32 bits, specify an address with a 16-bit or 32-bit address boundary. Operation cannot be guaranteed if other addresses are specified.

The initial value of these registers after a reset is undefined. They retain their previous values in standby mode.

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

12.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. These registers include count functions, and during a DMA transfer, indicate the next destination address.

To transfer data in 16 bits or in 32 bits, specify an address with a 16-bit or 32-bit address boundary. Operation cannot be guaranteed if other addresses are specified.

The initial value of these registers after a reset is undefined. They retain their previous values in standby mode.

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

12.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit readable/writable registers that specify the DMA transfer count (bytes, words, or longwords). The number of transfers is 1 when the setting is H'000001, and 16,777,216 (the maximum) when H'000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

Writing to their upper eight bits in DMATCR is invalid; 0 is returned if these bits are read.

The initial value of these registers after a reset is undefined. They retain their previous values in standby mode.

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

12.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)

DMA channel control registers 0–3 (CHCR0–CHCR3) are 32-bit readable/writable registers that specify the operation mode, transfer method, and other parameters for each channel. Writing to bits 31–24 and 7 in these registers is invalid; 0 is returned if these bits are read.

Bit 20 is only used in CHCR3; it is not used in CHCR0–CHCR2. Consequently, writing to this bit is invalid in CHCR0–CHCR2; 0 is returned if this bit is read. Bit 19 is only used in CHCR2; it is not used in CHCR0, CHCR1, or CHCR3. Consequently, writing to this bit is invalid in CHCR0, CHCR1, and CHCR3; 0 is returned if this bit is read. Bits 6 and 16–18 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; 0 returned if these bits are read.

These registers are initialized to H'00000000 by a reset. They retain their previous values in standby mode.

Bit:	31	...	21	20	19	18	17	16
Bit name:	—	...	—	DI	RO	—	AM	AL
Initial value:	0	...	0	0	0	0	0	0
R/W:	R	...	R	(R/W)	(R/W)	R	(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. The DI, RO, AM, AL, and DS bits are not included in some channels.

Bit 20—Direct/Indirect Select (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–CHCR2; 0 is returned if this bit is read.

Bit 20: DI	Description	
0	Direct address mode	(Initial value)
1	Indirect address mode	

Bit 19—Source Address Reload (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 returned 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 17—Acknowledge Mode (AM): Specifies whether $\overline{\text{DACK}}$ is output in the data read cycle or the 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 returned if this bit is read.

Bit 17: AM	Description	
0	$\overline{\text{DACK}}$ output in read cycle	(Initial value)
1	$\overline{\text{DACK}}$ output in write cycle	

Bit 16—Acknowledge Level (AL): Specifies whether the $\overline{\text{DACK}}$ (acknowledge) signal output is active-high or active-low active.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is returned if this bit is read.

Bit 16: AL	Description	
0	$\overline{\text{DACK}}$ output is active-low	(Initial value)
1	$\overline{\text{DACK output}}$ is active-high	

Bits 15 and 14—Destination Address Mode 1 and 0 (DM1, 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, +4 in 32-bit transfer)
1	0	Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer)
1	1	Illegal setting

Bits 13 and 12—Source Address Mode 1 and 0 (SM1, 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, +4 in 32-bit transfer)
1	0	Source address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer)
1	1	Reserved (illegal setting)

If the transfer source is specified by 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 incrementing or decrementing in indirect address mode depends on the SM1 and SM0 settings. In this case, however, the SAR3 increment or decrement value is +4, –4, or fixed at 0, regardless of the transfer data size specified in TS1 and TS0.

Bits 11 to 8—Resource Select 3–0 (RS3–RS0): RS3–RS0 specify which transfer requests will be sent to the DMAC.

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: 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	A/D converter	
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 (DS): Selects low-level or falling-edge deterction as the sampling method for the $\overline{\text{DREQ}}$ pin used in external request mode.

This bit is only valid in CHCR0 and CHCR1. Writing to this bit is invalid in CHCR2 and CHCR3; 0 is returned if this bit is read.

In channels 0 and 1, if an on-chip supporting 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 at low level	(Initial value)
1	$\overline{\text{DREQ}}$ detected at falling edge	

Bit 5—Transmit Mode (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 1 and 0 (TS1, TS0): TS1 and TS0 specify 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 (IE): Setting this bit to 1 generates an interrupt request at data transfer end (TE = 1) at the count specified in TCR.

Bit 2: IE	Description
0	Interrupt request is not generated even if data transfer ends at the specified count (Initial value)
1	Interrupt request is generated when data transfer ends at the specified count

Bit 1—Transfer End (TE): TE is set to 1 when data transfer ends at the count specified in DMATCR. If the IE bit is set to 1 at this time, an interrupt request is generated.

If data transfer ends due to an NMI interrupt or clearing of the DE bit or the DME bit in DMAOR before this bit is set to 1, it will not be set to 1. While this bit is set to 1, transfer is not enabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	Data transfer has not ended at the count specified in DMATCR (Initial value) Clearing conditions: Writing 0 after reading TE = 1, power-on reset, manual reset
1	Data transfer ends at the specified count

Bit 0—DMAC Enable (DE): Enables channel operation.

Bit 0: DE	Description	
0	Channel operation disabled	(Initial value)
1	Channel operation enabled	

If an auto-request is specified (in RS3–RS0), transfer starts when this bit is set to 1. In the case of an external request or an internal module request, transfer starts if the transfer request is generated after this bit is set to 1. Clearing this bit during transfer will terminate the 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.

12.2.5 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit readable/writable register that controls the DMAC transfer mode. Writing to bits 15–10 and bits 7–2 is invalid in this register; 0 is always returned if these bits are read.

DMAOR is initialized to H'0000 by a reset. It retains its previous value in 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 bit after 1 is read.

Bits 9 and 8—Priority Mode 1 and 0 (PR1, PR0): PR1 and PR0 select the priority level between channels when there are simultaneous transfer requests for multiple channels.

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 (NMIF): Indicates that an NMI interrupt occurred. This bit is set regardless of whether the DMAC is operating or halted. 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)
	Clearing conditions:	Writing 0 after reading NMIF = 1, power-on reset, manual reset	
1	NMI input. DMA transfer is disabled		
	This bit is set by generation of an NMI interrupt.		

Bit 0—DMA Master Enable (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 1, transfer is enabled in the corresponding channel. If this bit is cleared during transfer, transfers in all the channels will 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	DMA transfers disabled on all channels	(Initial value)
1	DMA transfers enabled on all channels	

12.3 Operation

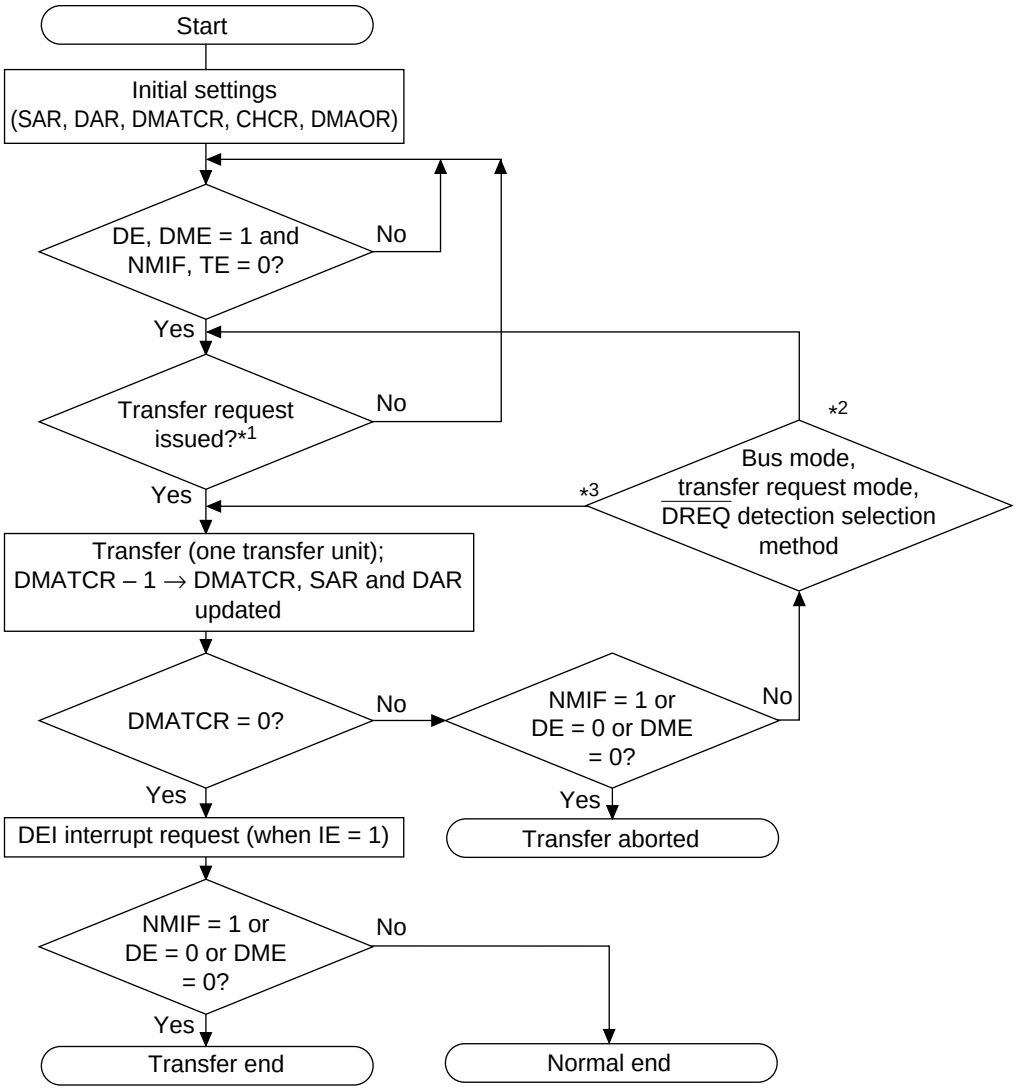
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto-request, external request, and on-chip module request. The dual address mode has a direct address transfer mode and an indirect address transfer mode. As the bus mode, burst mode or cycle steal mode can be selected.

12.3.1 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 arrives and transfer is enabled, the DMAC transfers one 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 in 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 in CHCR or the DME bit in DMAOR is changed to 0.

Figure 12.2 shows a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when NMIF and all TE bits are all 0 and the DE and DME bits are set to 1.
2. $\overline{\text{DREQ}}$ level detection in burst mode (external request) or cycle-steal mode.
3. $\overline{\text{DREQ}}$ edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 12.2 DMA Transfer Flowchart

12.3.2 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 supporting 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 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 supporting 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 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 in response to the request signal ($\overline{\text{DREQ}}$) of an external device. Choose the mode shown in table 12.3. 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 in CHCR0–CHCR3 (DS = 0 for level detection, DS = 1 for edge detection). The source of the transfer request does not have to be the data transfer source or destination.

Table 12.3 Selecting External Request Mode 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 supporting module (IRDA, SCIF, A/D converter, D/A converter, PCC, or I/O port)

On-Chip Module Request: In this mode a transfer is performed in response to the transfer request signal (interrupt request signal) of an on-chip module. There are six transfer request signals: the receive data full interrupts (RXI) and 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 12.4). When this mode is selected, if 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). If the transfer request is from the A/D converter, the data transfer source must be the A/D converter register.

Table 12.4 Selecting On-Chip Supporting Module Request Modes with the RS Bits

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 communication interface channels 1 and 2

ADDR: A/D data register of A/D converter

Note: External memory, memory-mapped external device, on-chip peripheral module (IRDA, SCIF, A/D converter, D/A converter, PCC, or I/O port)

When outputting transfer requests from on-chip supporting modules, the appropriate interrupt enable bits must be set to output the interrupt signals.

If the interrupt request signal of the on-chip supporting module is used as a DMA transfer request signal, an interrupt is sent to the CPU.

The DMA transfer request signals in table 12.4 are automatically withdrawn when the corresponding DMA transfer is performed. If cycle steal mode is being used, they are withdrawn at the first transfer; if burst mode is being used, they are withdrawn at the last transfer.

12.3.3 Channel Priority Order

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 priority bits PR1 and PR0 in the DMA operation register.

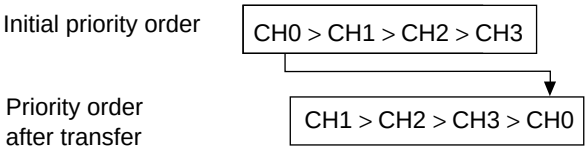
Fixed Modes: In these modes, the priority levels among the channels remain fixed. There are three kinds of fixed modes as follows:

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).

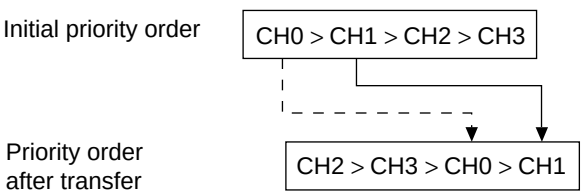
Round-Robin Mode: Each time one word, byte, or longword is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished rotates to the bottom of the priority order. Round-robin mode operation is shown in figure 12.3. The priority order of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after a reset.

(1) Channel 0 transfer



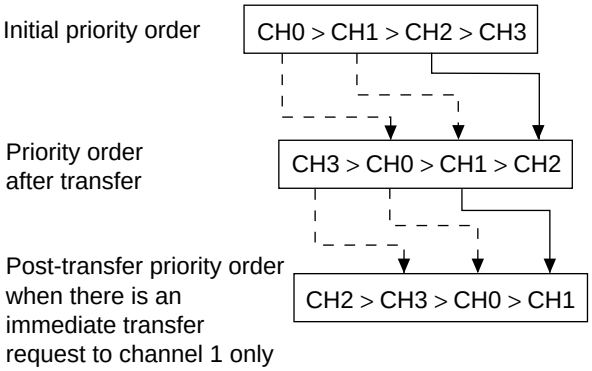
Channel 0 becomes bottom priority

(2) Channel 1 transfer



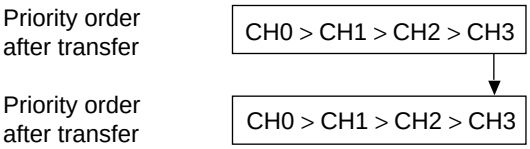
Channel 0 becomes bottom priority.
The priority of channel 0, which was higher than channel 3, is also shifted.

(3) Channel 2 transfer



Channel 2 becomes bottom priority.
The priorities 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 priorities of channels 0 and 3, which were higher than channel 1, are also shifted.

(4) Channel 3 transfer



Priority order does not change

Figure 12.3 Round-Robin Mode

Figure 12.4 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 for 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.

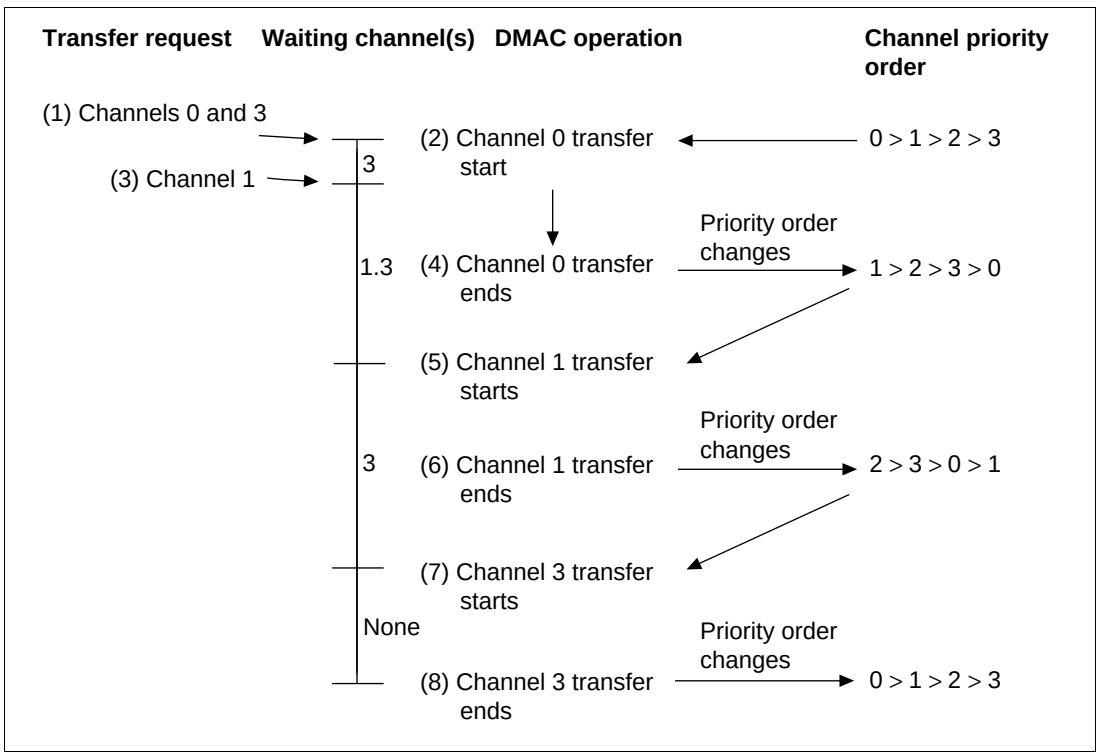


Figure 12.4 Changes in Channel Priority Order in Round-Robin Mode

12.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 12.5. In dual address mode, both the transfer source address and the transfer destination address are output. Dual address mode has a direct address mode and an indirect address mode. In direct address mode, an output address value is the data transfer target address; in indirect address mode, the value stored in the output address, not the output address value itself, is the data transfer target address. The data transfer timing depends on the bus mode, which may be cycle steal mode or burst mode.

Table 12.5 Supported DMA Transfers

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Supporting Module**
External device with DACK	Not available	Dual	Dual	Not available	Not available
External memory	Dual	Dual	Dual	Not available	Dual
Memory-mapped external device	Dual	Dual	Dual	Not available	Dual
On-chip memory	Not available	Not available	Not available	Not available	Not available
On-chip supporting module	Not available	Dual	Dual	Not available	Dual

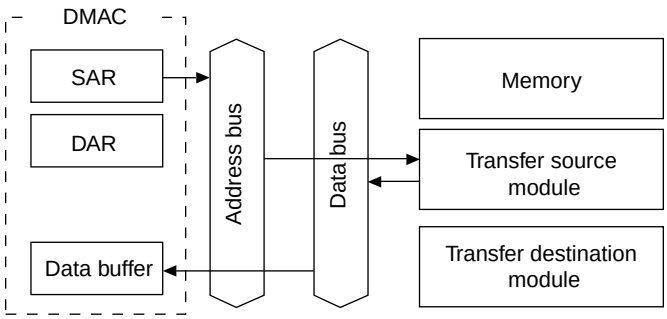
Notes: 1. Dual: Dual address mode
2. Dual address mode includes a direct address mode and an indirect address mode.
3. On-chip supporting module: IRDA, SCIF, A/D converter, D/A converter, PCC, or I/O port

Address Modes:

- Dual Address Mode

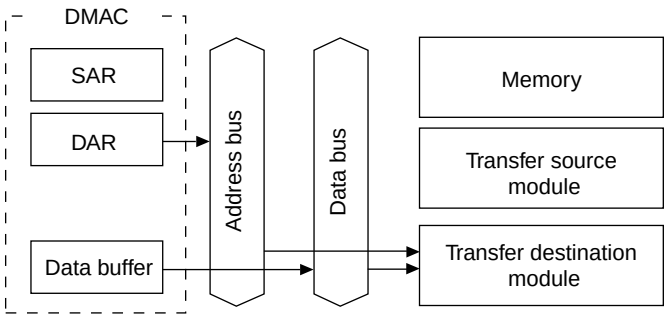
  In dual address mode, both the transfer source and destination are accessed (selectable) by an address. The source and destination can be located externally or internally. Dual address mode has (1) a direct address transfer mode and (2) an indirect address transfer mode.

  (1) In 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 12.5, 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 12.6 shows an example of the timing in this case.



The SAR value is an address, data is read from the transfer source module, and the data is temporarily stored in the DMAC.

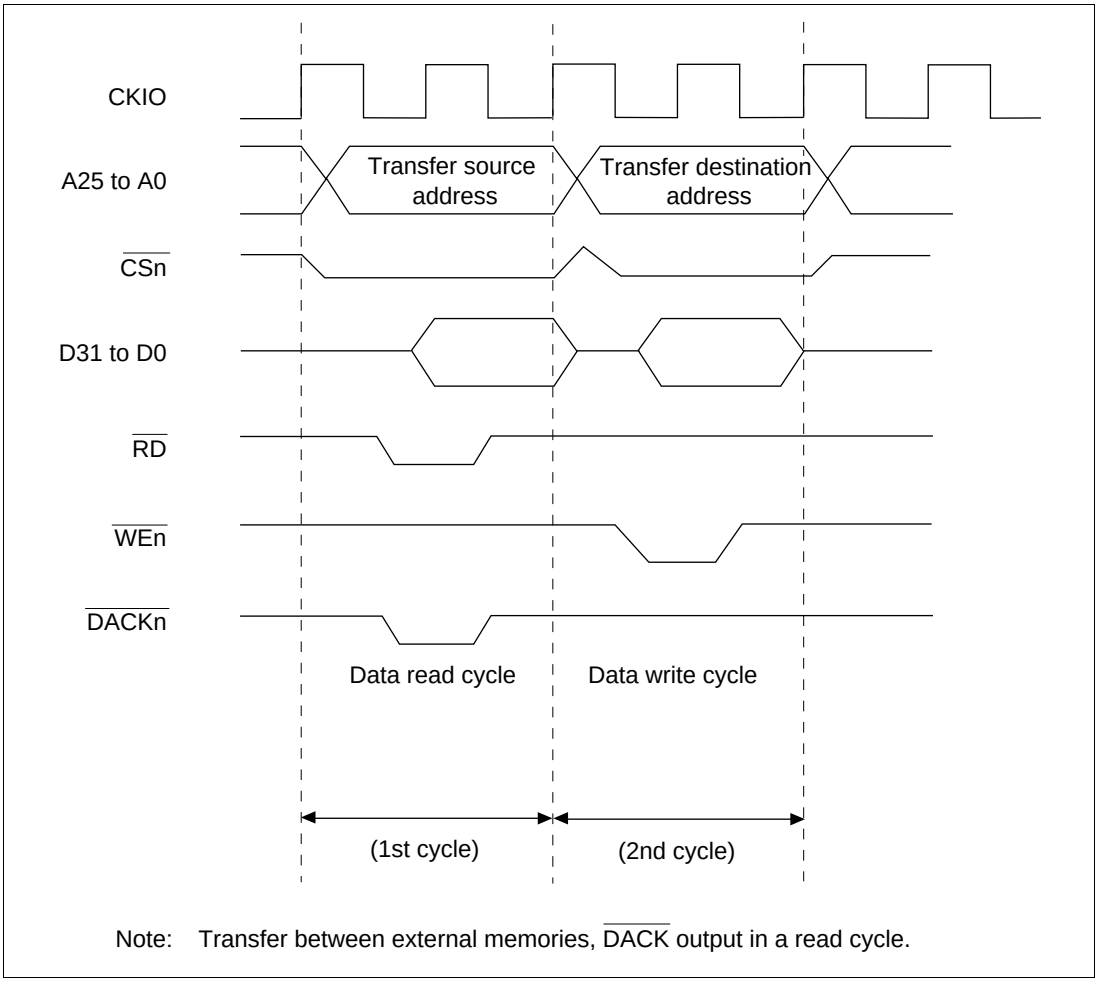
First bus cycle



The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 12.5 Indirect Address Mode Operation in Dual Address Mode



Note: Transfer between external memories, DACK output in a read cycle.

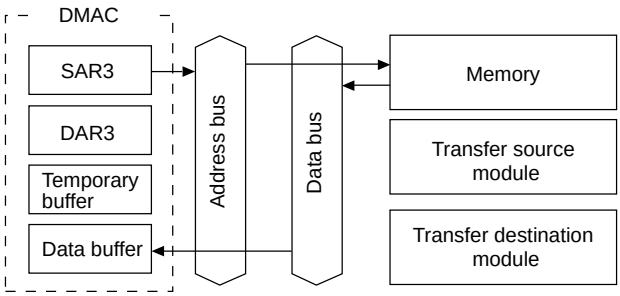
Figure 12.6 Example of DMA Transfer Timing in Direct Address Mode in Dual Address Mode

(2) In 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 12.7 shows one example. In this example, the transfer destination, the transfer source, and the storage destination of the indirect address are external memories in 16 bit areas, and transfer data is 16 or 8 bits. Figure 12.8 shows an example of the transfer timing.

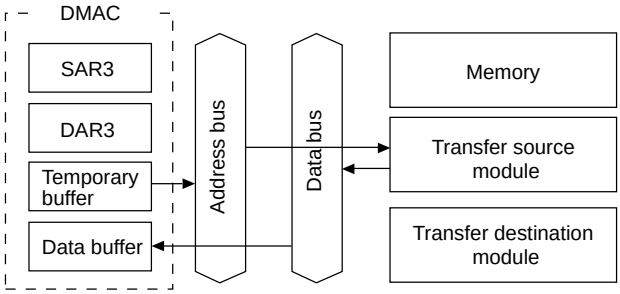
In this mode, one NOP cycle (CK1 cycle shown in figure 12.8) is required to output data read as an indirect address to the address bus.

If the transfer data is 32 bits, the third and fourth bus cycles shown in figure 12.8 are each required twice; a total of six bus cycles and one NOP cycle are thus required.



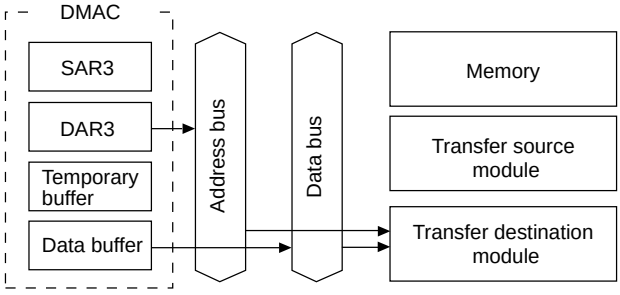
The value in SAR3 is used as an address, memory data is read, and its value is stored in the temporary buffer. The value read at this time is 32-bit because it is used as an address.
As the data bus connected to the external devices is 16 bits wide, two bus cycles are required.

First and second bus cycles



The value in the temporary buffer is used as an address, and data is read from the transfer source module to the data buffer.

Third bus cycle

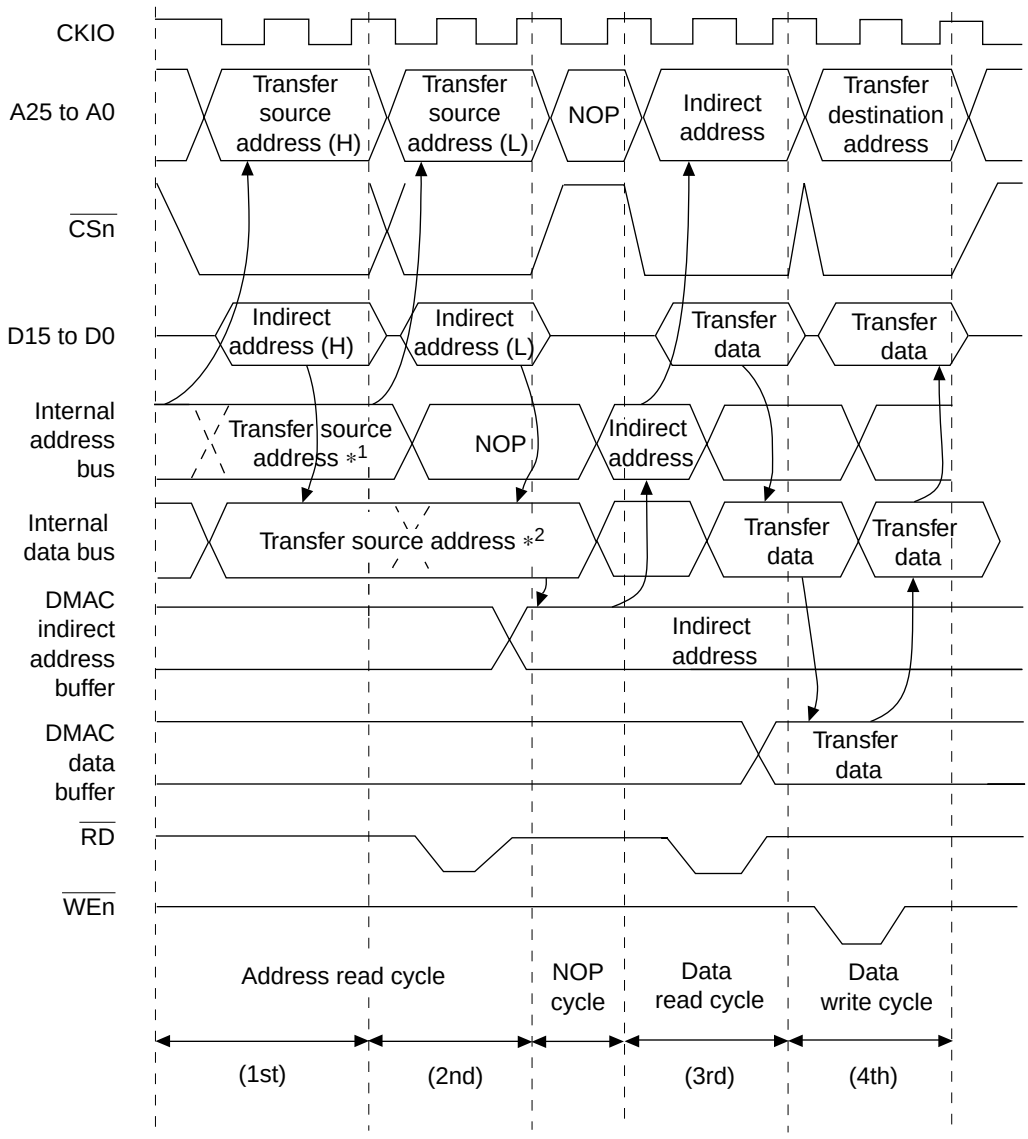


The value in DAR3 is used as an address, and the value in the data buffer is written to the transfer destination module.

Fourth bus cycle

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 12.7 Indirect Address Mode Operation in Dual Address Mode



Transfer between external memories

Notes: 1. The internal address bus value does not change, and is controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bus.

Figure 12.8 Example of Transfer Timing in Indirect Address Mode in Dual Address Mode

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 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 is obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

  In cycle steal mode, transfer areas are not affected regardless of settings of the transfer request source, transfer source, and transfer destination. Figure 12.9 shows an example of DMA transfer timing in cycle steal mode. Transfer conditions shown in the figure are:

  — Dual address mode
  — $\overline{\text{DREQ}}$ level detection

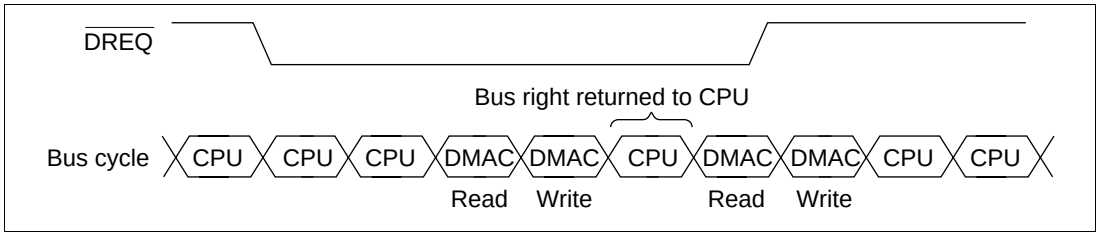


Figure 12.9 Example of of Transfer in Cycle Steal Mode

- Burst Mode

  Once the bus right is obtained, the transfer is performed continuously until the transfer end condition is satisfied. In 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 end of the DMAC transfer request that has already been accepted, even if the transfer end conditions have not been satisfied.

  Burst mode cannot be used when a serial communication interface (IRDA or SCIF) is the transfer request source.

Figure 12.10 shows an example of DMA transfer timing in burst mode.

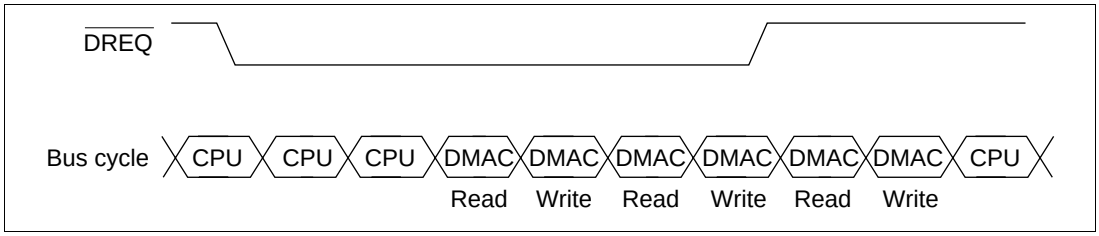


Figure 12.10 Example of Transfer in Burst Mode ($\overline{\text{DREQ}}$ Level Detection)

Relationship between Request Modes and Bus Modes by DMA Transfer Category: Table 12.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 12.6 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	All*1	B/C	8/16/32	0,1
	External device with DACK and memory-mapped external device	All*1	B/C	8/16/32	0, 1
	External device with DACK and on-chip peripheral module	All*2	B/C*3	8/16/32*4	0, 1
	External memory and external memory	All*1	B/C	8/16/32	0–3*5
	External memory and memory-mapped external device	All*1	B/C	8/16/32	0–3*5
	External memory and on-chip peripheral module	All*2	B/C*3	8/16/32*4	0–3*5
	Memory-mapped external device and memory-mapped external device	All*1	B/C	8/16/32	0–3*5
	Memory-mapped external device and on-chip peripheral module	All*2	B/C*3	8/16/32*4	0–3*5
	On-chip peripheral module and on-chip peripheral module	All*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 supporting module requests are all available. For on-chip supporting module requests, however, IRDA, SCIF, and the A/D converter cannot be specified as the transfer request source.

2. External requests, auto requests, and on-chip supporting module requests are all available. When the IRDA, SCIF, or A/D converter is also the transfer request source, however, the transfer destination or transfer source must be the IRDA, SCIF, or the A/D converter, respectively.

3. If the transfer request source is the IRDA or SCIF, cycle steal only.
4. The access size permitted when the transfer destination or source is an on-chip supporting module register.
5. If the transfer request is an external request, channels 0 and 1 only.

Bus Mode and Channel Priority Order: When channel 1 is transferring in burst mode and there is a transfer request to channel 0 with a higher priority, the channel 0 transfer 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 cycle steal mode or in burst mode.

If the priority is set in 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 cycle steal mode or in 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 fixed mode or in round-robin mode, the bus will not be given to the CPU since channel 1 is in burst mode. This example is illustrated in figure 12.11.

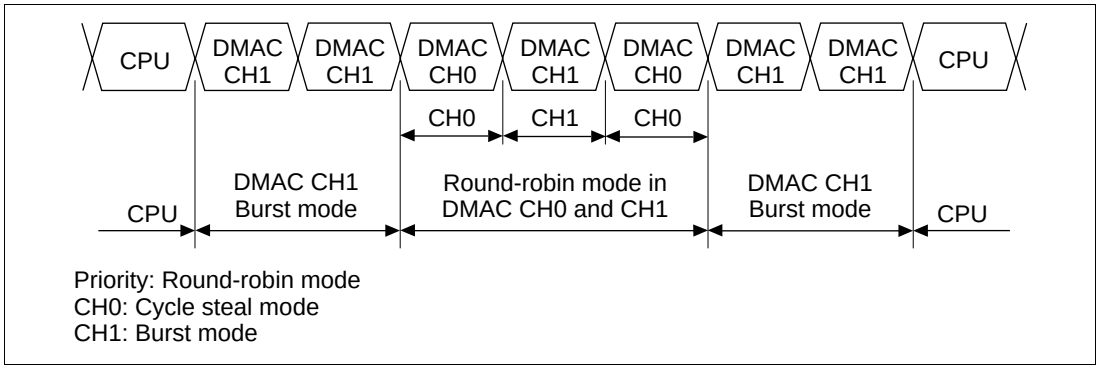


Figure 12.11 Bus Handling when Multiple Channels Are Operating

12.3.5 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 Controllers (BSC, BSCP).

$\overline{\text{DREQ}}$ 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 12.12 (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 12.13.

  $\overline{\text{DACK}}$ is output in a read in the example in figure 12.12, and in a write in the example in figure 12.13. 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 12.14, 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, Edge Detection

  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is only performed once.

  For example, in figure 12.15, 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.

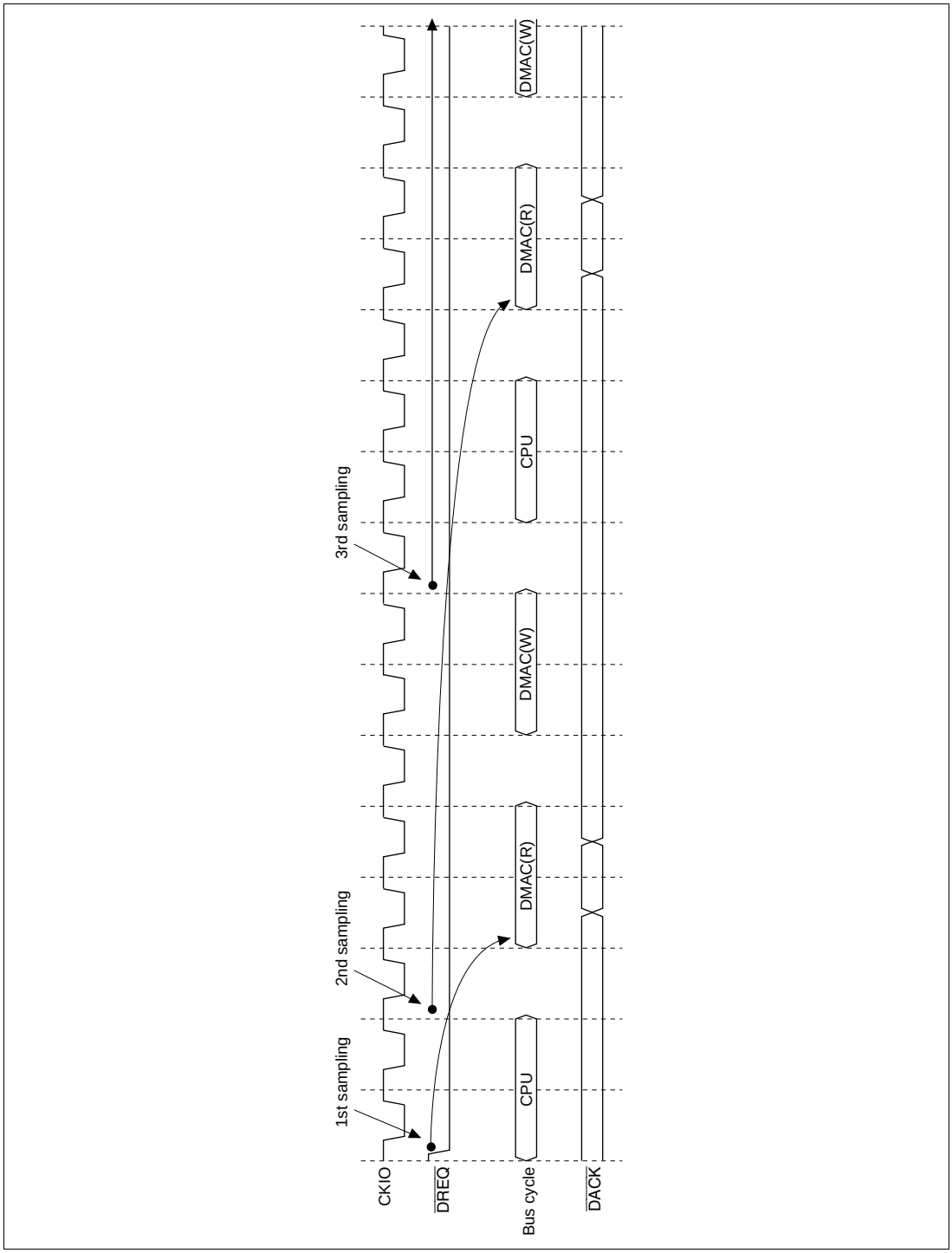


Figure 12.12 Cycle Steal Mode, Level Input (CPU Access: 2 Cycles)

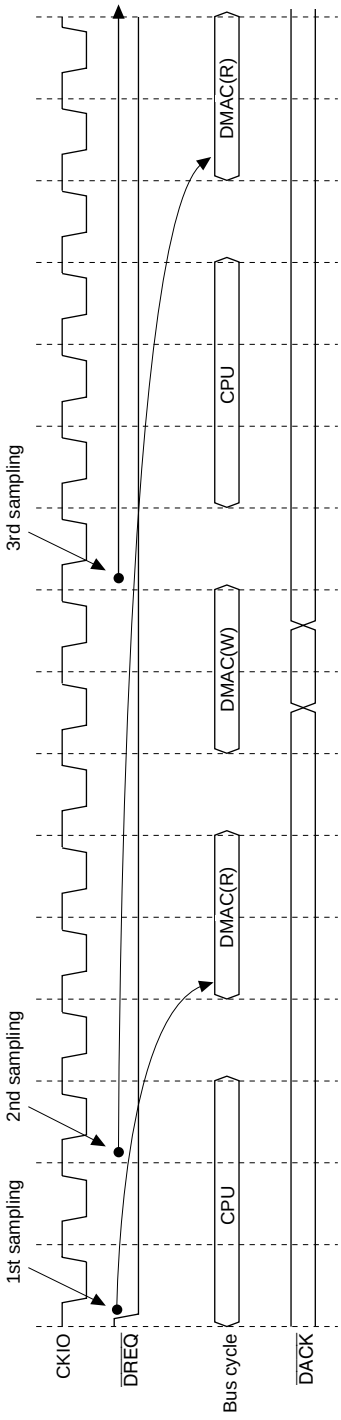


Figure 12.13 Cycle Steal Mode, Level Input (CPU Access: 3 Cycles)

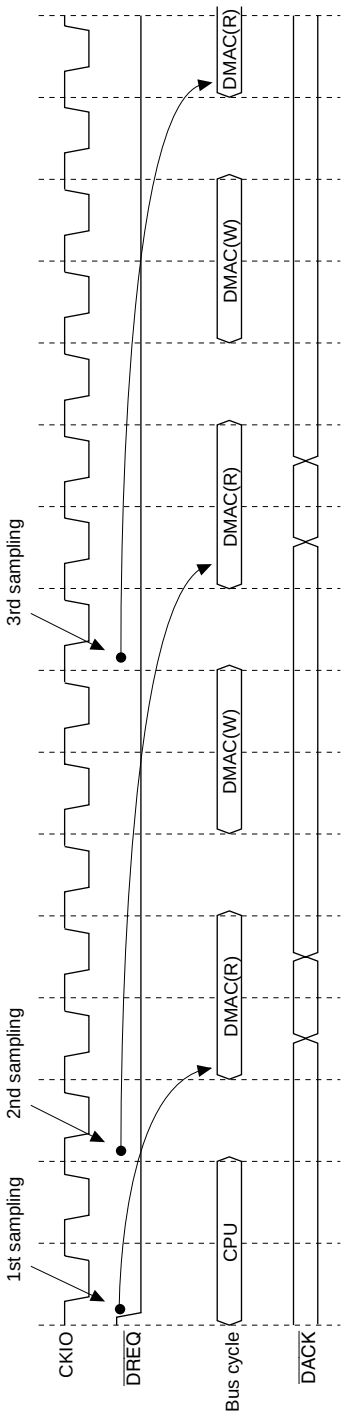


Figure 12.14 Burst Mode, Level Input

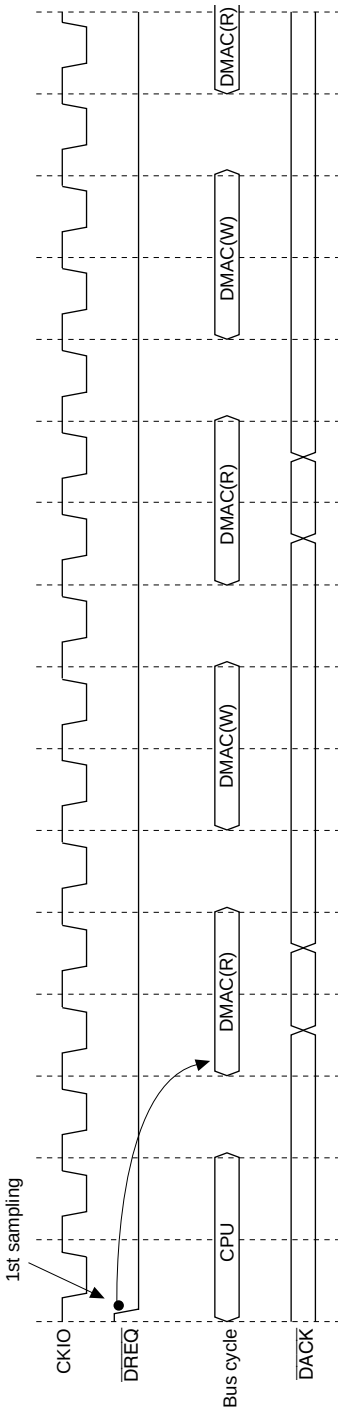


Figure 12.15 Burst Mode, Edge Input

12.3.6 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) every four transfers by setting the RO bit in CHCR2. Figure 12.16 shows this operation. Figure 12.17 shows a timing chart of the source address reload function, with the following operating conditions: burst mode, auto-request, 16-bit transfer data size, SAR2 incrementing, DAR2 fixed, reload function on, and use of channel 2 only.

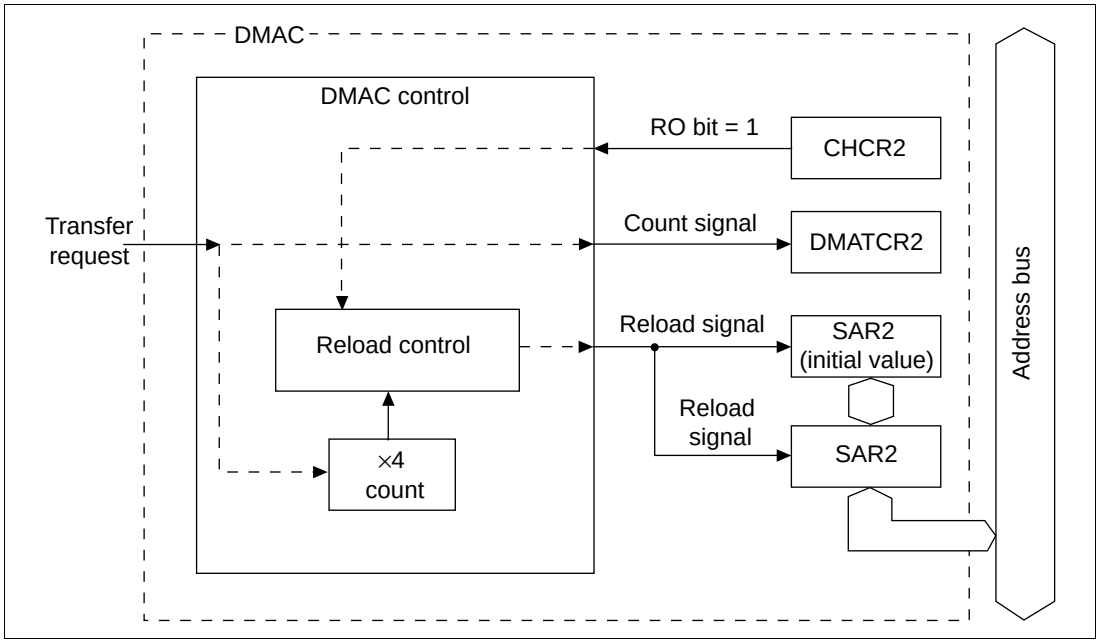


Figure 12.16 Source Address Reload Function

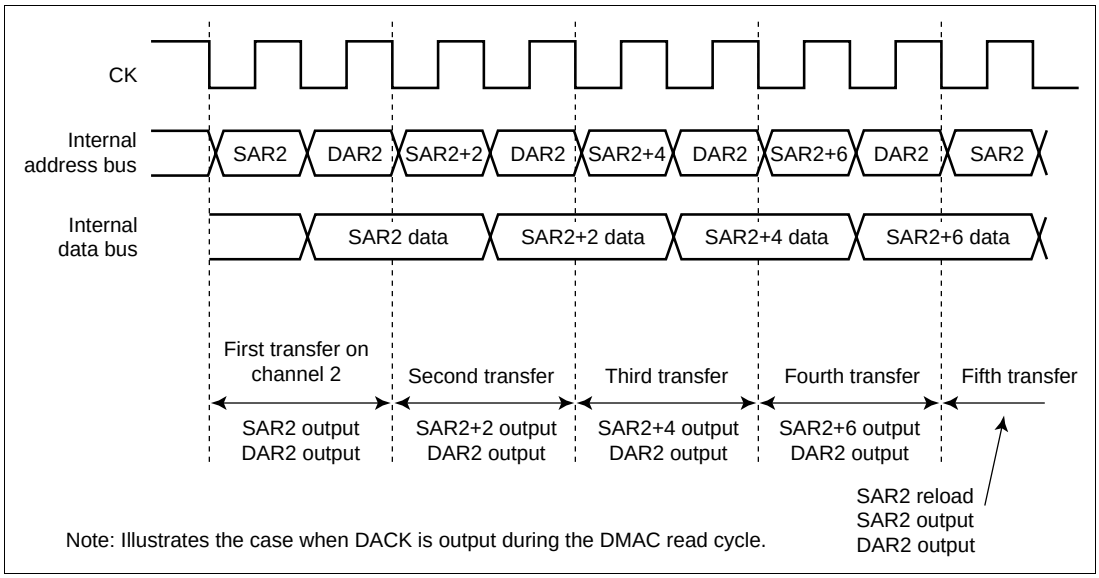


Figure 12.17 Timing Chart of Source Address Reload Function

The reload function can be used regardless of whether the transfer data size is 8, 16, or 32 bits.

DMATCR, which specifies the transfer count, is decremented by 1 each time a transfer ends regardless of whether the reload function is on or off. Consequently, be sure to specify a multiple of four in DMATCR when the reload function is on. Operation cannot be guaranteed if other values are specified.

Though the counters that count transfers in sets of four 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, as well as by a reset, the SAR2, DAR2, DMATCR2 registers are not reset. Therefore, if these sources are generated, the DMAC will contain counters that are initialized and counters that are not; malfunction will be caused by restarting the DMAC in that state. Consequently, if these sources occur other than for setting the TE bit during use of the reload function, set SAR2, DAR2, and DMATCR2 again.

12.3.7 DMA Transfer Ending Conditions

The DMA transfer ending conditions differ for individual channel ending and ending on all channels together. At transfer end, the following conditions are applied except in 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 cycle steal mode, the operation is the same regardless of whether the transfer request is detected by level or edge.

(b) Edge detection in 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 is the same as in cycle steal mode. With edge detection in burst mode, though only one transfer request is generated to start up the DMAC, stop request sampling is performed at the same timing as transfer request sampling in cycle steal mode. As a result, the period when a stop request is not sampled is regarded as the period when a transfer request is generated, and after performing the DMA transfer for this period, the DMAC stops operating.

(c) Level detection in burst mode (external request)

Same as (a) above.

(d) Bus timing when transfers are suspended

The transfer is suspended when one transfer ends. Even if transfer ending conditions are satisfied during a read in a direct address transfer in dual address mode, the subsequent write process is executed, and after the transfer in (a) to (c) above has been executed, DMAC operation halts.

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 in 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 CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) request is sent to the CPU. This transfer ending does not apply to (a) to (d) described above.
- When DE in 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 on All Channels Simultaneously: Transfers on all channels end 1) when the NMIF (NMI flag) bit is set to 1 in DMAOR, or 2) when the DME bit in 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 DMAOR and all channels stop their transfers according to the conditions in (a) to (d) described above, and pass the bus to another bus master. Consequently, even if the NMI bit is set to 1 during transfer, SAR, DAR, and DMATCR are updated. The TE bit is not set. To resume transfer after NMI interrupt exception handling, clear the NMIF bit to 0. If there are channels that should not be restarted, clear the corresponding CHCR DE bits.
- Transfers ending when DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in DMAOR aborts transfer on all channels. The TE bit is not set. All channels aborts their transfers according to the conditions in (a) to (d) in 12.3.7, as in the case of NMI interrupt generation. In this case, the values in SAR, DAR, and DMATCR are also updated.

12.4 Compare Match Timer (CMT)

12.4.1 Overview

The DMAC has an on-chip compare match timer (CMT) for generating DMA transfer requests. The CMT has a 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.
- Generates a DMA transfer request when a compare match occurs.

Block Diagram

Figure 12.18 shows a block diagram of the CMT.

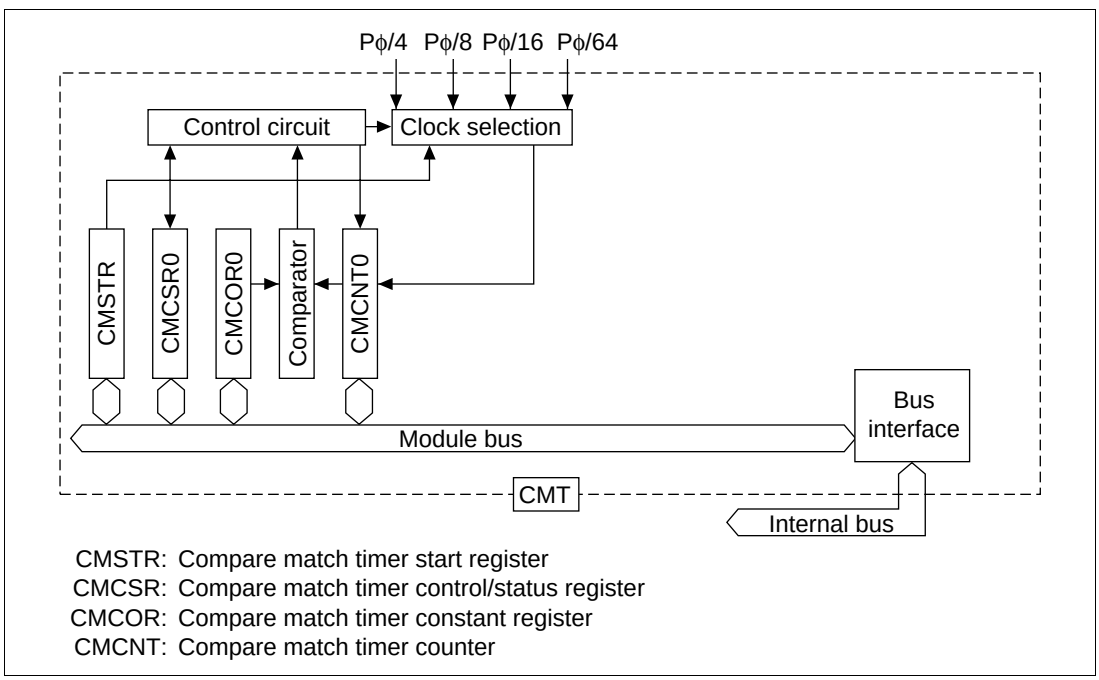


Figure 12.18 CMT Block Diagram

12.4.2 Register Configuration

Table 12.7 summarizes the CMT register configuration.

Table 12.7 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Compare match timer start register	CMSTR	R/W	H'0000	H'4000070	8, 16, 32
Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'4000072	8, 16, 32
Compare match counter 0	CMCNT0	R/W	H'0000	H'4000074	8, 16, 32
Compare match constant register 0	CMCOR0	R/W	H'FFFF	H'4000076	8, 16, 32

Note: Only 0 can be written to the CMF bit in CMCSR0, to clear the flag.

12.4.3 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). CMSTR is initialized to H'0000 by a reset, but 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:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	R/W

Bits 15 to 1—Reserved: Read-only bits, always read as 0.

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 interrupt enabling/disabling, and specifies the clock used for incrementation. CMCSR is initialized to H'0000 by a reset, but 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:	—	—	—	—	—	—	—	—

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/W

Note: Only 0 can be written to clear the flag.

Bits 15 to 8 and 6 to 2—Reserved: Read-only bits, always read as 0.

Bit 7—Compare Match Flag (CMF): Indicates whether or not the CMCNT and CMCOR values match.

Bit 7: CMF	Description	
0	CMCNT and CMCOR values match	(Initial value)
	Clearing condition: Write 0 to CMF after reading CMF = 1	
1	CMCNT and CMCOR values match	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock input to CMCNT from among the four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT begins incrementing on the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ/4	(Initial value)
	1	Pϕ/8	
1	0	Pϕ/16	
	1	Pϕ/64	

Compare Match Counter (CMCNT)

The compare match counter (CMCNT) is a 16-bit register used as an up-counter.

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT begins incrementing on that clock. When the CMCNT value matches that of the compare match constant register (CMCOR), CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a reset, but 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 period for a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a reset, but 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

12.4.4 Operation

Period Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT begins incrementing on 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 in CMCSR is set to 1. The CMCNT counter begins counting up again from H'0000.

Figure 12.19 shows the compare match counter operation.

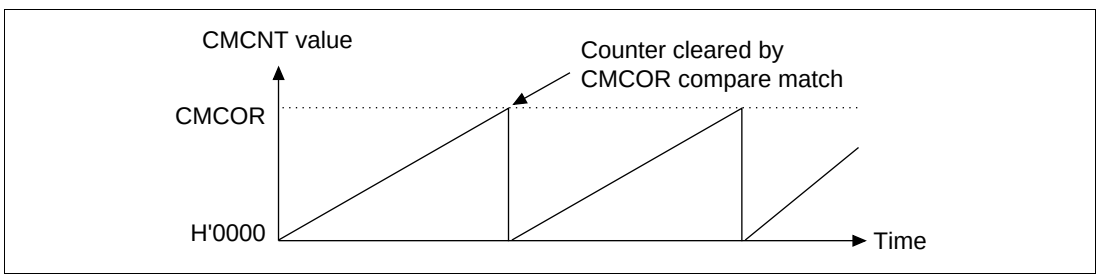


Figure 12.19 Counter Operation

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 and CKS0 bits in CMCSR. Figure 12.20 shows the timing.

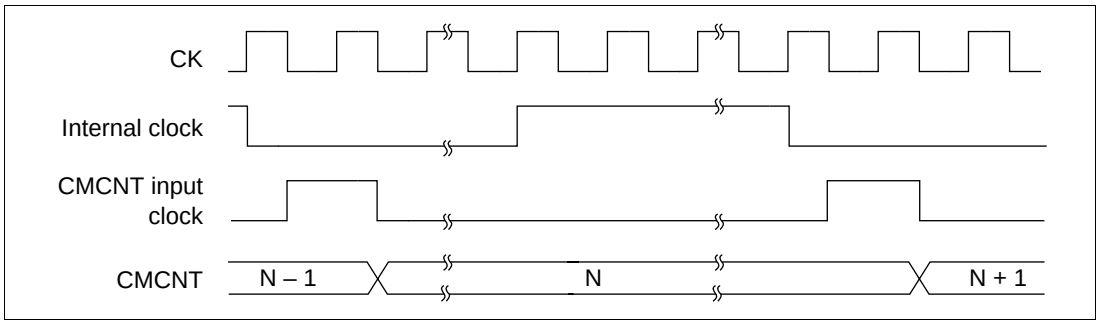


Figure 12.20 Count Timing

12.4.5 Compare Match

Timing of Compare Match Flag Setting

The CMF bit in CMCSR is set to 1 by the compare match signal generated when the CMCOR register and CMCNT counter values match. The compare match signal is generated in 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 12.21 shows the timing of CMF bit setting.

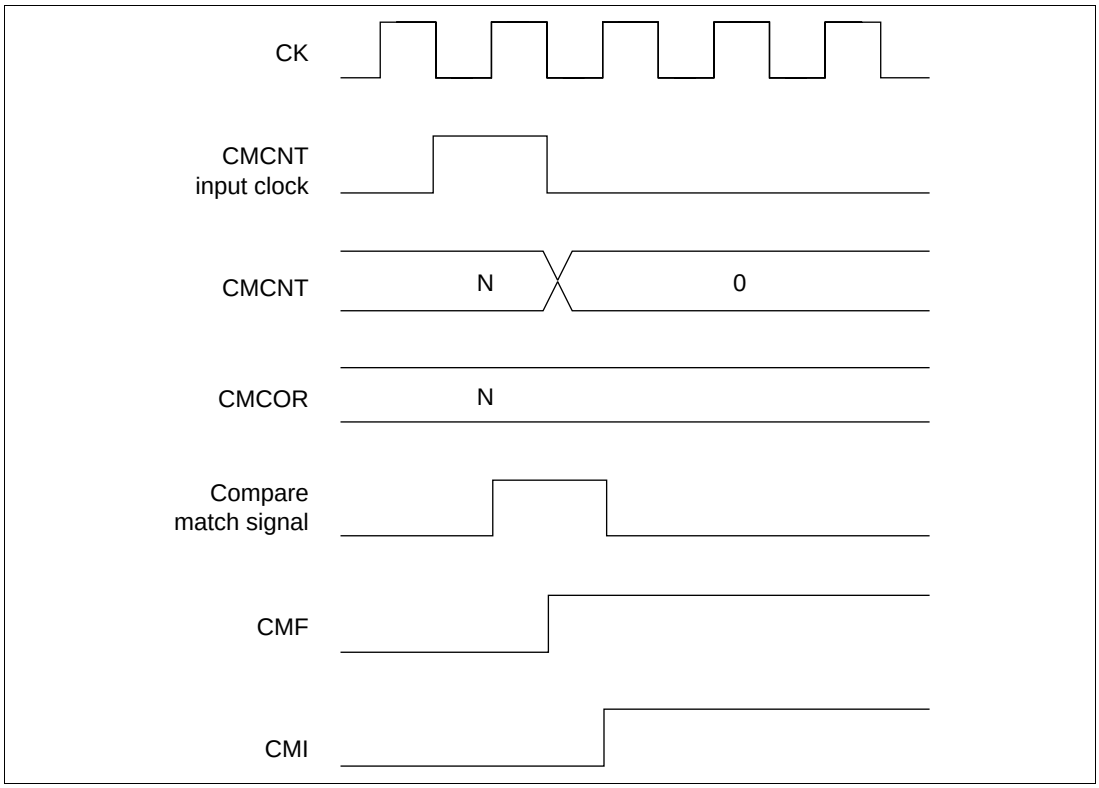


Figure 12.21 Timing of CMF Setting

Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by writing 0 to it after reading 1. Figure 12.22 shows the timing when the CMF bit is cleared by the CPU.

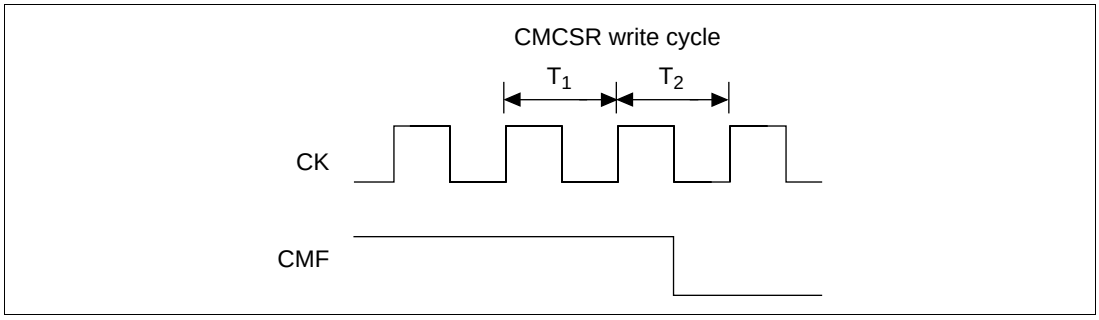


Figure 12.22 Timing of CMF Clearing by the CPU

12.5 Examples of Use

12.5.1 Example of DMA Transfer between On-Chip IRDA and External Memory

In this example, on-chip IRDA receive data is transferred to external memory using DMAC channel 3. Table 12.8 shows the transfer conditions and register settings. In addition, it is recommended that the trigger for the number of receive FIFO data in IRDA is set to 1 (RTRG1 = RTRG0 = 0 in SCFCR).

Table 12.8 Transfer Conditions and Register Settings for Transfer between On-Chip IRDA 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

12.5.2 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 A/D converter (transfer source) and external memory (transfer destination) with the address reload function on. Table 12.9 shows the transfer conditions and register settings.

Table 12.9 Transfer Conditions and Register Settings for Transfer between On-Chip A/D Converter Channel 2 and External Memory

Transfer Conditions	Register	Setting
Transfer source: on-chip A/D 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: longword		
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 every four transfers. In this example, when an interrupt request is generated from A/D converter, longword data is read from the register at address H'04000080 in the A/D converter, and is written to external memory address H'00400000. Since longword data has been transferred, the values in SAR and DAR are H'04000084 and H'003FFFFC, respectively. The bus is retained and data transfers are performed successively as this transfer is in burst mode.

After four transfers are completed, fifth and sixth transfers are performed if the address reload function is off, and the value in SAR is incremented from H'0400008C to H'04000090, H'04000094, and so on. If the address reload function is on, the DMA transfer stops after the fourth transfer, and 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 decremented regardless of whether the address reload function is on or off.

As a result, the values in the DMAC are as shown in table 12.10 when the fourth transfer ends, depending on whether the address reload function is on or off.

Table 12.10 Values in the DMAC after the Fourth Transfer

Items	Address Reload On	Address Reload Off
SAR	H'04000080	H'04000090
DAR	H'003FFFFC	H'003FFFF0
DMATCR	H'0000007C	H'0000007C
Bus	Released	Held
DMAC operation	Stops	Continues
Interrupt	Not generated	Not generated
Transfer request source flag clearing	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 burst mode when using the address reload function. This function may not be correctly executed in cycle steal mode.
4. Set a multiple of four in DMATCR when using the address reload function. This function may not be correctly executed if other values are specified.

12.5.3 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 by indirect address (transfer source) and the SCIF transmitter (transfer destination) . Table 12.11 shows the transfer conditions and register settings. In addition, the trigger for the number of transmit FIFO data is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Table 12.11 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: SCIF (TXI2)		
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 indirect address mode is set, data stored in the address set in SAR is not used as transfer source data. In the indirect address mode, 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 12.11, when an SCIF 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 indirect address mode, 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 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 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 operations are the same as in a normal dual address transfer.

12.6 Cautions

1. The DMA channel control registers (CHCR0–CHCR3) can be accessed using any data size. The DMA operation register (DMAOR) must be accessed using byte (8-bit) or word (16-bit) size; other registers must be accessed using word (16-bit) or longword (32-bit) size.
2. Before rewriting the RS0–RS3 bits in CHCR0–CHCR3, first clear the DE bit to 0 (when rewriting CHCR with a byte address, be sure to clear the DE bit to 0 in advance).
3. The NMIF bit in DMAOR will be set when the NMI interrupt is input, even if the DMAC is not operating.
4. When entering standby mode, the DME bit in DMAOR must be cleared to 0 and the transfer accepted by the DMAC must be terminated.
5. The on-chip modules which the DMAC can access are the IRDA, SCIF, A/D converter, D/A converter, PCC, and I/O ports. Do not access other modules with the DMAC.
6. When starting up the DMAC, set CHCR or DMAOR last. Normal operation cannot be guaranteed if other registers are specified last.
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, 0 should still be written to DMATCR, otherwise, normal DMA transfer may not be performed.
8. When using the address reload function, specify burst mode as the transfer mode. In cycle steal mode, normal DMA transfer may not be performed.
9. When using the address reload function, set a multiple of four in DMATCR. Normal operation cannot be guaranteed if other values are specified.
10. When detecting an external request at the falling edge, keep the external request pin high when setting the DMAC.
11. Do not access addresses H'4000062–H'400006F, as this space is not used in the DMAC. Accessing this space may cause malfunctions.

Section 13 Timer (TMU)

13.1 Overview

The SH7707 has a three-channel 32-bit timer unit (TMU).

13.1.1 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 supporting 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 SH7707 is in standby mode: When the RTC output clock is being used as the counter input clock, the SH7707 is still able to count in standby mode.
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the supporting 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 register so that the entire 32-bit data in TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels: Operate the SH7707 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.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the TMU.

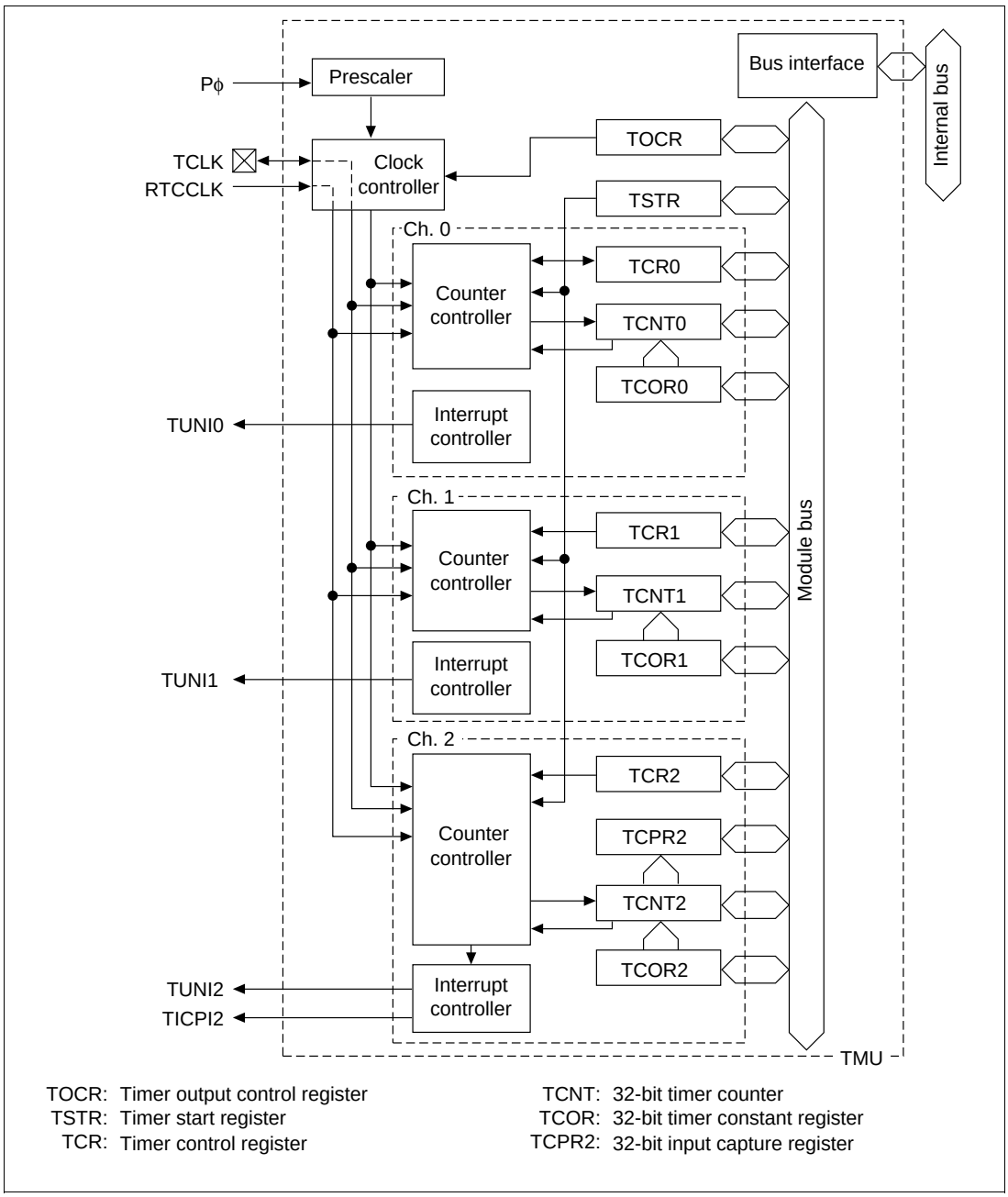


Figure 13.1 TMU Block Diagram

13.1.3 Pin Configuration

Table 13.1 shows the pin configuration of the TMU.

Table 13.1 Pin Configuration

Channel	Pin	I/O	Description
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/real-time clock (RTC) output pin

13.1.4 Register Configuration

Table 13.2 shows the TMU register configuration.

Table 13.2 TMU Register Configuration

Channel	Register	Abbreviation	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	Undefined	H'FFFFFEB8	32

13.2 TMU Registers

13.2.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit readable/writable register that selects whether to use the external TCLK pin as an external clock or input capture control 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/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

13.2.2 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable 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/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	TCNT2 count halted	(Initial value)
1	TCNT2 count started	

Bit 1—Counter Start 1 (STR1): Selects whether to run or halt timer counter 1 (TCNT1).

Bit 1: STR1	Description	
0	TCNT1 count halted	(Initial value)
1	TCNT1 count started	

Bit 0—Counter Start 0 (STR0): Selects whether to run or halt timer counter 0 (TCNT0).

Bit 0: STR0	Description	
0	TCNT0 count halted	(Initial value)
1	TCNT0 count started	

13.2.3 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 readable/writable 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/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/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/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 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 (Do not 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 enabling 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 and 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 edges	

Bits 2 to 0—Timer Prescalers 2 to 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	

13.2.4 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 readable/writable 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

13.2.5 Timer Counters (TCNT)

The timer counters are 32-bit readable/writable 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 SH7707 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 one time.

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

13.2.6 Input Capture Register (TCPR2)

The input capture register (TCPR2) is a read-only 32-bit register incorporated only in 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

13.3 TMU Operation

13.3.1 Overview

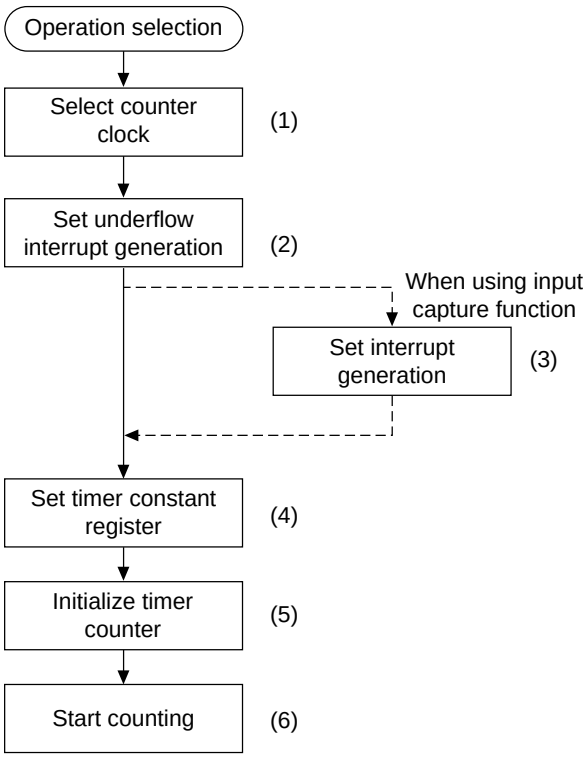
Each of the three channels has a 32-bit timer counter (TCNT) and a 32-bit timer constant register. The TCNT counter counts down. The auto-reload function enables synchronized counting and counting by external events. Channel 2 has an input capture function.

13.3.2 Basic Functions

Counter Operation: When one of bits STR0–STR2 in the timer start register (TSTR) is set, the corresponding timer counter (TCNT) starts counting. When 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 13.2):

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 relevant STR bit in the timer start register (TSTR) to 1 to start operation.



Note: When an interrupt has been generated, clear the flag that caused it in the interrupt handler. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 13.2 Setting the Count Operation

Auto-Reload Count Operation: Figure 13.3 shows the TCNT auto-reload operation.

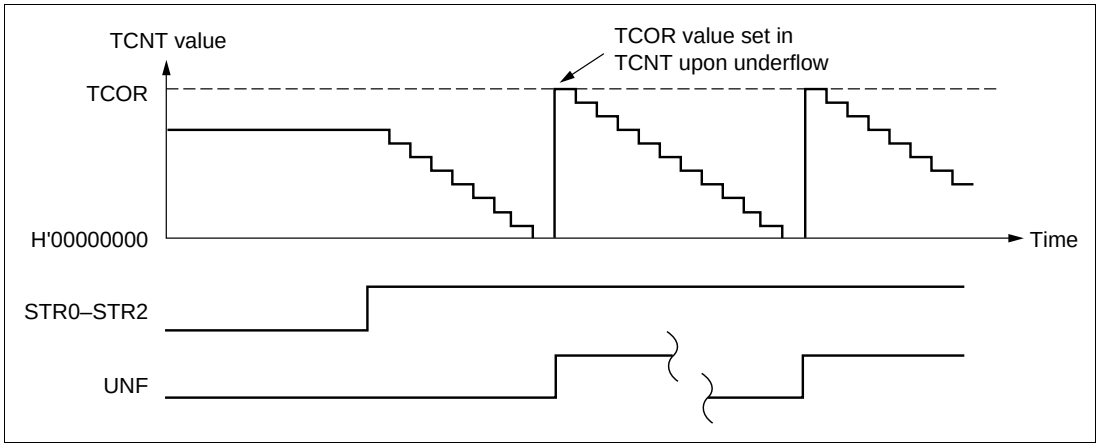


Figure 13.3 Auto-Reload Count Operation

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 13.4 shows the timing.

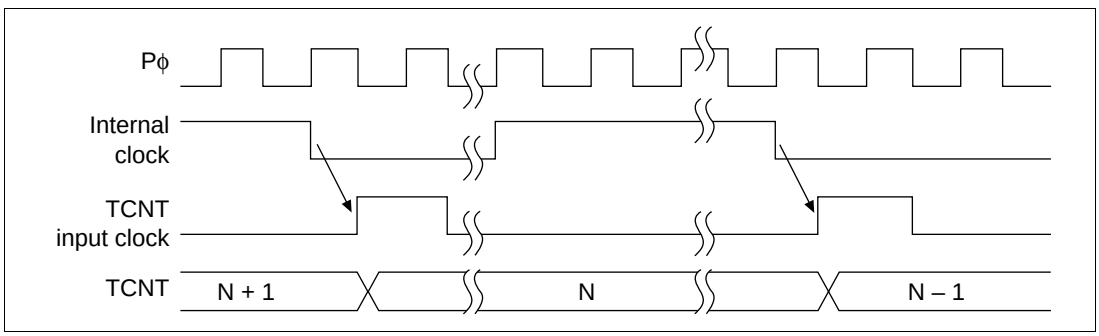


Figure 13.4 Count Timing when Internal Clock Is Operating

- 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: rising, falling or both edges may be selected. The pulse width of the external clock must be at least 1.5 peripheral clock cycles for single edges or 2.5 peripheral clock cycles for both edges. A shorter pulse width will result in inaccurate operation. Figure 13.5 shows the timing for both-edge detection.

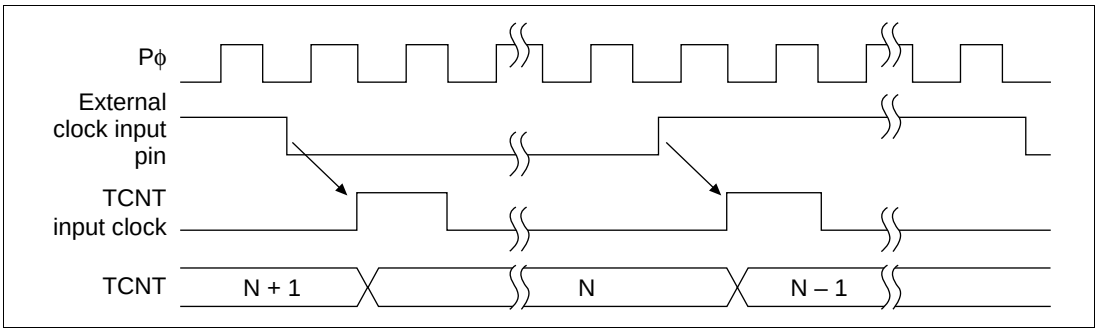


Figure 13.5 Count Timing when External Clock Is Operating (Both Edges Detected)

- On-Chip RTC Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the on-chip RTC clock as the timer clock. Figure 13.6 shows the timing.

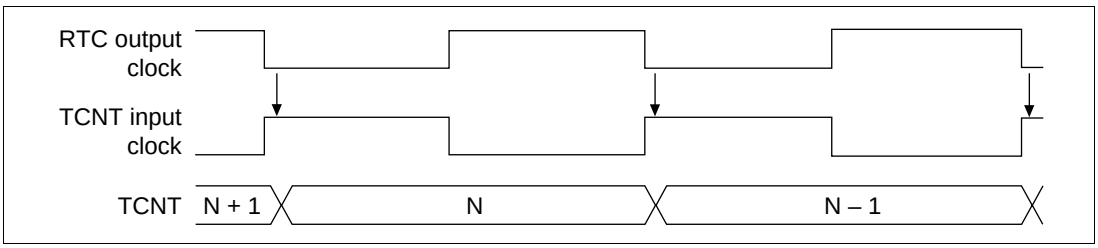


Figure 13.6 Count Timing when On-Chip RTC Clock Is Operating

Input Capture Function: Channel 2 has an input capture function (figure 13.7). 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 (TCR). Also, designate use of the input capture function and whether to generate interrupts on using it with the IPCE1–IPCE0 bits in TCR, and designate the use of either the rising or falling edge of the TCLK pin to set the timer counter (TNCT) value into the input capture register (TCPR) with the CKEG1–CKEG0 bits in TCR.

The input capture function cannot be used in standby mode.

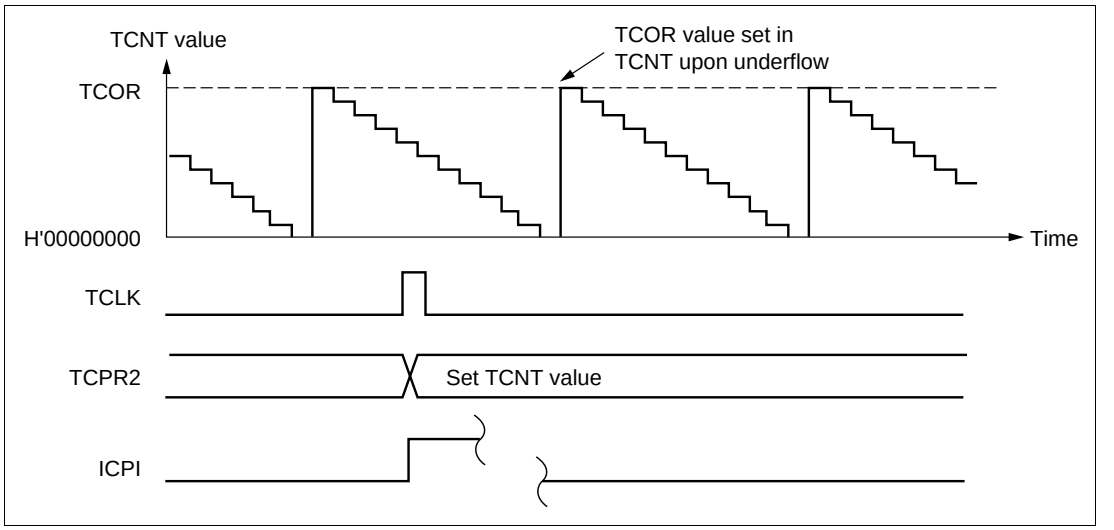


Figure 13.7 Operation Timing when Using the Input Capture Function (Using TCLK Rising Edge)

13.4 Interrupts

There are two sources of TMU interrupts: underflow interrupts and interrupts when using the input capture function.

13.4.1 Timing of Status Flag Setting

UNF is set to 1 when TCNT underflows (H'00000000 → H'FFFFFFFF). Figure 13.8 shows the timing.

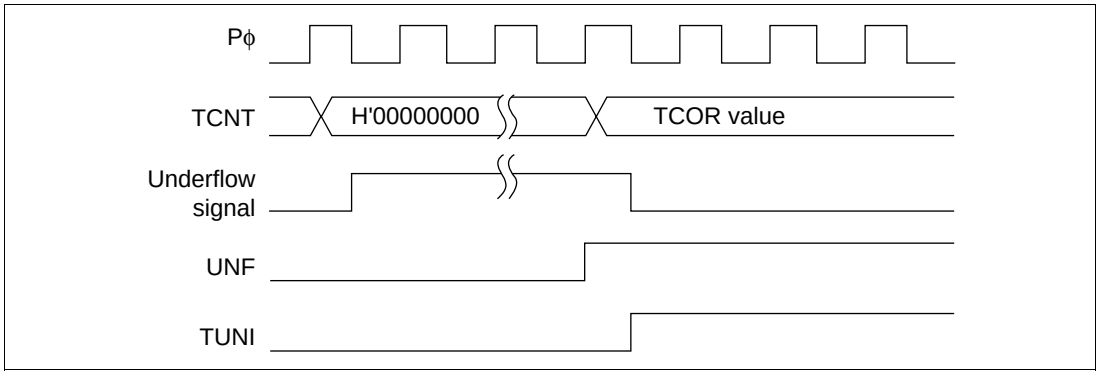


Figure 13.8 Timing of UNF Setting

13.4.2 Timing of Status Flag Clearing

The status flag can be cleared by writing 0 from the CPU. Figure 13.9 shows the timing.

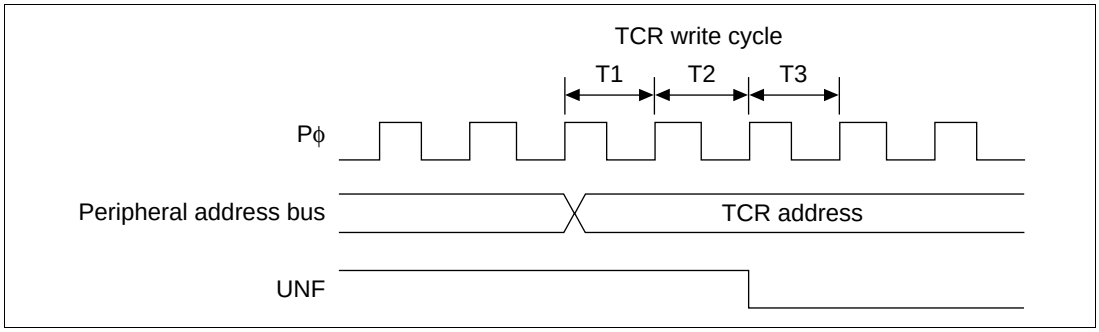


Figure 13.9 Timing of Status Flag Clearing

13.4.3 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 and INTEVT2) for these interrupts and interrupt handling is carried out according to the codes.

The relative priorities of channels can be changed using the interrupt controller (see section 6, Interrupt Controller). Table 13.3 lists TMU interrupt sources.

Table 13.3 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

13.5 Usage Notes

13.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits (STR2–STR0) for the channel in the timer start register (TSTR) to halt timer counting.

13.5.2 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 the TCNT count-down (with synchronization processing) is read.

Section 14 Real-Time Clock (RTC)

14.1 Overview

The SH7707 has a real-time clock (RTC) with its own 32.768-kHz crystal oscillator.

14.1.1 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

14.1.2 Block Diagram

The following abbreviations are used in the block diagram of the RTC (figure 14.1):

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 week alarm register
RWKCNT:	Day of 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

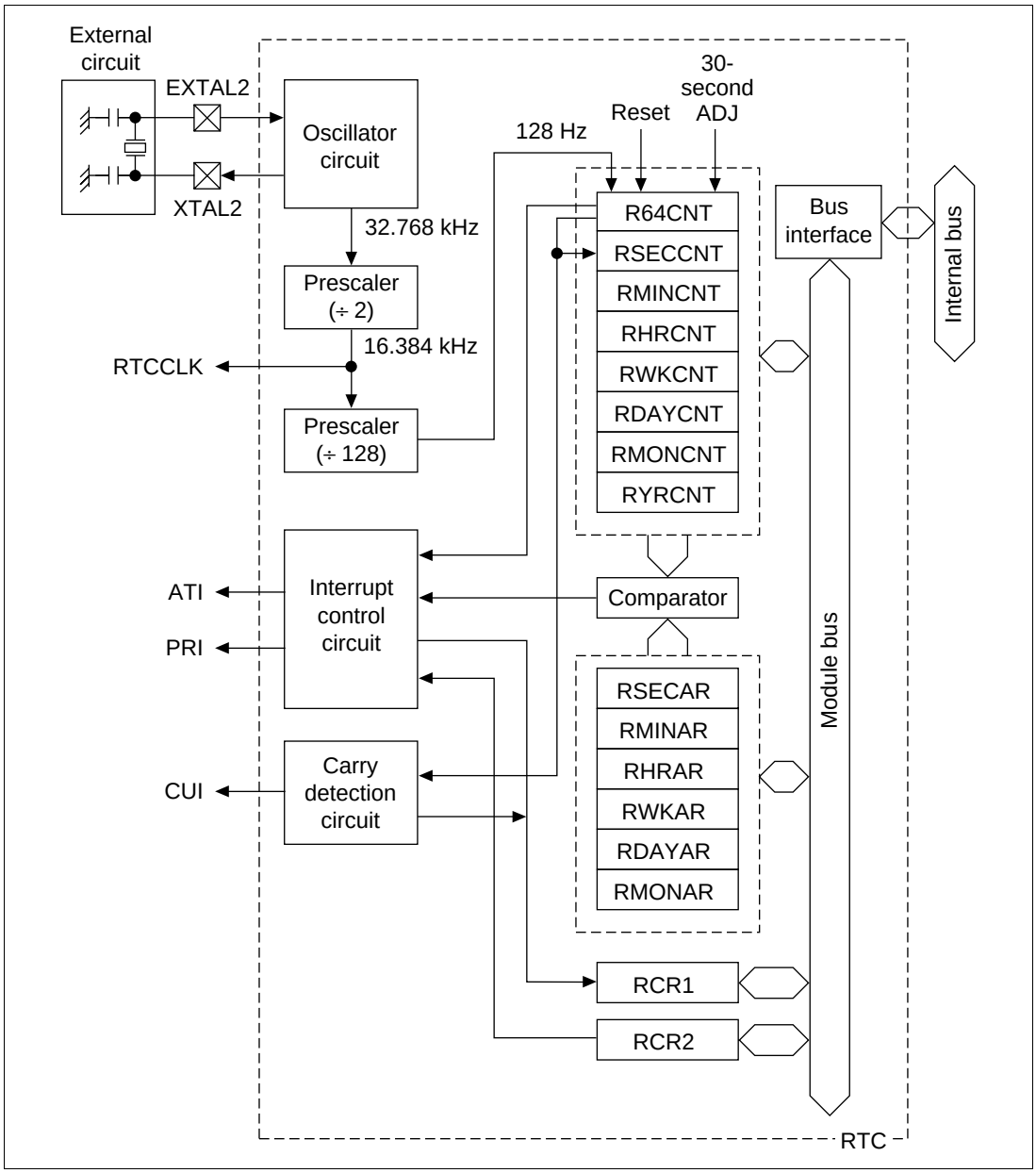


Figure 14.1 RTC Block Diagram

14.1.3 Pin Configuration

Table 14.1 shows the RTC pin configuration.

Table 14.1 RTC Pin Configuration

Pin	Abbreviation	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/real-time 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.

14.1.4 RTC Register Configuration

Table 14.2 shows the RTC register configuration.

Table 14.2 RTC Registers

Name	Abbreviation	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 bit in each register is initialized.

14.2 RTC Registers

14.2.1 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

14.2.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit readable/writable 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

14.2.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit readable/writable 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

14.2.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit readable/writable 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

14.2.5 Day of Week Counter (RWKCNT)

The day of week counter (RWKCNT) is an 8-bit readable/writable 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 14.3.

Table 14.3 Day of Week Codes (RWKCNT)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

14.2.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit readable/writable 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

14.2.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit readable/writable 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

14.2.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit readable/writable 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

14.2.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit readable/writable register used as 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 the ENB bit 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

14.2.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit readable/writable register used as 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 the ENB bit 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

14.2.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit readable/writable register used as 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 the ENB bit 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

14.2.12 Day of Week Alarm Register (RWKAR)

The day of week alarm register (RWKAR) is an 8-bit readable/writable register used as 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 the ENB bit 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 14.4.

Table 14.4 Day of Week Codes (RWKAR)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

14.2.13 Date Alarm Register (RDAYAR)

The date alarm register (RDAYAR) is an 8-bit readable/writable register used as 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 RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with the ENB bit 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

14.2.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit readable/writable register used as 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 RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter and alarm register comparison is performed only on those with the ENB bit 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

14.2.15 RTC Control Register 1 (RCR1)

The RTC control register 1 (RCR1) is an 8-bit readable/writable 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 enables 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 the 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.

14.2.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit readable/writable register for periodic interrupt control, 30-second adjustment, divider circuit resetting, and RTC count start/stop control. It is initialized to H'09 by a power-on reset. It is not initialized by a manual reset or 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 to 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	RTC crystal oscillator halted
1	RTC crystal oscillator runs (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	Normal operation	(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	Normal operation	(Initial value)
1	Divider circuit is reset	

Bit 0—Start Bit (START): Halts or 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.

14.3 RTC Operation

14.3.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

14.3.2 Setting the Time

Part (a) in figure 14.2 shows how to set the time when the clock is stopped. This is used when the entire calendar or clock is to be set.

Part (b) in figure 14.2 describes how to set the clock when the clock is running. This is used 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.

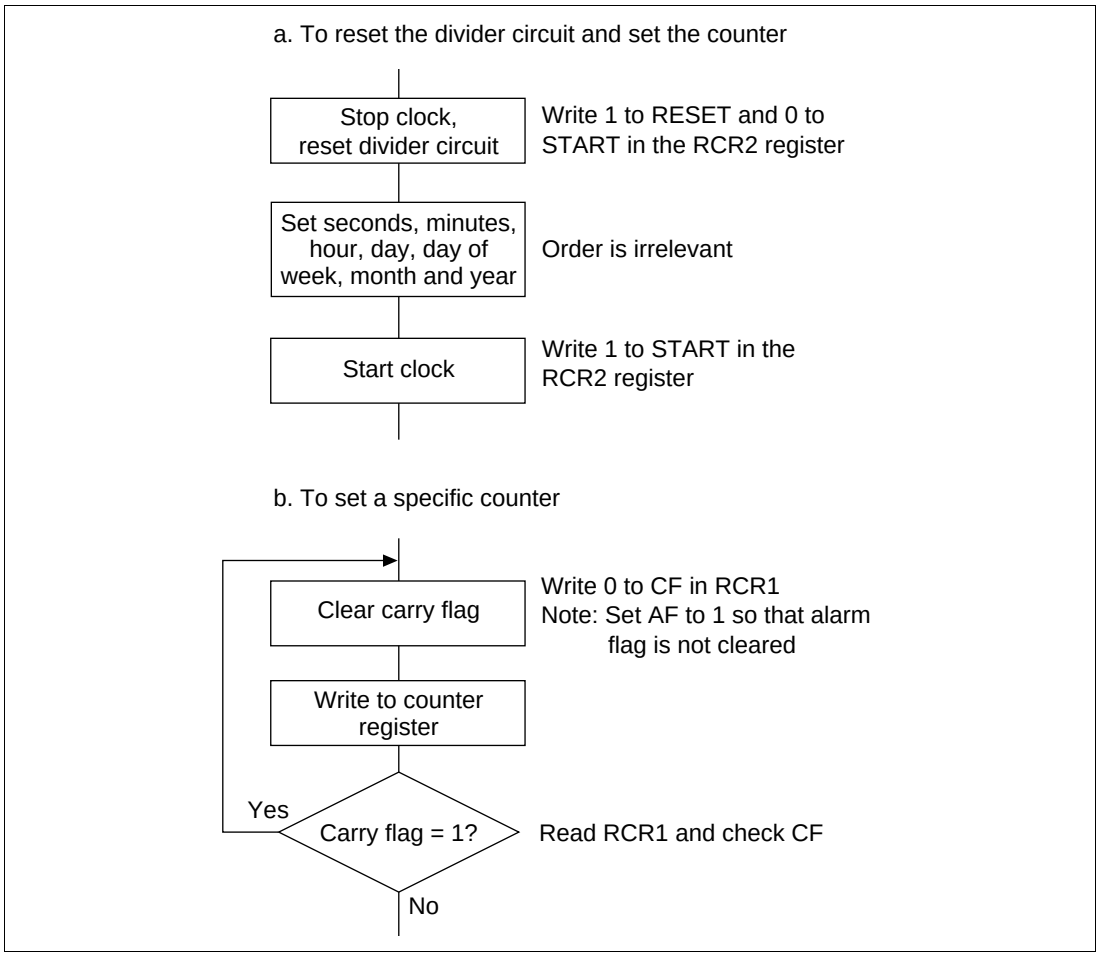


Figure 14.2 Setting the Time

14.3.3 Reading the Time

Figure 14.3 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 14.3 shows the method of reading the time without using interrupts; part (b) in figure 14.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

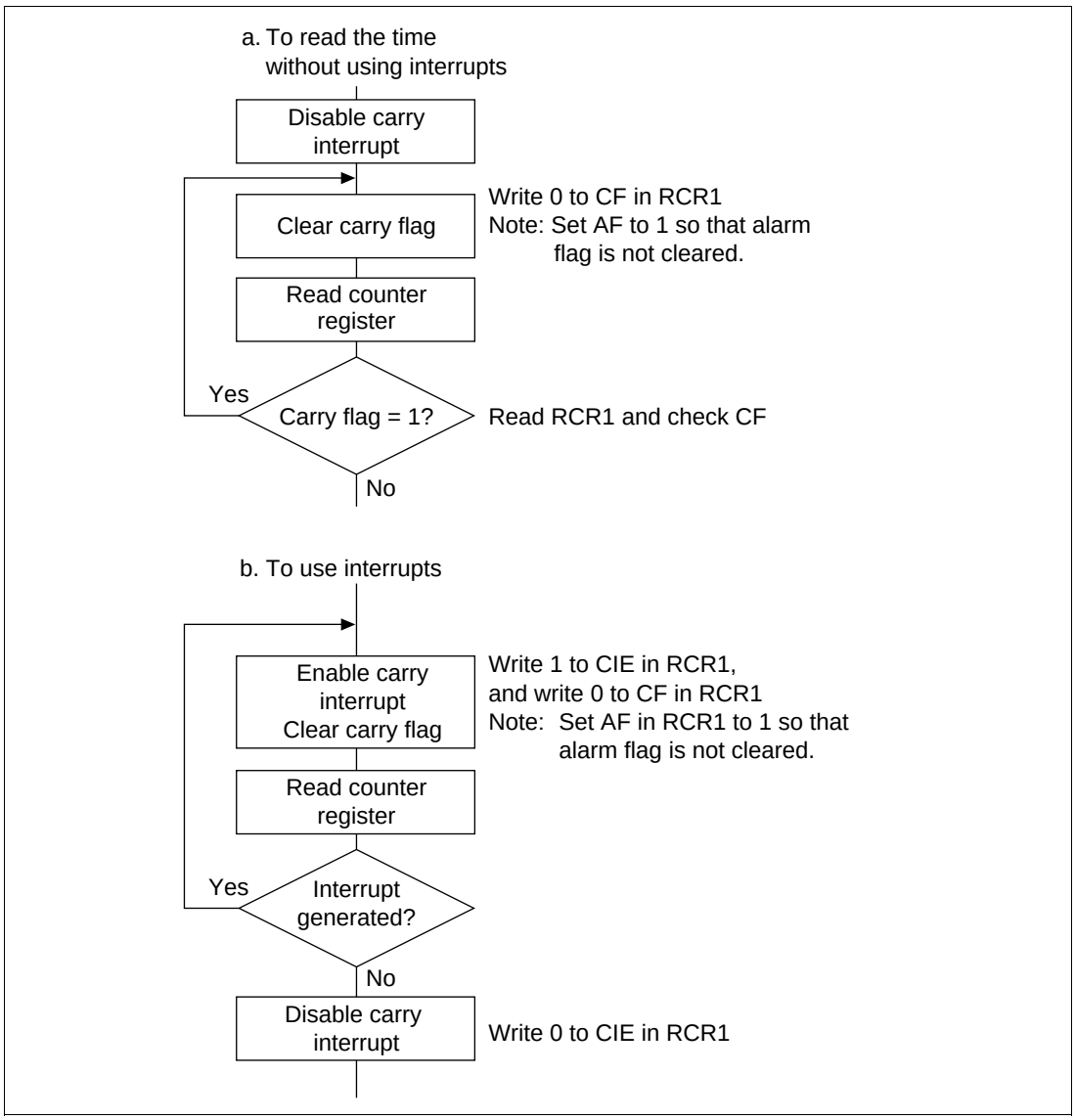


Figure 14.3 Reading the Time

14.3.4 Alarm Function

Figure 14.4 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 in which the alarm is set to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register in which the alarm is set 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 set in the AIE bit (bit 3) in RCR1, an interrupt is generated when an alarm occurs.

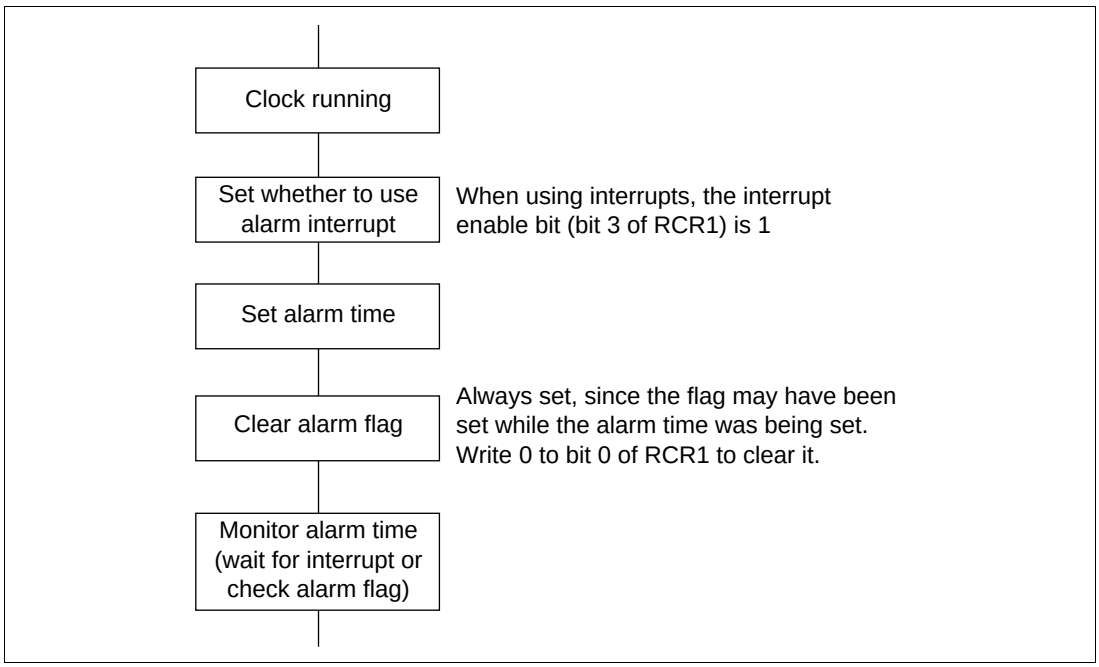


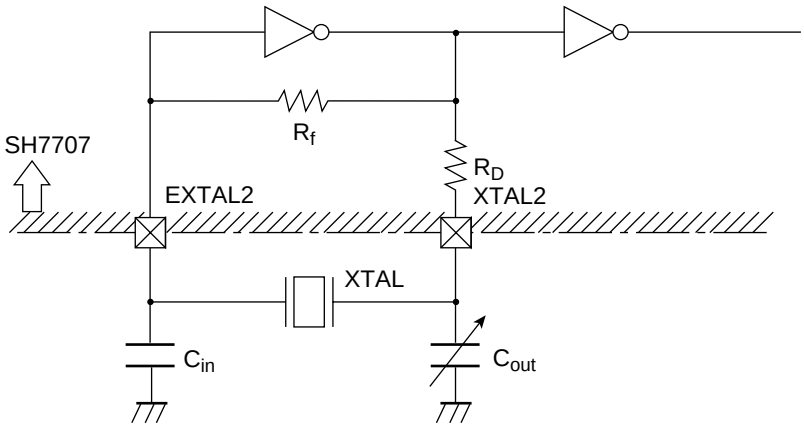
Figure 14.4 Using the Alarm Function

14.3.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 14.5, and the RTC crystal oscillator circuit in figure 14.5.

Table 14.5 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 the 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 (other than GND) and signal lines.

Figure 14.5 Example of Crystal Oscillator Circuit Connection

Section 15 Serial Communication Interface (SCI)

15.1 Overview

The SH7707 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 between two or more processors. The SCI supports a smart card interface, which is a serial communication feature for IC card interfaces that conforms to the ISO/IEC standard 7816-3 for identification cards. See section 16, Smart Card Interface, for more information.

15.1.1 Features

Select asynchronous or synchronous as the serial communications mode.

- Asynchronous mode:
  - Serial data communication is synchronized by the start-stop method 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
- Synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a 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.
- Built-in 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.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the SCI.

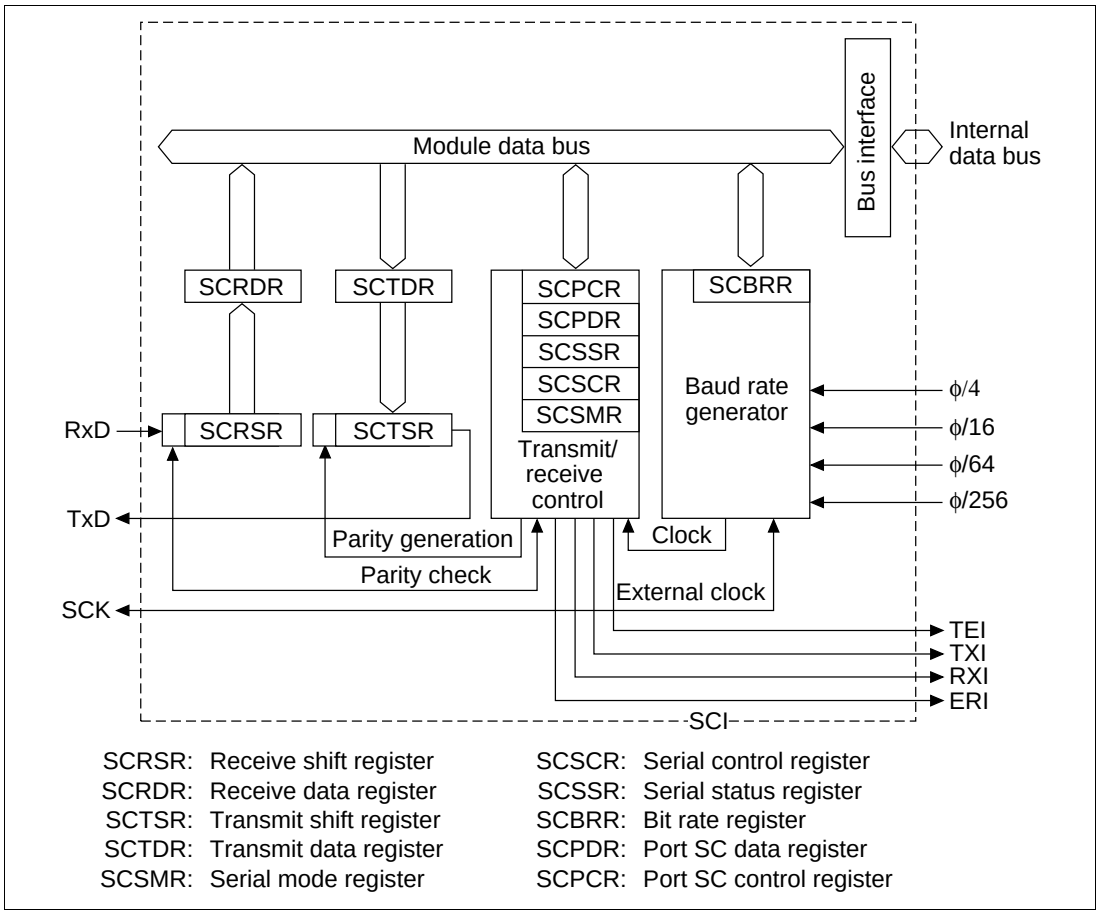
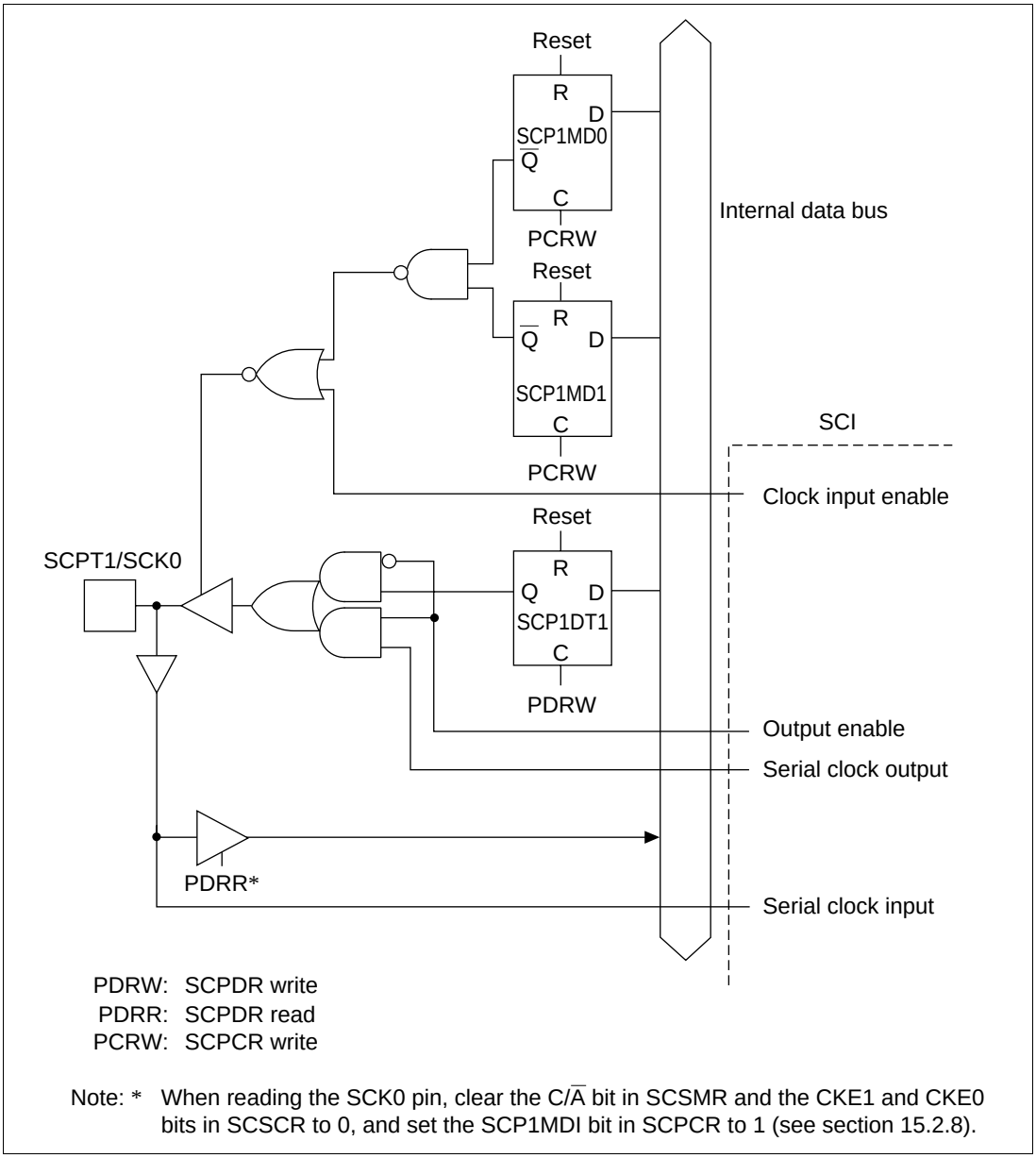


Figure 15.1 SCI Block Diagram

Figures 15.2, 15.3, and 15.4 show block diagrams of the SCI I/O port pins.



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 SCP1MDI bit in SCPCR to 1 (see section 15.2.8).

Figure 15.2 SCPT[1]/SCK0 Pin

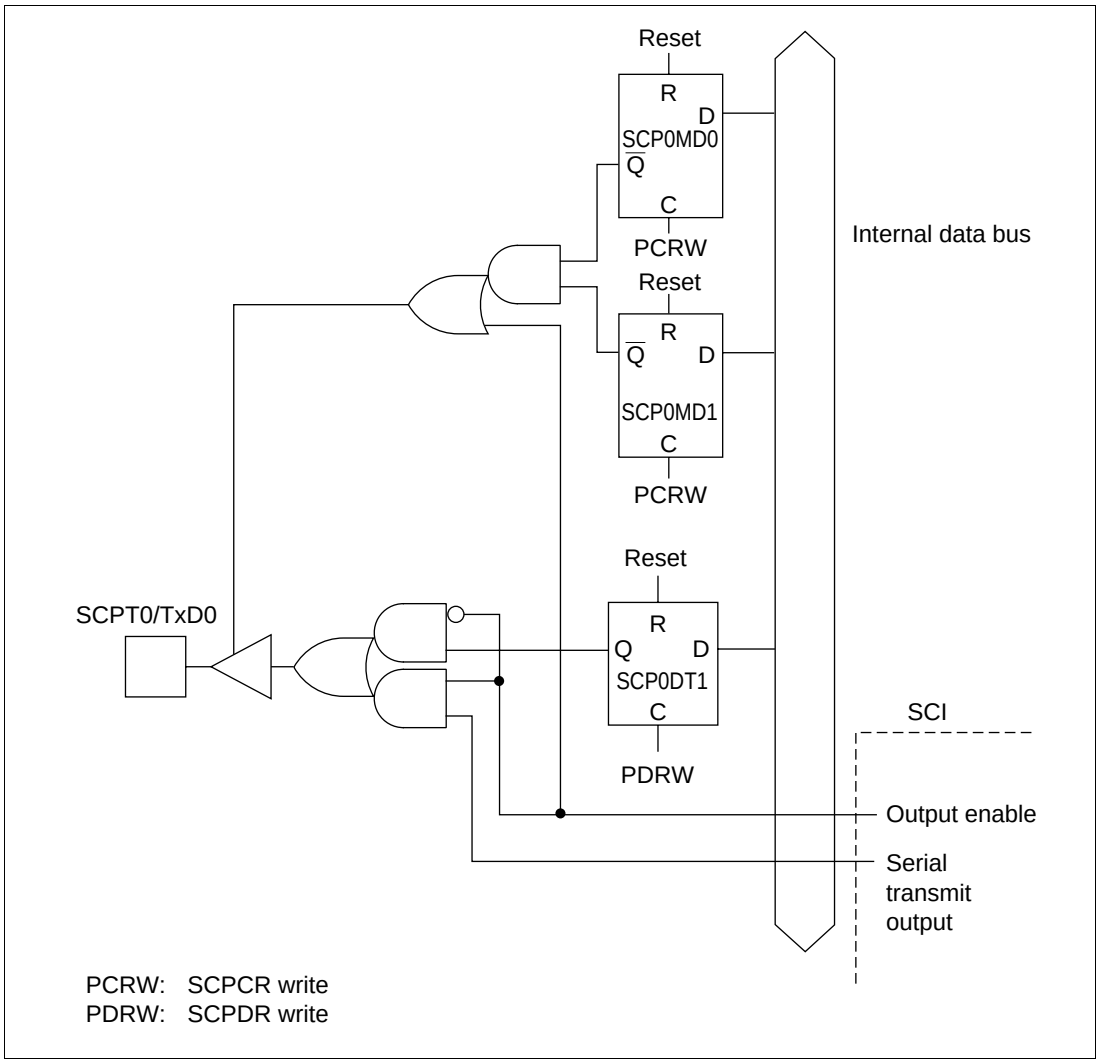


Figure 15.3 SCPT[0]/TxD0 Pin

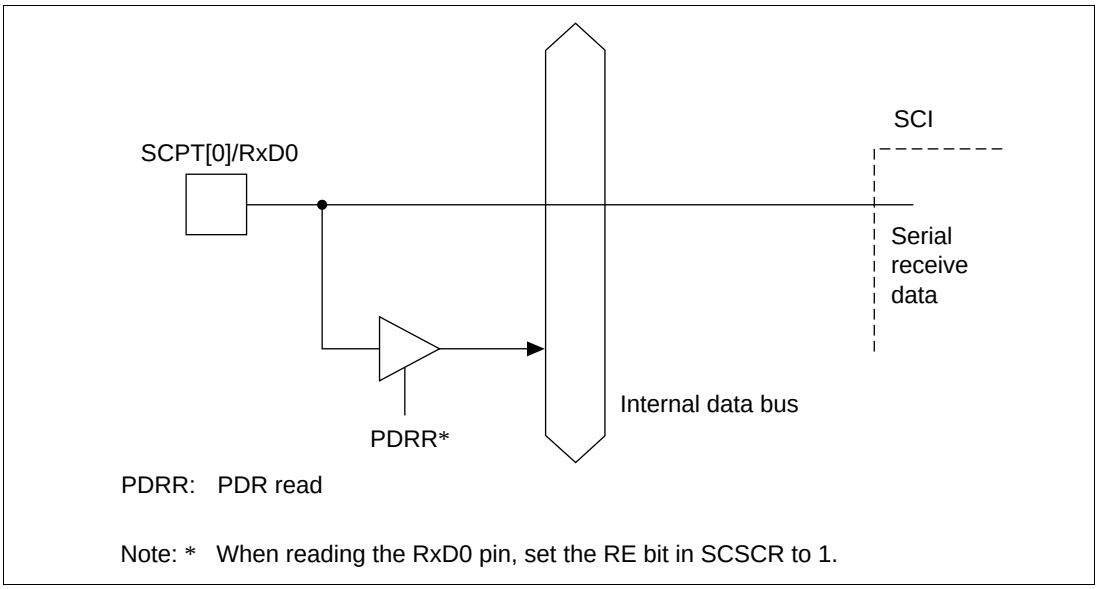


Figure 15.4 SCPT[0]/RxD0 Pin

15.1.3 Pin Configuration

The SCI has the serial pins summarized in table 15.1.

Table 15.1 SCI Pins

Pin Name	Abbreviation	I/O	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

15.1.4 Register Configuration

Table 15.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 15.2 SCI Registers

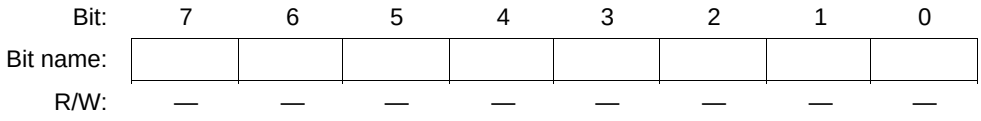
Address	Name	Abbreviation	R/W	Initial Value	Access Size
H'FFFFFE80	Serial mode register	SCSMR	R/W	H'00	8
H'FFFFFE82	Bit rate register	SCBRR	R/W	H'FF	8
H'FFFFFE84	Serial control register	SCSCR	R/W	H'00	8
H'FFFFFE86	Transmit data register	SCTDR	R/W	H'FF	8
H'FFFFFE88	Serial status register	SCSSR	R/(W)*	H'84	8
H'FFFFFE8A	Receive data register	SCRDR	R	H'00	8
H'4000136	Port SC data register	SCPDR	R/W	H'00	8
H'4000116	Port SC control register	SCPCR	R/W	H'A888	16

Note: Only 0 can be written, to clear the flags.

15.2 Register Descriptions

15.2.1 Receive Shift Register

The receive shift register (SCRSR) receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCRDR. The CPU cannot read or write to SCRSR directly.



15.2.2 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 SCRDR for storage. 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 to SCRDR. SCRDR is initialized to H'00 by a reset and in standby and 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

15.2.3 Transmit Shift Register

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR) into 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 SCTDR into SCTSR and starts transmitting again. If the TDRE bit in SCSSR is 1, however, the SCI does not load the SCTDR contents into SCTSR. The CPU cannot read or write to SCTSR directly.

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

15.2.4 Transmit Data Register

The transmit data register (SCTDR) is an 8-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 SCTDR into SCTSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in SCTDR during serial transmission from SCTSR.

The CPU can always read and write to SCTDR. SCTDR is initialized to H'FF by a reset and 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

15.2.5 Serial Mode Register

The serial mode register (SCSMR) is an 8-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 to SCSMR. SCSMR is initialized to H'00 by a reset and in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
Bit name:	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	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/$\overline{A}$): Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7: C/$\overline{A}$	**Description**
0	Asynchronous mode (Initial value)
1	Synchronous mode

Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data in asynchronous mode. In synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	**Description**
0	8-bit data (Initial value)
1	7-bit data. (When 7-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 asynchronous mode. In 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/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) 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 synchronous mode, and in asynchronous mode when parity addition and check is disabled.

Bit 4: O/$\overline{E}$	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 asynchronous mode. This setting is used only in asynchronous mode. It is ignored in 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 1-bit is added at the end of each transmitted character
1	Two stop bits In transmitting, two 1-bits 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 asynchronous mode; it is ignored in synchronous mode. For the multiprocessor communication function, see section 15.3.3, 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, CKS0): These bits select the internal clock source of the built-in baud rate generator. Four clock sources are available: Pϕ, Pϕ/4, Pϕ/16, and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 15.2.9, 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

15.2.6 Serial Control Register

The serial control register (SCSCR) operates the SCI transmitter/receiver, selects the serial clock output in asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'00 by a reset and 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 SCTDR to 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 SCRSR to 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 fixed 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 SCTDR. Select the transmit format in 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 asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receive format in SCSMR before setting RE to 1

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SCSMR) is set to 1. The MPIE setting is ignored in synchronous mode and when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (Initial value) MPIE is cleared to 0 when 0 is written in MPIE, 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 SCRSR to 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, 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 asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). Set CKE1 and CKE0 before selecting the SCI operating mode in the serial mode register (SCSMR). For further details on selection of the SCI clock source, see table 15.9 in section 15.3, 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)
		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
		Synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*2
		Synchronous mode	External clock, SCK pin used for synchronous clock input
	1	Asynchronous mode	External clock, SCK pin used for clock input*2
		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.

15.2.7 Serial Status Register

The serial status register (SCSSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate the SCI operating status.

The CPU can always read and write to 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. SCSSR is initialized to H'84 by a reset and 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: Only 0 can be written, to clear the flag.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from SCTDR into SCTSR and new serial transmit data can be written in 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 when 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, or when the TE bit in the serial control register (SCSCR) is cleared to 0, or when 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 receive data.

Bit 6: RDRF	Description
0	SCRDR does not contain valid receive data (Initial value) RDRF is cleared to 0 when the chip is reset or enters standby mode, or when software reads RDRF after it has been set to 1, then writes 0 in RDRF, or when data is read from SCRDR
1	SCRDR contains valid receive data RDRF is set to 1 when serial data is received normally and transferred from SCRSR to SCRDR

Note: 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*[1] (Initial value) ORER is cleared to 0 when the chip is reset or enters standby mode, or when 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 synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception aborted due to a framing error in 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 when 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 SCRDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In 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 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 when 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 SCRDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In 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, SCTDR did not contain valid data, so transmission has ended. TEND is a read-only bit.

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, 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 asynchronous mode. MPB is a read-only bit.

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 asynchronous mode. The MPBT setting is ignored in 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

15.2.8 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	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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/$\overline{A}$ bit in SCSMR and bits CKE1 and CKE0 in SCSCR to 0.

Bit 3: SCP1MD1	**Bit 2: 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 in SCSCR to 0.

Bit 1: SCP0MD1	Bit 0: 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. If the RE bit in SCSCR is set to 1, the RxD pin value can be read from the SCP0DT bit in SCPDR regardless of the values of the SCP0MD0 and SCP0MD1 bits in SCPCR. 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 port pins are shown in figures 15.2, 15.3, and 15.4.

15.2.9 Bit Rate Register (SCBRR)

The bit rate register (SCBRR) is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a reset and in module standby and standby modes. 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: $N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

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 ($0 \leq N \leq 255$)
$P\phi$: Operating frequency for supporting modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 15.3)

Table 15.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	$P\phi$	0	0
1	$P\phi/4$	0	1
2	$P\phi/16$	1	0
3	$P\phi/64$	1	1

Note: The bit rate error for asynchronous mode is given by the following formula:
Error (%) $= \{P\phi \times 10^6/[(N + 1) \times B \times 64 \times 2^{2n-1}] - 1\} \times 100$

Table 15.4 lists examples of SCBRR settings in asynchronous mode; table 15.5 lists examples of SCBRR settings in synchronous mode.

Table 15.4 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 15.4 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 15.4 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

	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 15.4 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 15.5 Bit Rates and SCBRR Settings in Synchronous Mode

Bit Rate (bits/s)	P φ (MHz)									
	4		8		16		28.7		30	
	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 transmission/reception not possible

Table 15.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is being used. Tables 15.7 and 15.8 list the maximum rates for external clock input.

Table 15.6 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 15.7 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 15.8 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

15.3 Operation

15.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. Asynchronous/synchronous mode and the transmission format are selected in the serial mode register (SCSMR), as shown in table 15.9. The SCI clock source is selected by the combination of the C/$\overline{A}$ bit in the serial mode register (SCSMR) and the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 15.10.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as 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 built-in baud rate generator, and can output a serial clock signal with a frequency equal to the bit rate.
  - — When an external clock is selected, the external clock input must have a frequency of 16 times the bit rate. (The built-in baud rate generator is not used.)

Synchronous Mode:

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - — When an internal clock is selected, the SCI operates using the built-in baud rate generator, and outputs a synchronous clock signal to external devices.
  - — When an external clock is selected, the SCI operates on the input serial clock. The built-in baud rate generator is not used.

Table 15.9 Serial Mode Register Settings and SCI Communication Formats

	SCSMR Settings					SCI Communication Format			
Mode	Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multipro-cessor 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
Synchronous	1	*	*	*	*	8-bit		Not set	None

Note: Asterisks (*) indicate don't-care bits.

Table 15.10 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSMR	SCSCR Settings		SCI Transmit/Receive Clock	
Mode	Bit 7 C/$\overline{A}$	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		
Synchronous mode	1	0	0	Internal	Outputs the serial clock
			1		
		1	0	External	Inputs the serial clock
			1		

15.3.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the 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 15.5 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 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 of 16 times the bit rate. Receive data is latched at the center of each bit.

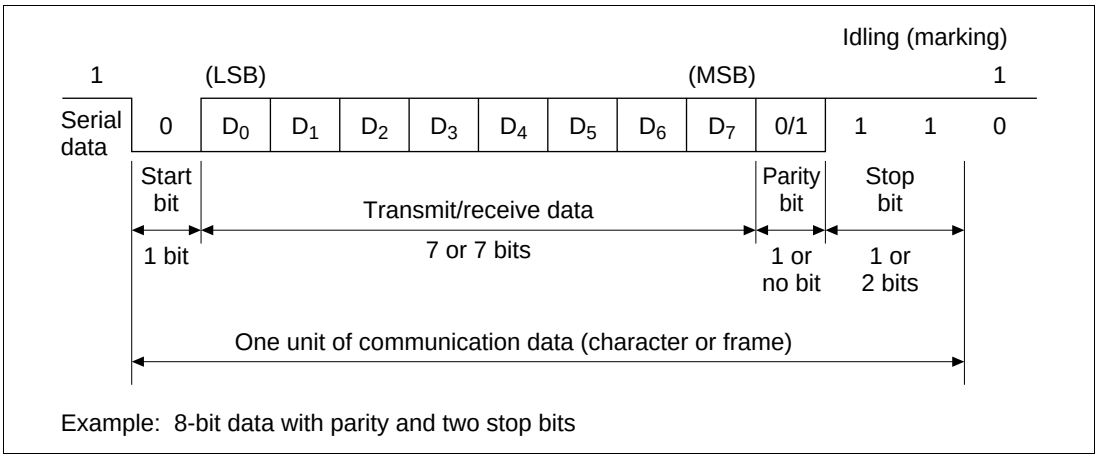


Figure 15.5 Data Format in Asynchronous Communication

Transmit/Receive Formats: Table 15.11 lists the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 15.11 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	
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 built-in 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 15.10).

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 15.6 so that the rising edge of the clock occurs at the center of each transmit data bit.

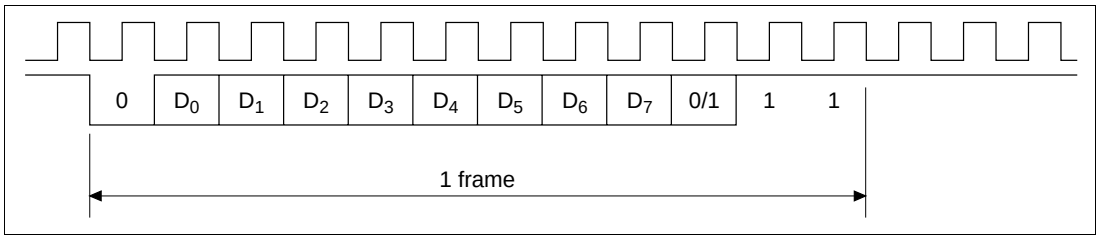


Figure 15.6 Output Clock and Serial Data Timing (Asynchronous Mode)

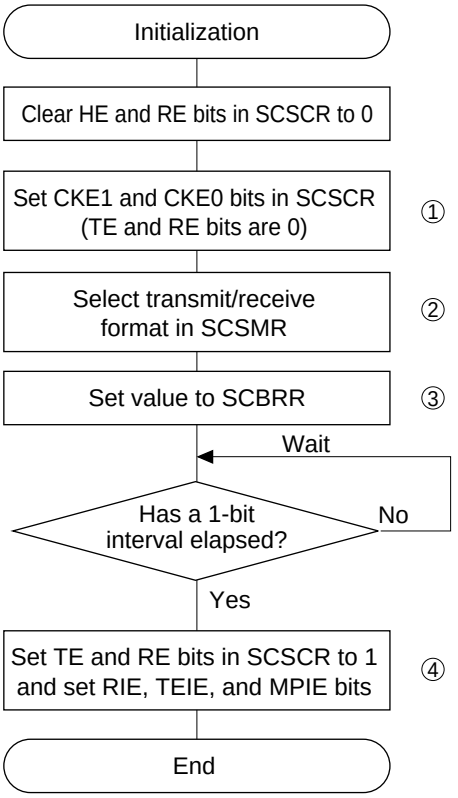
Transmitting and Receiving Data (SCI Initialization (Asynchronous Mode)): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

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 15.7 shows 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 in 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).

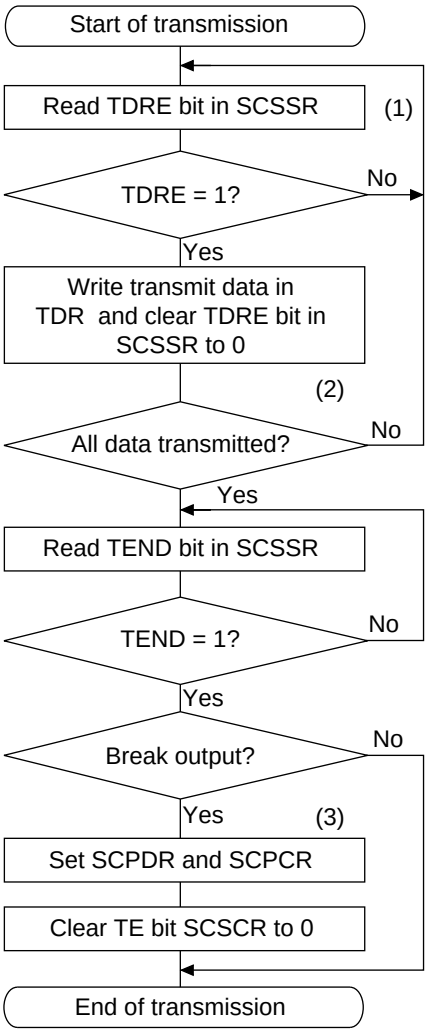


Note: Circled numbers refer to the description of the procedure in the text.

Figure 15.7 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 15.8 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 15.2.8.



Note: Circled numbers refer to the description of the procedure in the text.

Figure 15.8 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in SCSSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from SCTDR into the transmit shift register (SCTSR).
2. After loading the data from SCTDR into 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 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 SCTDR into 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 SCSSR, outputs the stop bit, then continues output of 1-bits (marking). If the transmit-end interrupt enable bit (TEIE) in SCSCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 15.9 shows an example of SCI transmit operation in asynchronous mode.

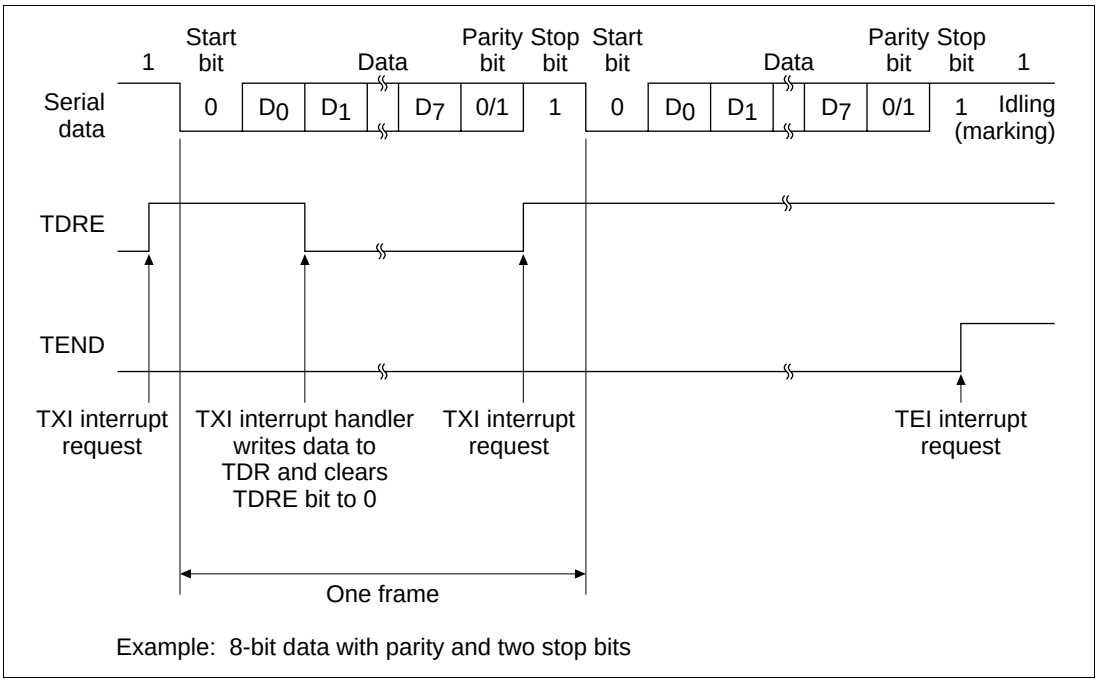
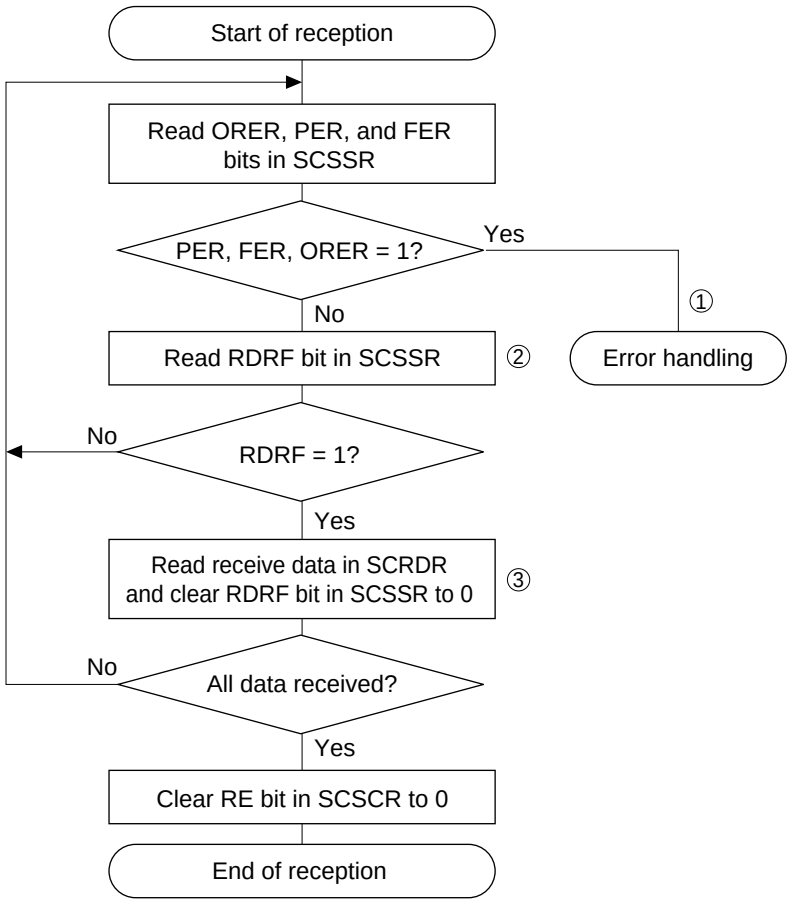


Figure 15.9 SCI Transmit Operation in Asynchronous Mode

Receiving Serial Data (Asynchronous Mode): Figure 15.10 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 in 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 remains 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.



Note: Circled numbers refer to the description of the procedure in the text.

Figure 15.10 Sample Flowchart for Receiving Serial Data

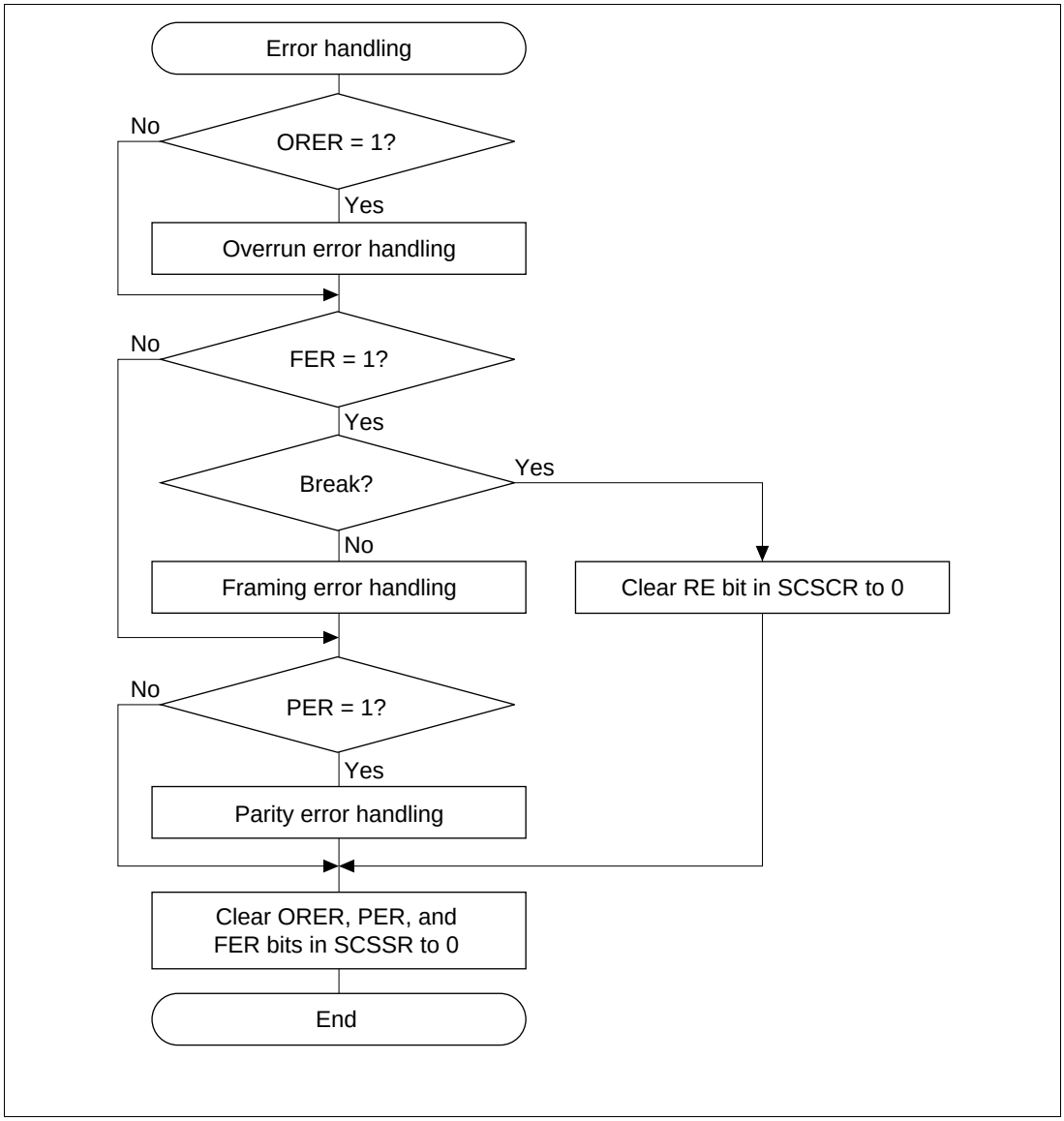


Figure 15.10 Sample Flowchart for Receiving Serial Data (cont)

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 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/$\overline{E}$ bit in 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 SCRSR into SCRDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in SCRDR. If one of the checks fails (receive error), the SCI operates as indicated in table 15.12.

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 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 SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 15.12 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 15.11 shows an example of SCI receive operation in asynchronous mode.

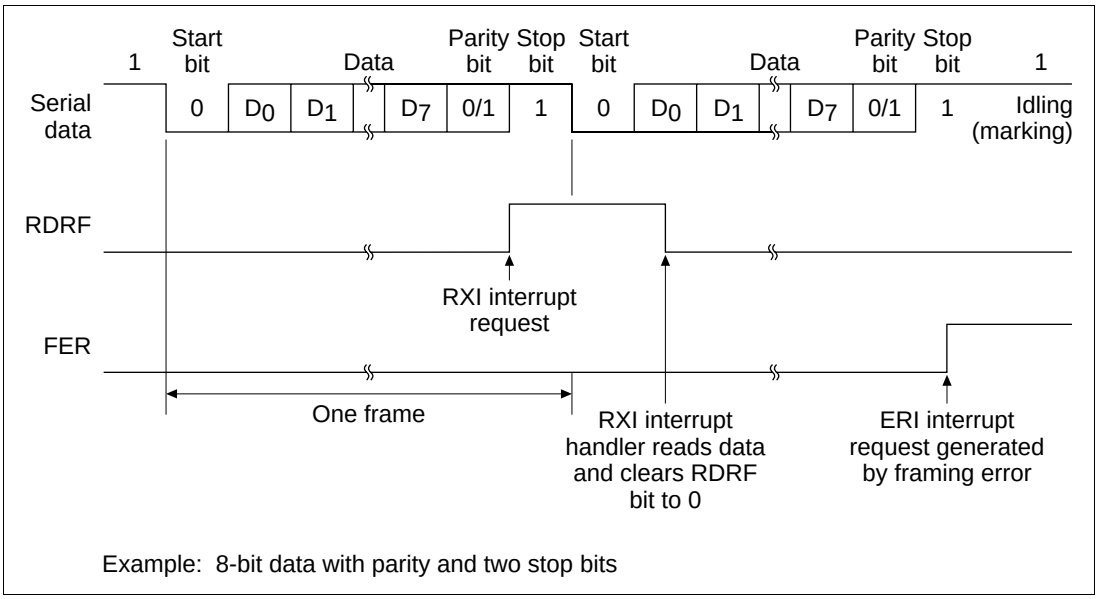


Figure 15.11 SCI Receive Operation

15.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in 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. They then compare the receive 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 15.12 shows an example of communication among processors using a multiprocessor format.

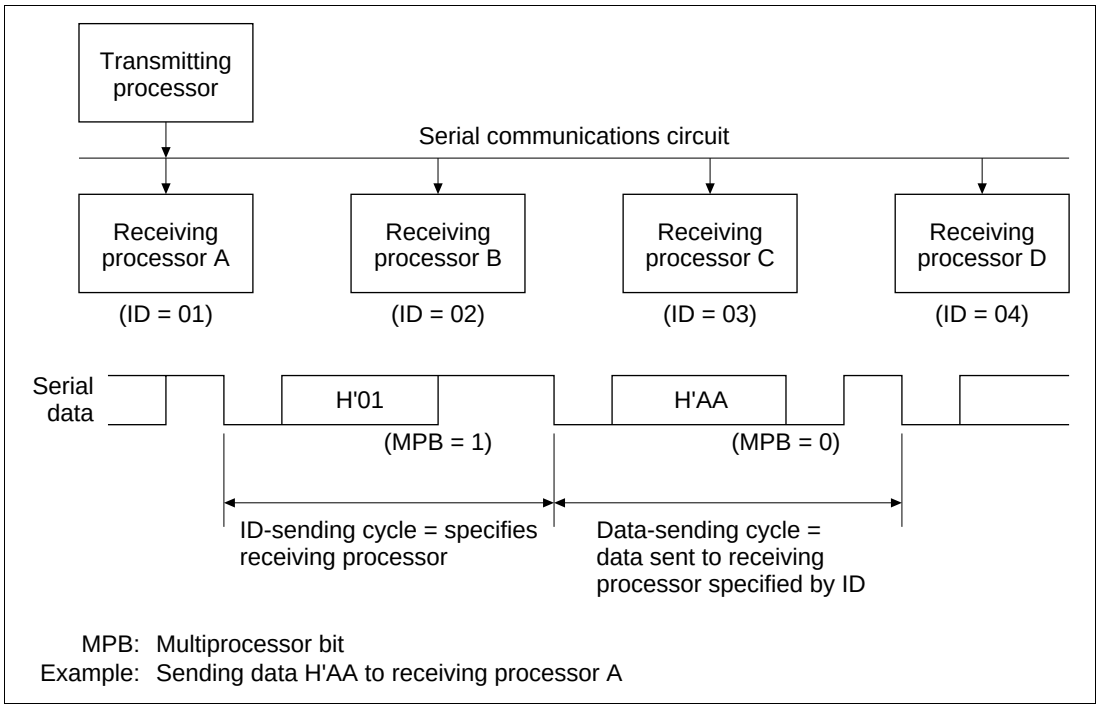


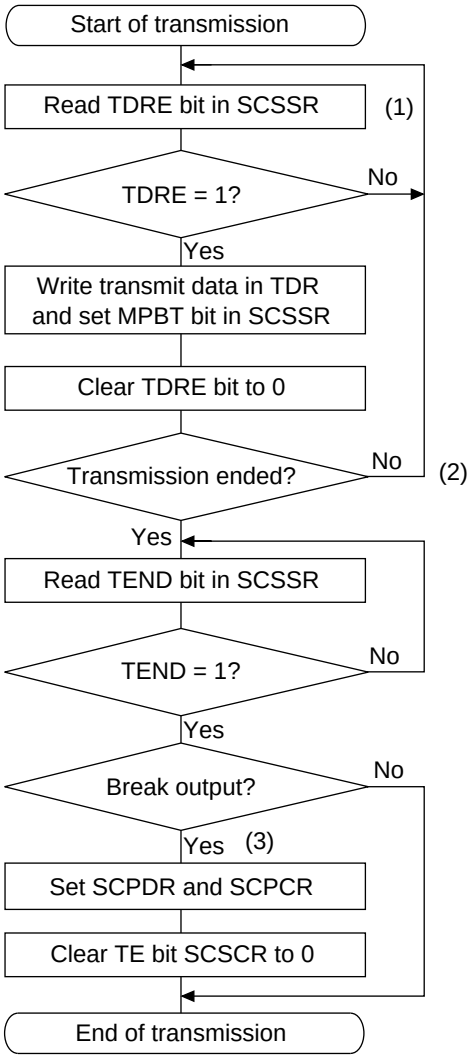
Figure 15.12 Communication among Processors Using Multiprocessor Format

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 15.11.

Clock: See the description in the asynchronous mode section.

Transmitting Multiprocessor Serial Data: Figure 15.13 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 15.2.8.



Note: Circled numbers refer to the description of the procedure in the text.

Figure 15.13 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from SCTDR into the transmit shift register (SCTSR).
2. After loading the data from SCTDR into SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in 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 SCTDR into SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in 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 SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 15.14 shows SCI transmission with a multiprocessor format.

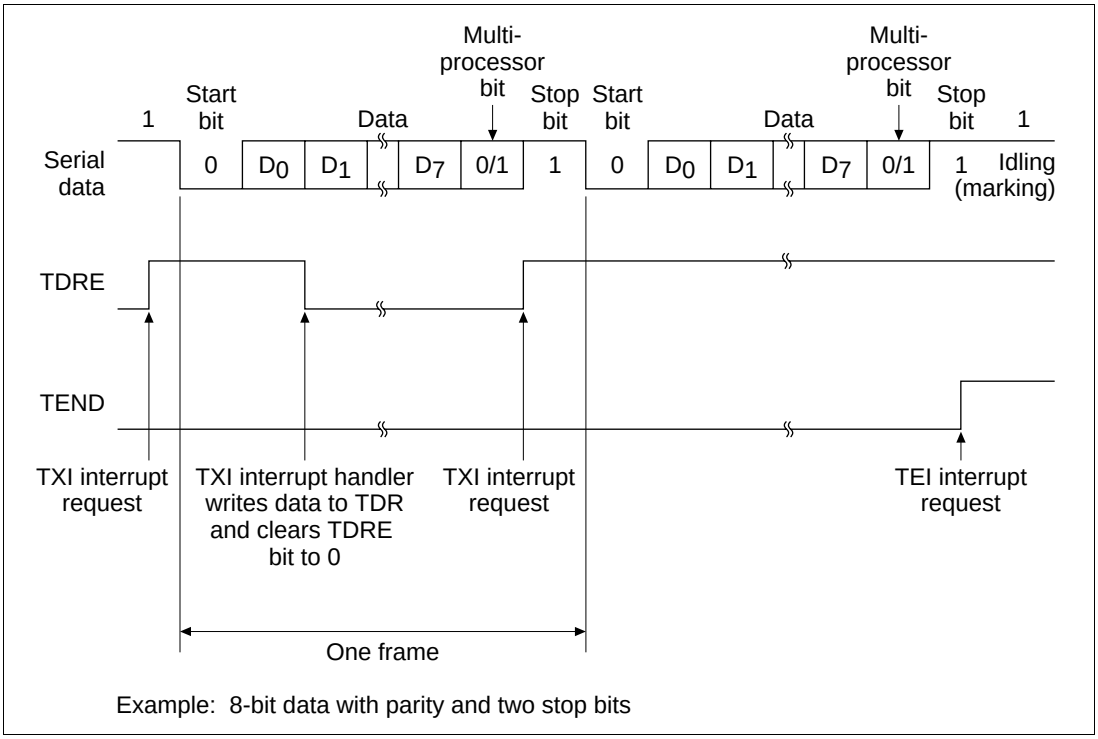


Figure 15.14 SCI Multiprocessor Transmit Operation

Receiving Multiprocessor Serial Data: Figure 15.15 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 handling, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remains set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.

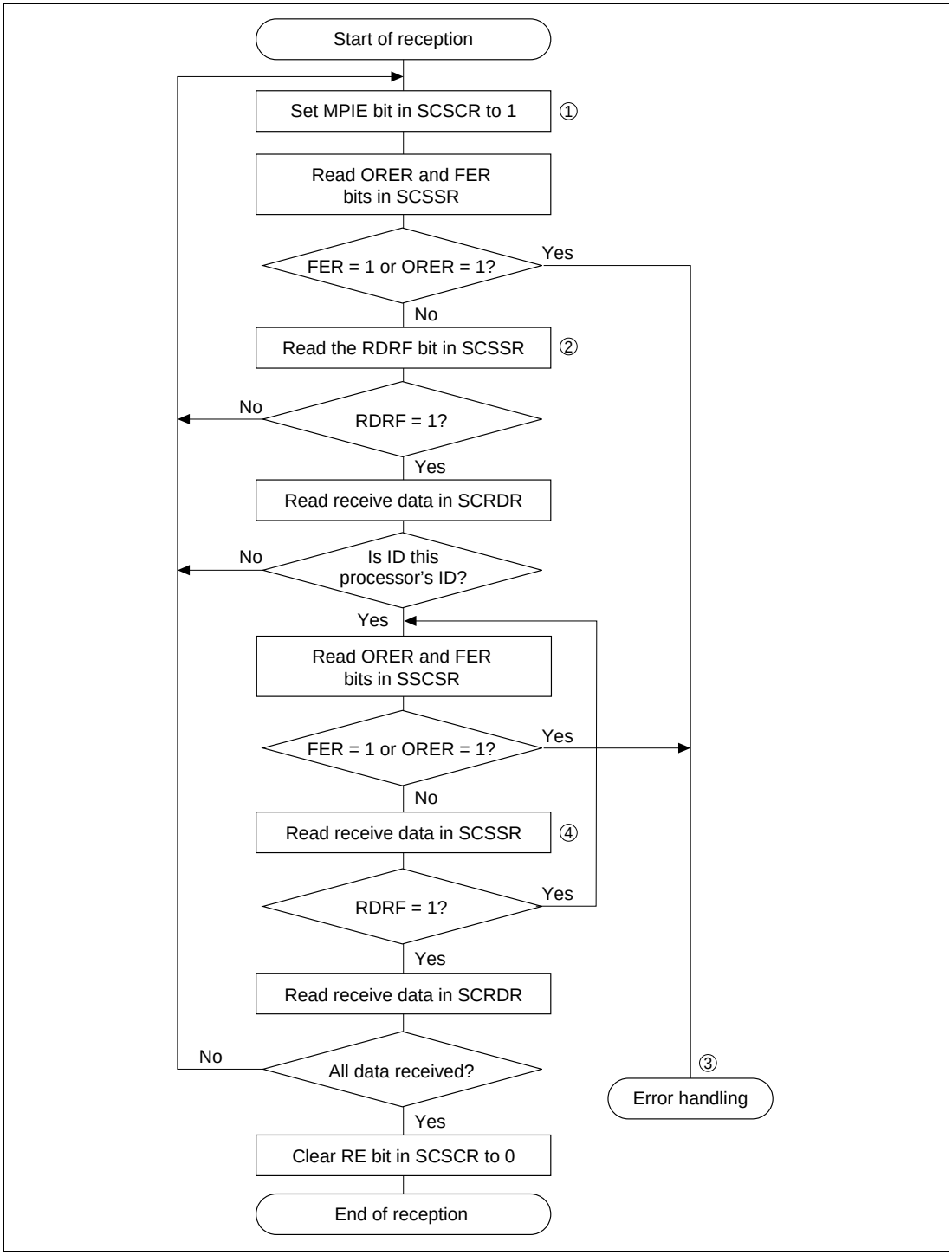


Figure 15.15 Sample Flowchart for Receiving Multiprocessor Serial Data

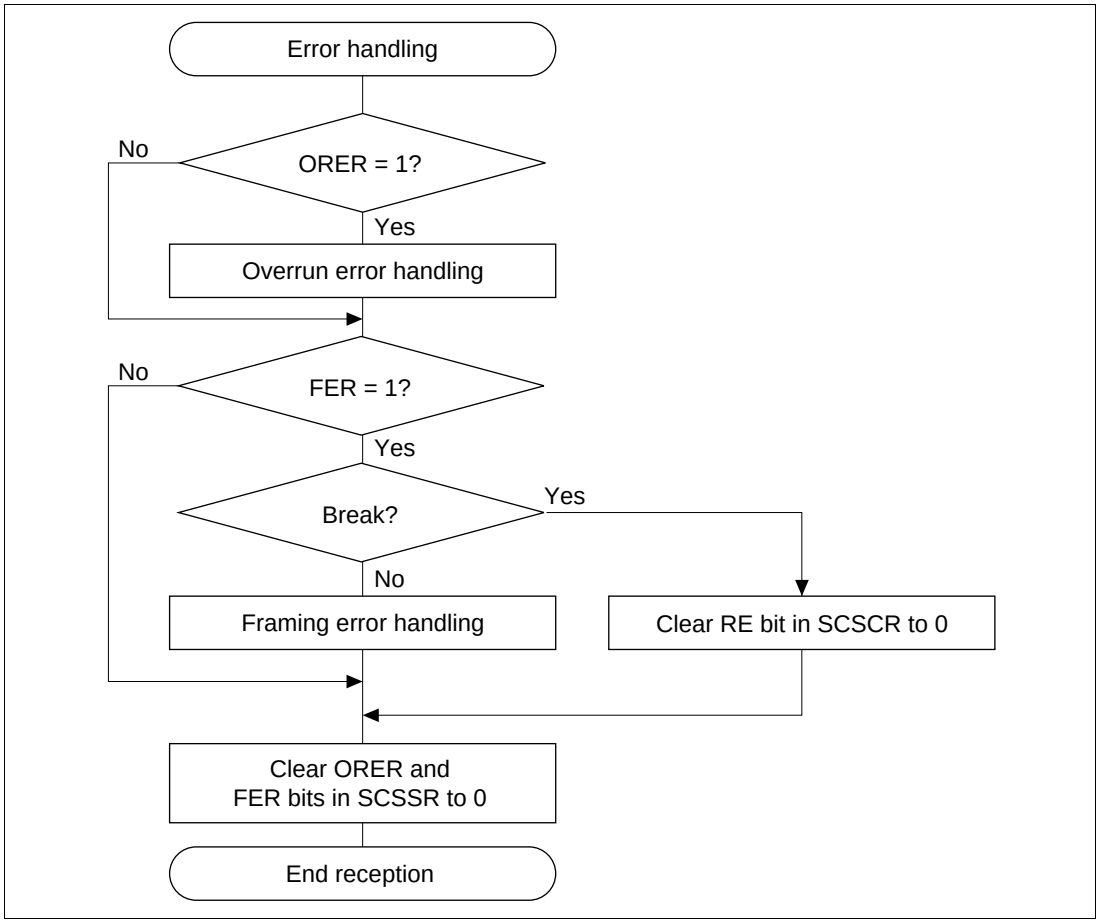
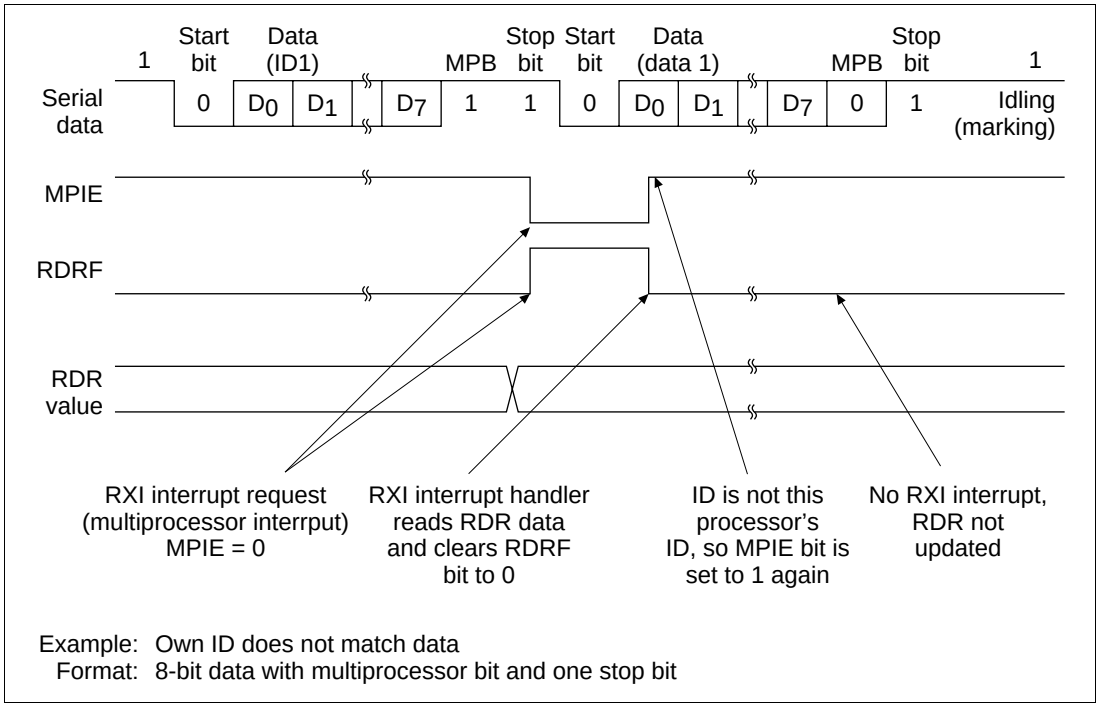


Figure 15.15 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figure 15.16 shows an example of SCI receive operation using a multiprocessor format.



Example: Own ID does not match data
Format: 8-bit data with multiprocessor bit and one stop bit

Figure 15.16 Example of SCI Receive Operation

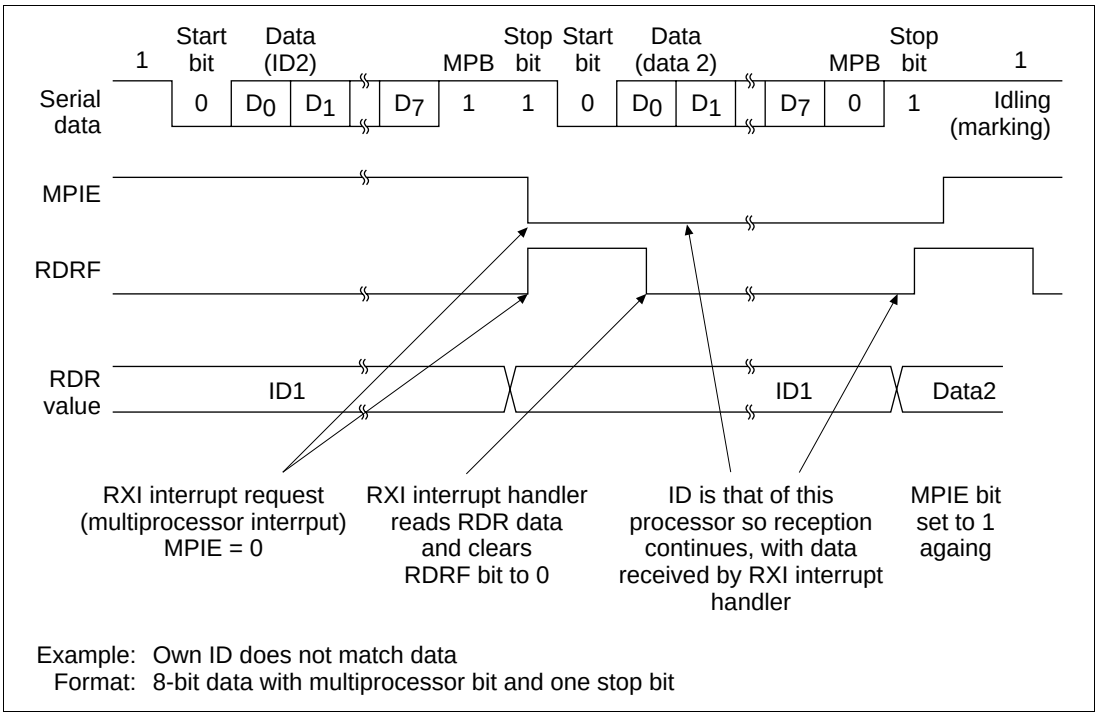


Figure 15.16 Example of SCI Receive Operation (cont)

15.3.4 Synchronous Operation

In 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 15.17 shows the general format in synchronous serial communication.

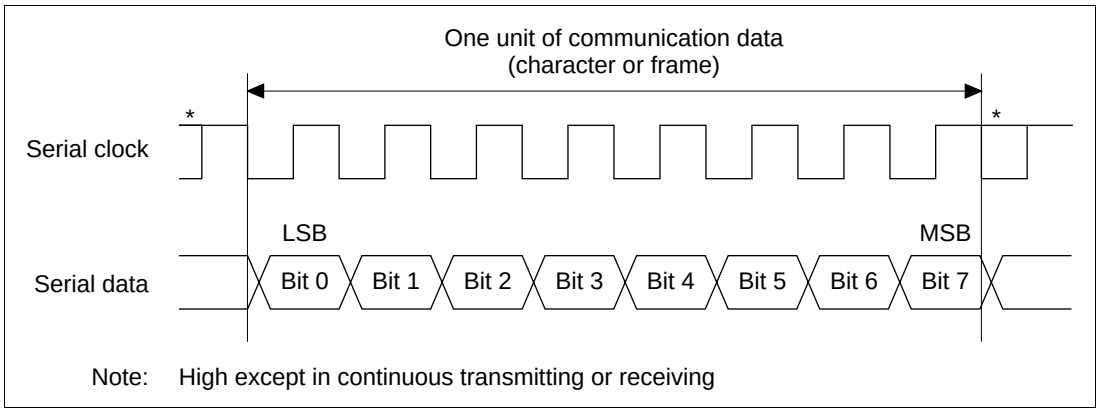


Figure 15.17 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data 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 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 built-in 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 15.10.

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 serial clock is output. To receive in 1-character units, select an external clock source.

Transmitting and Receiving Data: SCI Initialization (Synchronous Mode): Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the 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 15.18 shows 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.

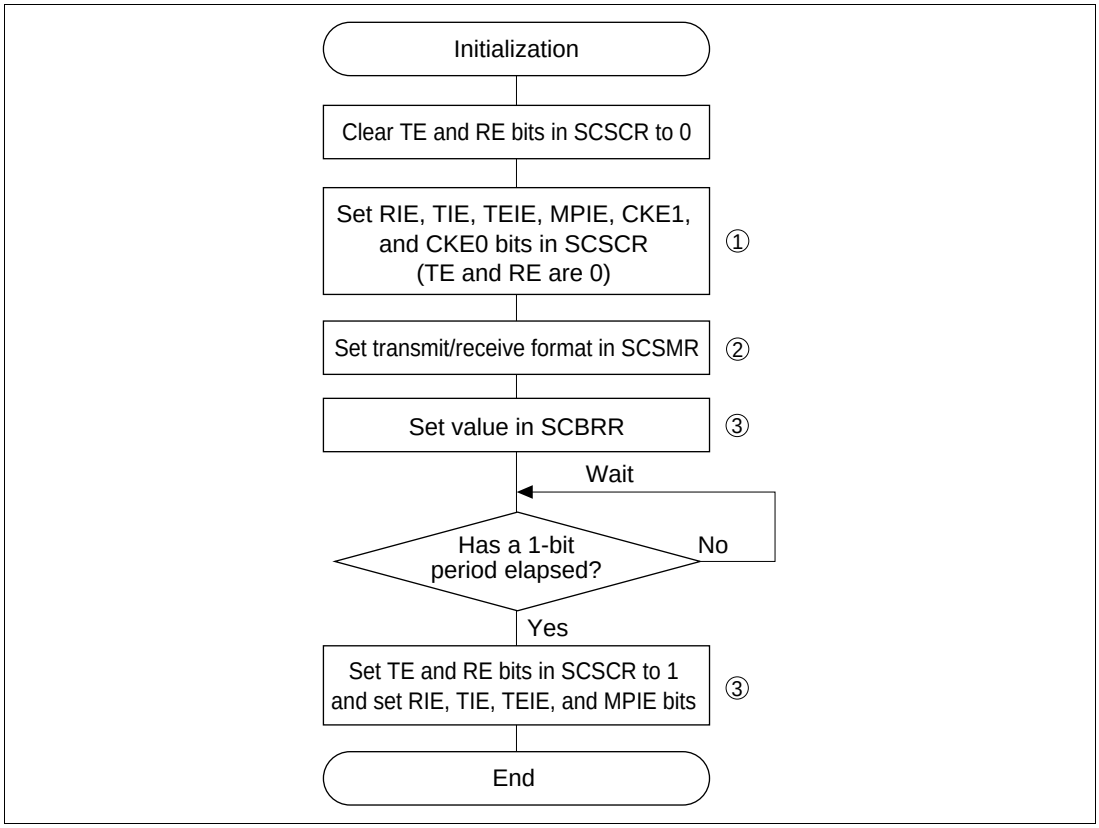


Figure 15.18 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 15.19 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.

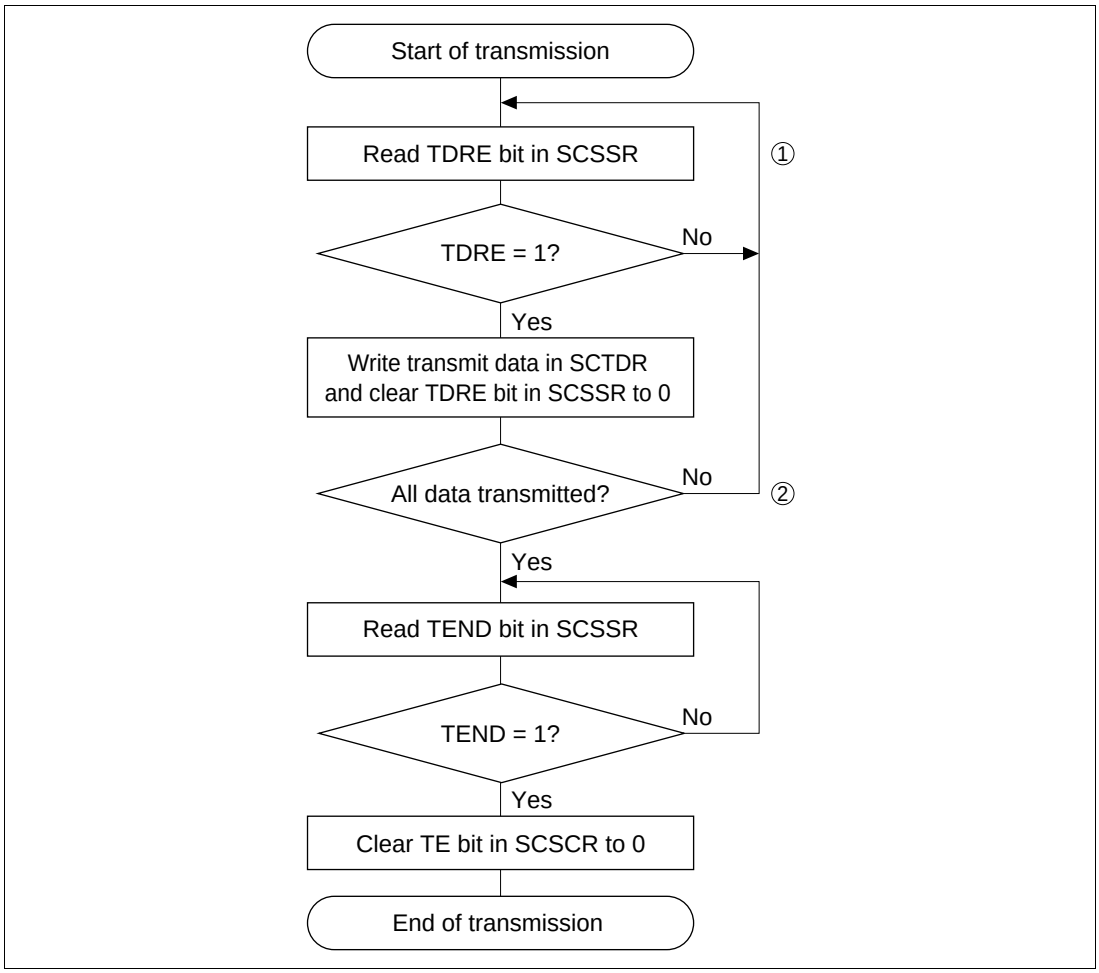


Figure 15.19 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data and loads this data from SCTDR into the transmit shift register (SCTSR).
2. After loading the data from SCTDR into SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in 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 is 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 SCTDR into SCTSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in 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 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 15.20 shows an example of SCI transmit operation.

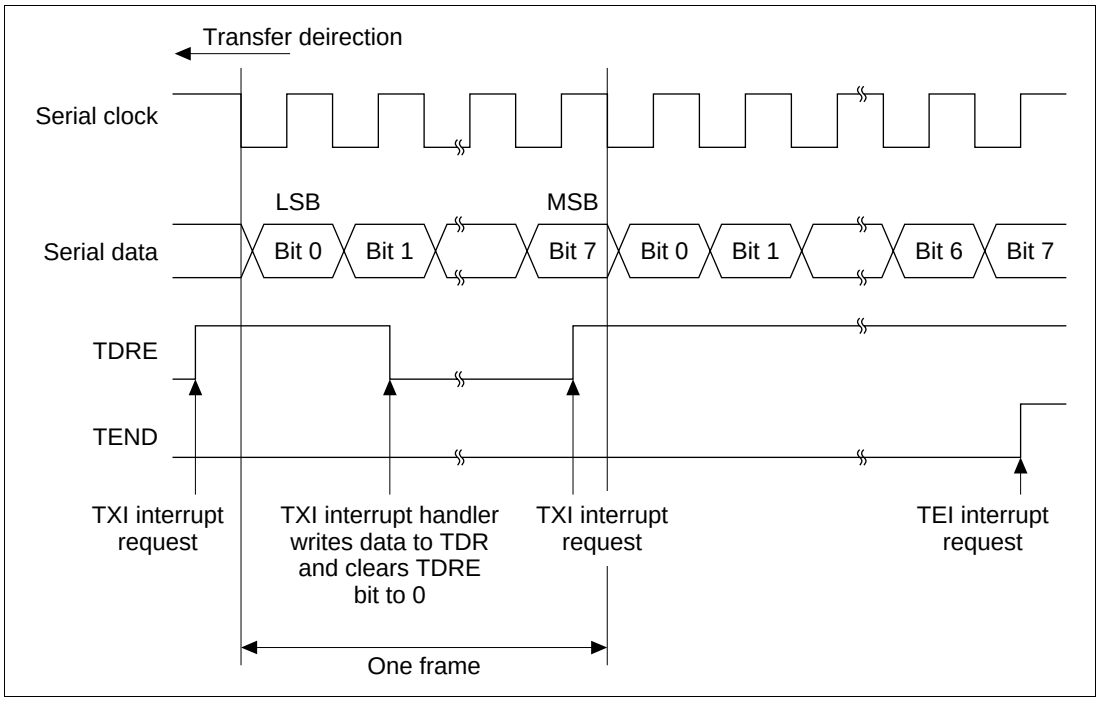


Figure 15.20 Example of SCI Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 15.21 shows a sample flowchart for receiving serial data. When switching from asynchronous mode to 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 MSB (bit 7) of the current frame is received.

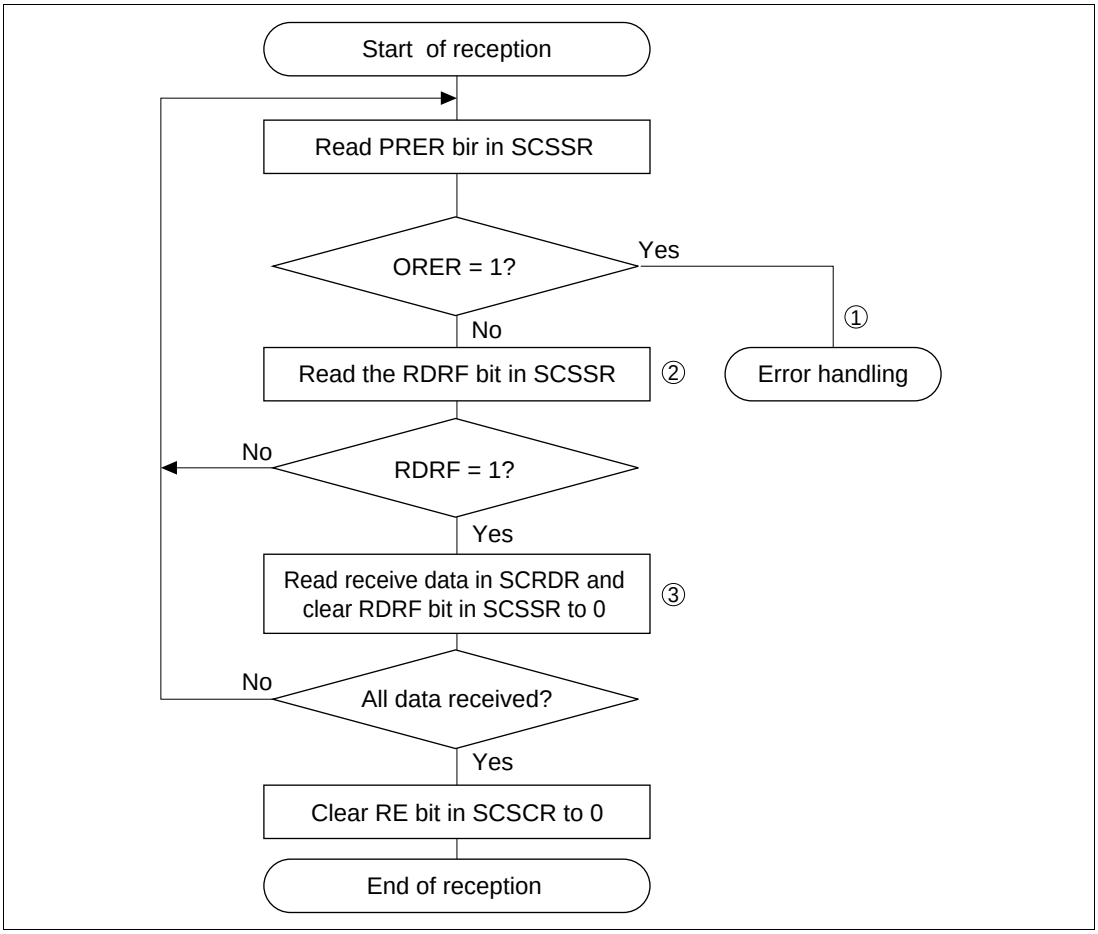


Figure 15.21 Sample Flowchart for Serial Receiving

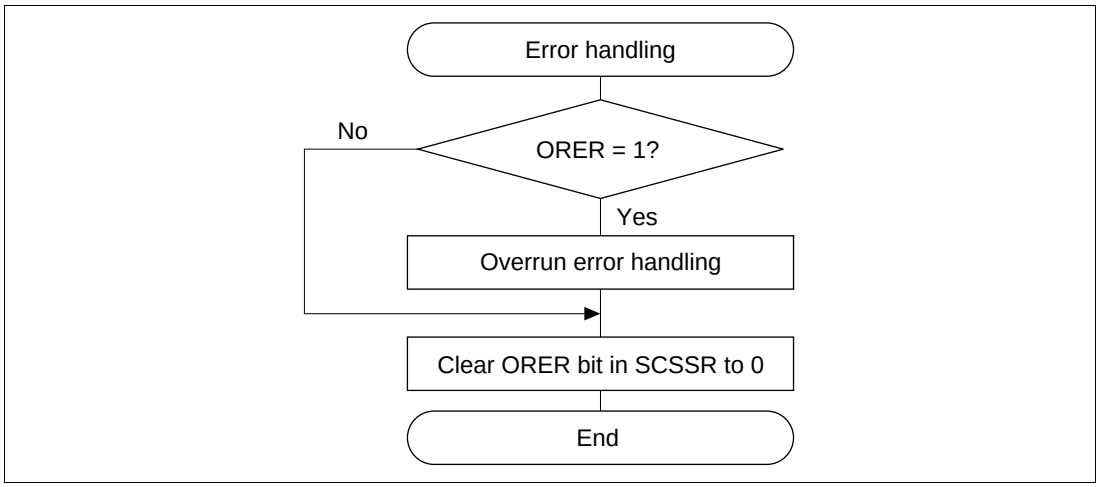


Figure 15.21 Sample Flowchart for Serial Receiving (cont)

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 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 SCRSR into SCRDR. If this check is passed, the SCI sets RDRF to 1 and stores the received data in SCRDR. If the check is not passed (receive error), the SCI operates as indicated in table 15.11. 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 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 SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 15.22 shows an example of SCI receive operation.

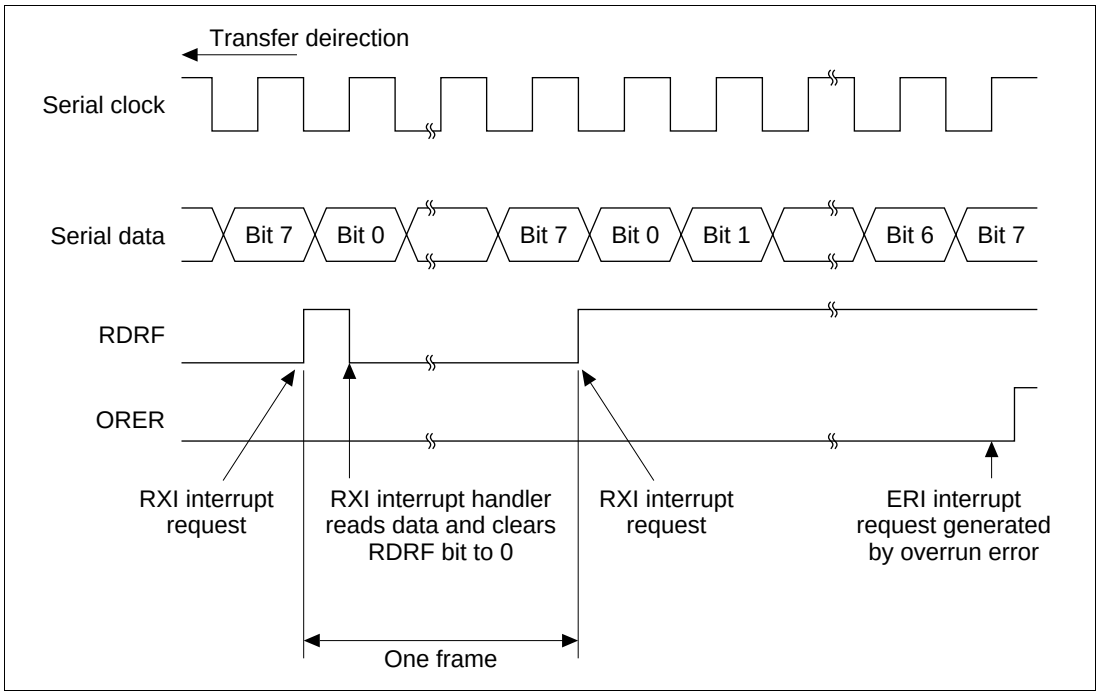
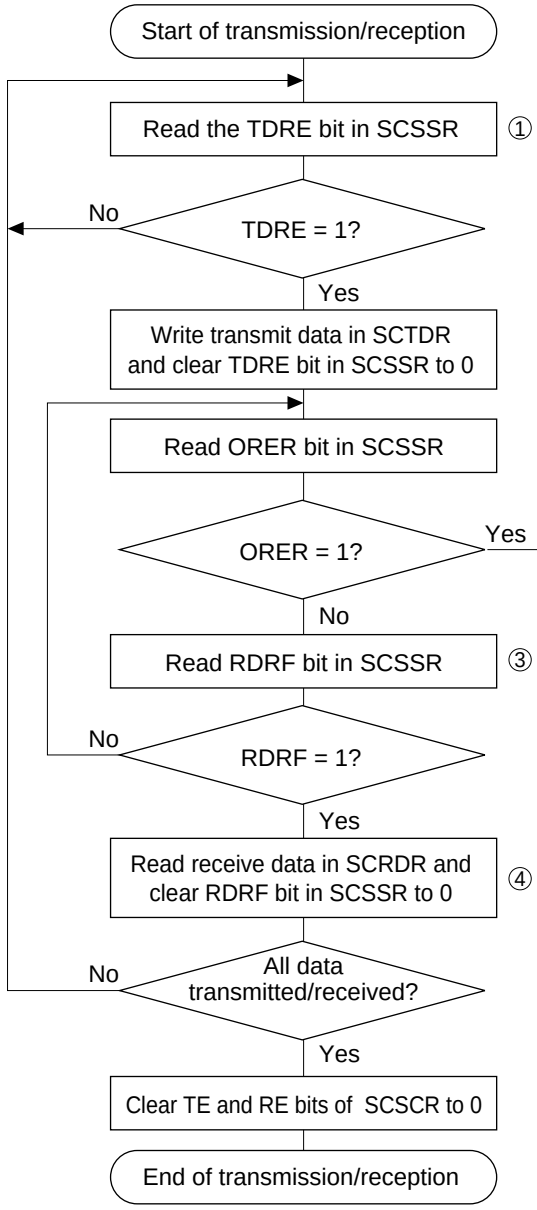


Figure 15.22 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 15.23 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 handling, 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.



Note: When switching from transmitting or receiving or to simulaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1.

Figure 15.23 Sample Flowchart for Serial Transmitting

15.4 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 15.13 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 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 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 SCSSR is set to 1.

TEI is requested when the TEND bit in 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 15.13 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is Cleared
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data register full (RDRF)	↓
TXI	Transmit data register empty (TDRE)	↓
TEI	Transmit end (TEND)	Low

See section 4, Exception Handling, for information on the priority order and relationship to non-SCI interrupts.

15.5 Usage Notes

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 SCTDR into SCTSR. The SCI sets TDRE to 1 when it transfers data from SCTDR to SCTSR. Data can be written to SCTDR regardless of the TDRE bit status. If new data is written in SCTDR when TDRE is 0, however, the old data stored in SCTDR will be lost because the data has not yet been transferred to SCTSR. Before writing transmit data to SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 15.14 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 SCRDR, so receive data is lost.

Table 15.14 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 in the SCP0DT bit in the port SC data register (SCPDR) and bits SCP0MD0 and SCP0MD1 in 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 is confirmed.

Receive Error Flags and Transmitter Operation (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 Asynchronous Mode: In 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 15.24).

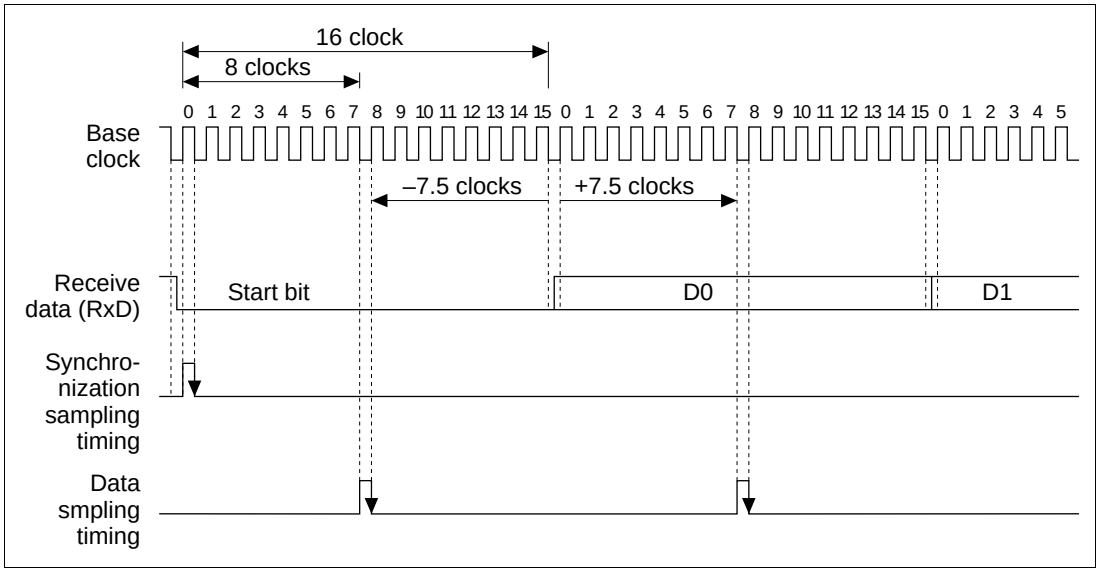


Figure 15.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

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 given by equation 2.

Equation 2:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Cautions on Synchronous External Clock Mode:

- Set TE = RE = 1 only when external clock SCK is 1.
- Do not set TE = RE = 1 until at least four clocks after 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 on 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 16 Smart Card Interface

16.1 Overview

As an added serial communication 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.

16.1.1 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 built-in baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communication-error interrupts are requested independently.

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the smart card interface.

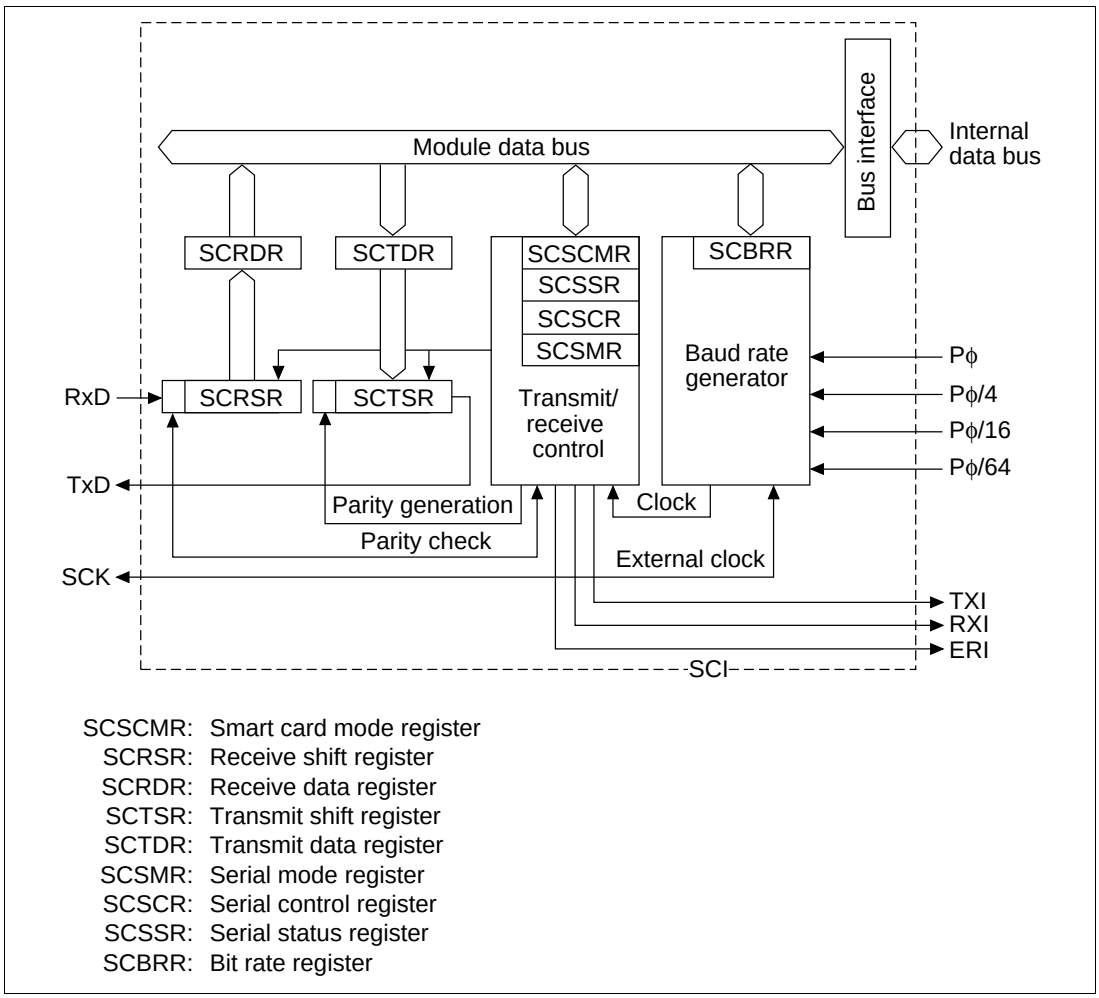


Figure 16.1 Smart Card Interface Block Diagram

16.1.3 Pin Configuration

Table 16.1 summarizes the smart card interface pins.

Table 16.1 Smart Card Interface Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	Output	Clock output
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

16.1.4 Smart Card Interface Register Configuration

Table 16.2 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 13, Serial Communication Interface.

Table 16.2 Registers

Name	Abbreviation	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.

16.2 Register Descriptions

This section describes the registers added for the smart card interface and the bits whose functions are changed.

16.2.1 Smart Card Mode Register (SCSCMR)

The smart card mode register (SCSCMR) is an 8-bit readable/writable 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/W	R/W	—	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

16.2.2 Serial Status Register (SCSSR)

In 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 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.

16.3 Operation

16.3.1 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 be transferred) from the end of the parity bit to the start of the next frame.
2. During reception, the card outputs a low-level error signal 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.

16.3.2 Pin Connections

Figure 16.2 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, 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.

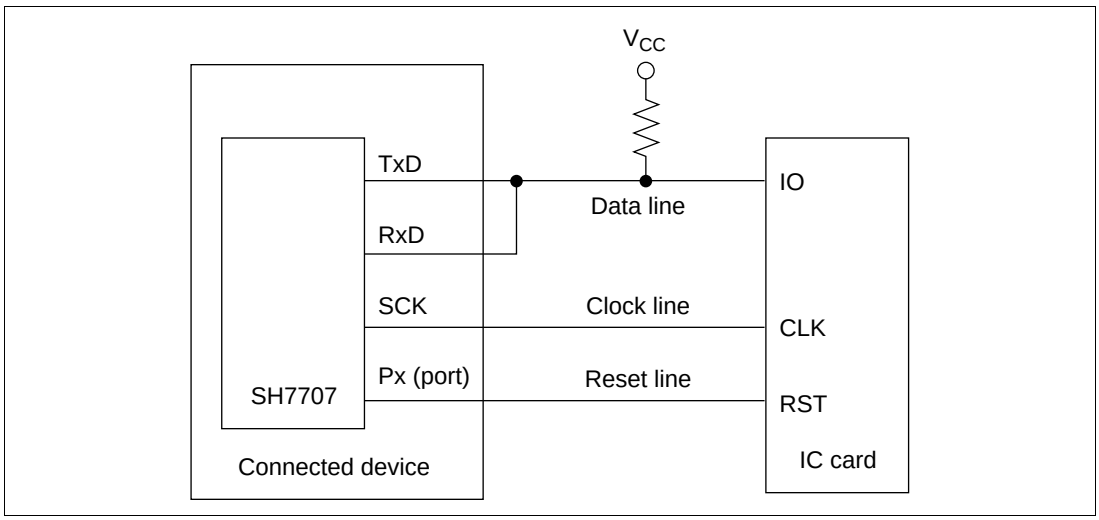


Figure 16.2 Pin Connection Diagram for Smart Card Interface

16.3.3 Data Format

Figure 16.3 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 device 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.

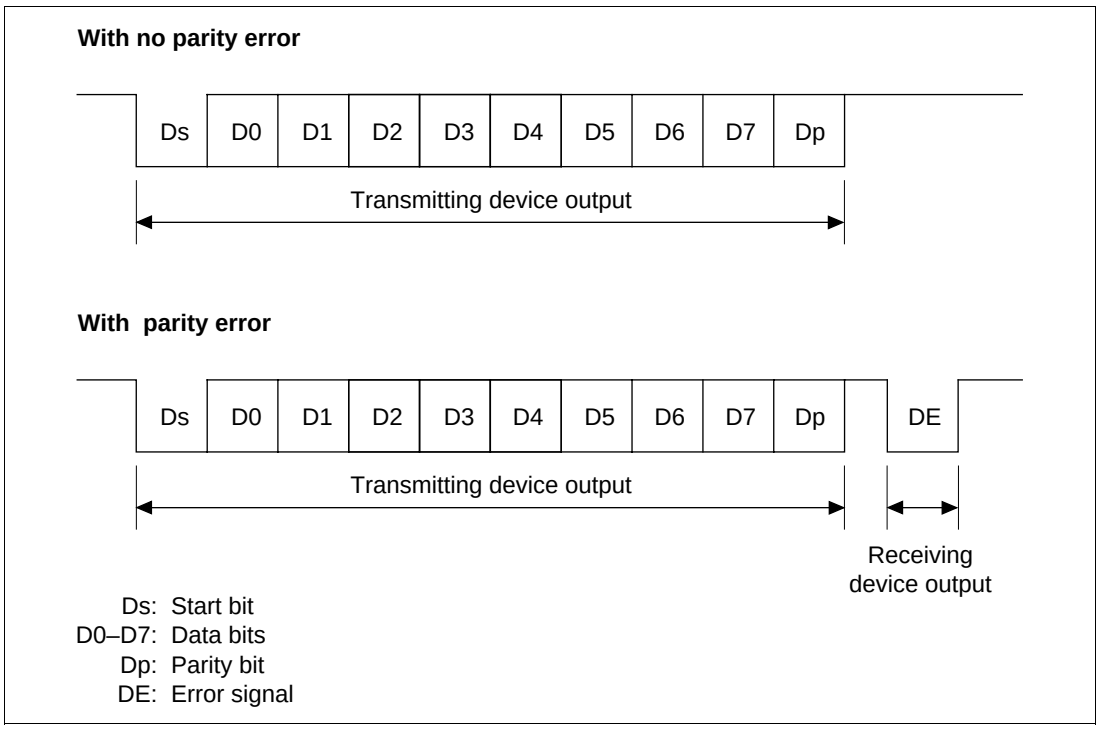


Figure 16.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is in the high-impedance state when not in use and is fixed high with a pull-up resistor.
2. The transmitting device starts transmission of one frame of data. The data frame starts with a start bit (Ds, low level), followed by eight data bits (D0–D7) and a parity bit (Dp).
3. On the smart card interface, the data line returns to the high-impedance state after this. The data line is pulled high with a pull-up resistor.
4. The receiving device checks parity. When the data is received normally with no parity errors, the receiving device then waits to receive the next data. When a parity error occurs, the receiving device outputs an error signal (DE, low level) and requests re-transfer of data. The receiving device returns the signal line to the high-impedance state after outputting the error signal for a specified period. The signal line is pulled high with a pull-up resistor.
5. The transmitting device 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.

16.3.4 Register Settings

Table 16.3 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 16.3 Register Settings for 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): Clear the O/$\overline{E}$ bit to 0 when the IC card uses the direct convention, or clear it to to 1 when using the inverse convention. Select the built-in baud rate generator clock source with the CKS1 and CKS0 bits (see section 16.3.5, Clock).
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 16.3.5, Clock, for the method of calculating 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 13, 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 cleared 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 16.4 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.

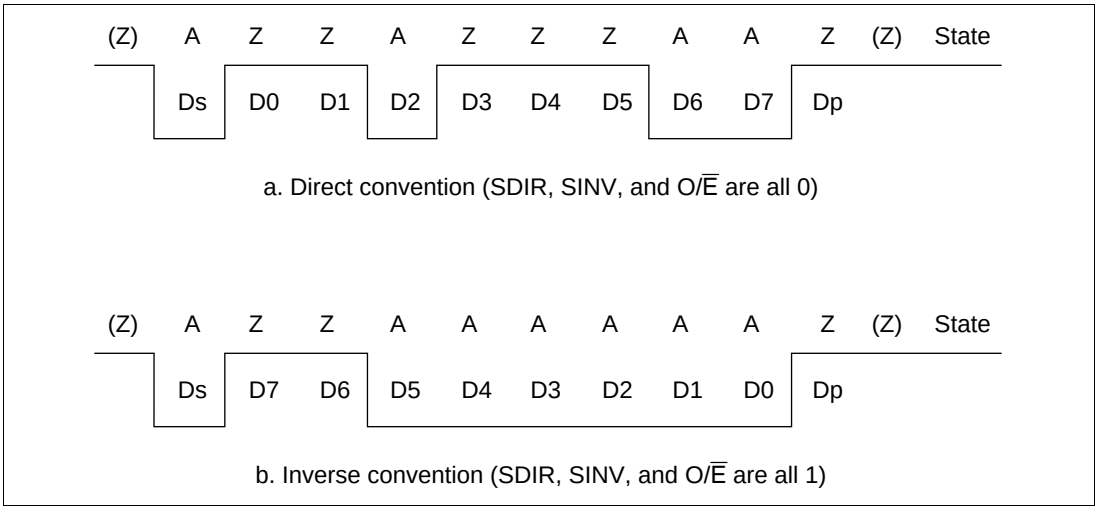


Figure 16.4 Waveform of Start Character

16.3.5 Clock

Only the internal clock generated by the built-in 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 16.5 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 N 255)
B = Bit rate (bit/s)
Pϕ = Supporting module operating frequency (MHz)*
n = 0–3 (table 16.4)

Table 16.4 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 16.5 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: Bit rates are rounded to one decimal place.

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 16.6 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 16.7 Maximum Bit Rates for Various 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 given by the following equation:

$$\text{Error}(\%) = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

Table 16.8 shows the relationship between transmit/receive clock register set values and output states on the smart card interface.

Table 16.8 Register Set Values and SCK Pin

	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{A}$**	**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.

16.3.6 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 16.5 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/$\overline{A}$ bit, parity bit (O/$\overline{E}$ 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.

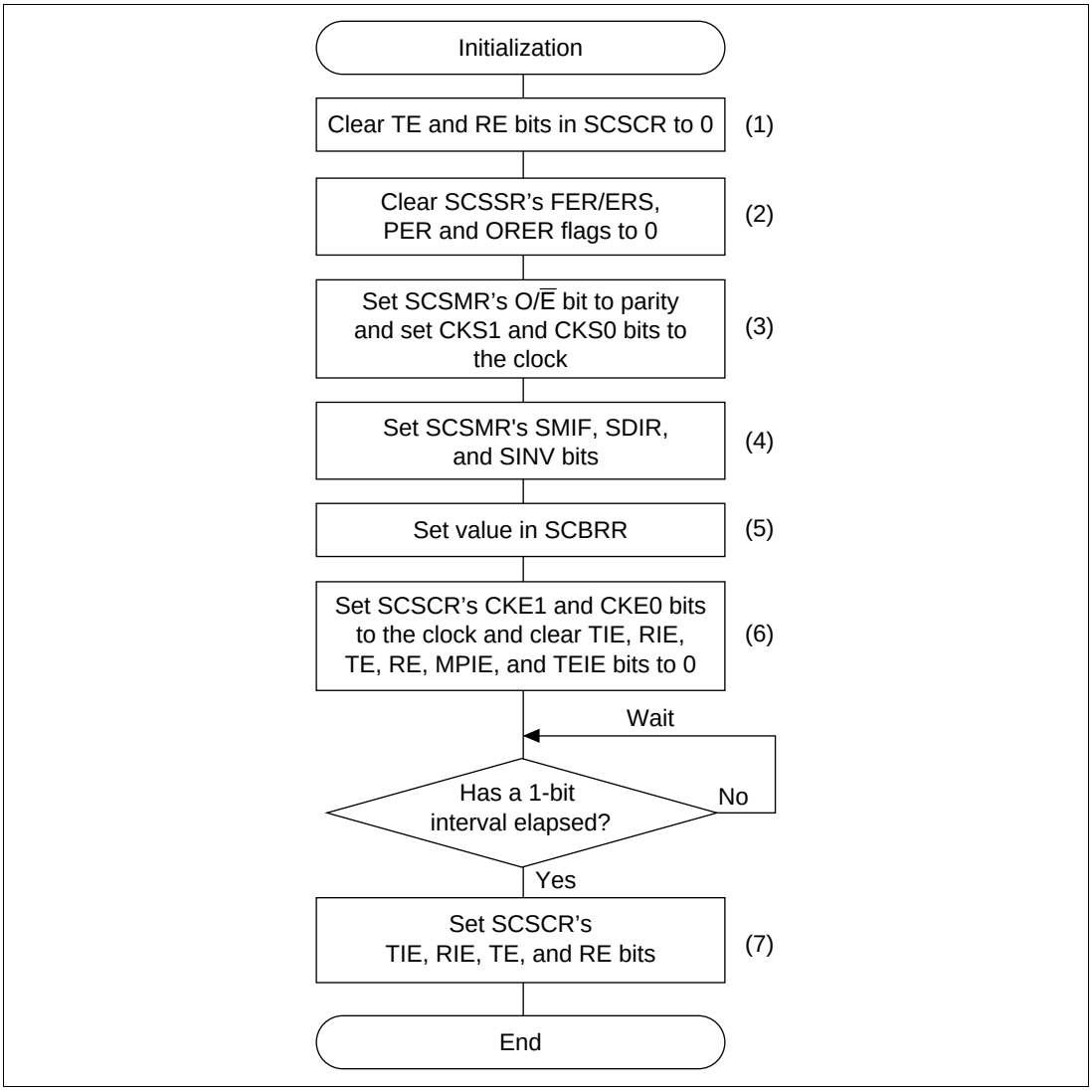


Figure 16.5 Initialization Flowchart (Example)

Serial Data Transmission: The handling procedures in smart card mode differ from ordinary SCI processing because data is retransmitted when an error signal is sampled during data transmission. This results in the transmission processing flowchart shown in figure 16.6.

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 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.

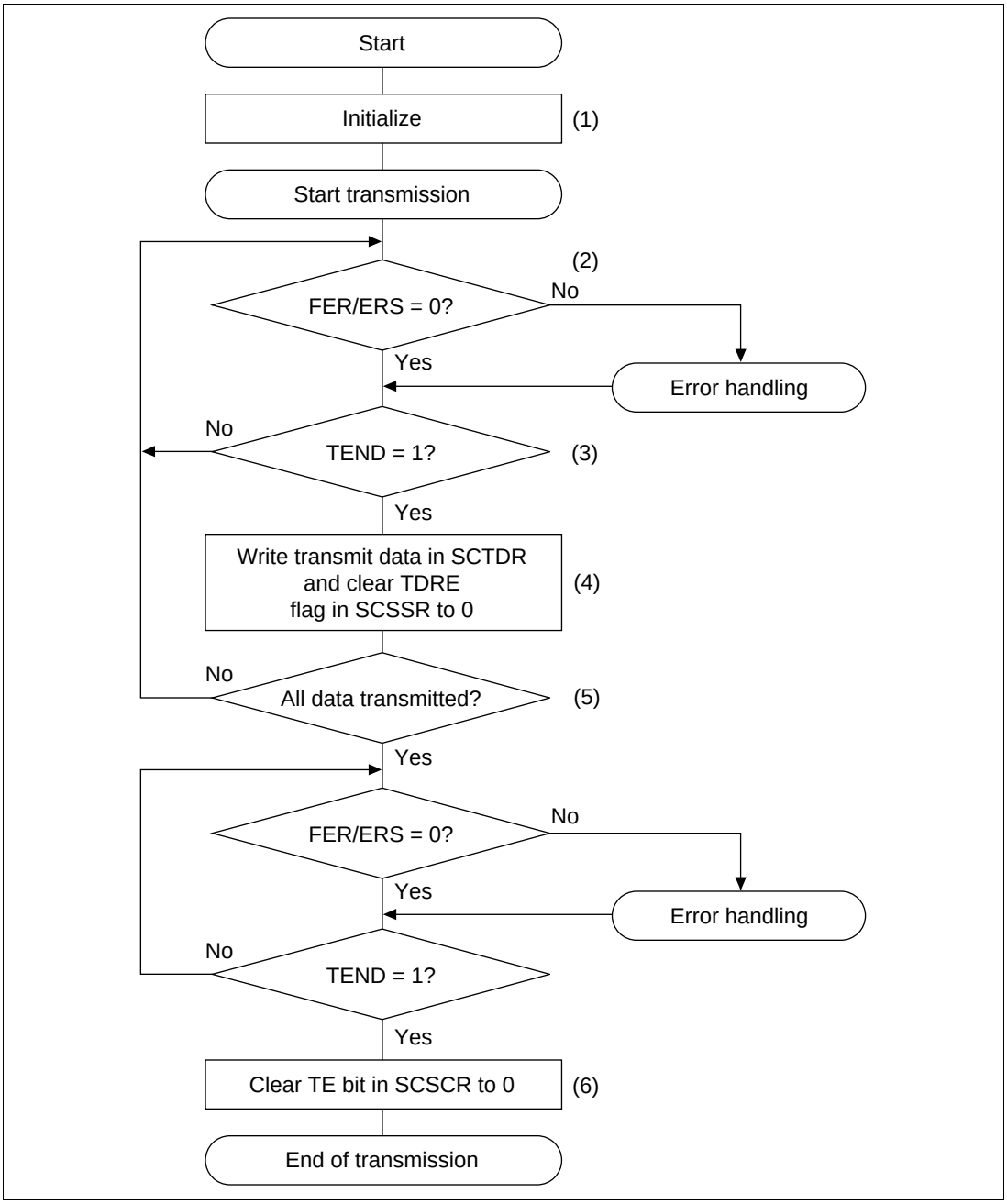


Figure 16.6 Transmission Flowchart

Serial Data Reception: The handling procedures in smart card mode are the same as in ordinary SCI processing. The reception processing flowchart is shown in figure 16.7.

1. Initialize the smart card interface mode as described above in Initialization and in figure 16.5.
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 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.

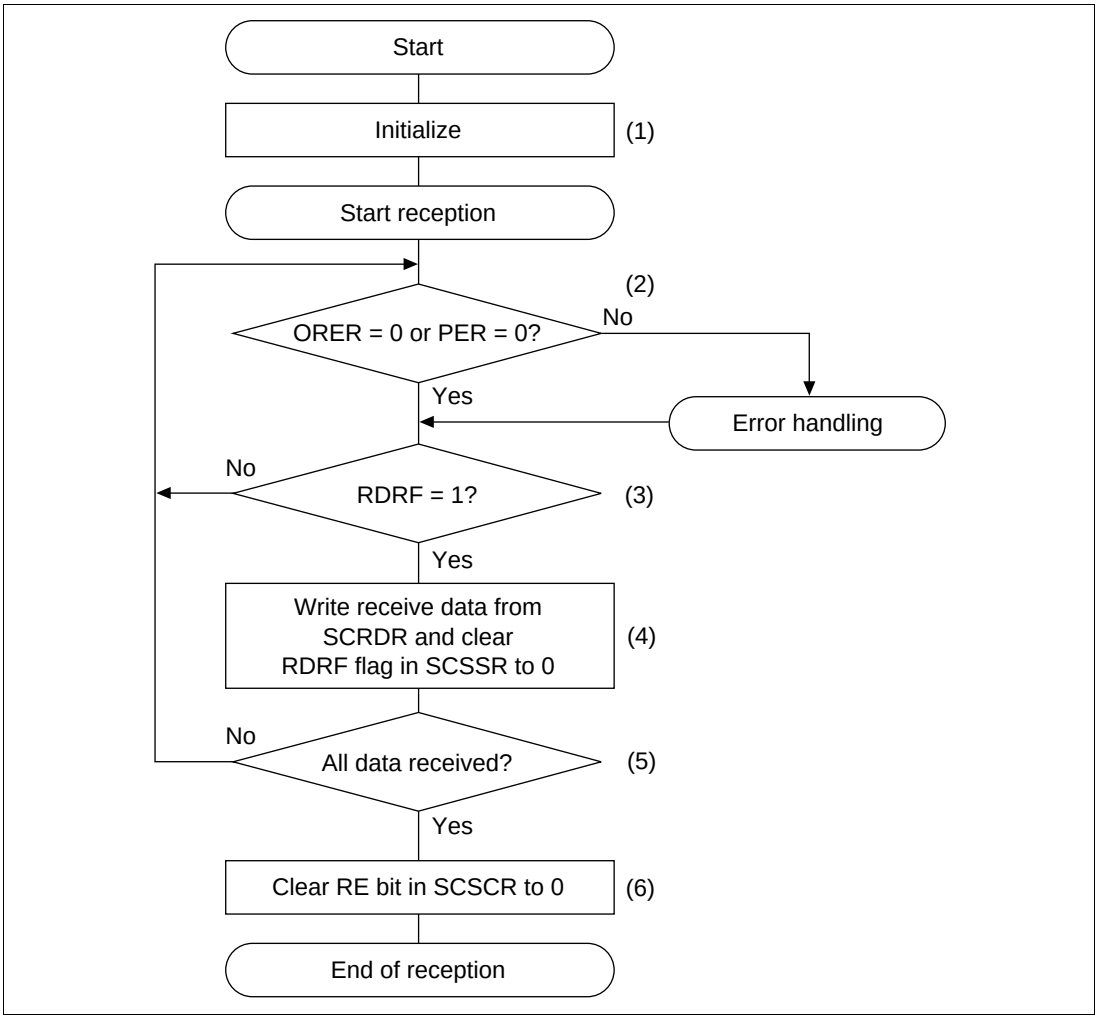


Figure 16.7 Reception Flowchart (Example)

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 16.9).

Table 16.9 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

16.4 Usage Notes

When the SCI is used as a smart card interface, be sure that all criteria in sections 16.4.1 and 16.4.2 are applied.

16.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI runs on a base 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 base clock cycle (figure 16.8).

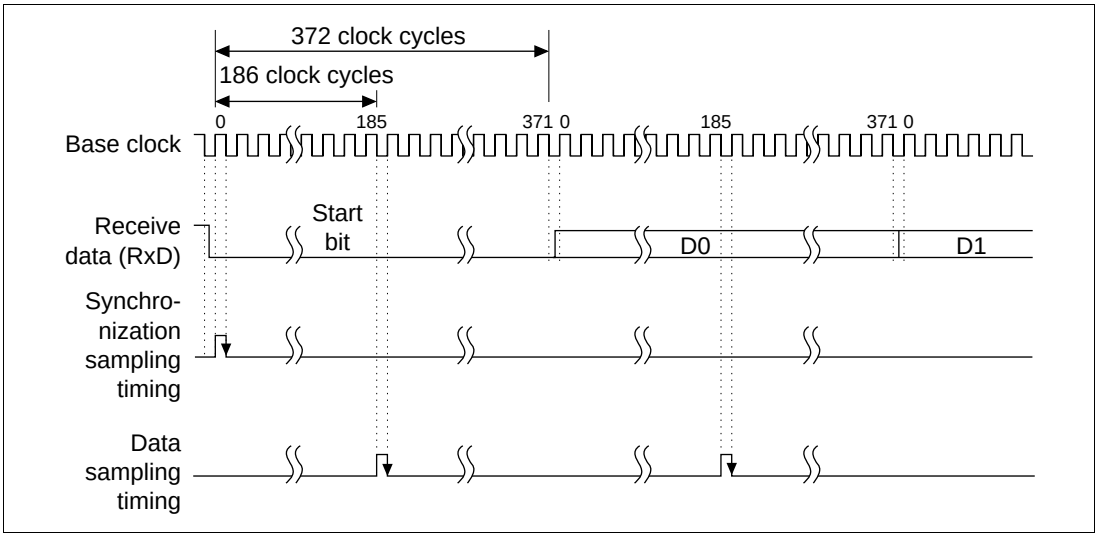


Figure 16.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is given by 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:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

16.4.2 Retransmission (Receive and Transmit Modes)

Retransmission by SCI in Receive Mode: Figure 16.9 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 for 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.

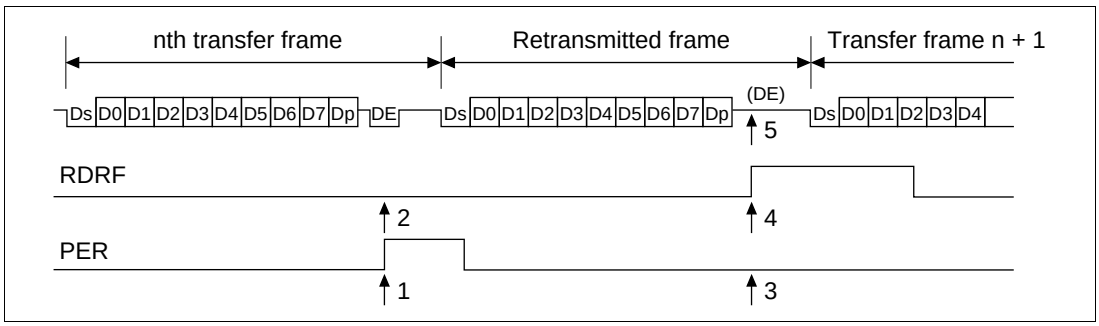


Figure 16.9 Retransmission in SCI Receive Mode

Retransmission by SCI in Transmit Mode: Figure 16.10 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 device. 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 for 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 device.
4. When no error signal is returned from the receiving device, 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.

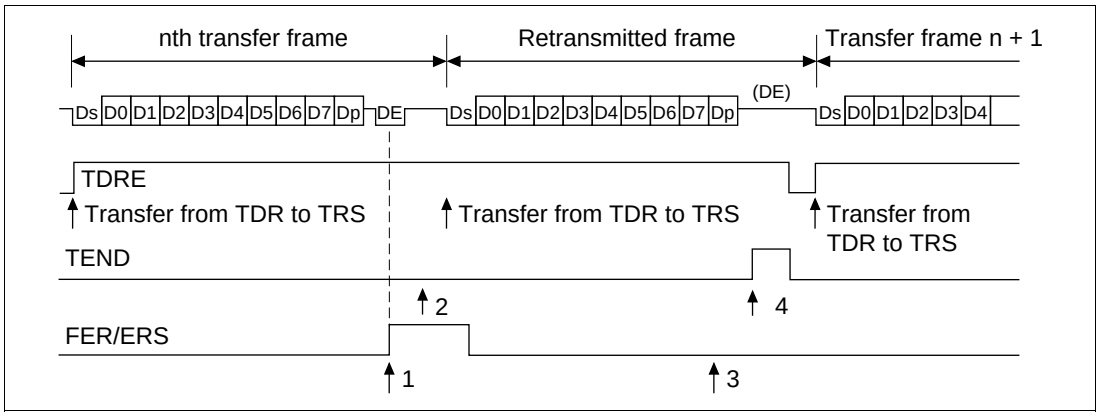


Figure 16.10 Retransmission in SCI Transmit Mode

Section 17 Serial Communication Interface with FIFO (SCIF)

17.1 Overview

The SH7707 has an on-chip 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 efficient, high-speed, continuous communication.

Note: IRDA module can also be used as another SCIF. For details, see section 18, IRDA.

17.1.1 Features

- Asynchronous serial communication:
  - — Serial data communication is performed by the start-stop method in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - — Data length: 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 following the generated framing error is the space 0 level, indicating a framing error. It is also detected by reading the RxD2 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 SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- Built-in baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK2 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 (RTS2 and CTS2)

- 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 ascertained.
- A time-out error (DR) can be detected in receiving.

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the SCIF.

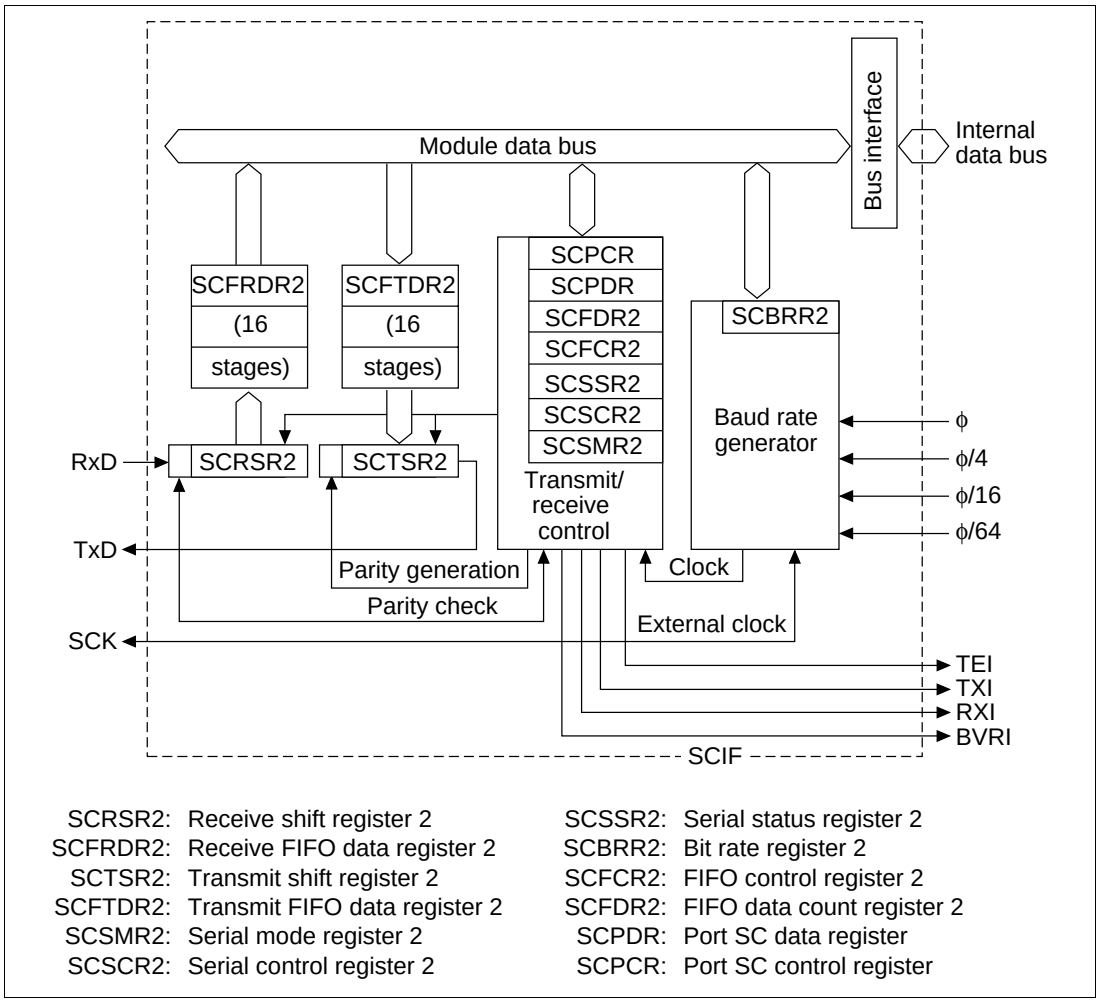


Figure 17.1 SCIF Block Diagram

Figures 17.2 to 17.4 show the SCIF I/O port pins.

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 19.3.12, Port SC Control Registers (SCPCR) and 20.13.2 Port SC Data Register (SCPDR).

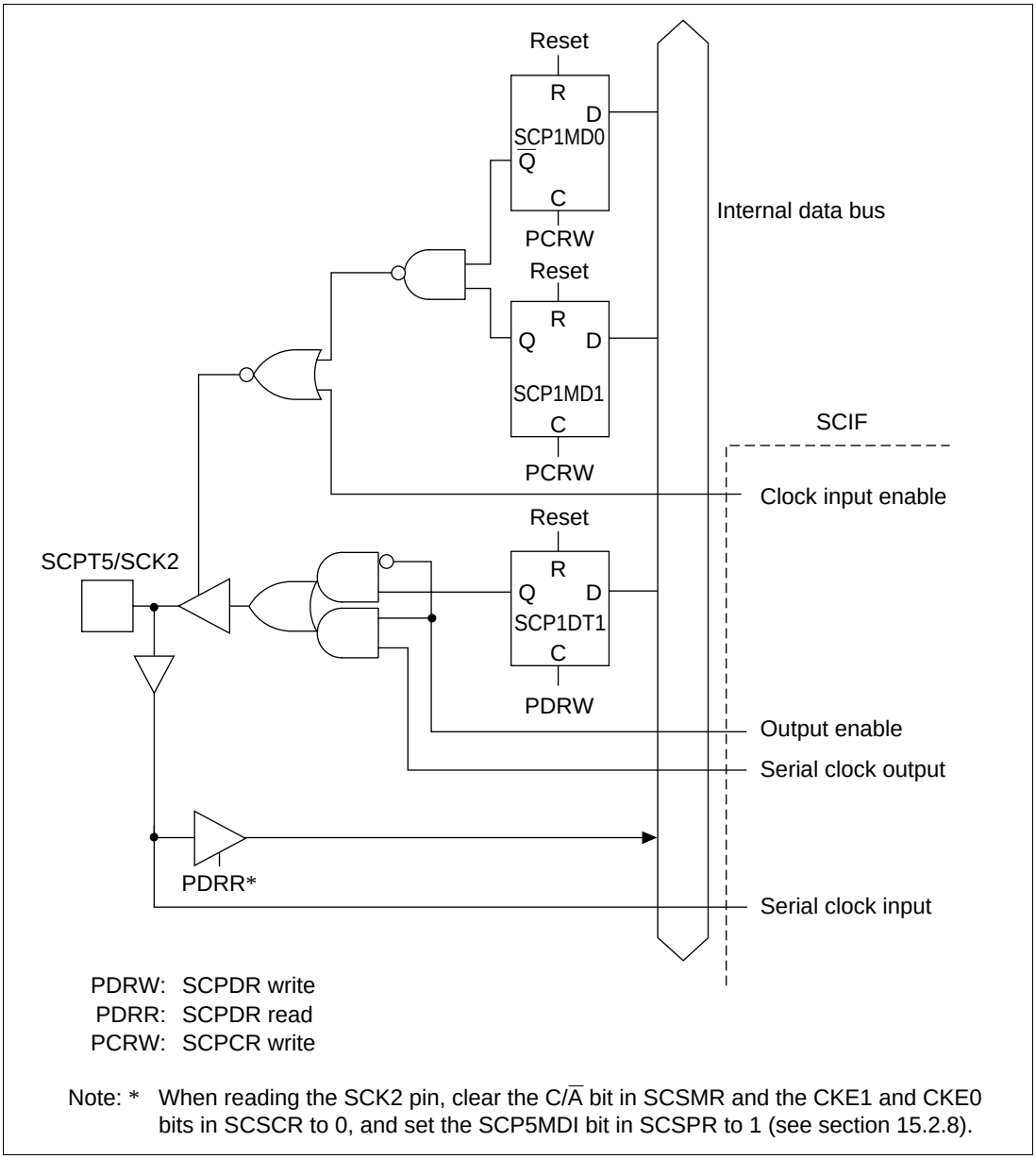


Figure 17.2 SCPT[5]/SCK2 Pin

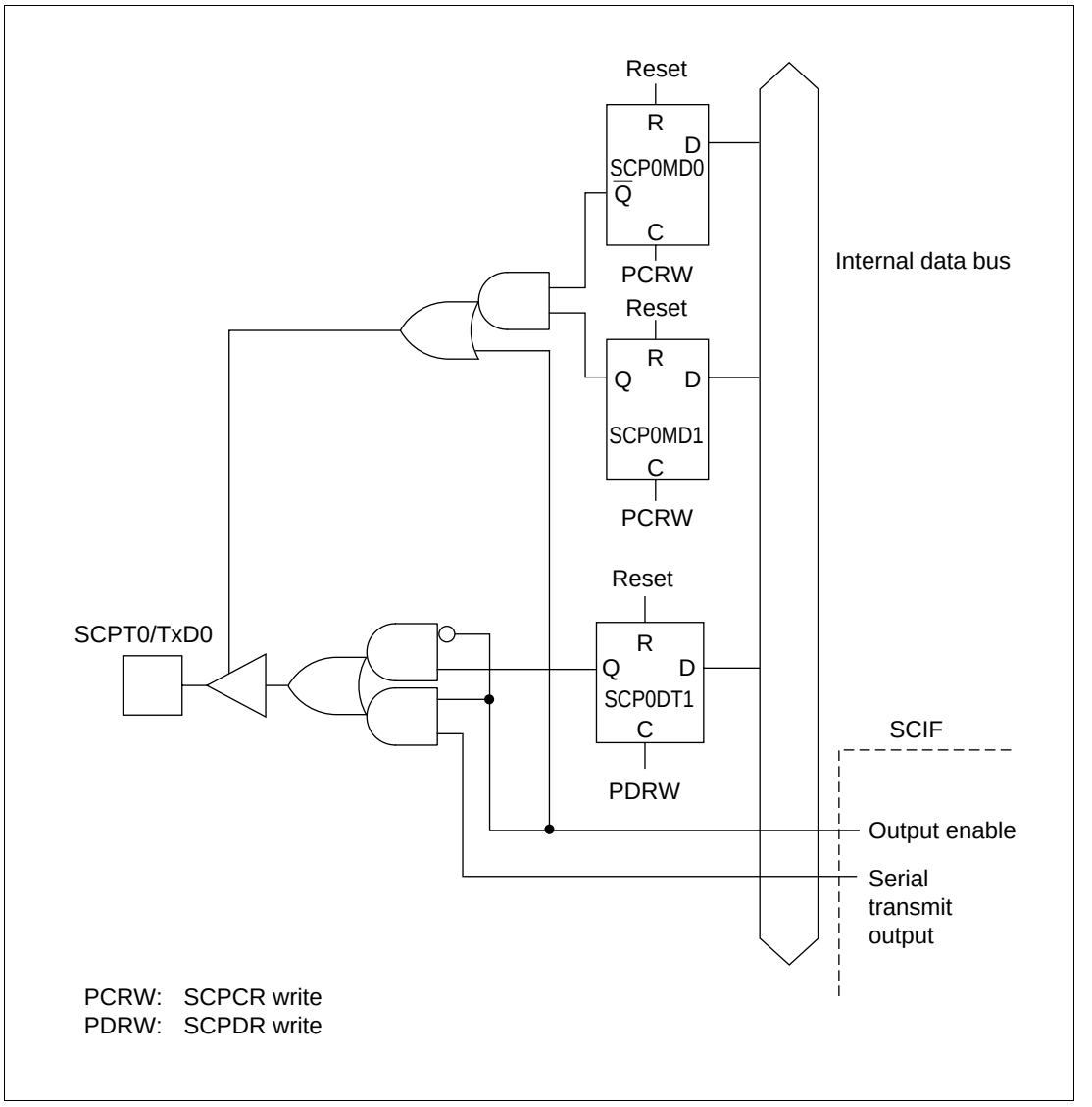


Figure 17.3 SCPT[4]/TxD2 Pin

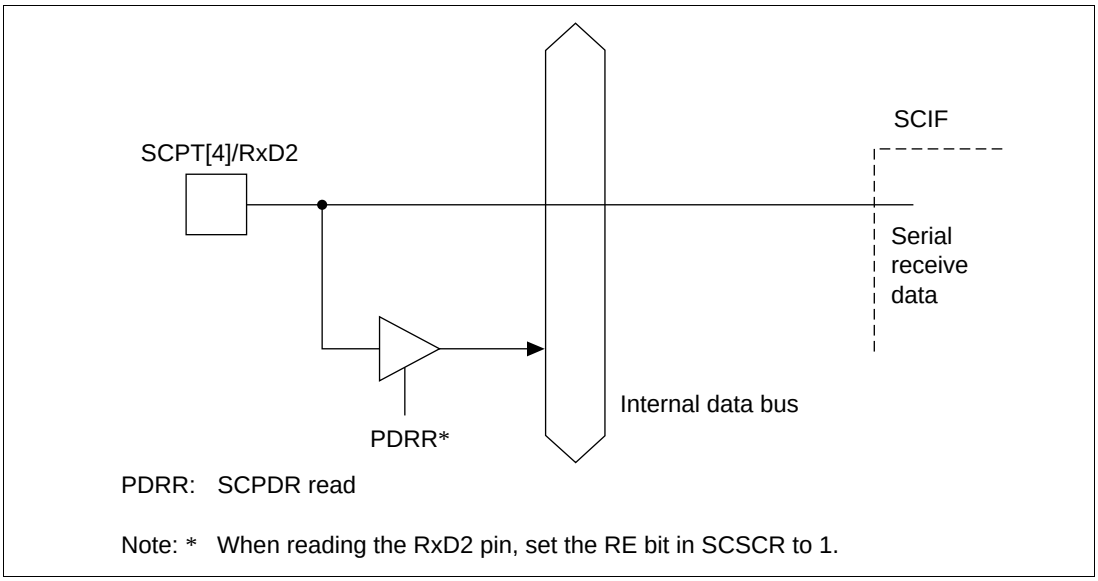


Figure 17.4 SCPT[4]/RxD2 Pin

17.1.3 Pin Configuration

The SCIF has the serial pins summarized in table 17.1.

Table 17.1 SCIF Pins

Pin Name	Abbreviation	I/O	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	RTS2	Output	Request to send
Clear to send pin	CTS2	Input	Clear to send

17.1.4 Register Configuration

Table 17.2 summarizes the SCIF internal registers. These registers select the communication mode (asynchronous or synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 17.2 SCIF Registers

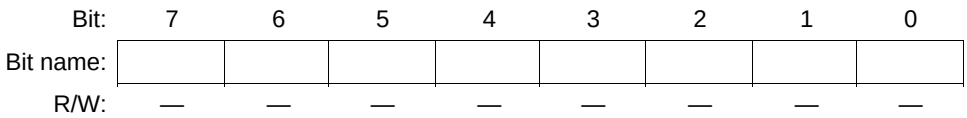
Register Name	Abbreviation	R/W	Initial Value	Address	Access size
Serial mode register 2	SCSMR2	R/W	H'00	H'4000150	8 bits
Bit rate register 2	SCBRR2	R/W	H'FF	H'4000152	8 bits
Serial control register 2	SCSCR2	R/W	H'00	H'4000154	8 bits
Transmit FIFO data register 2	SCSFDR2	W	—	H'4000156	8 bits
Serial status register 2	SCSSR2	R/(W)*	H'0060	H'4000158	16 bits
Receive data FIFO register 2	SCFRDR2	R	Undefined	H'400015A	8 bits
FIFO control register 2	SCFCR2	R/W	H'00	H'400015C	8 bits
FIFO data count register 2	SCFDR2	R	H'0000	H'400015E	16 bits

Note: Only 0 can be written, to clear the flags.

17.2 Register Descriptions

17.2.1 Receive Shift Register 2 (SCRSR2)

The receive shift register 2 (SCRSR2) receives serial data. Data input at the RxD2 pin is loaded into SCRSR2 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 SCFRDR2, which is a receive FIFO register. The CPU cannot read or write to SCRSR2 directly.



17.2.2 Receive FIFO Data Register 2 (SCFRDR2)

The 16-byte receive FIFO data register 2 (SCFRDR2) 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 2 (SCRSR2) into SCFRDR2 for storage. Continuous receiving is enabled until 16 bytes are stored.

The CPU can read but not write to SCFRDR2. When data is read without receive data in the receive FIFO data register, the value is undefined. When the received data fills this register, 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

17.2.3 Transmit Shift Register 2 (SCTSR2)

The transmit shift register 2 (SCTSR2) transmits serial data. The SCIF loads transmit data from the transmit FIFO data register 2 (SCFTDR2) into SCTSR2, then transmits the data serially from the TxD2 pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR2 into SCTSR2 and starts transmitting again. The CPU cannot read or write to SCTSR2 directly.

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

17.2.4 Transmit FIFO Data Register 2 (SCFTDR2)

The transmit FIFO data register 2 (SCFTDR2) is a 16-byte, 8-bit-length FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register 2 (SCTSR2) is empty, it moves transmit data written in SCFTDR2 into SCTSR2 and starts serial transmission. Continuous serial transmission is performed until the SCFTDR2 is empty. The CPU can always write to SCFTDR2.

When SCFTDR2 is full of transmit data (16 bytes), more data cannot be written. If writing is attempted, 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

17.2.5 Serial Mode Register 2 (SCSMR2)

The serial mode register 2 (SCSMR2) 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 to SCSMR2. SCSMR2 is initialized to H'00 by a reset and 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 7-bit or 8-bit data in asynchronous mode.

Bit 6: CHR	Description
0	8-bit data (Initial value)
1	7-bit data. (When seven-bit data is selected, the MSB (bit 7) of the transmit FIFO data register 2 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 1-bit is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two 1-bits 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 built-in 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

17.2.6 Serial Control Register 2 (SCSCR2)

The serial control register 2 (SCSCR2) operates the SCIF transmitter/receiver, selects the serial clock output in asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR2. SCSCR2 is initialized to H'00 by a reset and 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 2 (SCFTDR2) to the transmit shift register 2 (SCTSR2), when the quantity of data in the

transmit FIFO register 2 becomes less than the specified transmission trigger number, and when the TDFE flag in the serial FIFO status register 2 (SCFSR2) 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 a quantity of transmit data greater than the specified transmission trigger number to SCFTDR2 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 serial receive data is transferred from the receive shift register 2 (SCRSR2) to the receive FIFO data register 2 (SCFRDR2), when the quantity of data in the receive FIFO register 2 exceeds the specified receive trigger number, and when the RDRF flag in SCSSR2 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. When RDF is set, read 1 from the RDF flag and clear it to 0, after reading the received data from SCRDR2 until the quantity of received data is less than the specified receive trigger number
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 transmit data into SCFTDR2. Select the transmit format in SCSMR2 and SCFCR2 and reset 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 RDF). These flags retain their previous values
1	Receiver enabled. Serial reception starts when a start bit is detected. Select the receive format in SCSMR before setting RE to 1

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCIF clock source and enable or disable clock output from the SCK2 pin. Depending on the combination of CKE1 and CKE0, the SCK2 pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only when the SCIF is operating with the internal clock (CKE1 = 0). The CKE0 setting is ignored when an external clock source is selected (CKE1 = 1). Set CKE1 and CKE0 before selecting the SCIF operating mode in the serial mode register 2 (SCSMR2). For further details on selection of the SCIF clock source, see table 17.8 in section 17.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description
0	0	Internal clock, SCK2 pin used for input pin (input signal is ignored)(Initial value)
	1	Internal clock, SCK2 pin used for clock output*1
1	0	External clock, SCK2 pin used for clock input*2
	1	External clock, SCK2 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.

17.2.7 Serial Status Register 2 (SCSSR2)

The serial status register 2 (SCSSR2) is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO register data, and the lower 8 bits indicate the SCIF operating status.

The CPU can always read and write to SCSSR2, but cannot write 1 in the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits. SCSSR2 is initialized to H'0060 by a reset and 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: Only 0 can be written, to clear the flag.

Bit 7—Receive Error (ER): Indicates that a framing error or a parity error has occurred. *1

Bit 7: ER	Description
0	Receiving is in progress or has ended normally 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 at the end of reception of one data byte*2, or when the total number of 1s in the receive data and in the parity bit does not match the even/odd parity specification by the O/E bit in SCSMR2

Notes: 1. Clearing the RE bit to 0 in SCSCR2 does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR2 and the receive operation is continued. Whether or not the data read from SCRDR2 includes a receive error can be detected by the FER and PER bits in SCSSR2.

2. In 2-stop-bit mode, only the first stop bit is checked.

Bit 6—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the SCFTDR2 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 SCTDR2
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 2 (SCSCR2), or SCFTDR2 does not contain receive data when the last bit of a one-byte serial character is transmitted

Bit 5—Transmit FIFO Data Register Empty (TDFE): Indicates that data has been transferred from the transmit FIFO data register 2 (SCFTDR2) to the transmit shift register 2 (SCTSR2), the quantity of data in SCFTDR2 is equal to or less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO control register 2 (SCFCR2), and writing of transmit data to SCFTDR2 is enabled.

Bit 5: TDFE	Description
0	The quantity of transmit data written to SCFTDR2 is greater than the specified transmission triggers number. TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR2, software reads TDFE after it has been set to 1, then writes 0 in TDFE
1	The quantity of transmit data in SCFTDR2 is equal to or less than the specified transmission trigger number. TDFE is set to 1 by a reset or in standby mode, or when the quantity of transmit data in SCFTDR2 is equal to or less than the specified transmission trigger number as a result of transmission*

Note: Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If writing of additional data is attempted, the data is ignored. The quantity of data in SCFTDR2 is indicated by the upper 8 bits of SCFDR2.

Bit 4—Break Detection (BRK): Indicates that a break signal has been detected in the receive 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 when software reads BRK after it has been set to 1, then writes 0 in BRK
1	The break signal is received* BRK is set to 1 when data including a framing error is received and a framing error occurs with space (0) in the subsequent receive data

Note: When a break is detected, transfer of the received data (H'00) to SCFRDR2 stops after detection. When the break ends and the receive signal goes to 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 SCFRDR2. After this, however, no received data is transferred until a break ends with the received signal at 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 2 (SCFRDR2).

Bit 3: FER	Description
0	No receive framing error occurred in the data read from SCFRDR2 (Initial value) FER is cleared to 0 when the chip is reset or enters standby mode, or when no framing error is present in the data read from SCFRDR2
1	A receive framing error occurred in the data read from SCFRDR2 FER is set to 1 when a framing error is present in the data read from SCFRDR2

Bit 2—Parity Error (PER): Indicates a parity error in the data read from the receive FIFO data register 2 (SCFRDR2).

Bit 2: PER	Description
0	No receive parity error occurred in the data read from SCFRDR2 (Initial value) PER is cleared to 0 when the chip is reset or enters standby mode, or when no parity error is present in the data read from SCFRDR2
1	A receive framing error occurred in the data read from SCFRDR2 PER is set to 1 when a parity error is present in the data read from SCFRDR2

Bit 1—Receive FIFO Data Register Full (RDF): Indicates that received data has been transferred to the receive FIFO data register 2 (SCFRDR2), and the quantity of data in SCFRDR2 reaches or exceeds the receive trigger number specified by the RTRG1 and RTRG0 bits in FIFO control register 2 (SCFCR2).

Bit 1: RDF	Description
0	The quantity of transmit data written to SCFRDR2 is less than the specified number of receive trigger number (Initial value) RDF is cleared to 0 by a reset or in standby mode, or cleared when SCFRDR is read until the quantity of receive data in SCFRDR2 is less than the specified receive trigger number, or when 1 is read from RDF, and 0 is then written
1	The quantity of receive data in SCFRDR2 is equal to or greater than the specified receive trigger number RDF is set to 1 when a quantity of receive data which is equal to or greater than the specified receive trigger number is stored in SCFRDR2*

Note: Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be read when RDF is 1 is the specified receive trigger number. If reading is attempted after all data in SCFRDR2 has been read, the data will be undefined. The quantity of receive data in SCFRDR2 is indicated by the lower 8 bits of SCFDR2.

Bit 0—Receive Data Ready (DR): Indicates that the receive FIFO data register 2 (SCFRDR2) holds a quantity of data less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 etu since the last stop bit.

Bit 0: DR	Description
0	Receiving is in progress, or no received data remains in SCFRDR2 after reception has ended normally (Initial value) DR is cleared to 0 when the chip is reset or enters standby mode, or when software reads DR after it has been set to 1, then writes 0 in DR
1	Next receive data has not been received DR is set to 1 when SCFRDR2 holds a quantity of data less than the specified receive trigger number and the next data has not yet been received after the elapse of 15 etu since the last stop bit*

Note: This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (etu: elemental time at 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 to 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 2 (SCFRDR2). The value indicated by bits 15 to 12 represents the number of parity errors in SCFRDR2.

Bits 11 to 8—Number of Framing Errors (FER): Indicates the quantity of data including a framing error in the received data stored in SCFRDR2. The value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR2.

17.2.8 Bit Rate Register 2 (SCBRR2)

The bit rate register 2 (SCBRR2) is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register 2 (SCSMR2), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR2. SCBRR2 is initialized to H'FF by a reset and in module standby and 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

The SCBRR2 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: SCBRR2 setting for baud rate generator ($0 \leq N \leq 255$)

$P\phi$: Operating frequency for supporting modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 17.3.)

Table 17.3 SCSMR2 Settings

		SCSMR2 Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	$P\phi$	0	0
1	$P\phi/4$	0	1
2	$P\phi/16$	1	0
3	$P\phi/64$	1	1

Note: The bit rate error is given by the following equation:

$$\text{Error } (\%) = \{P(\phi \times 10^6)/[(N + 1) \times B \times 64 \times 2^{2n-1}] - 1\} \times 100$$

Table 17.4 lists examples of SCBRR2 settings.

Table 17.4 Bit Rates and SCBRR2 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

	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

Table 17.4 Bit Rates and SCBRR2 Settings (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 17.4 Bit Rates and SCBRR2 Settings (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

	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 17.4 Bit Rates and SCBRR2 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 17.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is being used. Table 17.6 list the maximum rates for external clock input.

Table 17.5 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 17.6 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

17.2.9 FIFO Control Register 2 (SCFCR2)

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 2 (SCFCR2) resets the quantity of data in the transmit and receive FIFO registers, sets the trigger data, number and contains the enable bit for loop-back testing. SCFCR2 can read and written by the CPU at all times. It is initialized to H'00 by a reset, by the module standby function, and in standby mode.

Bits 7 and 6—Receive FIFO Data Trigger (RTRG1, RTRG0): These bits set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register 2 (SCSSR2). The RDF flag is set when the quantity of receive data stored in the receive FIFO register 2 (SCFRDR2) exceeds the trigger number as shown below.

Bit 7: RTRG1	Bit 6: RTRG0	Receive Trigger Number	
0	0	1	(Initial value)
0	1	4	
1	0	8	
1	1	14	

Bits 5 and 4—Transmit FIFO Data Trigger (TTRG1, TTRG0): These bits set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register 2 (SCSSR2). The TDFE flag is set when the quantity of transmit data in the transmit FIFO data register 2 (SCFTDR2) falls below the trigger number specified by these bits, as shown below.

Bit 5: TTRG1	Bit 4: TTRG0	Transmit Trigger Number	
0	0	8 (8)	(Initial value)
0	1	4 (12)	
1	0	2 (14)	
1	1	1 (15)	

Note: Values in brackets indicate the number of empty SCFTDR2 bytes when the flag is set.

Bit 3—Modem Control Enable (MCE): Enables modem control signals CTS2 and RTS2.

Bit 3: MCE	Description	
0	Modem signals disabled*	(Initial value)
1	Modem signals enabled	

Note: CTS2 is fixed at active-0 regardless of the input value, and RTS2 is also fixed at 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	Reset operation disabled*	(Initial value)
1	Reset operation enabled	

Note: Resetting operates in resets and in 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	Reset operation disabled*	(Initial value)
1	Reset operation enabled	

Note: Resetting operates in resets and in standby mode.

Bit 0—Loop-Back Test (LOOP): Internally connects the transmit output pin (TXD2) and receive input pin (RXD2) and enables loop-back testing.

Bit 0: LOOP	Description	
0	Loop-back testing disabled	(Initial value)
1	Loop-back testing enabled	

17.2.10 FIFO Data Count Register 2 (SCFDR2)

The FIFO data count register 2 (SCFDR2) is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register 2 (SCFTDR2) and the receive FIFO data register 2 (SCFRDR2). It indicates the quantity of transmit data in SCFTDR2 in the upper eight bits, and the quantity of receive data in SCFRDR2 in the lower eight bits. SCFDR2 can always be read by 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

SCFDR2 indicates the quantity of receive data stored in SCFRDR2. H'00 means that there is no receive data, and H'10 means that SCFRDR2 is filled with receive data.

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

SCFDR2 indicates the quantity of non-transmitted data stored in SCFTDR2. H'00 means that there is no transmit data, and H'10 means that SCFTDR2 is filled with transmit data.

17.3 Operation

17.3.1 Overview

For serial communication, the SCIF uses an asynchronous mode in which characters are synchronized individually. Refer to section 15.3.2, SCI, for details of asynchronous mode operation. The SCIF has a 16-byte FIFO buffer for both transmit and receive operations, reducing CPU overhead and enabling continuous high-speed communication. It also uses RTS2 and CTS2 modem control signals. The transmission format is selected in the serial mode register 2 (SCSMR2), as shown in table 17.6. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register 2 (SCSCR2), as shown in table 17.7.

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as 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 bytes is displayed for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the built-in 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 of 16 times the bit rate. (The built-in baud rate generator is not used.)

Table 17.6 Serial Mode Register Settings and SCIF Communication Formats

	SCSMR2 Settings					SCIF 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 17.7 SCSMR2 and SCSCR2 Settings and SCIF Clock Source Selection

	SCSCR2 Settings		SCIF Transmit/Receive Clock	
Mode	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK2 Pin Function
Asynchronous mode	0	0	Internal	SCIF does not use the SCK2 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		

17.3.2 Serial Operation

Transmit/Receive Formats: Table 17.8 lists the eight communication formats that can be selected. The format is selected by settings in the serial mode register 2 (SCSMR2).

Table 17.8 Serial Communication Formats

SCSMR2 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 built-in baud rate generator or an external clock input from the SCK2 pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register 2 (SCSMR2) and bits CKE1 and CKE0 in the serial control register 2 (SCSCR2) (table 17.7).

When an external clock is input at the SCK2 pin, it must have a frequency equal to 16 times the desired bit rate.

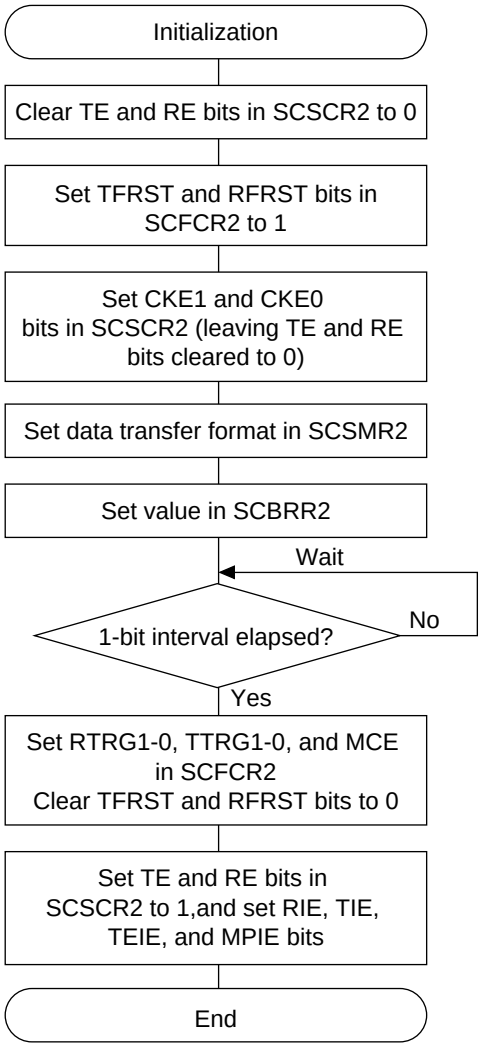
When the SCIF operates on an internal clock, it can output a clock signal at the SCK2 pin. The frequency of this output clock is 16 times the bit rate.

Transmitting and Receiving Data (SCIF Initialization): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register 2 (SCSCR2), 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 2 (SCTSR2). Clearing TE and RE to 0, however, does not initialize the serial status register 2 (SCSSR2), transmit FIFO data register 2 (SCFTDR2), or receive FIFO data register 2 (SCFRDR2), which retain their previous contents. Clear TE to 0 after all transmit data are transmitted and the TEND flag in SCSSR2 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 SCFCR2 to 1 and reset SCFTDR2 before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 17.5 shows a sample flowchart for initializing the SCIF. The procedure for initializing the SCIF is shown in the figure.



1. Set the clock selection in SCSCR2.
   Be sure to clear bits RIE TIE, TE, and RE to 0.
   When clock output is selected, the clock is output immediately after SCSCR2 settings are made.

2. Set the data transfer format in SCSMR2.

3. Write a value corresponding to the bit rate into the bit rate register 2 (SCBRR2).
   (Not necessary if an external clock is used.)

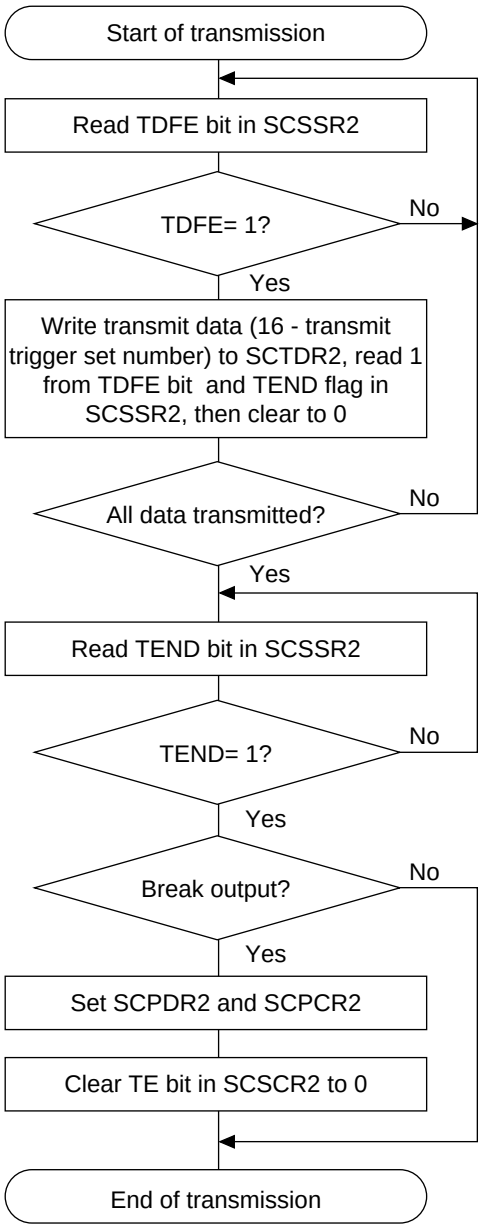
4. Wait at least one bit interval, then set the TE bit or RE bit in SCSCR2 to 1. Also set the RIE and TIE bits.
   Setting the TE and RE bits enables the TxD2 and RxD2 pins to be used. When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 17.5 Sample SCIF Initialization Flowchart

- Serial data transmission

Figure 17.6 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 2 (SCSSR2) and check that the TDFE flag is set to 1, then write transmit data to the transmit FIFO data register 2 (SCFTDR2), 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 setting).

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 SCFTDR2, 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 2 (SCPDR2) and port SC control register 2 (SCPCR2), then clear the TE bit to 0 in the serial control register2 (SCSCR2). For information on SCPDR2 and SCPCR2, see section 17.1.2.

   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 SCFTDR2 indicated by the upper 8 bits of the FIFO data register2 (SCFDR2).

Figure 17.6 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register 2 (SCFTDR2), the SCIF transfers the data from SCFTDR2 to the transmit shift register 2 (SCTSR2) and starts transmitting. Confirm that the TDFE flag in the serial status register 2 (SCSSR2) is set to 1 before writing transmit data to SCFTDR2. The number of data bytes that can be written is (16 - transmit trigger setting).

2. When data is transferred from SCFTDR2 to SCTSR2 and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR2. When the number of transmit data bytes in SCFTDR2 falls below the transmit trigger number set in the FIFO control register 2 (SCFCR2), the TDFE flag is set. If the TIE bit in the serial control register 2 (SCSR2) 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 TxD2 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 SCFTDR2 transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR2 to SCTSR2, 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 SCSSR2 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 17.7 shows an example of the operation for transmission.

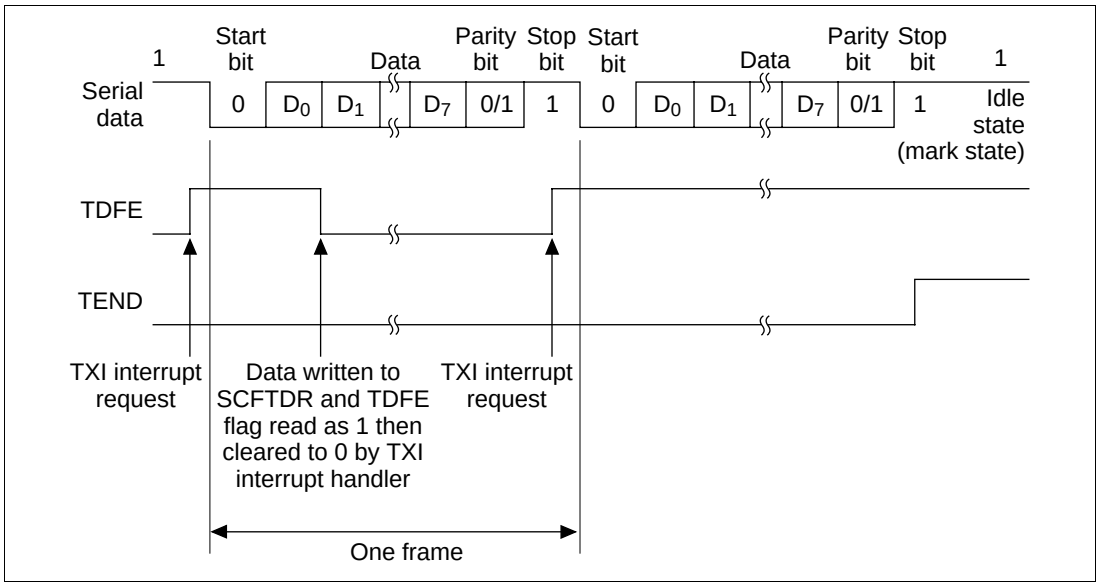


Figure 17.7 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the CTS2 input value. When CTS2 is 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When CTS2 is 0, the next transmit data is output starting from the start bit.

Figure 17.8 shows an example of the operation when modem control is used.

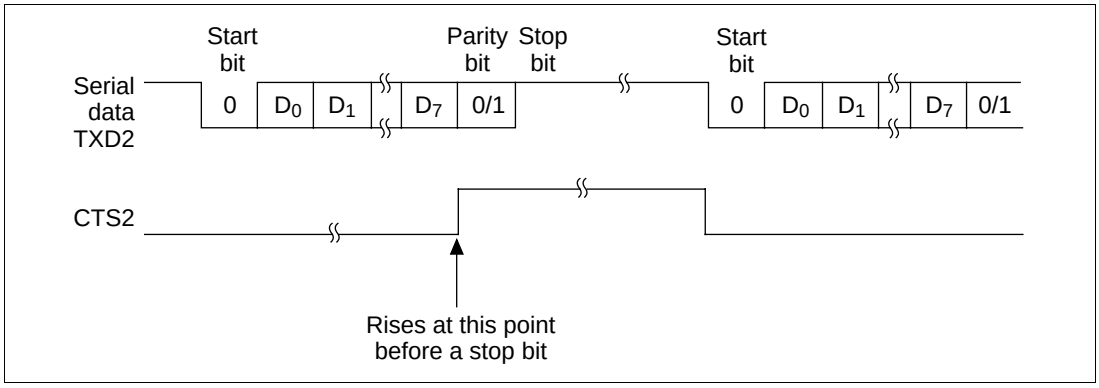
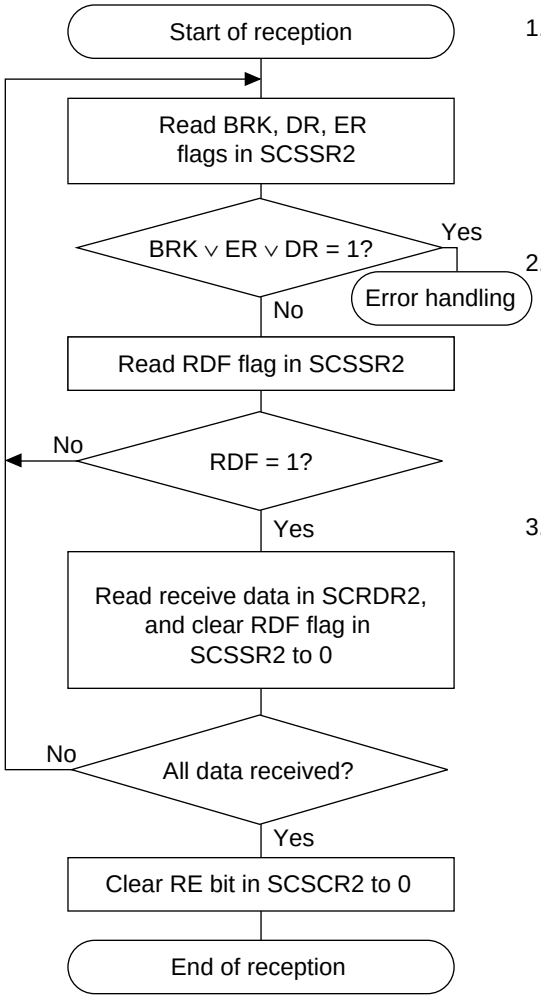


Figure 17.8 Example of Operation Using Modem Control (CTS2)

- Serial data reception

Figure 17.9 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

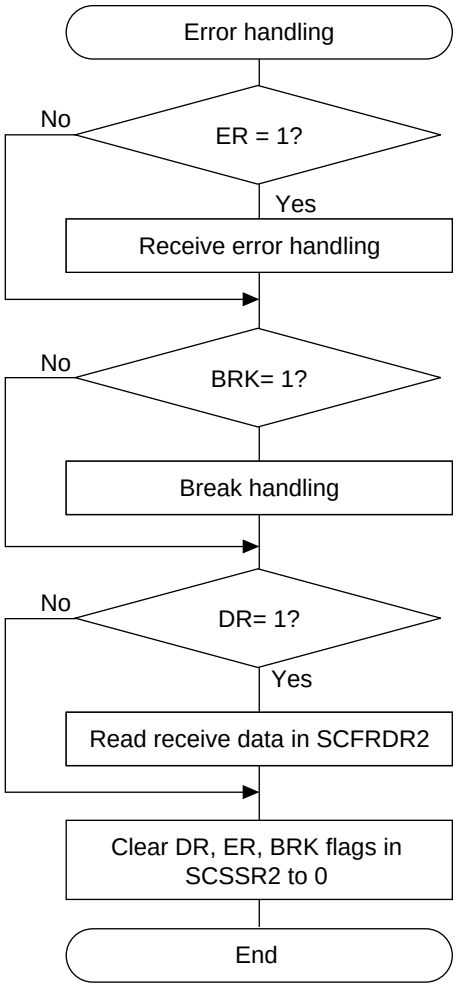


1. Receive error handling and break detection: Read the DR, ER, and BRK flags in SCSSR2 to identify any error, perform the appropriate error handling, then clear the DR, ER, and BRK flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD2 pin.

2. SCIF status check and receive data read : Read the serial status register 2 (SCSSR2) and check that RDF = 1, then read the receive data in the receive FIFO data register 2 (SCFRDR2), 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 SCFRDR2, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR2 can be ascertained by reading the lower bits of SCFDR2.

Figure 17.9 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR2 can be ascertained from the FER and PER bits in SCSSR2.

2. When a break signal is received, receive data is not transferred to SCFRDR2 while the BRK flag is set. However, note that the last data in SCFRDR2 is H'00 and the break data in which a framing error occurred is stored.

Figure 17.10 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 SCRSR2 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 2 (SCRSR2) to SCFRDR2.
   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 SCFRDR2.

   Note: Reception is not suspended when a receive error occurs.

4. If the RIE bit in SCSCR2 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 SCSCR2 is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.
   If the RIE bit in SCSCR2 is set to 1 when the BRK flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 17.11 shows an example of the operation for reception.

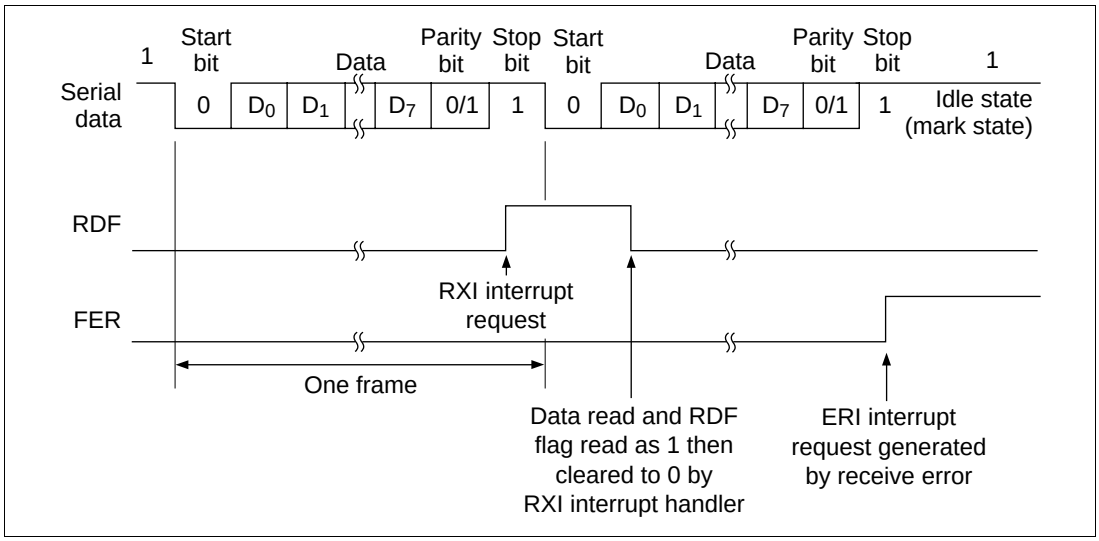


Figure 17.11 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the RTS2 signal is output when SCFRDR2 is empty. When RTS2 is 0, reception is possible. When RTS2 is 1, this indicates that SCFRDR2 is full and reception is not possible.

Figure 17.12 shows an example of the operation when modem control is used.

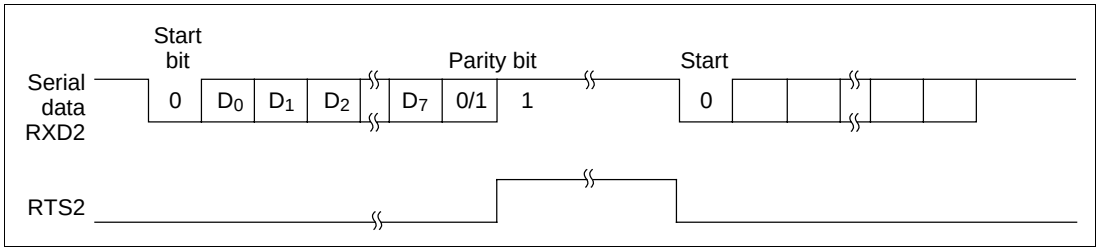


Figure 17.12 Example of Operation Using Modem Control (RTS2)

17.4 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 17-9 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 SCSCR2. 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 2 (SCSSR2) 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 2 (SCFTDR2) by the DMAC.

When the RDF flag in SCSSR2 is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed when the RDF flag in SCSSR2 is set to 1. The RDF flag is automatically cleared to 0 when data is read from the receive data register 2 (SCFRDR2) by the DMAC.

When the ER flag in SCSSR2 is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR2 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 SCFRDR2.

Table 17.9 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data register 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 register empty flag (TDFE)	Possible	Low

See section 4, Exception Handling, for priorities and the relationship with non-SCIF interrupts.

17.5 Usage Notes

Note the following when using the SCIF.

1. SCFTDR2 Writing and the TDFE Flag: The TDFE flag in the serial status register 2 (SCSSR2) is set when the number of transmit data bytes written in the transmit FIFO data register 2 (SCFTDR2) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register 2 (SCFCR2). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR2 can be written, allowing efficient continuous transmission. However, if the number of data bytes written in SCFTDR2 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 SCFTDR2 contains more than the transmit trigger number of transmit data bytes.
   The number of transmit data bytes in SCFTDR2 can be found from the upper 8 bits of the FIFO data count register 2 (SCFDR2).

2. SCFRDR2 Reading and the RDF Flag: The RDF flag in the serial status register 2 (SCSSR2) is set when the number of receive data bytes in the receive FIFO data register 2 (SCFRDR2) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register 2 (SCFCR2). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR2, allowing efficient continuous reception. However, if the number of data bytes in SCFRDR2 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 SCFRDR2 can be found from the lower 8 bits of the FIFO data count register 2 (SCFDR2).

3. Break Detection and Processing: Break signals can be detected by reading the RxD2 pin directly when a framing error (FER) is detected. In the break state the input from the RxD2 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 SCFRDR2 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 TxD2 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 SPB4DT 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 TxD2 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 17.13.

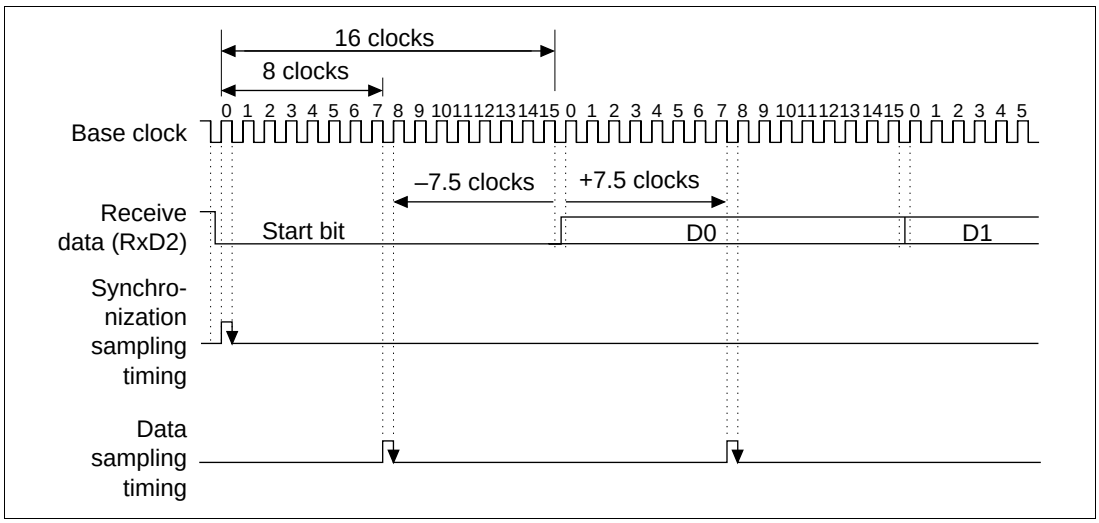


Figure 17.13 Receive Data Sampling Timing in Asynchronous Mode

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:

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \cdots\cdots (2)$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 18 IRDA

18.1 Overview

The SH7707 has an on-chip Infrared Data Association (IRDA) interface which is based on the IrDA 1.0 system and can perform infrared communication. It can also be used as the SCIF by means of register settings.

18.1.1 Features

- Based on the IrDA 1.0 system
- Asynchronous serial communication
  - Data length: Eight bits
  - Stop bit length: One bit
  - Parity bit: None
- Built-in 16-stage FIFO buffers for both transmit and receive
- Built-in baud rate generator with selectable bit rates
- Guard functions to protect the receiver during transmission
- Clock supply halted to reduce power consumption when using not IrDA

IRDA has also the same function as SCIF, except modem control functions. (IRDA has no RTS and CTS pins). See section 17.1.1, Features, for details.

18.1.2 Block Diagram

Figure 18.1 shows a block diagram of the IRDA.

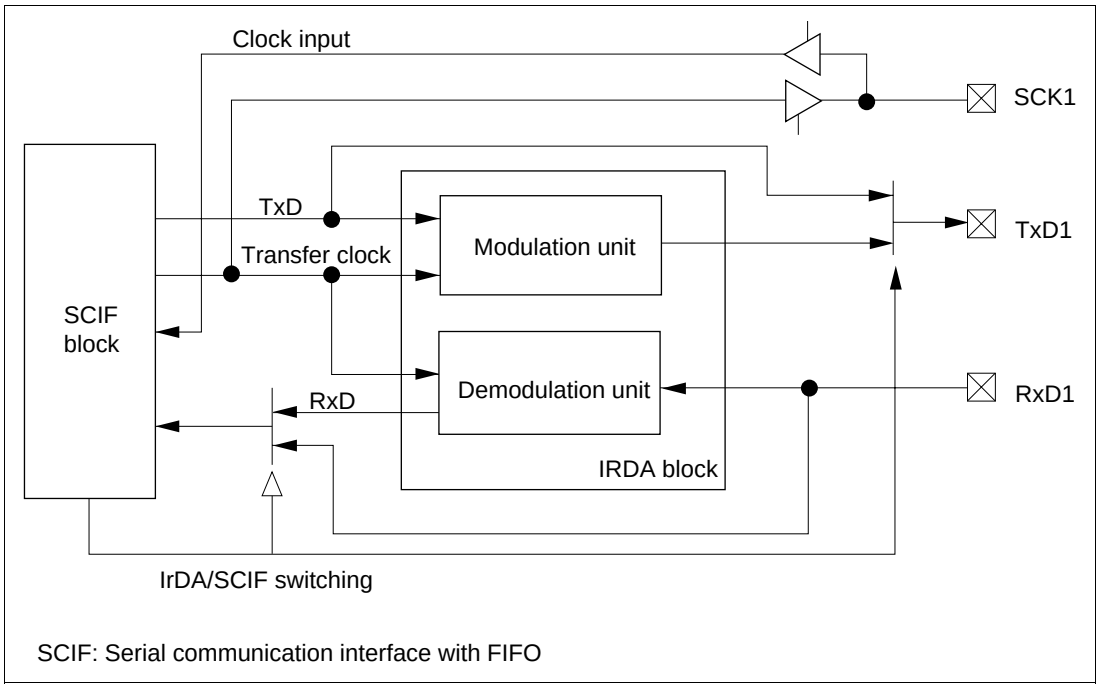


Figure 18.1 IRDA Block Diagram

Figures 18.2 to 18.4 show the IRDA I/O port pins.

IRDA 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 19.3.12, Port SC Control Register (SCPCR) and 20.13.2, Port SC Data Register (SCPDR)

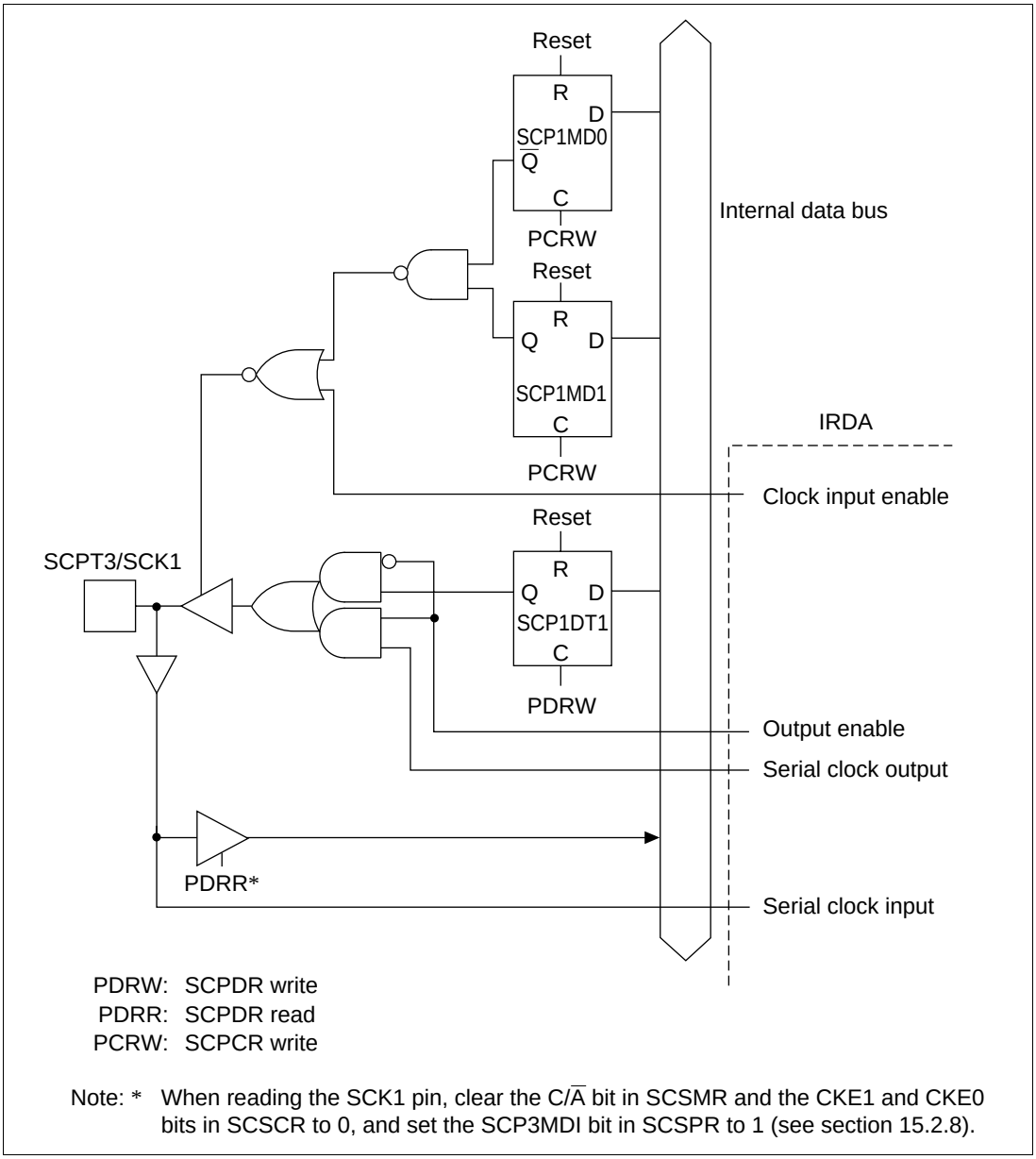


Figure 18.2 SCPT[3]/SCK1 Pin

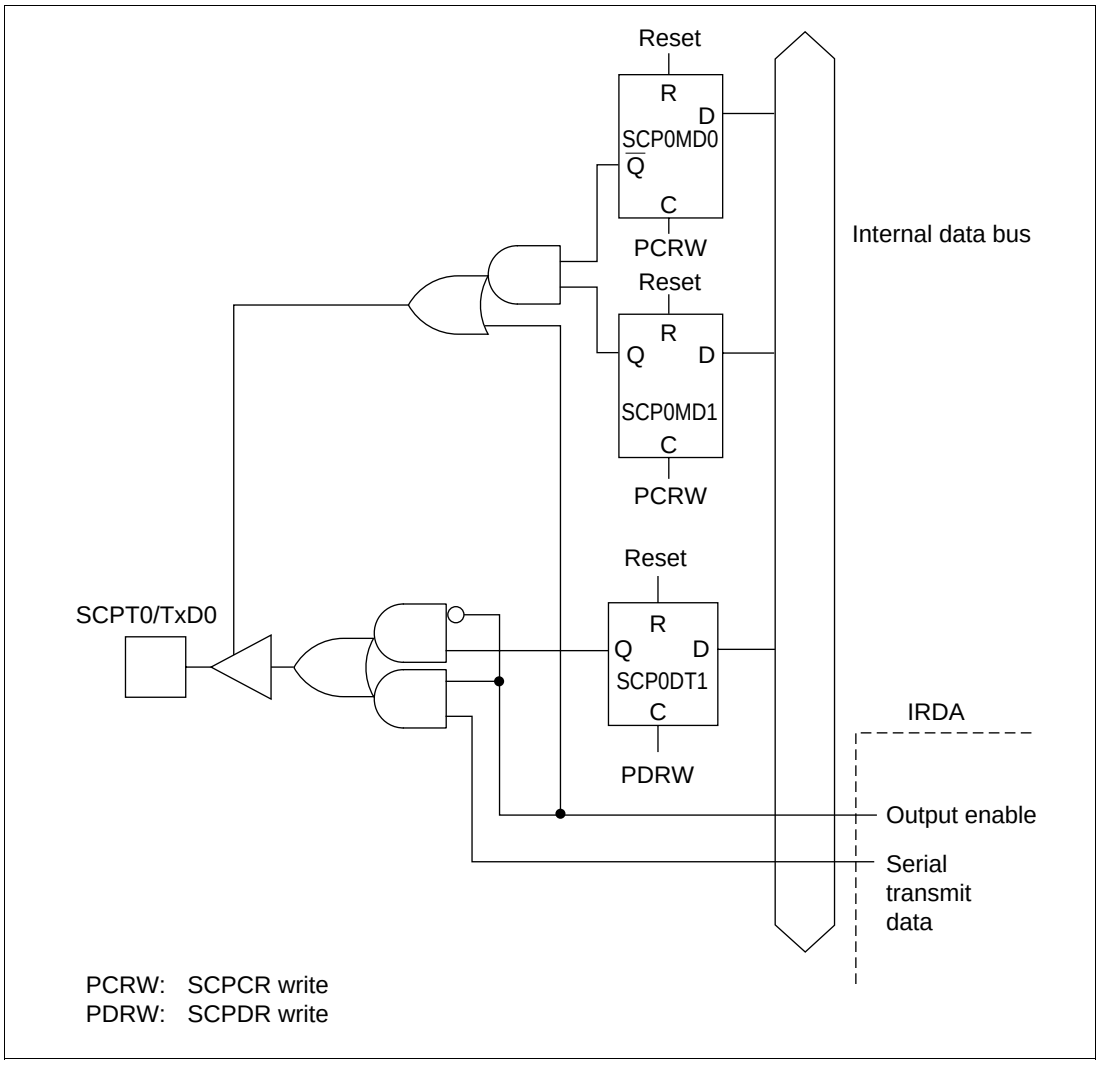


Figure 18.3 SCPT[2]/TxD1 Pin

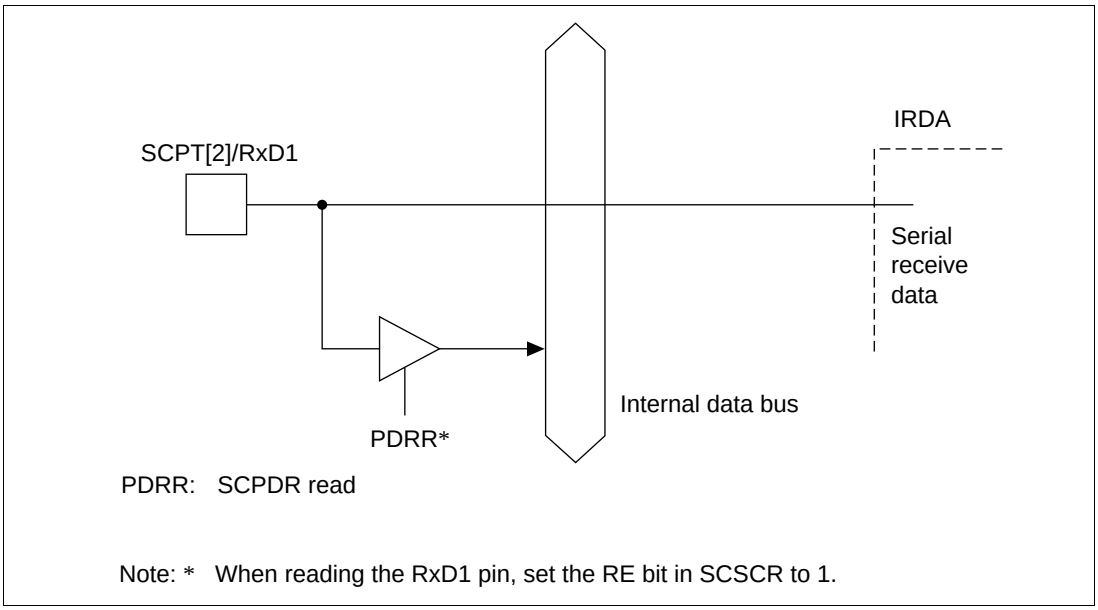


Figure 18.4 SCPT[2]/RxD1 Pin

18.1.3 Pin Configuration

The IRDA module has the serial pins summarized in table 18.1.

Table 18.1 IRDA Pins

Pin Name	Signal Name	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

18.1.4 Register Configuration

The IRDA has the internal registers shown in table 18.2. By using these registers, IrDA or SCIF mode, the data format, and the bit rate can be specified, and the transmit and receive sections can be controlled.

Table 18.2 IRDA Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Serial mode register 1	SCSMR1	R/W	H'00	H'4000140	8 bits
Bit rate register 1	SCBRR1	R/W	H'FF	H'4000142	8 bits
Serial control register 1	SCSCR1	R/W	H'00	H'4000144	8 bits
Transmit FIFO data register 1	SCFTDR1	W	—	H'4000146	8 bits
Serial status register 1	SCSSR1	R/(W)*	H'0060	H'4000148	16 bits
Receive data FIFO register 1	SCFRDR1	R	Undefined	H'400014A	8 bits
FIFO control register 1	SCFCR1	R/W	H'00	H'400014C	8 bits
FIFO data count set register 1	SCFDR1	R	H'0000	H'400014E	16 bits

Note: * Only 0 can be written, to clear the flags.

18.2 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 17, Serial Communication Interface with FIFO, for details of these registers.

18.2.1 Serial Mode Register 1 (SCSMR1)

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

The serial mode register 1 (SCSMR1) is an 8-bit register which can select IrDA or SCIF mode, specify the SCIF serial communication format, select the IrDA output pulse width, and select the 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 at 0. This register functions in the same way as the SCSMR register in the SCIF by setting the IRMOD bit to 0; therefore, this module can also operates as the SCIF.

SCSMR is initialized to H'00 by a power-on reset, manual reset, stoppage by the module standby function, or standby mode.

Bit 7—IrDA Mode (IRMOD): Selects whether this module operates as an IrDA serial communication interface or as an SCIF.

Bit 7: IRMOD	Description	
0	Operates as SCIF	(Initial value)
1	Operates as IRDA	

Bits 6 to 3—IRDA Clock Select Bits (ICK3–ICK0)

Bit 2—Output Pulse Width Select (PSEL)

The output pulse width select bit (PSEL) selects an output pulse width of IRDA that is 3/16 of the bit length for 115 kbps or 3/16 of the bit length for the selected baud rate.

The IRDA clock select bits should be set properly to fix the output pulse width at 3/16 of the bit length for 115 kbps by setting the PSEL bit to 1.

Bit 2: PSEL	Description	
0	Pulse width: 3/16 of selected bit length	(Initial value)
1	Pulse width: 3/16 of 115 kbps bit length	

It is necessary to generate a fixed clock pulse, IRCLK, by dividing the Pϕ clock by 1/2N+2, with the value of N determined by the setting of ICK3-ICK0.

Example:

Pϕ clock: 14.7456 MHz
IRCLK: 921.6 kHz (fixed)
N: Setting of ICK3-ICK0 ($0 \leq N \leq 15$)

$$N = \frac{P\text{ø}}{2 \times IRCLK} = 7$$

Thus, N is 7.

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal baud rate generator clock source. Pϕ, Pϕ/4, Pϕ/16, or Pϕ/64 can be selected.

Refer to section 17.2.9, Bit Rate Register, for the relationship between 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 clock	
1	1	Pϕ/64 clock	

Note: Pϕ: Peripheral clock

18.3 Operation

The IRDA module can perform infrared communication conforming to IrDA 1.0 by connecting an infrared transmit/receive unit. The serial communication interface unit includes a 16-stage FIFO buffer in the transmit section and the receive section, enabling CPU overhead to be reduced and continuous high-speed communication to be performed. This module also supports DMAC data transfer. The IRDA module differs from the SCIF described in section 17 in that it does not include modem control signals RTS and CTS.

Refer to section 17.3, SCIF Operation, for SCIF mode operation.

18.3.1 Overview

The IRDA module modifies TxD/RxD transmit/receive data waveforms to satisfy the IrDA 1.0 specification for infrared communication.

In the IrDA 1.0 specification, communication is first performed at the rate of 9600 bps, and the communication rate is changed. However, the communication rate cannot be automatically changed in this module, and therefore the communication rate must be checked and the appropriate rate set 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 SH7707's RxD pin is active-high, a (Schmitt) inverter must be inserted when connecting an active-low IRDA module.

18.3.2 Transmission

The waveform of the serial output signal (UART frame) from the SCIF block is modified and the signal is converted into the IR frame serial output signal by the IRDA block, as shown in figure 18.5.

When serial data is 0, a 3/16-bit width pulse of the IR frame is generated and output. When serial data is 1, a pulse is not output.

An infrared LED is driven with this signal, modulated into 3/16 width.

18.3.3 Reception

The 3/16-bit width pulse of the IR frame that was received is demodulated and converted into a UART frame, as shown in figure 18.5.

Demodulation to 0 is performed for pulse input, and demodulation to 1 is performed for no pulse input.

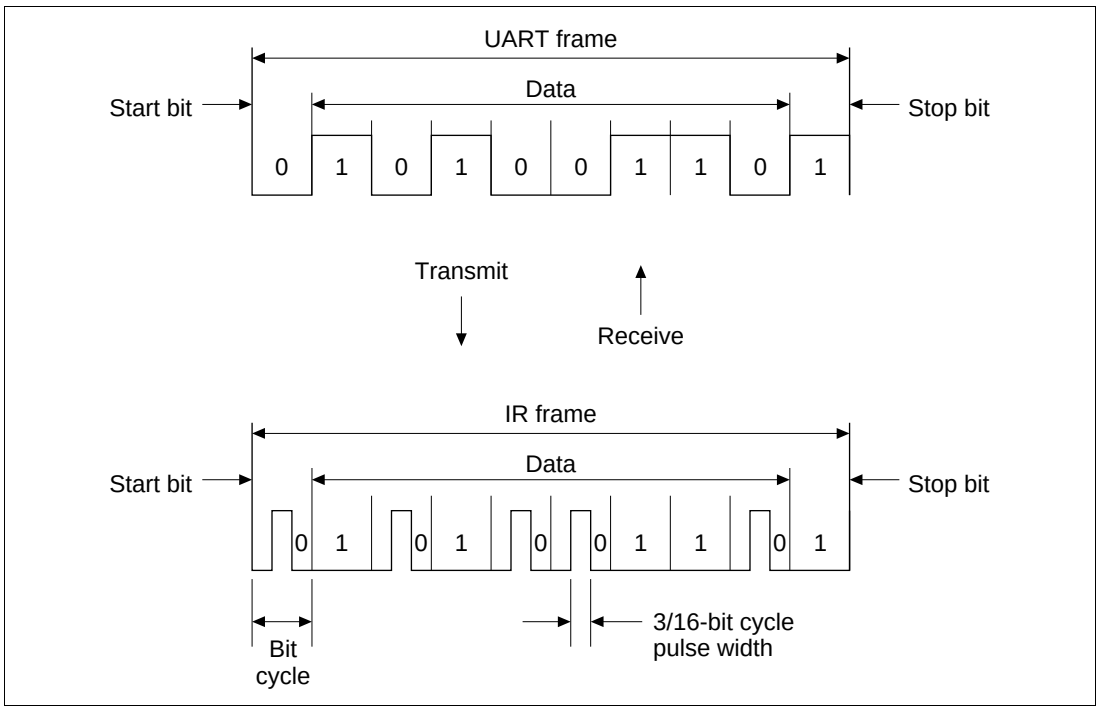


Figure 18.5 Transmit/Receive Operation

Section 19 Pin Function Controller

19.1 Overview

The pin function controller (PFC) is composed of registers for selecting the function of multiplexed pins and the input/output direction. The pin function and input/output direction can be selected for each pin individually without regard to the operating mode of the SH7707. Table 19.1 lists the multiplexed pins.

Table19.1 Multiplexed Pins

Port	Function 1 (Related Module)	Function 2 (Related Module)	Pin No. (FP-208A)	Pin No. (CSP-216)
A	PTA7 I/O (port)/PINT7 input (INTC)	D23 I/O (data bus)	23	B13
A	PTA6 I/O (port)/PINT6 input (INTC)	D22 I/O (data bus)	24	A13
A	PTA5 I/O (port)/PINT5 input (INTC)	D21 I/O (data bus)	25	B14
A	PTA4 I/O (port)/PINT4 input (INTC)	D20 I/O (data bus)	26	A14
A	PTA3 I/O (port)/PINT3 input (INTC)	D19 I/O (data bus)	28	A15
A	PTA2 I/O (port)/PINT2 input (INTC)	D18 I/O (data bus)	30	A16
A	PTA1 I/O (port)/PINT1 input (INTC)	D17 I/O (data bus)	31	B17
A	PTA0 I/O (port)/PINT0 input (INTC)	D16 I/O (data bus)	32	A17
B	PTB7 I/O (port)/PINT15 input (INTC)	D31 I/O (data bus)	13	B08
B	PTB6 I/O (port)/PINT14 input (INTC)	D30 I/O (data bus)	14	A08
B	PTB5 I/O (port)/PINT13 input (INTC)	D29 I/O (data bus)	15	B09
B	PTB4 I/O (port)/PINT12 input (INTC)	D28 I/O (data bus)	16	A09
B	PTB3 I/O (port)/PINT11 input (INTC)	D27 I/O (data bus)	17	B10
B	PTB2 I/O (port)/PINT10 input (INTC)	D26 I/O (data bus)	18	A10
B	PTB1 I/O (port)/PINT9 input (INTC)	D25 I/O (data bus)	20	A11
B	PTB0 I/O (port)/PINT8 input (INTC)	D24 I/O (data bus)	22	A12
C	PTC7 I/O (port)	UD3 output (LCDC)	177	V02
C	PTC6 I/O (port)	UD2 output (LCDC)	178	V01
C	PTC5 I/O (port)	UD1 output (LCDC)	179	U02
C	PTC4 I/O (port)	UD0 output (LCDC)	180	U01
C	PTC3 I/O (port)	LD3 output (LCDC)	185	P02
C	PTC2 I/O (port)	LD2 output (LCDC)	186	P01
C	PTC1 I/O (port)	LD1 output (LCDC)	187	N02
C	PTC0 I/O (port)	LD0 output (LCDC)	188	N01

Table19.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Pin No. (FP-208A)	Pin No. (CSP-216)
D	PTD7 I/O (port)	$\overline{\text{DACK1}}$ output (DMAC)	115	AH23
D	PTD6 input (port)	$\overline{\text{DREQ1}}$ input (DMAC)	192	L01
D	PTD5 I/O (port)	$\overline{\text{DACK0}}$ output (DMAC)	114	AJ24
D	PTD4 input (port)	$\overline{\text{DREQ0}}$ input (DMAC)	191	L02
D	PTD3 I/O (port)	CL1 output (LCDC)	182	T01
D	PTD2 I/O (port)	CL2 output (LCDC)	184	R01
D	PTD1 I/O (port)	FLM output (LCDC)	189	M02
D	PTD0 I/O (port)	DON output (LCDC)	190	M01
E	PTE7 I/O (port)	PCCRD$\overline{\text{WR}}$ output (PCMCIA)	94	AB29
E	PTE6 I/O (port)	$\overline{\text{PCCREG}}$ output (PCMCIA)	116	AJ23
E	PTE5 I/O (port)	$\overline{\text{CE2B}}$ output (PCMCIA)	104	AG29
E	PTE4 I/O (port)	$\overline{\text{CE2A}}$ output (PCMCIA)	103	AG28
E	PTE3 I/O (port)	$\overline{\text{PCC0DRV}}$ output (PCMCIA)	117	AH22
E	PTE2 I/O (port)	PCC0RESET output (PCMCIA)	118	AJ22
E	PTE1 I/O (port)	$\overline{\text{PCC1DRV}}$ output (PCMCIA)	119	AH21
E	PTE0 I/O (port)	PCC1RESET output (PCMCIA)	120	AJ21
F	PTF7 input (port)	PCC1WP input (PCMCIA)	136	AJ13
F	PTF6 input (port)	PCC1READY input (PCMCIA)	137	AH12
F	PTF5 input (port)	PCC1BVD1 input (PCMCIA)	138	AJ12
F	PTF4 input (port)	PCC1BVD2 input (PCMCIA)	139	AH11
F	PTF3 input (port)	$\overline{\text{PCC1CD1}}$ input (PCMCIA)	140	AJ11
F	PTF2 input (port)	$\overline{\text{PCC1CD2}}$ input (PCMCIA)	141	AH10
F	PTF1 input (port)	$\overline{\text{PCC1VS1}}$ input (PCMCIA)	142	AJ10
F	PTF0 input (port)	$\overline{\text{PCC1VS2}}$ input (PCMCIA)	143	AH09
G	PTG7 input (port)	PCC0WP input (PCMCIA)	126	AJ18
G	PTG6 input (port)	PCC0READY input (PCMCIA)	127	AH17
G	PTG5 input (port)	PCC0BVD1 input (PCMCIA)	128	AJ17
G	PTG4 input (port)	PCC0BVD2 input (PCMCIA)	129	AH16
G	PTG3 input (port)	$\overline{\text{PCC0CD1}}$ input (PCMCIA)	130	AJ16
G	PTG2 input (port)	$\overline{\text{PCC0CD2}}$ input (PCMCIA)	131	AH15
G	PTG1 input (port)	$\overline{\text{PCC0VS1}}$ input (PCMCIA)	133	AH14
G	PTG0 input (port)	$\overline{\text{PCC0VS2}}$ input (PCMCIA)	135	AH13

Table19.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Pin No. (FP-208A)	Pin No. (CSP-216)
H	PTH7 I/O (port)	TCLK I/O (timer)	159	AG02
H	PTH6 input (port)	$\overline{\text{PCC0WAIT}}$ input (PCMCIA)	124	AJ19
H	PTH5 input (port)	$\overline{\text{PCC1WAIT}}$ input (PCMCIA)	125	AH18
H	PTH4 input (port) IRQ4 input (INTC)	$\overline{\text{ADTRG}}$ input (A/D converter) IRQ4 input (INTC)	12	A07
H	PTH3 input (port) IRQ3 input (INTC)	IRQ3 input (INTC) $\overline{\text{IRL3}}$ input (INTC)	11	B07
H	PTH2 input (port) IRQ2 input (INTC)	IRQ2 input (INTC) $\overline{\text{IRL2}}$ input (INTC)	10	A06
H	PTH1 input (port) IRQ1 input (INTC)	IRQ1 input (INTC) $\overline{\text{IRL1}}$ input (INTC)	9	B06
H	PTH0 input (port) IRQ0 input (INTC)	IRQ0 input (INTC) $\overline{\text{IRL0}}$ input (INTC)	8	A05
J	PTJ7 I/O (port)	STATUS1 output (SYSC)	158	AH01
J	PTJ6 I/O (port)	STATUS0 output (SYSC)	157	AH02
J	PTJ5 I/O (port)	$\overline{\text{CASHH}}$ output (BSC) $\overline{\text{CAS2H}}$ output (BSC)	113	AH24
J	PTJ4 I/O (port)	$\overline{\text{CASHL}}$ output (BSC) $\overline{\text{CAS2L}}$ output (BSC)	112	AJ25
J	PTJ3 I/O (port)	$\overline{\text{CASLH}}$ output (BSC)	110	AJ26
J	PTJ2 I/O (port)	$\overline{\text{CASLL}}$ output (BSC)	108	AJ27
J	PTJ1 I/O (port)	$\overline{\text{RAS2}}$ output (BSC)	107	AH27
J	PTJ0 I/O (port)	$\overline{\text{RAS}}$ output (BSC)	106	AJ28
K	PTK7 I/O (port)	$\overline{\text{WE3}}$ output (BSC) $\overline{\text{ICIOWR}}$ output (BSC)	92	AA29
K	PTK6 I/O (port)	$\overline{\text{WE2}}$ output (BSC) $\overline{\text{ICIORD}}$ output (BSC)	91	AA28
K	PTK5 I/O (port)	NC	105	AH28
K	PTK4 I/O (port)	$\overline{\text{BS}}$ output (BSC)	87	W28
K	PTK3 I/O (port)	$\overline{\text{CS5}}$ output (BSC) $\overline{\text{CE1A}}$ output (BSC)	101	AF28
K	PTK2 I/O (port)	$\overline{\text{CS4}}$ output (BSC)	100	AE29
K	PTK1 I/O (port)	$\overline{\text{CS3}}$ output (BSC)	99	AE28
K	PTK0 I/O (port)	$\overline{\text{CS2}}$ output (BSC)	98	AD29

Table19.1 Multiplexed Pins (cont)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Pin No. (FP-208A)	Pin No. (CSP-216)
L	PTL7 input (port)	AN7 input (A/D converter) DA0 output (D/A converter)	207	C02
L	PTL6 input (port)	AN6 input (A/D converter) DA1 output (D/A converter)	206	D01
L	PTL5 input (port)	AN5 input (A/D converter)	204	E01
L	PTL4 input (port)	AN4 input (A/D converter)	203	E02
L	PTL3 input (port)	AN3 input (A/D converter)	202	F01
L	PTL2 input (port)	AN2 input (A/D converter)	201	F02
L	PTL1 input (port)	AN1 input (A/D converter)	200	G01
L	PTL0 input (port)	AN0 input (A/D converter)	199	G02
SCPT	SCPT7 input (port) IRQ5 input (INTC)	CTS2 input (SCIF) IRQ5 input (INTC)	176	W01
SCPT	SCPT6 I/O (port)	RTS2 output (SCIF)	170	AB01
SCPT	SCPT5 I/O (port)	SCK2 I/O (SCIF)	169	AB02
SCPT	SCPT4 input (port)	RxD2 input (SCIF)	174	Y01
	SCPT4 output (port)	TxD2 output (SCIF)	168	AC01
SCPT	SCPT3 I/O (port)	SCK1 I/O (IRDA)	167	AC02
SCPT	SCPT2 input (port)	RxD1 input (IRDA)	172	AA01
	SCPT2 output (port)	TxD1 output (IRDA)	166	AD01
SCPT	SCPT1 I/O (port)	SCK0 I/O (SCI)	165	AD02
SCPT	SCPT0 input (port)	RxD0 input (SCI)	171	AA02
	SCPT0 output (port)	TxD0 output (SCI)	164	AE01

Note: SCPT0, SCPT2, and SCPT4 have the same data register for access, although they have different input pins and output pins.

19.2 Register Configuration

Table 19.2 summarizes the registers of the pin function controller.

Table19.2 Pin Function Controller Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A control register	PACR	R/W	H'0000	H'4000100	16
Port B control register	PBCR	R/W	H'0000	H'4000102	16
Port C control register	PCCR	R/W	H'AAAA	H'4000104	16
Port D control register	PDCR	R/W	H'AAAA	H'4000106	16
Port E control register	PECR	R/W	H'AAAA	H'4000108	16
Port F control register	PFCR	R/W	H'AAAA	H'400010A	16
Port G control register	PGCR	R/W	H'AAAA	H'400010C	16
Port H control register	PHCR	R/W	H'AAAA	H'400010E	16
Port J control register	PJCR	R/W	H'0000	H'4000100	16
Port K control register	PKCR	R/W	H'0000	H'4000112	16
Port L control register	PLCR	R/W	H'0000	H'4000114	16
SC port control register	SCPCR	R/W	H'A888	H'4000116	16

19.3 Register Descriptions

19.3.1 Port A Control Register (PACR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0
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:	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0
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 port A control register (PACR) is a 16-bit readable/writable register that selects the pin functions. PACR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

When port A is set to port output or port input, set either 8 or 16 bits as the memory bus width for all areas.

Bits 15 and 14—PA7 Mode 1 and 0 (PA7MD1, PA7MD0)
Bits 13 and 12—PA6 Mode 1 and 0 (PA6MD1, PA6MD0)
Bits 11 and 10—PA5 Mode 1 and 0 (PA5MD1, PA5MD0)
Bits 9 and 8—PA4 Mode 1 and 0 (PA4MD1, PA4MD0)
Bits 7 and 6—PA3 Mode 1 and 0 (PA3MD1, PA3MD0)
Bits 5 and 4—PA2 Mode 1 and 0 (PA2MD1, PA2MD0)
Bits 3 and 2—PA1 Mode 1 and 0 (PA1MD1, PA1MD0)
Bits 1 and 0—PA0 Mode 1 and 0 (PA0MD1, PA0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PAnMD1	Bit 2n: PAnMD0	Pin Function*	
0	0	Hiz	(Initial value)
	1	PTA[n] output	
1	0	PTA[n]/PINT[n] input (pullup MOS: on)	
	1	PTA[n]/PINT[n] input (pullup MOS: off)	

(n = 7–0)

Note: When the data bus width for all areas is 8 bits or 16 bits.

19.3.2 Port B Control Register (PBCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
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:	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0
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 port B control register (PBCR) is a 16-bit readable/writable register that selects the pin functions. PBCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

When port B is set to port output or port input, set either 8 or 16 bits as the memory bus width for all areas.

Bits 15 and 14—PB7 Mode 1 and 0 (PB7MD1, PB7MD0)
Bits 13 and 12—PB6 Mode 1 and 0 (PB6MD1, PB6MD0)
Bits 11 and 10—PB5 Mode 1 and 0 (PB5MD1, PB5MD0)
Bits 9 and 8—PB4 Mode 1 and 0 (PB4MD1, PB4MD0)
Bits 7 and 6—PB3 Mode 1 and 0 (PB3MD1, PB3MD0)
Bits 5 and 4—PB2 Mode 1 and 0 (PB2MD1, PB2MD0)
Bits 3 and 2—PB1 Mode 1 and 0 (PB1MD1, PB1MD0)
Bits 1 and 0—PB0 Mode 1 and 0 (PB0MD1, PB0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PBnMD1	Bit 2n: PBnMD0	Pin Function*	
0	0	Hiz	(Initial value)
	1	PTB[n] output	
1	0	PTB[n]/PINT[n+8] input (pullup MOS: on)	
	1	PTB[n]/PINT[n+8] input (pullup MOS: off)	

(n = 7–0)

Note: When the data bus width for all areas is 8 bits or 16 bits.

19.3.3 Port C Control Register (PCCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0
Initial value:	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

The port C control register (PCCR) is a 16-bit readable/writable register that selects the pin functions. PCCR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PC7 Mode 1 and 0 (PC7MD1, PC7MD0)
Bits 13 and 12—PC6 Mode 1 and 0 (PC6MD1, PC6MD0)
Bits 11 and 10—PC5 Mode 1 and 0 (PC5MD1, PC5MD0)
Bits 9 and 8—PC4 Mode 1 and 0 (PC4MD1, PC4MD0)
Bits 7 and 6—PC3 Mode 1 and 0 (PC3MD1, PC3MD0)
Bits 5 and 4—PC2 Mode 1 and 0 (PC2MD1, PC2MD0)
Bits 3 and 2—PC1 Mode 1 and 0 (PC1MD1, PC1MD0)
Bits 1 and 0—PC0 Mode 1 and 0 (PC0MD1, PC0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PCnMD1	Bit 2n: PCnMD0	Pin Function	
0	0	Other function	
	1	PTC[n] output	
1	0	PTC[n] input (pullup MOS: on)	(Initial value)
	1	PTC[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.4 Port D Control Register (PDCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0
Initial value:	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

The port D control register (PDCR) is a 16-bit readable/writable register that selects the pin functions. PDCR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PD7 Mode 1 and 0 (PD7MD1, PD7MD0)
Bits 11 and 10—PD5 Mode 1 and 0 (PD5MD1, PD5MD0)
Bits 7 and 6—PD3 Mode 1 and 0 (PD3MD1, PD3MD0)
Bits 5 and 4—PD2 Mode 1 and 0 (PD2MD1, PD2MD0)
Bits 3 and 2—PD1 Mode 1 and 0 (PD1MD1, PD1MD0)
Bits 1 and 0—PD0 Mode 1 and 0 (PD0MD1, PD0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PDnMD1	Bit 2n: PDnMD0	Pin Function	
0	0	Other function	
	1	PTD[n] output	
1	0	PTD[n] input (pullup MOS: on)	(Initial value)
	1	PTD[n] input (pullup MOS: off)	

(n = 7, 5, 3–0)

Note: “Other function” indicates Function 2 in table 19.1.

Bits 13 and 12—PD6 Mode 1 and 0 (PD6MD1, PD6MD0)
Bits 9 and 8—PD4 Mode 1 and 0 (PD4MD1, PD4MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PDnMD1	Bit 2n: PDnMD0	Pin Function	
0	0	Other function	
	1	Reserved	
1	0	PTD[n] input (pullup MOS: on)	(Initial value)
	1	PTD[n] input (pullup MOS: off)	

(n = 6, 4)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.5 Port E Control Register (PECR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0
Initial value:	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

The port E control register (PECR) is a 16-bit readable/writable register that selects the pin functions. PECR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PE7 Mode 1 and 0 (PE7MD1, PE7MD0)
Bits 13 and 12—PE6 Mode 1 and 0 (PE6MD1, PE6MD0)
Bits 11 and 10—PE5 Mode 1 and 0 (PE5MD1, PE5MD0)
Bits 9 and 8—PE4 Mode 1 and 0 (PE4MD1, PE4MD0)
Bits 7 and 6—PE3 Mode 1 and 0 (PE3MD1, PE3MD0)
Bits 5 and 4—PE2 Mode 1 and 0 (PE2MD1, PE2MD0)
Bits 3 and 2—PE1 Mode 1 and 0 (PE1MD1, PE1MD0)
Bits 1 and 0—PE0 Mode 1 and 0 (PE0MD1, PE0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PEnMD1	Bit 2n: PEnMD0	Pin Function	
0	0	Other function	
	1	PTE[n] output	
1	0	PTE[n] input (pullup MOS: on)	(Initial value)
	1	PTE[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.6 Port F Control Register (PFCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0
Initial value:	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

The port F control register (PFCR) is a 16-bit readable/writable register that selects the pin functions. PFCR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PF7 Mode 1 and 0 (PF7MD1, PF7MD0)
Bits 13 and 12—PF6 Mode 1 and 0 (PF6MD1, PF6MD0)
Bits 11 and 10—PF5 Mode 1 and 0 (PF5MD1, PF5MD0)
Bits 9 and 8—PF4 Mode 1 and 0 (PF4MD1, PF4MD0)
Bits 7 and 6—PF3 Mode 1 and 0 (PF3MD1, PF3MD0)
Bits 5 and 4—PF2 Mode 1 and 0 (PF2MD1, PF2MD0)
Bits 3 and 2—PF1 Mode 1 and 0 (PF1MD1, PF1MD0)
Bits 1 and 0—PF0 Mode 1 and 0 (PF0MD1, PF0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PFnMD1	Bit 2n: PFnMD0	Pin Function	
0	0	Other function	
	1	Reserved	
1	0	PTF[n] input (pullup MOS: on)	(Initial value)
	1	PTF[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.7 Port G Control Register (PGCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PG7MD1	PG7MD0	PG6MD1	PG6MD0	PG5MD1	PG5MD0	PG4MD1	PG4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0
Initial value:	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

The port G control register (PGCR) is a 16-bit readable/writable register that selects the pin functions. PGCR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PG7 Mode 1 and 0 (PG7MD1, PG7MD0)
Bits 13 and 12—PG6 Mode 1 and 0 (PG6MD1, PG6MD0)
Bits 11 and 10—PG5 Mode 1 and 0 (PG5MD1, PG5MD0)
Bits 9 and 8—PG4 Mode 1 and 0 (PG4MD1, PG4MD0)
Bits 7 and 6—PG3 Mode 1 and 0 (PG3MD1, PG3MD0)
Bits 5 and 4—PG2 Mode 1 and 0 (PG2MD1, PG2MD0)
Bits 3 and 2—PG1 Mode 1 and 0 (PG1MD1, PG1MD0)
Bits 1 and 0—PG0 Mode 1 and 0 (PG0MD1, PG0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PGnMD1	**Bit 2n: PGnMD0**	**Pin Function**	
0	0	Other function	
	1	Reserved	
1	0	PTG[n] input (pullup MOS: on)	(Initial value)
	1	PTG[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.8 Port H Control Register (PHCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PH7MD1	PH7MD0	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
Bit name:	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0
Initial value:	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

The port H control register (PHCR) is a 16-bit readable/writable register that selects the pin functions. PHCR is initialized to H'AAAA by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PH7 Mode 1 and 0 (PH7MD1, PH7MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PHnMD1	Bit 2n: PHnMD0	Pin Function	
0	0	Other function	
	1	PTH[n] output	
1	0	PTH[n] input (pullup MOS: on)	(Initial value)
	1	PTH[n] input (pullup MOS: off)	

(n = 7)

Note: “Other function” indicates Function 2 in table 19.1.

Bits 13 and 12—PH6 Mode 1 and 0 (PH6MD1, PH6MD0)
Bits 11 and 10—PH5 Mode 1 and 0 (PH5MD1, PH5MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PHnMD1	Bit 2n: PHnMD0	Pin Function	
0	0	Other function	
	1	Reserved	
1	0	PTH[n] input (pullup MOS: on)	(Initial value)
	1	PTH[n] input (pullup MOS: off)	

(n = 6, 5)

Note: “Other function” indicates Function 2 in table 19.1.

Bits 9 and 8—PH4 Mode 1 and 0 (PH4MD1, PH4MD0)
Bits 7 and 6—PH3 Mode 1 and 0 (PH3MD1, PH3MD0)
Bits 5 and 4—PH2 Mode 1 and 0 (PH2MD1, PH2MD0)
Bits 3 and 2—PH1 Mode 1 and 0 (PH1MD1, PH1MD0)
Bits 1 and 0—PH0 Mode 1 and 0 (PH0MD1, PH0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PHnMD1	Bit 2n: PHnMD0	Pin Function	
0	0	Other function	
	1	Reserved	
1	0	PTH[n] input (pullup MOS: on)/IRQ[n] input	(Initial value)
	1	PTH[n] input (pullup MOS: off)/IRQ[n] input	

(n = 4–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.9 Port J Control Register (PJCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PJ7MD1	PJ7MD0	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4MD0
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:	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0
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 port J control register (PJCR) is a 16-bit readable/writable register that selects the pin functions. PJCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PJ7 Mode 1 and 0 (PJ7MD1, PJ7MD0)
Bits 13 and 12—PJ6 Mode 1 and 0 (PJ6MD1, PJ6MD0)
Bits 11 and 10—PJ5 Mode 1 and 0 (PJ5MD1, PJ5MD0)
Bits 9 and 8—PJ4 Mode 1 and 0 (PJ4MD1, PJ4MD0)
Bits 7 and 6—PJ3 Mode 1 and 0 (PJ3MD1, PJ3MD0)
Bits 5 and 4—PJ2 Mode 1 and 0 (PJ2MD1, PJ2MD0)
Bits 3 and 2—PJ1 Mode 1 and 0 (PJ1MD1, PJ1MD0)
Bits 1 and 0—PJ0 Mode 1 and 0 (PJ0MD1, PJ0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PJnMD1	Bit 2n: PJnMD0	Pin Function	
0	0	Other function	(Initial value)
	1	PTJ[n] output	
1	0	PTJ[n] input (pullup MOS: on)	
	1	PTJ[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.10 Port K Control Register (PKCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PK7MD1	PK7MD0	PK6MD1	PK6MD0	PK5MD1	PK5MD0	PK4MD1	PK4MD0
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:	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK0MD0
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 port K control register (PKCR) is a 16-bit readable/writable register that selects the pin functions. PKCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

Bits 15 and 14—PK7 Mode 1 and 0 (PK7MD1, PK7MD0)
Bits 13 and 12—PK6 Mode 1 and 0 (PK6MD1, PK6MD0)
Bits 11 and 10—PK5 Mode 1 and 0 (PK5MD1, PK5MD0)
Bits 9 and 8—PK4 Mode 1 and 0 (PK4MD1, PK4MD0)
Bits 7 and 6—PK3 Mode 1 and 0 (PK3MD1, PK3MD0)
Bits 5 and 4—PK2 Mode 1 and 0 (PK2MD1, PK2MD0)
Bits 3 and 2—PK1 Mode 1 and 0 (PK1MD1, PK1MD0)
Bits 1 and 0—PK0 Mode 1 and 0 (PK0MD1, PK0MD0)

These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): PKnMD1	**Bit 2n: PKnMD0**	**Pin Function**	
0	0	Other function	(Initial value)
	1	PTK[n] output	
1	0	PTK[n] input (pullup MOS: on)	
	1	PTK[n] input (pullup MOS: off)	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.11 Port L Control Register (PLCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PL7MD1	PL7MD0	PL6MD1	PL6MD0	PL5MD1	PL5MD0	PL4MD1	PL4MD0
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:	PL3MD1	PL3MD0	PL2MD1	PL2MD0	PL1MD1	PL1MD0	PL0MD1	PL0MD0
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 port L control register (PLCR) is a 16-bit readable/writable register that selects the pin functions. PLCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

When the DA0 and DA1 pins are used as D/A converter outputs, clear the PL7MD0 and PL7MD1 bits and the PL6MD0 and PL6MD1 bits in PLCR to 0.

When pin AD(n) is used as a A/D converter input, clear the PLnMD1 and PLnMD0 bits in PLCR to 0.

Bits 15 and 14—PL7 Mode 1 and 0 (PL7MD1, PL7MD0)
Bits 13 and 12—PL6 Mode 1 and 0 (PL6MD1, PL6MD0)
Bits 11 and 10—PL5 Mode 1 and 0 (PL5MD1, PL5MD0)
Bits 9 and 8—PL4 Mode 1 and 0 (PL4MD1, PL4MD0)
Bits 7 and 6—PL3 Mode 1 and 0 (PL3MD1, PL3MD0)
Bits 5 and 4—PL2 Mode 1 and 0 (PL2MD1, PL2MD0)
Bits 3 and 2—PL1 Mode 1 and 0 (PL1MD1, PL1MD0)
Bits 1 and 0—PL0 Mode 1 and 0 (PL0MD1, PL0MD0)

These bits select the pin functions.

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function	
0	0	Other function	(Initial value)
	1	Reserved	
1	0	PTL[n] input	
	1	PTL[n] input	

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

19.3.12 Port SC Control Register (SCPCR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	SCP7MD1	SCP7MD0	SCP6MD1	SCP6MD0	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0
Initial value:	1	0	1	0	1	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:	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCP0MD0
Initial value:	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

The port SC control register (SCPCR) is a 16-bit readable/writable register that selects the pin functions. The setting of SCPCR is valid only when transmit/receive operations are disabled by the SCSCR register setting. SCPCR is initialized to H'A888 by a power-on reset, but is not initialized by a manual reset, or in standby mode or sleep mode.

When the TE bit in SCSCR is set to 1, TxD[2:0] output state has priority over the SCPCR setting for the TxD[2:0] pins. The SCPT[4:2:0] signals are not output.

When the RE bit in SCSCR is set to 1, the input state has priority over the SCPCR setting for the RxD[2:0] pins. RxD[2:0] input and SCPT[4:2:0] pin status reading are possible.

Bits 15 and 14—SCP7 Mode 1 and 0 (SCP7MD1, SCP7MD0): These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): SCPnMD1	**Bit 2n: SCPnMD0**	**Pin Function**	
0	0	CTS2 input/IRQ5 input	
	1	Reserved	
1	0	SCPT7 input (pullup MOS: on)/IRQ5 input	(Initial value)
	1	SCPT7 input (pullup MOS: off)/IRQ5 input	

(n = 7)

Bits 13 and 12—SCP6 Mode 1 and 0 (SCP6MD1, SCP6MD0): These bits select the pin functions and perform input pullup MOS control.

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function	
0	0	RTS2 output	
	1	SCPT6 output	
1	0	SCPT6 input (pullup MOS: on)	(Initial value)
	1	SCPT6 input (pullup MOS: off)	

(n = 6)

Bits 11 and 10—SCP5 Mode 1 and 0 (SCP5MD1, SCP5MD0): These bits select the pin functions and perform input pullup MOS control. When using the SCPT5 function of the SCK2/SCPT5 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	No function	
	1	SCPT5 output	
1	0	SCPT5 input (pullup MOS: on)	(Initial value)
	1	SCPT5 input (pullup MOS: off)	

(n = 5)

Note: Clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bits 9 and 8—SCP4 Mode 1 and 0 (SCP4MD1, SCP4MD0): These bits select the pin functions and perform input pullup MOS control.

The SCPT4 data register is also used in SCIF transmission/reception (see section 17).

When using the SCPT4 function of the TxD2/SCPT4 pin independently of SCIF transmission/reception, clear the TE bit in SCSCR to 0.

When using the SCPT4 function of the RxD2/SCPT4 pin independently of SCIF transmission/reception, clear the RE bit and TE bit in SCSCR to 0.

TxD2

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	HiZ	(Initial value)
	1	SCPT4 output	
1	0	HiZ	
	1	HiZ	

(n = 4)

Note: Clear the TE bit in SCSCR to 0.

RxD2

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	SCPT4 input (pullup MOS: off)	(Initial value)
	1	HiZ	
1	0	SCPT4 input (pullup MOS: on)	
	1	SCPT4 input (pullup MOS: off)	

(n = 4)

Note: Clear the RE bit and TE bit in SCSCR to 0.

Bits 7 and 6—SCP3 Mode 1 and 0 (SCP3MD1, SCP3MD0): These bits select the pin functions and perform input pullup MOS control.

When using the SCPT3 function of the SCK1/SCPT3 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function	
0	0	No function	
	1	SCPT3 output	
1	0	SCPT3 input (pullup MOS: on)	(Initial value)
	1	SCPT3 input (pullup MOS: off)	

(n = 3)

Note: Clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bits 5 and 4—SCP2 Mode 1 and 0 (SCP2MD1, SCP2MD0): These bits select the pin functions and perform input pullup MOS control.

The SCPT2 data register is also used in IRDA transmission/reception (see section 18).

When using the SCPT2 function of the TxD1/SCPT2 pin independently of IRDA transmission/reception, clear the TE bit in SCSCR to 0.

When using the SCPT2 function of the RxD1/SCPT2 pin independently of IRDA transmission/reception, clear the RE bit and TE bit in SCSCR to 0.

TxD1

Bit (2n+1): SCPnMD1	**Bit 2n: SCPnMD0**	**Pin Function***	
0	0	HiZ	(Initial value)
	1	SCPT2 output	
1	0	HiZ	
	1	HiZ	

(n = 2)

Note: Clear the TE bit in SCSCR to 0.

RxD1

Bit (2n+1): SCPnMD1	**Bit 2n: SCPnMD0**	**Pin Function***	
0	0	SCPT2 input (pullup MOS: off)	(Initial value)
	1	HiZ	
1	0	SCPT2 input (pullup MOS: on)	
	1	SCPT2 input (pullup MOS: off)	

(n = 2)

Note: Clear the RE bit and TE bit in SCSCR to 0.

Bits 3 and 2—SCP1 Mode 1 and 0 (SCP1MD1, SCP1MD0): These bits select the pin functions and perform input pullup MOS control.

When using the SCPT1 function of the SCK0/SCPT1 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	No function	
	1	SCPT1 output	
1	0	SCPT1 input (pullup MOS: on)	(Initial value)
	1	SCPT1 input (pullup MOS: off)	

(n = 1)

Note: Clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0.

Bits 1 and 0—SCP0 Mode 1 and 0 (SCP0MD1, SCP0MD0): These bits select the pin functions and perform input pullup MOS control.

The SCPT0 data register is also used in SCI transmission/reception (see section 15).

When using the SCPT0 function of the TxD0/SCPT0 pin independently of SCI transmission/reception, clear the TE bit in SCSCR to 0.

When using the SCPT0 function of the RxD0/SCPT0 pin independently of SCI transmission/reception, clear the RE bit and TE bit in SCSCR to 0.

TxD0

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	HiZ	(Initial value)
	1	SCPT0 output	
1	0	HiZ	
	1	HiZ	

(n = 0)

Note: Clear the TE bit in SCSCR to 0.

RxD0

Bit (2n+1): SCPnMD1	Bit 2n: SCPnMD0	Pin Function*	
0	0	SCPT0 input (pullup MOS: off)	(Initial value)
	1	HiZ	
1	0	SCPT0 input (pullup MOS: on)	
	1	SCPT0 input (pullup MOS: off)	

(n = 0)

Note: Clear the RE bit and TE bit in SCSCR to 0.

Section 20 I/O Port

20.1 Overview

The SH7707 has twelve 8-bit ports (ports A to L and SC). All port pins are multiplexed with other pin functions (the pin function controller (PFC) handles the selection of the pin functions and pullup MOS control). Each port has a data register which stores the data for the pins.

20.2 Port A

Port A is an 8-bit input/output port with the pin configuration shown in figure 20.1. Each pin has an input pullup MOS, which is controlled by the port A control register (PACR) in PFC.

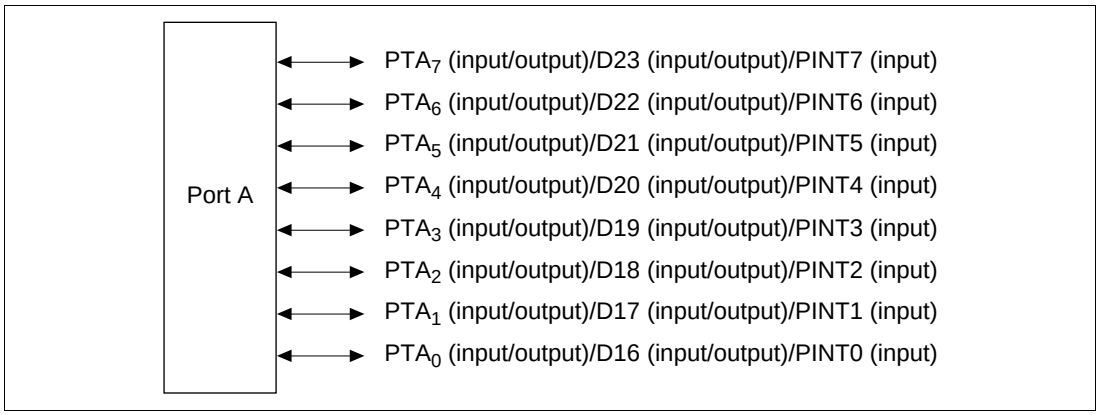


Figure 20.1 Port A

20.2.1 Register Description

Table 20.1 summarizes the register of port A.

Table 20.1 Port A Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'00	H'4000120	8

20.2.2 Port A Data Register (PADR)

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

The port A data register (PADR) is an 8-bit readable/writable register that stores data for pins PTA7–PTA0. Bits PA7DT–PA0DT correspond to pins PTA7–PTA0. 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 20.2 shows the function of PADR.

PADR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.2 Port A Data Register (PADR) Read/Write Operations

PAnMD1	PAnMD0	Pin State*	Read*	Write*
0	0	Hiz	PADR value	Can write to PADR, but it has no effect on pin state
	1	Output	PADR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PADR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PADR, but it has no effect on pin state

(n = 7–0)

Note: When the data bus width for all areas is 8 bits or 16 bits.

20.3 Port B

Port B is an 8-bit input/output port with the pin configuration shown in figure 20.2. Each pin has an input pullup MOS, which is controlled by the port B control register (PBCR) in the PFC.

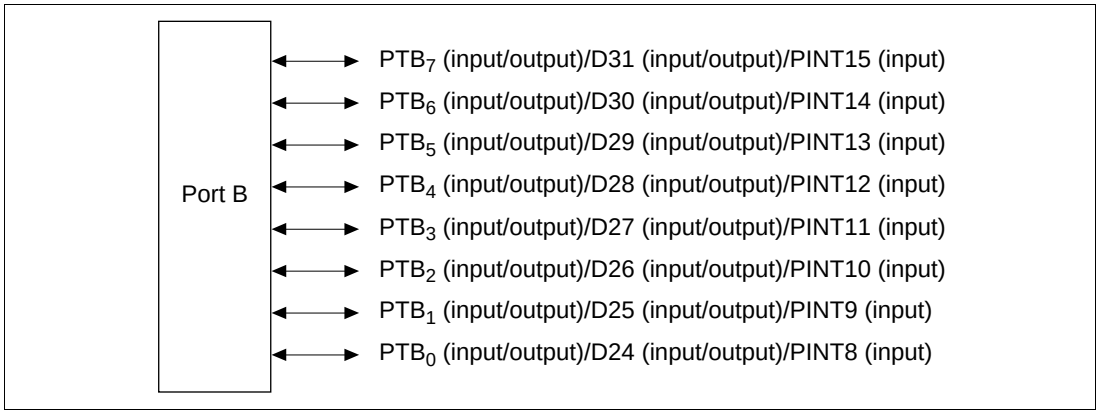


Figure 20.2 Port B

20.3.1 Register Description

Table 20.3 summarizes the register of port B.

Table 20.3 Port B Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'00	H'4000122	8

20.3.2 Port B Data Register (PBDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB07DT
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 port B data register (PBDR) is an 8-bit readable/writable register that stores data for pins PTB7–PTB0. Bits PB7DT–PB0DT correspond to bits PTB7–PTB0. 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 20.4 shows the function of PBDR.

PBDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.4 Port B Data Register (PBDR) Read/Write Operations

PAnMD1	PAnMD0	Pin State*	Read*	Write*
0	0	Hiz	PBDR value	Can write to PBDR, but it has no effect on pin state
	1	Output	PBDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PBDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PBDR, but it has no effect on pin state

(n = 7–0)

Note: When the data bus width for all areas is 8 bits or 16 bits.

20.4 Port C

Port C is an 8-bit input/output port with the pin configuration shown in figure 20.3. Each pin has an input pullup MOS, which is controlled by the port C control register (PCCR) in the PFC.

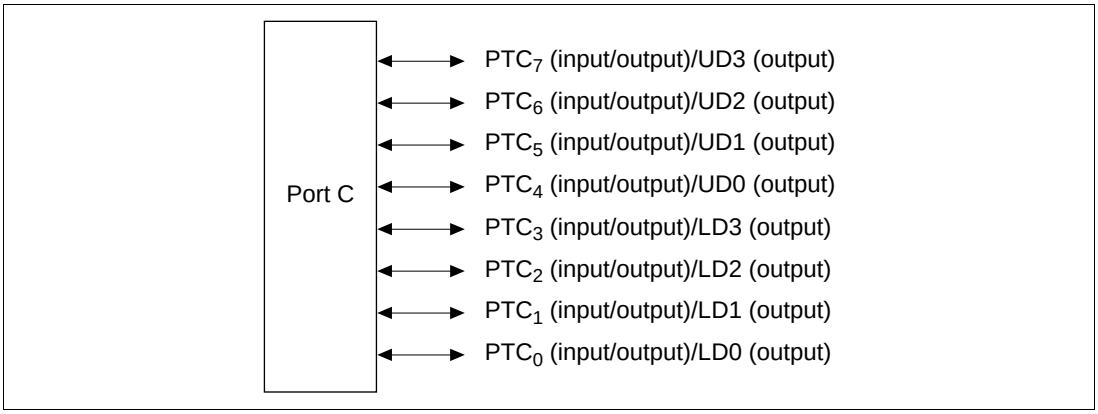


Figure 20.3 Port C

20.4.1 Register Description

Table 20.4.1 summarizes the register of port C.

Table 20.5 Port C Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'00	H'4000124	8

20.4.2 Port C Data Register (PCDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port C data register (PCDR) is an 8-bit readable/writable register that stores data for pins PTC7–PTC0. Bits PC7DT–PC0DT correspond to pins PTC7–PTC0. 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 20.6 shows the function of PCDR.

PCDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.6 Port C Data Register (PCDR) Read/Write Operation

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Can write to PCDR, but it has no effect on pin state
	1	Output	PCDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PCDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PCDR, but it has no effect on pin state

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.5 Port D

Port D is a 6-bit input/output and 2-bit input port with the pin configuration shown in figure 20.4. Each pin has an input pullup MOS, which is controlled by the port D control register (PDCR) in the PFC.

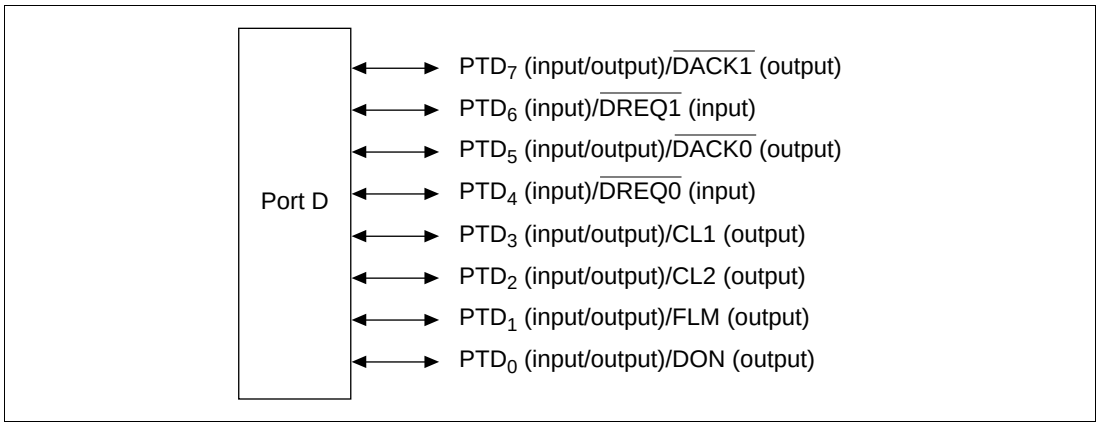


Figure 20.4 Port D

20.5.1 Register Description

Table 20.7 summarizes the register of port D.

Table 20.7 Port D Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register	PDDR	R/W or R	H'00	H'4000126	8

20.5.2 Port D Data Register (PDDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W

The port D data register (PDDR) is a 6-bit readable/writable and 2-bit read-only register that stores data for pins PTD7–PTD0. Bits PD7DT–PD0DT correspond to pins PTD7–PTD0. 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 20.8 shows the function of PDDR.

PDDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bits 6 and 4 are read other than in general-purpose input mode.

Table 20.8 Port D Data Register (PDDR) Read/Write Operations

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Can write to PDDR, but it has no effect on pin state
	1	Output	PDDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PDDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PDDR, but it has no effect on pin state

(n = 7, 5, 3, 2, 1, 0)

Note: “Other function” indicates Function 2 in table 19.1.

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no effect on pin state)
	1	Reserved	Low level	Ignored (no effect on pin state)
1	0	Input (pullup MOS on)	Pin state	Ignored (no effect on pin state)
	1	Input (pullup MOS off)	Pin state	Ignored (no effect on pin state)

(n = 6, 4)

Note: “Other function” indicates Function 2 in table 19.1.

20.6 Port E

Port E is an 8-bit input/output port with the pin configuration shown in figure 20.5. Each pin has an input pullup MOS, which is controlled by port E control register (PECR) in the PFC.

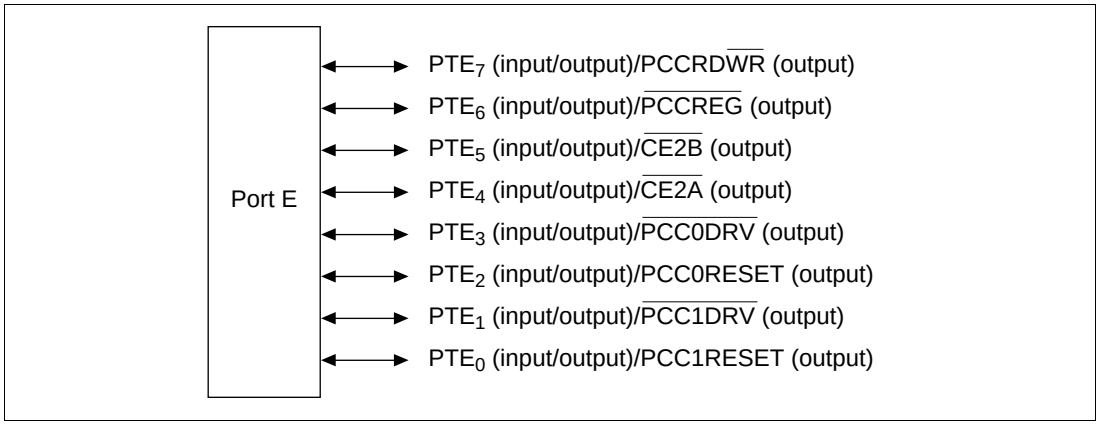


Figure 20.5 Port E

20.6.1 Register Description

Table 20.9 summarizes the register of port E.

Table 20.9 Port E Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'00	H'4000128	8

20.6.2 Port E Data Register (PEDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
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 port E data register (PEDR) is an 8-bit readable/writable register that stores data for pins PTE7–PTE0. Bits PE7DT–PE0DT correspond to pins PTE7–PTE0. 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 20.10 shows the function of PEDR.

PEDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.10 Port E Data Register (PEDR) Read/Write Operations

PEnMD1	PEnMD0	Pin State	Read	Write
0	0	Other function	PEDR value	Can write to PEDR, but it has no effect on pin state
	1	Output	PEDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PEDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PEDR, but it has no effect on pin state

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.7 Port F

Port F is an 8-bit input port with the pin configuration shown in figure 20.6. Each pin has an input pullup MOS, which is controlled by the port F control register (PFCR) in the PFC.

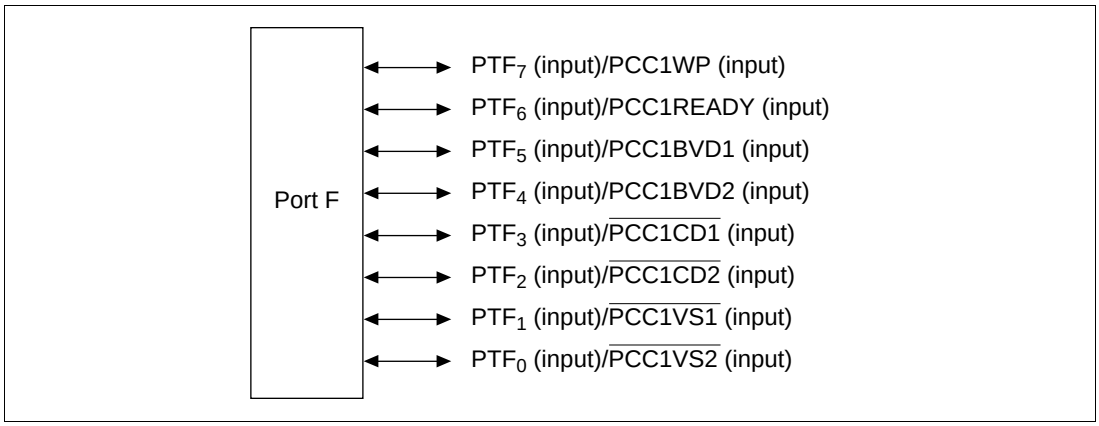


Figure 20.6 Port F

20.7.1 Register Description

Table 20.11 summarizes the register of port F.

Table 20.11 Port F Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R	H'00	H'400012A	8

20.7.2 Port F Data Register (PFDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PF7DT	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The port F data register (PFDR) is an 8-bit read-only register that stores data for pins PTF7–PTF0. Bits PF7DT–PF0DT correspond to pins PTF7–PTF0. When the function is general input port, if the port is read the corresponding pin level is read. Table 20.12 shows the function of PFDR.

Table 20.12 Port F Data Register (PFDR) Read/Write Operations

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no effect on pin state)
	1	Reserved	H'00	Ignored (no effect on pin state)
1	0	Input (pullup MOS on)	Pin state	Ignored (no effect on pin state)
	1	Input (pullup MOS off)	Pin state	Ignored (no effect on pin state)

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.8 Port G

Port G is an 8-bit input port with the pin configuration shown in figure 20.7. Each pin has an input pullup MOS, which is controlled by the port G control register (PGCR) in the PFC.

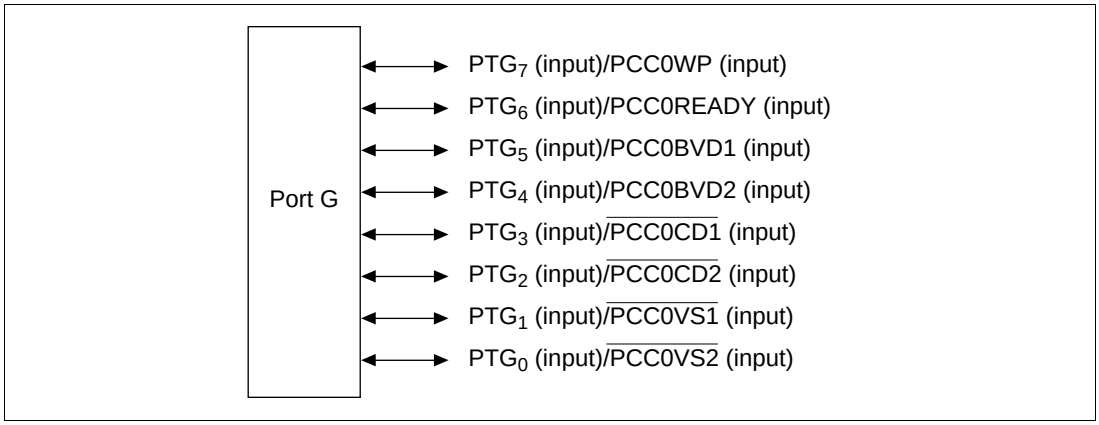


Figure 20.7 Port G

20.8.1 Register Description

Table 20.13 summarizes the register of port G.

Table 20.13 Port G Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register	PGDR	R	H'00	H'400012C	8

20.8.2 Port G Data Register (PGDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The port G data register (PGDR) is an 8-bit read-only register that stores data for pins PTG7–PTG0. Bits PG7DT–PG0DT correspond to pins PTG7–PTG0. When the function is general input port, if the port is read the corresponding pin level is read. Table 20.14 shows the function of PGDR.

Table 20.14 Port G Data Register (PGDR) Read/Write Operations

PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no effect on pin state)
	1	Reserved	H'00	Ignored (no effect on pin state)
1	0	Input (pullup MOS on)	Pin state	Ignored (no effect on pin state)
	1	Input (pullup MOS off)	Pin state	Ignored (no effect on pin state)

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.9 Port H

Port H is a 1-bit input/output and 7-bit input port with the pin configuration shown in figure 20.9. Each pin has an input pullup MOS, which is controlled by the port H control register (PHCR) in the PFC.

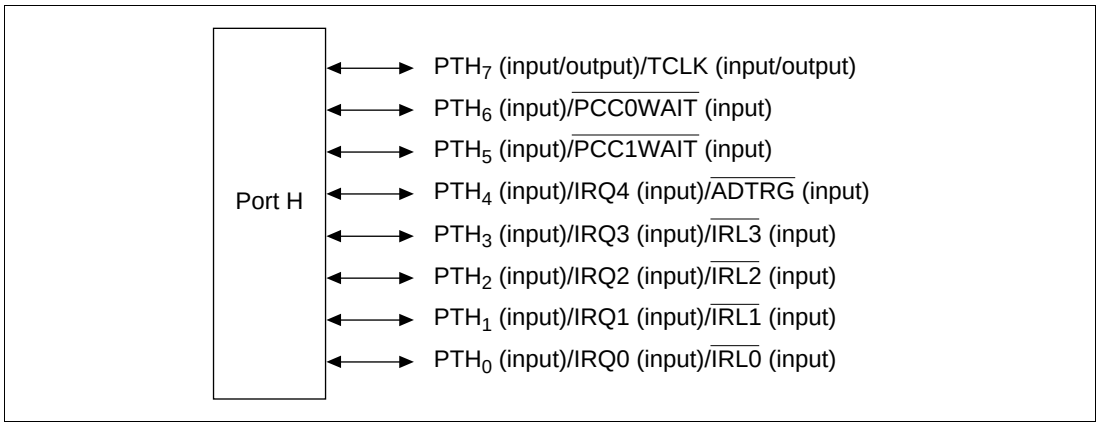


Figure 20.8 Port H

20.9.1 Register Description

Table 20.15 summarizes the register of port H.

Table 20.15 Port H Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R/W or R	H'00	H'400012E	8

20.9.2 Port H Data Register (PHDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

The port H data register (PHDR) is a 1-bit readable/writable and 7-bit read-only register that stores data for pins PTH7–PTH0. Bits PH7DT–PH0DT correspond to pins PTH7–PTH0. 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 20.16 shows the function of PHDR.

PHDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bits 6 to 0 are read other than in general-purpose input mode.

Table 20.16 Port H Data Register (PHDR) Read/Write Operations

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	PHDR value	Can write to PHDR, but it has no effect on pin state
	1	Output	PHDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PHDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PHDR, but it has no effect on pin state

(n = 7)

Note: “Other function” indicates Function 2 in table 19.1.

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no effect on pin state)
	1	Reserved	Low level	Ignored (no effect on pin state)
1	0	Input (pullup MOS on)	Pin state	Ignored (no effect on pin state)
	1	Input (pullup MOS off)	Pin state	Ignored (no effect on pin state)

(n = 6–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.10 Port J

Port J is an 8-bit input/output port with the pin configuration shown in figure 20.9. Each pin has an input pullup MOS, which is controlled by the port J control register (PJCR) in the PFC.

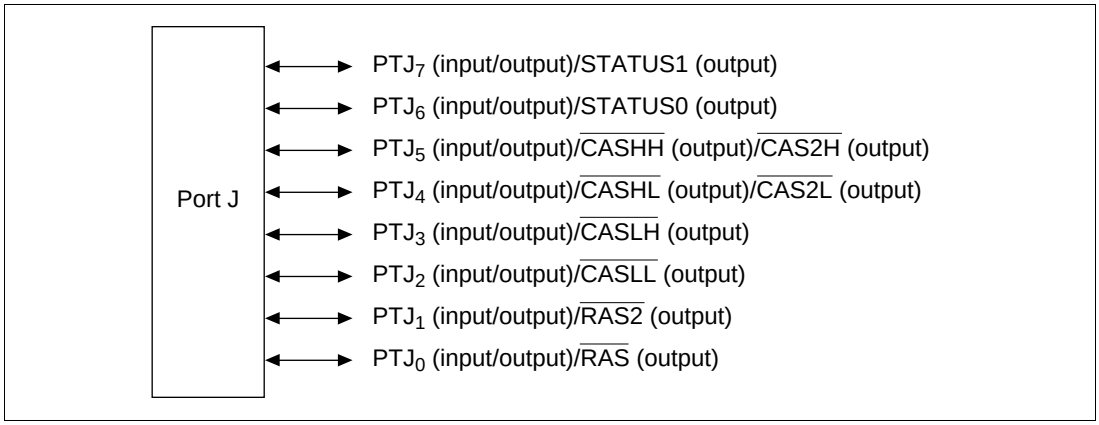


Figure 20.9 Port J

20.10.1 Register Description

Table 20.17 summarizes the register of port J.

Table 20.17 Port J Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port J data register	PJDR	R/W	H'00	H'4000130	8

20.10.2 Port J Data Register (PJDR)

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

The port J data register (PJDR) is an 8-bit readable/writable register that stores data for pins PTJ7–PTJ0. Bits PJ7DT–PJ0DT correspond to pins PTJ7–PTJ0. 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 20.18 shows the function of PJDR.

PJDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.18 Port J Data Register (PJDR) Read/Write Operations

PJnMD1	PJnMD0	Pin State	Read	Write
0	0	Other function	PJDR value	Can write to PJDR, but it has no effect on pin state
	1	Output	PJDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PJDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PJDR, but it has no effect on pin state

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.11 Port K

Port K is an 8-bit input/output port with the pin configuration shown in figure 20.10. Each pin has an input pullup MOS, which is controlled by the port K control register (PKCR) in the PFC.

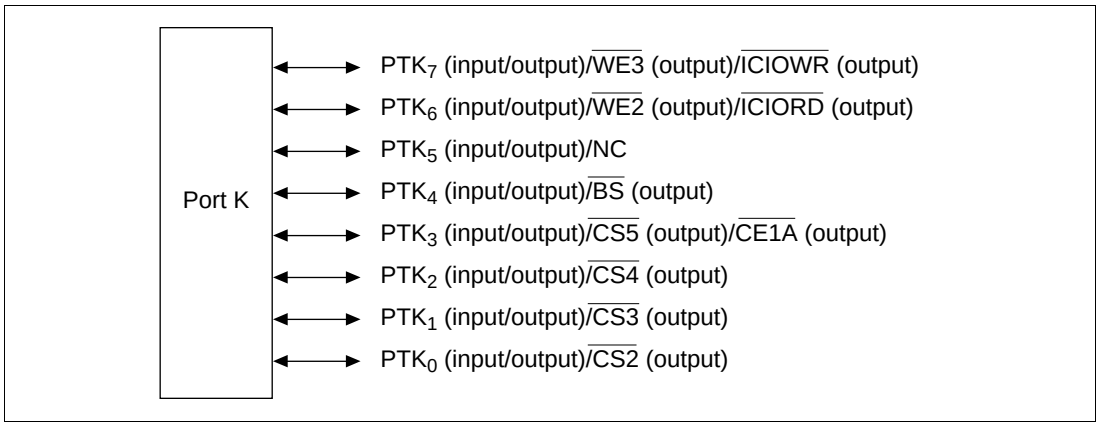


Figure 20.10 Port K

20.11.1 Register Description

Table 20.19 summarizes the register of port K.

Table 20.19 Port K Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port K data register	PKDR	R/W	H'00	H'4000132	8

20.11.2 Port K Data Register (PKDR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port K data register (PKDR) is an 8-bit readable/writable register that stores data for pins PTK7–PTK0. Bits PK7DT–PK0DT correspond to pins PTK7–PTK0. 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 20.20 shows the function of PKDR.

PKDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Table 20.20 Port K Data Register (PKDR) Read/Write Operations

PKnMD1	PKnMD0	Pin State	Read	Write
0	0	Other function	PKDR value	Can write to PKDR, but it has no effect on pin state
	1	Output	PKDR value	Value written is output by pin
1	0	Input (pullup MOS on)	Pin state	Can write to PKDR, but it has no effect on pin state
	1	Input (pullup MOS off)	Pin state	Can write to PKDR, but it has no effect on pin state

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.12 Port L

Port L is an 8-bit input port with the pin configuration shown in figure 20.11.

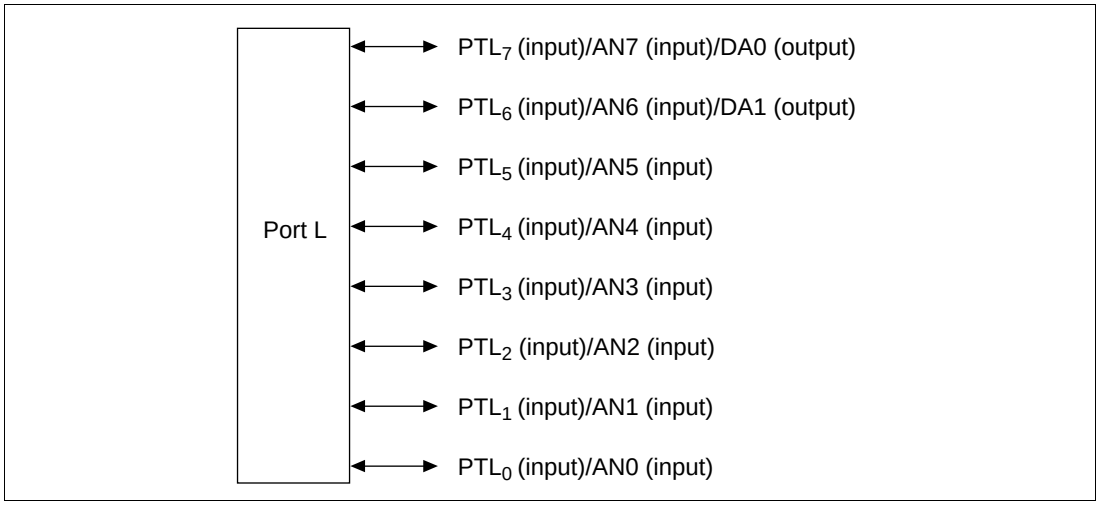


Figure 20.11 Port L

20.12.1 Register Description

Table 20.21 summarizes the register of port L.

Table 20.21 Port L Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port L data register	PLDR	R	H'00	H'4000134	8

20.12.2 Port L Data Register (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

The port L data register (PLDR) is an 8-bit read-only register that stores data for pins PTL7–PTL0. Bits PL7DT–PL0DT correspond to pins PTL7–PTL0. When the function is general input port, if the port is read the corresponding pin level is read. Table 20.22 shows the function of PLDR.

Table 20.22 Port L Data Register (PLDR) Read/Write Operations

PLnMD1	PLnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no effect on pin state)
	1	Reserved	H'00	Ignored (no effect on pin state)
1	0	Input	Pin state	Ignored (no effect on pin state)
	1	Input	Pin state	Ignored (no effect on pin state)

(n = 7–0)

Note: “Other function” indicates Function 2 in table 19.1.

20.13 SC Port

The SC port is a 4-bit input/output, 3-bit output, and 4-bit input port with the pin configuration shown in figure 20.12. Each input pin has an input pullup MOS, which is controlled by the SC port control register (SCPCR) in the PFC.

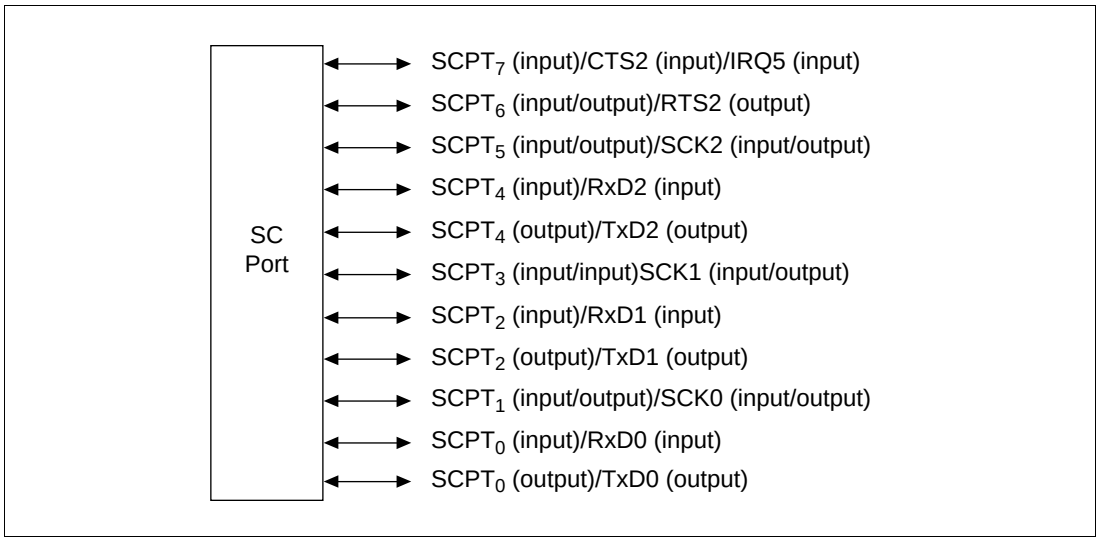


Figure 20.12 SC Port

20.13.1 Register Description

Table 20.23 summarizes the register of the SC port.

Table 20.23 SC Port Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
SC port data register	SCPDR	R/W or R	H'00	H'4000136	8

20.13.2 SC Port Data Register (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	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SC port data register (SCPDR) is a 7-bit readable/writable and 1-bit read-only register that stores data for pins SCPT7–SCPT0. Bits SCP7DT–SCP0DT correspond to pins SCPT7–SCPT0. 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 20.24 shows the function of SCPDR.

SCPDR is initialized to H'00 by a power-on reset. It retains its previous value in standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bit 7 is read other than in general-purpose input mode.

Table 20.24 SC Port Data Register (SCPDR) Read/Write Operations

SCPDT7

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	CTS2/IRQ5 input	Low level	Ignored (no effect on pin status)
	1	Reserved	Low level	Ignored (no effect on pin status)
1	0	SCPT7/IRQ5 input (pullup MOS: on)	Pin status	Ignored (no effect on pin status)
	1	SCPT7/IRQ5 input (pullup MOS: off)	Pin status	Ignored (no effect on pin status)

SCPDT6

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	RTS2 input	SCPDR value	Can write to SCPDR, but it has no effect on pin status.
	1	SCPT6 output	SCPDR value	Value written is output by pin.
1	0	SCPT6 input (pullup MOS: on)	Pin status	Can write to SCPDR, but it has no effect on pin status.
	1	SCPT6 input (pullup MOS: off)	Pin status	Can write to SCPDR, but it has no effect on pin status.

The function of pins SCPT4, SCPT2, and SCPT0 depends on the SCPCR and SCSCR settings. See section 19.3.12.

SCPDTn

(n = 5–0)

Pin State	Read	Write
SCPT(n) output	SCPDR value	Value written is output by pin.
SCPT(n) input (pullup MOS: on)	Pin status	Can write to SCPDR, but it has no effect on pin status.
SCPT(n) input (pullup MOS: off)	Pin status	Can write to SCPDR, but it has no effect on pin status.

In conditions other than those above, if the RE bit in SCSCR is set to 1, RxD[2:0] become inputs, taking precedence over the SCPCR setting, and the status of pins RxD[2:0] can be read.

Section 21 LCD Controller

21.1 Overview

The LCD controller (LCDC: liquid crystal display controller) reads display data from the system memory and displays it on an LCD.

The LCDC incorporates a dedicated 2-channel DMA controller (direct memory access controller). It can display a maximum of 16 gradations/colors by fetching dot data from system memory by DMA transfer and controlling the output data frame by frame by means of FRC (frame rate control).

LCD displays supported by the LCDC, with a maximum resolution of 640 × 480 dots, are STN type single-screen monochrome and dual-screen monochrome, birefringent reflective color, TFT/TFD color, and STN color.

21.1.1 Features

The LCDC has the following features:

- Built-in 2-channel DMA controller, enabling display of STN monochrome single-screen, STN monochrome dual-screens, birefringent reflective color STN color, or TFT/TFD color
- Maximum display capability of 16 gradations for STN monochrome, 16 colors for birefringent reflective color, 16/4096 colors for STN color, and 16/64 colors for TFT/TFD color
- Maximum resolution of 640 × 480 dots
- Frame extraction control using the space-modulation FRC (frame rate control) method provides a low-flicker graduated display
- Ratio of CKIO clock frequency to dot clock frequency can be selected in five steps, providing support for various clock frequencies up to 30 MHz
- Programmable display screen size, retrace line width, display clock width, output position, etc., facilitate optimization for the panel used and the operating conditions
- Indirect specification of display control registers, palette registers, and DMA control registers, using built-in address registers
- In 4-level gray scale display mode, any 4 of 16 gradation levels can be selected, for a graduated display matching the characteristics of the panel used

21.1.2 Display Modes

The LCDC has the display modes shown below.

Table 21.1 Display Modes

		LCD Data Output				
Mode	**Mode Name**	**Screen Config.**	**Data Transfer**	**Max. Colors**	**Bits/Dot*1**	**Max. Resolution**
A	STN monochrome	Single	4-bit	16 level gray scale	1, 2, or 4	640 × 480
B	STN monochrome	Single	8-bit	16 level gray scale	1, 2, or 4	640 × 480
C	STN monochrome	Dual	4 × 2-bit	16 level gray scale	1, 2, or 4	640 × 480
D	STN color	Single	8-bit *5	16/4096 colors*2	1, 2, or 4	640 × 480
E	Reflective color (birefringent)	Single	4-bit output	16 colors	1, 2, or 4	640 × 480
F	Reflective color (birefringent)	Single	8-bit output	16 colors	1, 2, or 4	640 × 480
G	TFT/TFD color	Single	6-bit output*4	16/64 colors*3	1, 2, or 4	640 × 480

Notes:
1. The built-in palette registers are used in all the above modes, and therefore the palette registers must also be initialized. Set palette registers LCDPR0 and LCDPR1 in 1-bit/dot mode, LCDPR0 to LCDPR3 in 2-bits/dot mode, and LCDPR0 to LCDPR15 in 4-bits/dot mode, according to the display colors.
2. In mode D, 16 of 4096 colors can be displayed simultaneously.
3. In mode G, 16 of 64 colors can be displayed simultaneously.
4. In mode G, FRC (frame rate control)) is not performed, and the 2-bit display data for each of red, green, and blue, set in the palette registers is output.
5. In mode D, R/G/B 3-bit × 8-dot data is output in three lots of 8 bits.

21.1.3 Block Diagram

Figure 21.1 shows a block diagram of the LCDC.

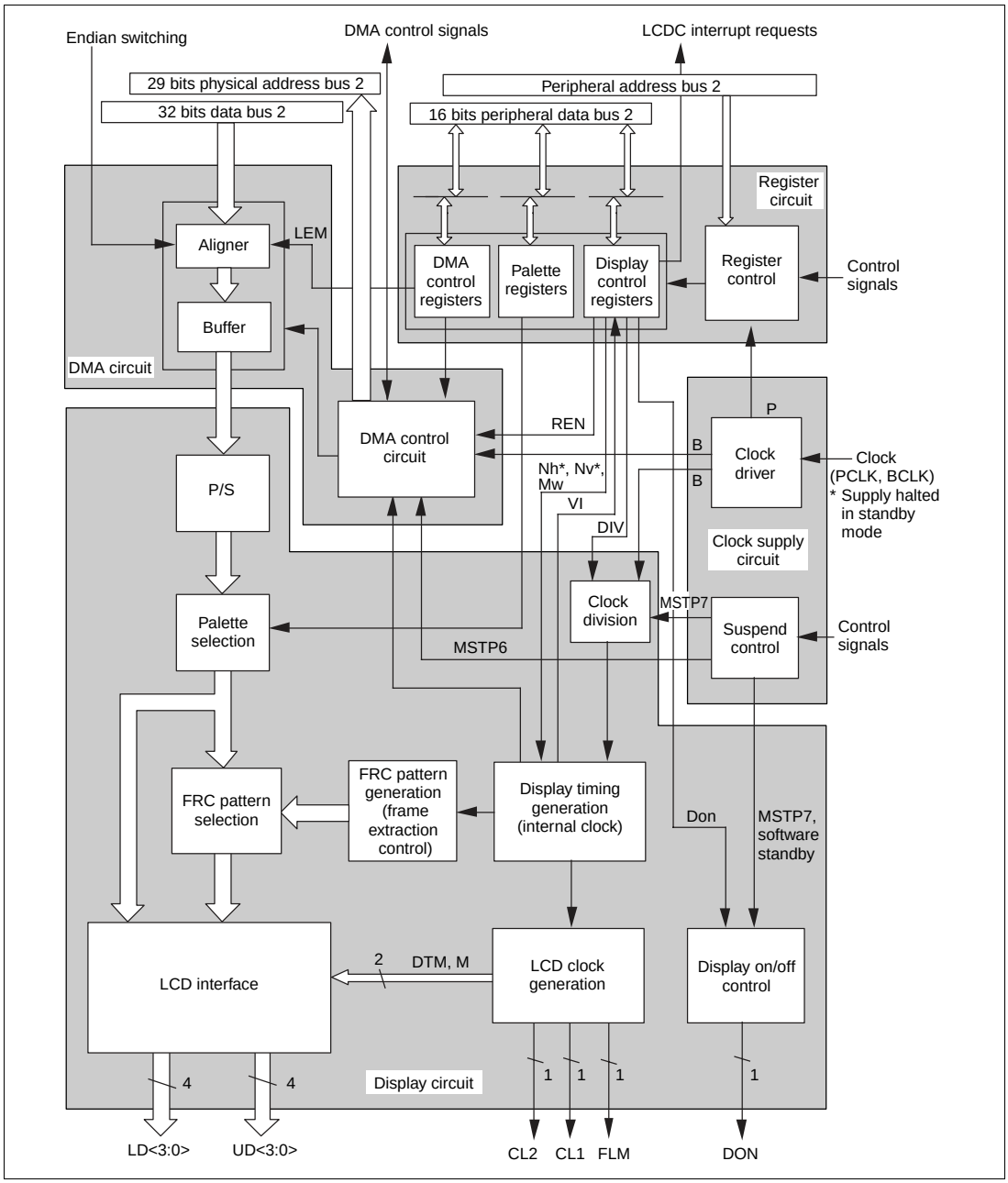


Figure 21.1 LCDC Block Diagram

21.1.4 Pin Configuration

The LCDC pins are listed in table 21.2, with a summary of their functions.

Table 21.2 LCDC Pins

Description	Abbreviation	I/O	Function
Display data latch clock	CL1	Output	Outputs clock used by X driver to latch one line of display data
Display data shift clock	CL2	Output	Outputs display data shift clock for X driver
First line marker	FLM	Output	Outputs first line mark signal for Y driver
LCD display control	DON	Output	Outputs signal for LCD driver display on/off control
Upper data [3:0]	UD[3:0]	Output	In dual-screen mode, output upper panel display data UD[3:0] (mode C)
			In 4-bit output mode, output display data D[3:0] (modes A, E)
			In 6-bit output mode, output display data upper 4 bits D[5:2] (mode G)
			In 8-bit output mode, output display data upper 4 bits D[7:4] (modes B, D, F)
Lower data [3:0]	LD[3:0]	Output	In dual-screen mode, output lower panel display data LD[3:0] (mode C)
			In 4-bit output mode, LD[3:2] output 0, LD[1] outputs DTM, LD[0] outputs M (modes A, E)
			In 6-bit output mode, LD[3:2] output display data lower 2 bits D[1:0], LD[1] outputs DTM, LD[0] outputs M (mode G)
			In 8-bit output mode, output display data lower 4 bits D[7:4] (modes B, D, F)

Table 21.3 shows the switched-output signals from the LD1 and LD0 pins in modes A, E and G.

Table 21.3 Switched-Output Signals

Description	Abbreviation	I/O	Function
Display timing	DTM	Output	High output during display data output, low output during horizontal/vertical retrace line interval Independent of CC[1:0]
LCD driving signal alternation	M	Output	Outputs signal for AC conversion of LCD drive signals

21.1.5 Register Configuration

The LCDC has one address register (LCDAR) and data registers. There are three kinds of data registers: display control registers (LCDDR), palette registers (LCDPR), and DMA control registers (LCDDMR). Addresses from the CPU are assigned only to these four. For individual registers within the data registers, indirect addressing by means of the address register is used. For example, to write to palette register LCDPR3, 3 is first written to the address register (LCDAR), then a write is performed to the palette register (LCDPR).

Table 21.4 summarizes the LCDC's registers. Tables 21.5 to 21.7 show the detailed configuration of the display control registers, palette registers, and DMA control registers.

Table 21.4 LCDC Registers

Name	Abbreviation	R/W	Initial Value*1	Address	Access Size
Address register	LCDAR	R/W	H'000	H'40000C0	16
Display control registers	LCDDR	R/W*2	H'000	H'40000C2	16
Palette registers	LCDPR	R/W	H'000	H'40000C6	16
DMA control registers	LCDDMR	R/W	H'000	H'40000CE	16

Notes: 1. Initial value after a power-on reset. After a manual reset, the register value is retained (except for certain bits).
2. Bit 1 of LCDDR13 cannot be read. If read, a value of 0 will always be returned. Bit 1 can be cleared to 0 by a write access to LCDDR13.

Table 21.5 Display Control Registers

Address Register					Register No.	Register Name	Program Unit	Data Bits																
								15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LCDDR0	Display register 1																	*1 Don
0	0	0	0	1	LCDDR1	Display register 2							DIV2	DIV1	*2 DIV0	CC1	*3 CC0	*4 REN	GR1	*5 GR0	LM2	LM1	*6 LM0
0	0	0	1	0	LCDDR2	Horizontal character count	4 dots	(Horizontal display character count)								(Horizontal total character count)							
								Nhd7	Nhd6	Nhd5	Nhd4	Nhd3	Nhd2	Nhd1	Nhd0	Nht7	Nht6	Nht5	Nht4	Nht3	Nht2	Nht1	Nht0
0	0	0	1	1	LCDDR3	Horizontal synchroniza-tion	4 dots					(Horizontal synchronization width)				(Horizontal synchronization position)							
												Nhsw3	Nhsw2	Nhsw1	Nhsw0	Nhsp7	Nhsp6	Nhsp5	Nhsp4	Nhsp3	Nhsp2	Nhsp1	Nhsp0
0	0	1	0	0	LCDDR4	Vertical total line count	Line									(Vertical total line count)							
														Nvt9	Nvt8	Nvt7	Nvt6	Nvt5	Nvt4	Nvt3	Nvt2	Nvt1	Nvt0
0	0	1	0	1	LCDDR5	Vertical display line count	Line									(Vertical display line count)							
														Nvd9	Nvd8	Nvd7	Nvd6	Nvd5	Nvd4	Nvd3	Nvd2	Nvd1	Nvd0
0	0	1	1	0	LCDDR6	Vertical synchroniza-tion	Line	(AC line count)								(Vertical synchronization position)							
								Mw4	Mw3	Mw2	Mw1	Mw0		Nvsp9	Nvsp8	Nvsp7	Nvsp6	Nvsp5	Nvsp4	Nvsp3	Nvsp2	Nvsp1	Nvsp0
0	0	1	1	1	LCDDR7	Reserved																	
0	1	0	0	0	LCDDR8	Reserved																	
0	1	0	0	1	LCDDR9	Reserved																	
0	1	0	1	0	LCDDR10	Reserved																	
0	1	0	1	1	LCDDR11	Reserved																	
0	1	1	0	0	LCDDR12	Reserved																	
0	1	1	0	1	LCDDR13	LCDC interrupt																*7 VI	*8 VIE
0	1	1	1	0	LCDDR14	Reserved																	
0	1	1	1	1	LCDDR15	Reserved																	

Notes:
1. Don is the display on/off switch bit. It controls the DON output pin.
2. Bits DIV[2:0] specify the clock division ratio. Set the division ratio of the display clock with respect to CKIO.
3. Bits CC[1:0] are clock control bits. They specify CL1 and CL2 clock mask control during the retrace interval.
4. REN (DMA Request Enable) is the DMA transfer enable bit. Setting this bit to 1 starts output of a display data transfer request.
5. Bits GR[1:0] specify the number of gradations/colors.
6. Bits LM[2:0] are display mode specification bits.
7. VI is the LCDC interrupt request flag bit. It always returns 0 when read, and can be cleared to 0 by a write access.
8. Shading indicates reserved bits or registers. Data written to these areas is invalid. Reading a reserved bit in LCDDR0–LCDDR6 or LCDDR13 will return 0, and reading a reserved register will return an undefined value.

Table 21.6 Palette Registers

Address Register					Register No.	Register Name	Data Bits													
							15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LCDPR0	Palette register 0				*1		(Palette register 0)										*2 *3
0	0	0	0	1	LCDPR1	Palette register 1						(Palette register 1)										
0	0	0	1	0	LCDPR2	Palette register 2						(Palette register 2)										
0	0	0	1	1	LCDPR3	Palette register 3						(Palette register 3)										
0	0	1	0	0	LCDPR4	Palette register 4						(Palette register 4)										
0	0	1	0	1	LCDPR5	Palette register 5						(Palette register 5)										
0	0	1	1	0	LCDPR6	Palette register 6						(Palette register 6)										
0	0	1	1	1	LCDPR7	Palette register 7						(Palette register 7)										
0	1	0	0	0	LCDPR8	Palette register 8						(Palette register 8)										
0	1	0	0	1	LCDPR9	Palette register 9						(Palette register 9)										
0	1	0	1	0	LCDPR10	Palette register 10						(Palette register 10)										
0	1	0	1	1	LCDPR11	Palette register 11						(Palette register 11)										
0	1	1	0	0	LCDPR12	Palette register 12						(Palette register 12)										
0	1	1	0	1	LCDPR13	Palette register 13						(Palette register 13)										
0	1	1	1	0	LCDPR14	Palette register 14						(Palette register 14)										
0	1	1	1	1	LCDPR15	Palette register 15						(Palette register 15)										

Notes: 1. The most significant 4 bits [15:12] are reserved. Writes are invalid, but 0 must be written. A read will return 0.

2. In monochrome mode and reflective color mode (modes A, B, C, E, and F), only the lower 4 bits [3:0] are used, and 0 must be written to the upper 8 bits [11:4]. In color 6-bit output mode (mode G), only the lower 6 bits [5:0] are used, and 0 must be written to the upper 6 bits [11:7]. In color STN mode (mode D), write blue data to bits [3:0], green data to bits [7:4], and red data to bits [11:8].

3. In 1-bit/dot mode, only LCDPR0 and LCDPR1 are used. In 2-bits/dot mode, only LCDPR0–LCDPR3 are used. In 4-bits/dot mode, registers LCDPR0–LCDPR15 are all used.

Table 21.7 DMA Control Registers

Address Register					Register No.	Register Name	Program Unit	Data Bits																
								15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LCDDMR0	Display start address (lower)	Memory address	UMA15	UMA14	UMA13	(Display start address (low)) UMA12	UMA11	UMA10	UMA9	UMA8	UMA7	UMA6	UMA5	UMA4	UMA3	UMA2	UMA1	UMA0
0	0	0	0	1	LCDDMR1	Display start address (upper)	Memory address	UMA31	*1 UMA30	UMA29	(Display start address (high)) UMA28	UMA27	UMA26	UMA25	UMA24	UMA23	UMA22	UMA21	UMA20	UMA19	UMA18	UMA17	UMA16
0	0	0	1	0	LCDDMR2	Lower panel display start address (lower)	Memory address	LMA15	LMA14	LMA13	(Lower panel display start address (low)) LMA12	LMA11	LMA10	LMA9	LMA8	LMA7	LMA6	LMA5	LMA4	LMA3	LMA2	LMA1	LMA0
0	0	0	1	1	LCDDMR3	Lower panel display start address (upper)	Memory address	LMA31	*2 LMA30	LMA29	(Lower panel display start address (high)) LMA28	LMA27	LMA26	LMA25	LMA24	LMA23	LMA22	LMA21	LMA20	LMA19	LMA18	LMA17	LMA16
0	0	1	0	0	LCDDMR4	Transfer word count	16 bytes	*3 LEM	*4	TM13	(Transfer word count) TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
0	0	1	0	1	LCDDMR5	Reserved																	
0	0	1	1	0	LCDDMR6	Reserved																	
0	0	1	1	1	LCDDMR7	Reserved																	
0	1	0	0	0	LCDDMR8	Reserved																	
0	1	0	0	1	LCDDMR9	Reserved																	
0	1	0	1	0	LCDDMR10	Reserved																	
0	1	0	1	1	LCDDMR11	Reserved																	
0	1	1	0	0	LCDDMR12	Reserved																	
0	1	1	0	1	LCDDMR13	Reserved																	
0	1	1	1	0	LCDDMR14	Reserved																	
0	1	1	1	1	LCDDMR15	Reserved																	

Notes: 1. Write 0 to the most significant 3 bits.
2. Write 0 to the most significant 3 bits.
3. LEM (Little-Endian Mode) functions only in little-endian mode. Basic mode is set when LEM = 0, and linear byte address mode when LEM = 1.
4. Invalid bit. Writes are invalid, but 0 must be written. A read will return 0.

21.2 Register Descriptions

21.2.1 Address Register (LCDAR)

The address register (LCDAR) is used to indirectly specify one of the LCDC's data registers. The data registers consist of eight display control registers, 16 palette registers, and five DMA control registers. When the CPU accesses an LCDC register, it can choose from only four alternatives: address register, display control register, palette register, or DMA control register. To read or write to a specific data register, the number of the register to be accessed is first written to the address register, then the corresponding data register is accessed. For example, to set the vertical total line count in display control register 4, 4 is first written to the address register. Once a value has been written to the address register, it remains there until a power-on reset is executed or a new value is written. Thus different data registers assigned to the same number can be accessed in succession.

The address register is a readable/writable register with a 5-bit configuration. Note that 0 must be written to the most significant bit (bit 4).

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

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

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

Bits 15 to 5—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bit 4—Reserved: This bit can be read or written, but 0 must always be written. Writing 1 has the same effect operationally as writing 0.

Bits 3 to 0—Data Bits: These bits can be read or written. Normal operation is not guaranteed if indirect access is performed to a nonexistent register number. Set a value of 0, 1, 2, 3, 4, 5, 6, or 13 to designate a display control register, a value from 0 to 15 to designate a palette register, and a value from 0 to 4 to designate a DMA control register.

21.2.2 Display Control Registers (LCDDR)

Display Register 1: This register contains the LCD on/off switch bit. The set value is output from the DON pin. If the LCD driver used has a display on/off pin such as $\overline{\text{DISPOFF}}$, display control can be performed using the SH7707's DON pin. When the Don bit is set to 1, display is turned on; if the Don bit is cleared to 0, display is turned off, but the output display data does not change.

When Module Stop 7 (MSTP7) is set to 1, the DON pin output is 0 regardless of the Don bit setting.

The Don bit is cleared to 0 by a power-on reset or a manual reset.

Bit:	15	14	13	12	11	10	9	8
LCDDR0	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
LCDDR0	—	—	—	—	—	—	—	Don
Initial value:	—	—	—	—	—	—	—	0
R/W:	—	—	—	—	—	—	—	R/W

Bits 15 to 1—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bit 0—Display On/Off Switch (Don)

Bit 0: Don	Description
1	LCD on
0	LCD off (Initial value)

Display Register 2: This register specifies various LCDC operating modes. It is divided into five fields: clock division ratio (DIV[2:0]), clock control (CC[1:0]), DMA transfer enable (REN), gradation/color (GR[1:0]), and display mode (LM[2:0]).

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

Bit:	15	14	13	12	11	10	9	8
LCDDR1	—	—	—	—	—	DIV2	DIV1	DIV0
Initial value:	—	—	—	—	—	0	0	0
R/W:	—	—	—	—	—	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
LCDDR1	CC1	CC0	REN	GR1	GR0	LM2	LM1	LM0
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 11—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 10 to 8—Clock Division Ratio (DIV2 to DIV0): These bits specify the display clock division ratio with respect to the CKIO clock. This is determined by the ratio of the dot clock to the CKIO clock. See section 21.3.4 for details.

	Bit 10: DIV2	Bit 9: DIV1	Bit 8: DIV0	Description
	1	1	1	Illegal settings
	1	1	0	
	1	0	1	
	1	0	0	Dot clock frequency = CKIO × 1/8
	0	1	1	Dot clock frequency = CKIO × 1/4
	0	1	0	Dot clock frequency = CKIO × 1/2
	0	0	1	Dot clock frequency = CKIO × 1
*	0	0	0	Dot clock frequency = CKIO × 2 (Initial value)

Note: In TFT/TFD color (mode G), STN color (mode D), and STN monochrome (modes A, B, C) 16-gradation modes, and reflective color (modes E, F) 16-color modes, do not set DIV[2:0] = 000. When DIV[2:0] = 000 is set in STN monochrome (modes A, B, C) 2 or 4-gradation modes, set an upper limit of CKIO = 20 MHz. When DIV[2:0] = 001 is set in TFT/TFD color mode (mode G), set an upper limit of CKIO = 20 MHz.

Bits 7 to 6—Clock Control (CC[1:0]): These bits specify whether CL1 and CL2 clock output is to be performed during the retrace line interval.

Bit 7: CC1	Bit 6: CC0	Description
1	1	Illegal settings
	0	CL1 is not output during vertical retrace line interval*
		CL2 is not output during retrace line interval
0	1	CL1 is output during vertical retrace line interval
		CL2 is output during retrace line interval
	0	CL1 is output during vertical retrace line interval
		CL2 is not output during retrace line interval (Initial value)

Note: CL1 is output in the first line after the start of the vertical retrace line interval.

Bit 5—DMA Transfer Enable (REN): DMA transfer start enable bit used to display the display data held in the frame memory on the LCD panel. DMA transfer of the display data is started when REN = 1. Transfer requests are not output from the LCDC when REN = 0.

Bit 5: REN	Description
1	DMA transfer enabled
0	DMA transfer disabled (Initial value)

Bits 4 and 3—Gradation/Color (GR[1:0]): These bits specify the number of monochrome gradations or number of colors in the various display modes.

Bit 4: GR1	Bit 3: GR0	Description
1	1	16 gradations (colors) (4 bits/dot)
	0	4 gradations (colors) (2 bits/dot)
0	1	Monochrome (2 colors) (1 bit/dot)
	0	Illegal setting (Initial value)

Bits 2 to 0—Display Mode (LM[2:0]): These bits specify the display mode.

Bit 2: LM2	Bit 1: LM1	Bit 0: LM0	Description	
1	1	1	Illegal settings	—
		0		
	0	1	STN color	Mode D
		0	TFT/TFD color	Mode G
0	1	1	Illegal setting	—
		0	Monochrome / dual screens / 4 × 2-bit output	Mode C
	0	1	Monochrome / single screen / 8-bit output	Modes B, F
		0	Monochrome / single screen / 4-bit output (Initial value)	Modes A, E

Horizontal Character Count: This register is set with the number of display characters in the horizontal screen direction and the total number of horizontal display characters. The upper 8 of the 16 bits correspond to the horizontal display character count (Nhd), and the lower 8 bits to the horizontal total character count (Nht). The difference between Nhd and Nht is the horizontal retrace line interval; this must be at least 8 dots. Ensure that (Nhd + 2) Nht.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

Horizontal display character count (Nhd)

Bit:	15	14	13	12	11	10	9	8
LCDDR2	Nhd7	Nhd6	Nhd5	Nhd4	Nhd3	Nhd2	Nhd1	Nhd0
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

Horizontal total character count (Nht)

Bit:	7	6	5	4	3	2	1	0
LCDDR2	Nht7	Nht6	Nht5	Nht4	Nht3	Nht2	Nht1	Nht0
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 8—Horizontal Display Character Count (Nhd): In this field, set one less than the number of horizontal display dots converted to 4-dot units.

Example: For a horizontal display screen size of 640 dots:

Nhd = (640 ÷ 4) – 1 = 159 (B'10011111)

Bits 7 to 0—Horizontal Total Character Count (Nht): In this field, set one less than the total number of horizontal dots converted to 4-dot units.

Example: For a horizontal display screen size of 640 dots, and a horizontal retrace line interval of 8 dots:

Nht = ((640 + 8)÷ 4) – 1 = 161 (B'10100001)

Horizontal Synchronization: This register specifies the CL1 clock width and output position. The four data bits 11–8 correspond to the horizontal synchronization width (Nhsw), and the eight data bits 7–0, to the horizontal synchronization position (Nhsp).

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

					Horizontal synchronization width (Nhsw)			
Bit:	15	14	13	12	11	10	9	8
LCDDR3	—	—	—	—	Nhsw3	Nhsw2	Nhsw1	Nhsw0
Initial value:	—	—	—	—	0	0	0	0
R/W:	—	—	—	—	R/W	R/W	R/W	R/W

	Horizontal synchronization position (Nhsp)							
Bit:	7	6	5	4	3	2	1	0
LCDDR3	Nhsp7	Nhsp6	Nhsp5	Nhsp4	Nhsp3	Nhsp2	Nhsp1	Nhsp0
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 12—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 11 to 8—Horizontal Synchronization Width (Nhsw): In this field, set one less than the CL1 clock width converted to 4-dot units. An odd number must be set for display modes B, D, F, and G (monochrome/single screen/8-bit output, STN color, reflective color/single screen/8-bit output, TFT/TFD color). See figure 21.2 for details.

Bits 7 to 0—Horizontal Synchronization Position (Nhsp): In this field, set one less than the CL1 clock output position converted to 4-dot units. For display modes B, D, F, and G (monochrome/ single screen/8-bit output, STN color, reflective color/single screen/8-bit output, TFT/TFD color), the setting must be made so as to give an even number. See figure 21.2 for details.

Example: For display mode B, with a horizontal display screen size of 640 dots, a horizontal retrace line interval of 16 dots, CL1 output start position at the 641st dot, and a width of 8 dots:

Nhd = (640 ÷ 4) – 1 = 159 (B'10011111)
Nht = ((640 + 16)÷ 4) – 1 = 163 (B'10100011)
Nhsp = 159 + 1 = 160 (B'10100000)
Nhsw = (8 ÷ 4) – 1 = 1 (B'0001)

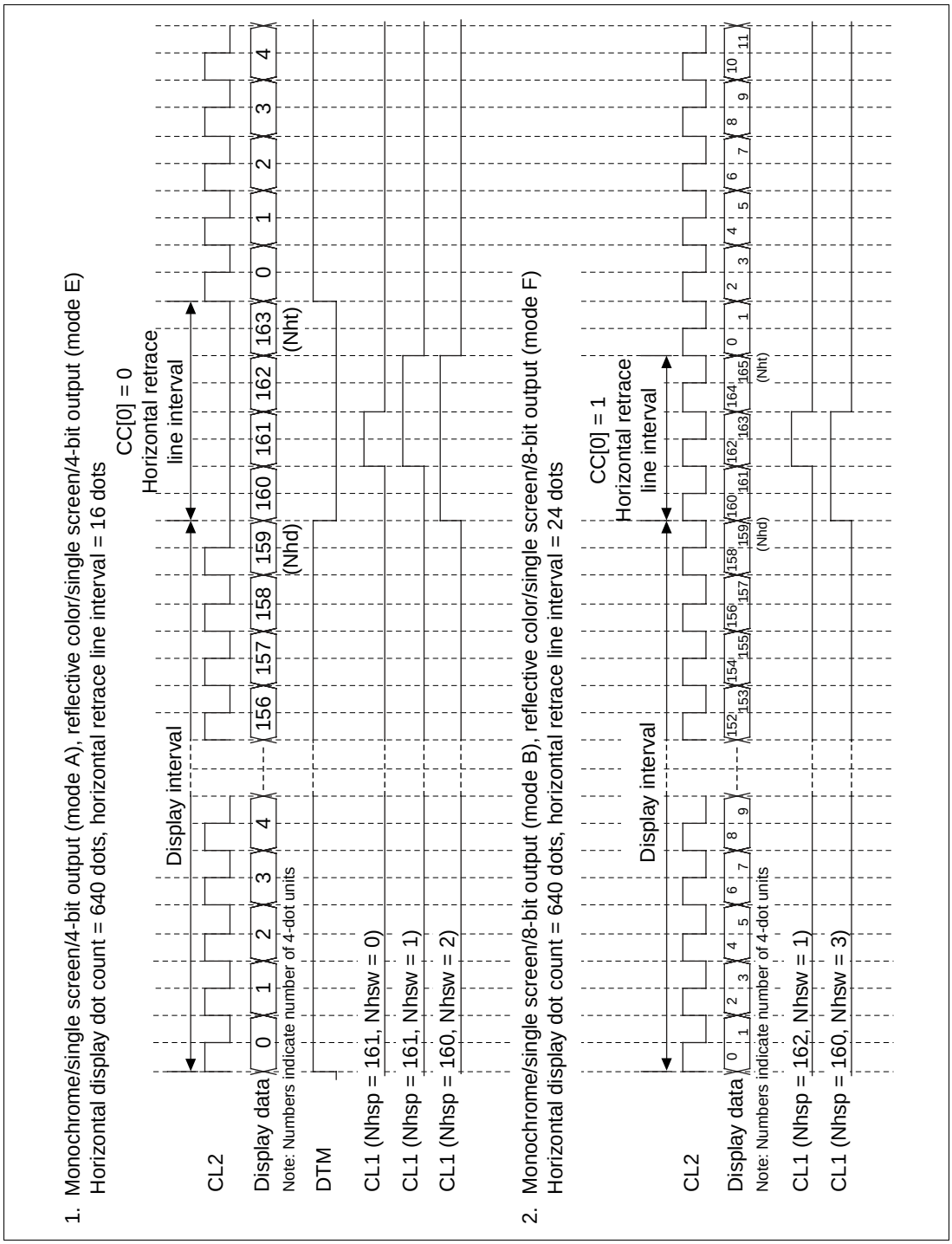


Figure 21.2 Examples of Nhsp and Nhsw Settings

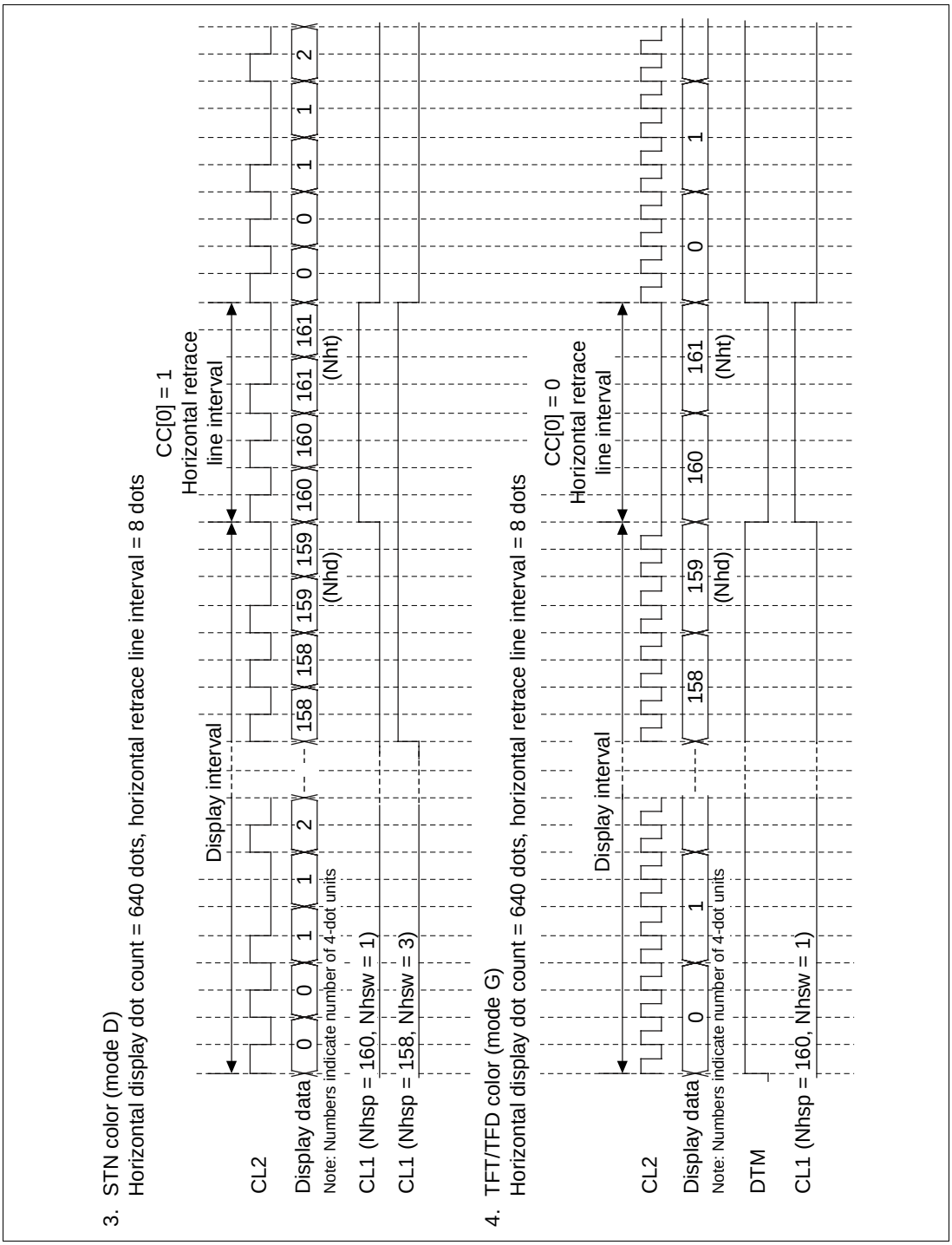


Figure 21.2 Examples of Nhsp and Nhsw Settings (cont)

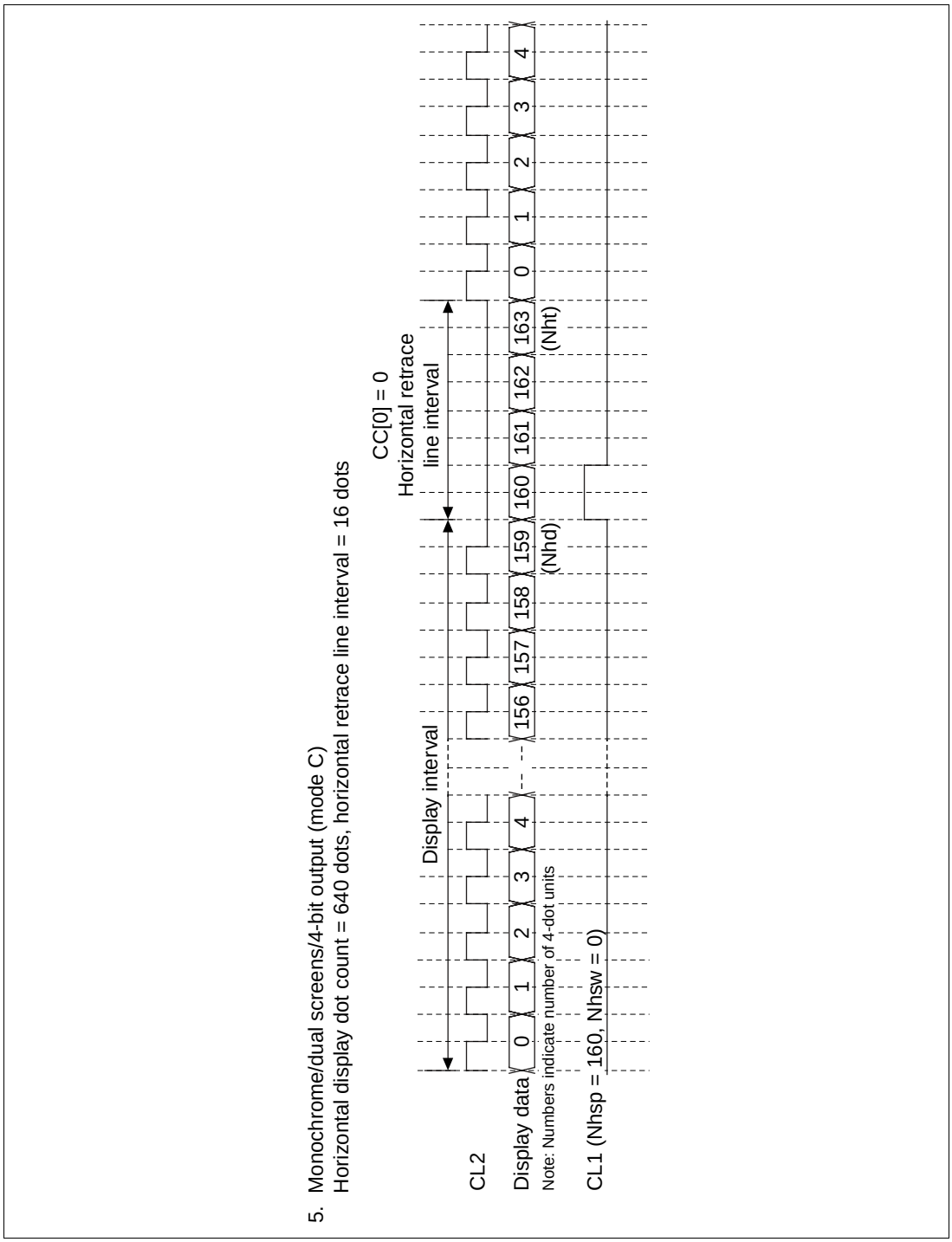


Figure 21.2 Examples of Nhsp and Nhsw Settings (cont)

Vertical Total Line Count: The vertical total line count register (Nvt) is set with the total number of lines in the vertical screen direction, including the vertical retrace line interval. The difference between Nvd and Nvt is the vertical retrace line interval. Ensure that Nvd Nvt.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

							Vertical total line count (Nvt)									
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR4	—	—	—	—	—	—	Nvt9	Nvt8	Nvt7	Nvt6	Nvt5	Nvt4	Nvt3	Nvt2	Nvt1	Nvt0
Initial value:	—	—	—	—	—	—	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

Bits 15 to 10—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 9 to 0—Vertical Total Line Count (Nvt): In this field, set one less than the total number of lines in the vertical direction.

Example: For a vertical display screen size of 200 lines, and a vertical retrace line interval of 4 lines:

Nvt = (200 + 4) – 1 = 203 (B'0011001011)
Nvd = 200 – 1 = 199 (B'0011000111)

Vertical Display Line Count: This register is set with the number of display lines in the vertical screen direction. This vertical display line count (Nvd) forms a pair with the vertical total line count (Nvt), and must be set so that Nvd Nvt. The difference between Nvd and Nvt is the vertical retrace line interval, and this can be used for fine adjustment of the frame frequency.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

							Vertical display line count (Nvd)									
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR5	—	—	—	—	—	—	Nvd9	Nvd8	Nvd7	Nvd6	Nvd5	Nvd4	Nvd3	Nvd2	Nvd1	Nvd0
Initial value:	—	—	—	—	—	—	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

Bits 15 to 10—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 9 to 0—Vertical Display Line Count (Nvd): In this field, set one less than the number of display lines in the vertical direction. Make the setting so that Nvd Nvt.

Examples: For a vertical display screen size of 480 lines with no vertical retrace lines:

Nvt = (480 + 0) – 1 = 479 (B'0111011111)
Nvd = 480 – 1 = 479 (B'0111011111)

For a vertical display screen size of 480 lines, and 4 vertical retrace lines:

Nvt = (480 + 4) – 1 = 483 (B'111100011)
Nvd = 480 – 1 = 479 (B'0111011111)

Vertical Synchronization: This register is composed of the AC line count register, which specifies the AC frequency of LCD driving signal alternation (M), and the vertical synchronization position register, which specifies the FLM clock output position. The five data bits 15–11 correspond to the AC line count (Mw), and the ten data bits 9–0, to the vertical synchronization position (Nvsp).

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

	AC line count (Mw)						Vertical synchronization position (Nvsp)									
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR6	Mw4	Mw3	Mw2	Mw1	Mw0	—	Nvsp 9	Nvsp 8	Nvsp 7	Nvsp 6	Nvsp 5	Nvsp 4	Nvsp 3	Nvsp 2	Nvsp 1	Nvsp 0
Initial value:	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

Bits 15 to 11—AC Line Count (Mw): In this field, set the H/L interval of repeated output of the LCD driving signal alternation (M) as one less than the CL1 line count. The presence or absence of CL1 in the vertical retrace line interval specified by CC[1] is immaterial. Mw cannot be set to 0.

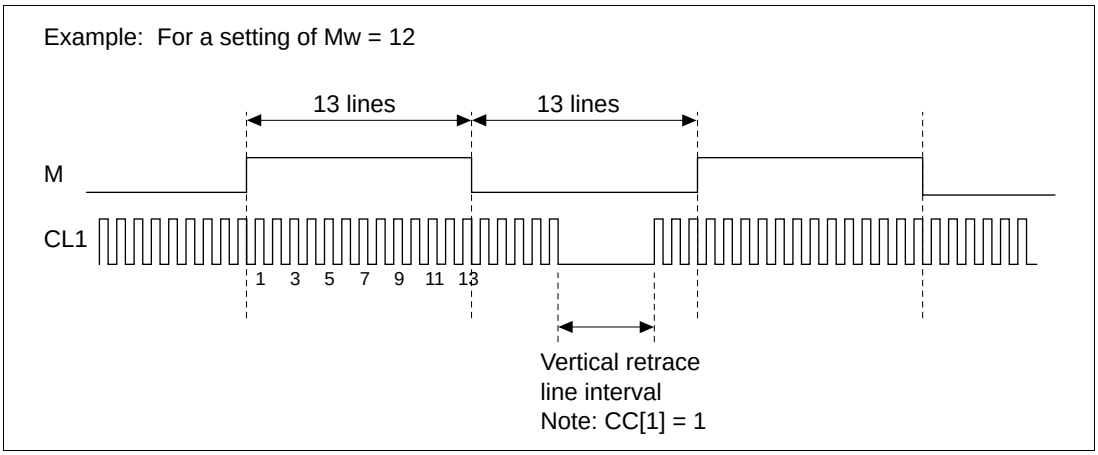


Figure 21.3 Output of LCD Driving Signal Alternation (M)

Bit 10—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 9 to 0—Vertical Synchronization Position (Nvsp): Specifies the FLM clock output position. For example, the clock can be output in the first line if Nvsp is set to 0, or in the second line if Nvsp is set to 1. The width is fixed at the length of one line.

Examples: Figure 21.4 shows examples for Nvsp settings of 0 and 1 for display mode A, a 480 × 240 screen, 8-dot horizontal retrace line interval, and 2 vertical retrace line interval.

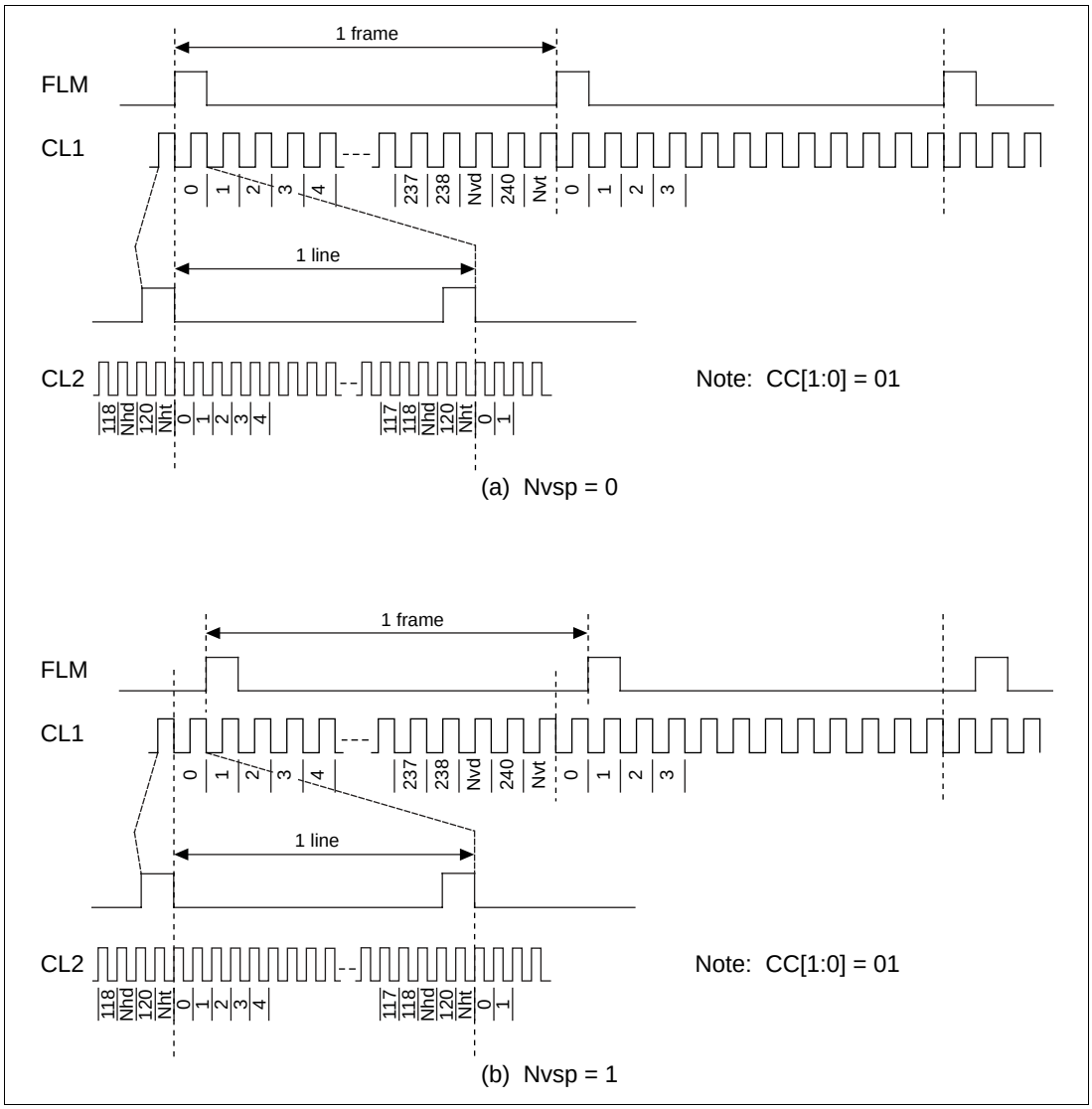
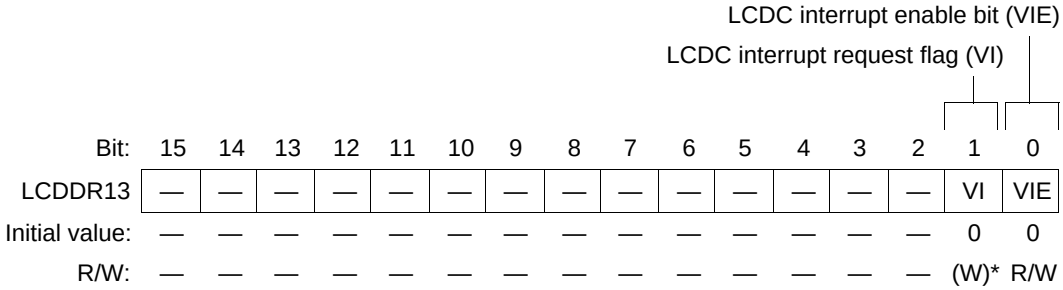


Figure 21.4 FLM Clock Position

LCDC Interrupt: This register consists of the LCDC interrupt request flag (VI) and LCDC interrupt enable bit (VIE).

Both bits are cleared to 0 by a power-on reset. In a manual reset, VI is cleared to 0 but VIE retains its value.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VI	VIE
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0
R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(W)*	R/W

Note: Can only be cleared to 0 by a write access to LCDDR13.

Bits 15 to 2—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bit 1—LCDC Interrupt Request Flag (VI): VI is set to 1 at the start position of the vertical retrace line interval. If there is no vertical retrace line interval (Nvd = Nvt), VI is always 0. When VI is set to 1 and the LCDC interrupt enable bit (VIE) is also 1, the LCDC outputs an LCDC interrupt request to the CPU.

The VI bit cannot be read. A read will return 0.

The VI bit can be cleared to 0 by a write access to the LCDDR13 register.

Bit 1: VI	**Description**	
1	LCDC interrupt request generated when VIE = 1	
0	LCDC interrupt request not generated	(Initial value)

Bit 0—LCDC Interrupt Enable Bit (VIE): LCDC interrupts are enabled when VIE = 1, and disabled when VIE = 0.

Bit 0: VIE	**Description**	
1	LCDC interrupts enabled	
0	LCDC interrupts disabled	(Initial value)

21.2.3 Palette Registers (LCDPR)

These registers specify, as gradation numbers, the display colors corresponding to display data written to memory. The gradation numbers run from 0 to 15, enabling specification of a white display, black display, and 14 intermediate colors. Palette register settings are necessary in all display modes (modes A to G).

Display data is processed in 1-, 2-, or 4-bit units, according to the gradation/color mode set with bits GR[0:1] in display control register LCDDR1. Each unit is called dot data, as it corresponds to

one display screen dot, and the respective modes are called the 1-bit/dot, 2-bits/dot, and 4-bits/dot mode.

The LCDC selects the palette register with the same number as the dot data value, and outputs the display color corresponding to the gradation number set in that register. For intermediate shades corresponding to gradation numbers, FRC (frame rate control) is used to vary the integration voltage applied to the cells to create a state of semi-luminance. However, in the TFT/TFD color mode (mode G), the set value of the selected palette register is output as display data without passing through the FRC block.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

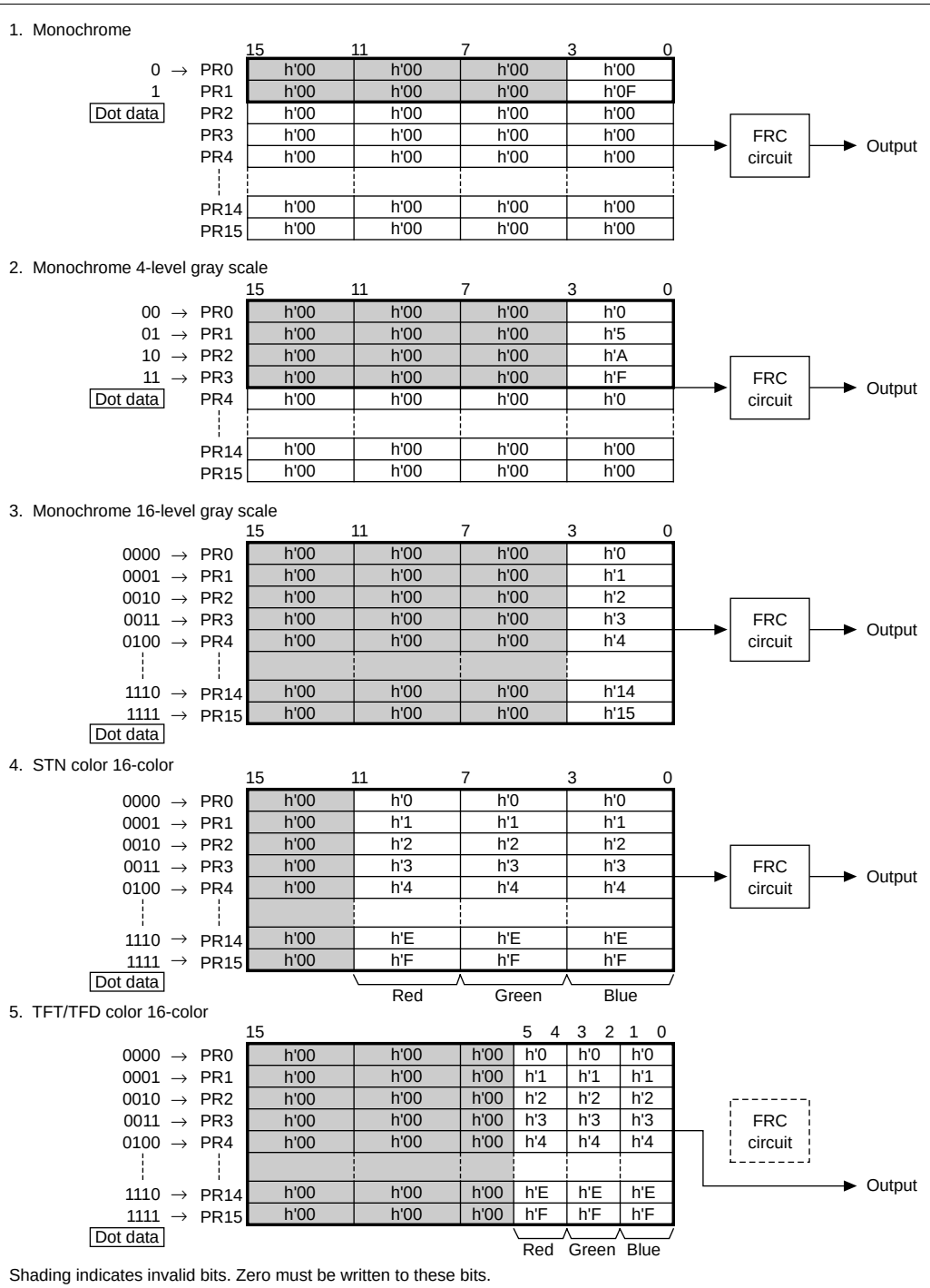
					Palette											
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDPR0–LCDPR15	—	—	—	—	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Initial value:	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 12—Reserved: Writes are invalid, but 0 must always be written. A read will return 0.

Bits 11 to 0—Data Bits: These bits can be read or written. Set LCDPR0 and LCDPR1 in 1-bit/dot mode, LCDPR0–LCDPR3 in 2-bits/dot mode, and LCDPR0–LCDPR15 in 4-bits/dot mode. Write 0 to unused palette registers.

In STN monochrome mode and reflective color mode (modes A, B, C, E, F), only the lower 4 bits [3:0] are used, and 0 should be written to the upper 8 bits [11:4]. In TFT/TFD color mode (mode G), only the lower 6 bits [5:0] are used: blue data is defined as being written to bits [1:0], green data to bits [3:2], and red data to bits [5:4]. Write 0 to the upper 6 bits [11:6]. In STN color mode (mode D), write blue data to bits [3:0], green data to bits [7:4], and red data to bits [11:8].

Figure 21.5 shows some examples of palette register settings.



Shading indicates invalid bits. Zero must be written to these bits.

Figure 21.5 Examples of Palette Register Settings

21.2.4 DMA Control Registers (LCDDMR)

These registers specify the DMA transfer start address and amount of transmit data to enable the LCDC to fetch in display data written in memory. There is also a bit for switching between the little-endian basic mode and linear byte address mode.

Display Start Address (Low): This register specifies part of the screen start address in single-screen mode, or the upper panel display start address in dual-screen mode. The upper panel display start address (low) register (LCDDMR0) is set with the lower 16 bits of the 32-bit specified address.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

	Display start address (low)															
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR0	UMA 15	UMA 14	UMA 13	UMA 12	UMA 11	UMA 10	UMA 9	UMA 8	UMA 7	UMA 6	UMA 5	UMA 4	UMA 3	UMA 2	UMA 1	UMA 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Display Start Address (Low): To perform LCDC DMA transfer in 16-byte units, set UMA[3:0] to 0000. This is to prevent two DRAM RAS cycles occurring within one DMA access, enabling efficient use of the memory bus.

Display Start Address (High): This register specifies part of the screen start address in single-screen mode, or the upper panel display start address in dual-screen mode. The upper panel display start address (high) register (LCDDMR1) is set with the upper 16 bits of the 32-bit specified address.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

	Display start address (high)															
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR1	UMA 31	UMA 30	UMA 29	UMA 28	UMA 27	UMA 26	UMA 25	UMA 24	UMA 23	UMA 22	UMA 21	UMA 20	UMA 19	UMA 18	UMA 17	UMA 16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Display Start Address (High): All bits are readable/writable, but as the SH7707 supports a 29-bit physical address space, the most significant 3 bits are actually ignored.

Lower Panel Display Start Address (Low): This register specifies part of the lower panel display start address in dual-screen mode. The lower panel display start address (low) register (LCDDMR2) is set with the lower 16 bits of the 32-bit specified address.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

Lower panel display start address (low)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR2	LMA 15	LMA 14	LMA 13	LMA 12	LMA 11	LMA 10	LMA 9	LMA 8	LMA 7	LMA 6	LMA 5	LMA 4	LMA 3	LMA 2	LMA 1	LMA 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Lower Panel Display Start Address (Low): To perform LCDC DMA transfer in 16-byte units, set UMA[3:0] to 0000. This is to prevent two DRAM RAS cycles occurring within one DMA access, enabling efficient use of the memory bus.

Lower Panel Display Start Address (High): This register specifies part of the lower panel display start address in dual-screen mode. The lower panel display start address (high) register (LCDDMR3) is set with the upper 16 bits of the 32-bit specified address.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.

Lower panel display start address (high)

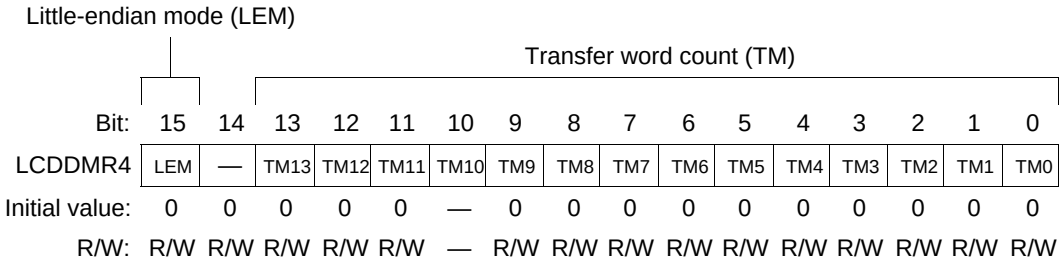
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR3	LMA 31	LMA 30	LMA 29	LMA 28	LMA 27	LMA 26	LMA 25	LMA 24	LMA 23	LMA 22	LMA 21	LMA 20	LMA 19	LMA 18	LMA 17	LMA 16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Lower Panel Display Start Address (High): All bits are readable/writable, but as the SH7707 supports a 29-bit physical address space, the most significant 3 bits are actually ignored.

Transfer Word Count: This register specifies the number of words in a DMA transfer, and also the little-endian mode. Data bit 15 is the little-endian mode (LEM) bit, and the 14 data bits 13–0 contain the transfer word count.

This register must not be modified during a display operation.

All bits are cleared to 0 by a power-on reset, but retain their values in a manual reset.



Bit 15—Little-Endian Mode (LEM): Functions only when little-endian mode is used. The data fetched into the LCDC by DMA transfer can be replaced byte by byte at a time. When display data is written to memory in byte units, setting LEM to 1 enables the corresponding screen to be obtained in address order. When data is written in longword units, LEM should be cleared to 0. The default setting is 0 (basic mode). In big-endian mode the LEM bit has no function, but should be written with 0. See section 21.3.2 for details.

Bit 0: LEM	Description	
1	Linear byte address mode	
0	Basic mode	(Initial value)

Bits 13 to 0—Transfer Word Count (TM): In this field, the total amount of data for one screen to be transferred by DMA is set as one less than the number of words to be transferred. As 16 bytes are transferred in one DMA transfer (32 × 4, burst), the transfer word count is given by the following equations.

Examples: 640 × 480 screen size, monochrome 16-level gray scale, single-screen display:

$$(\underbrace{640 \times 480}_{\text{Total dots}} \times \underbrace{4}_{\text{Bits/dot}}) \div (\underbrace{16 \times 8}_{\text{16B}}) - 1 = 9599\ (\text{H'257F})$$

640 × 480 (640 × 240 × 2) display screen size, monochrome, dual-screen display:

$$(\underbrace{640 \times 480}_{\text{Total dots}} \times \underbrace{1}_{\text{Bits/dot}}) \div (\underbrace{16 \times 8}_{\text{16B}}) - 1 = 2399\ (\text{H'95F})$$

21.3 Operation

21.3.1 Gradation Processing

With a basic monochrome display, the dots making up the LCD screen are driven with one frame as one cycle, and ON display dots are driven ON every frame. Thus, there are two effective application voltages for the liquid crystal, V_{ON} and V_{OFF}. With a graduated display, on the other hand, a number of frames are handled as one cycle, and if an OFF-drive frame is inserted for an ON-driven dot, the effective voltage applied to that dot falls to a level between V_{ON} and V_{OFF}, and the transmittivity of the liquid crystal changes. This is the principle of FRC (frame rate control). For example, imagine that a white disk is divided into 16 equal segments, a number of these are painted black, and the disk is rotated at high speed about its center. An intermediate shade between white and black will be obtained, determined by the number of black segments. The number of black segments corresponds, so to speak, to the gradation number. One revolution of this disk is equivalent to one gradation control cycle, and one segment corresponds to the display-ON state in one frame cycle. Varying the proportional illumination in this way produces different shades in the eye of the viewer. For this reason, FRC is also referred to as frame extraction control.

The SH7707/LCDC can display up to 16 gradations, and performs frame extraction control in 16-frame units. That is, if a certain display dot is OFF for all 16 frames, the effective voltage becomes V_{OFF} and the transmittivity of the corresponding liquid crystal becomes T_{OFF}, whereas if all 16 frames are ON, the effective voltage becomes V_{ON} and the liquid crystal transmittivity becomes T_{ON}. In the case of intermediate shades, on the other hand, the liquid crystal ON drive ratio varies between 1/15 and 14/15, and the transmittivity varies between T_1 and T_{14}. However, since the relationship between the liquid crystal effective voltage and transmittivity is not linear, the intervals from T_0 to T_{15} are not equal. Also, since the liquid crystal characteristics vary from panel to panel, the shade may appear different. The optimum gradation levels should therefore be selected for the panel used. Figures 21.6 and 21.7 show the principles for display of up to 4 gradations.

If all dots are simultaneously turned ON or OFF in a frame, flicker will result. To prevent this problem the SH7707/LCDC employs a method known as space-modulation FRC, in which a number of adjacent dots are grouped together, and the ON/OFF timing is shifted within the group.

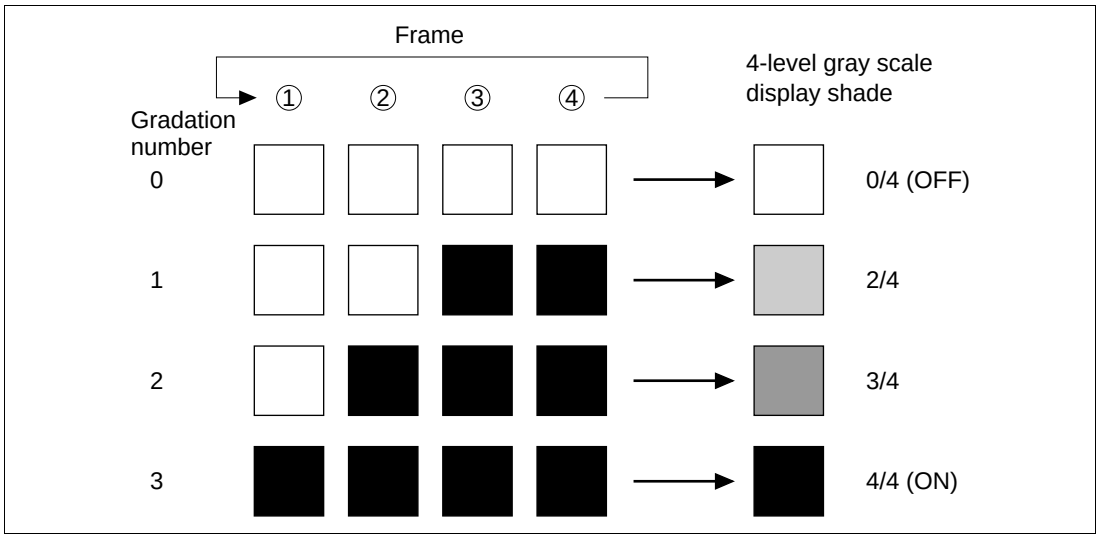


Figure 21.6 Principle of FRC (Frame Rate Control)

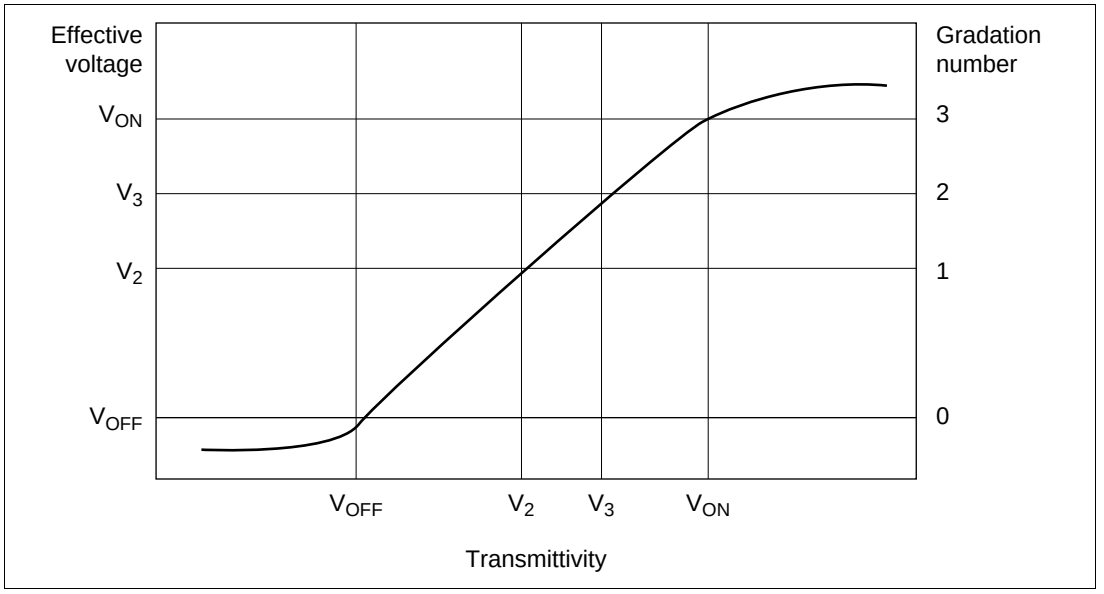


Figure 21.7 Relationship between Liquid Crystal Effective Voltage, Gradation Number, and Transmittivity

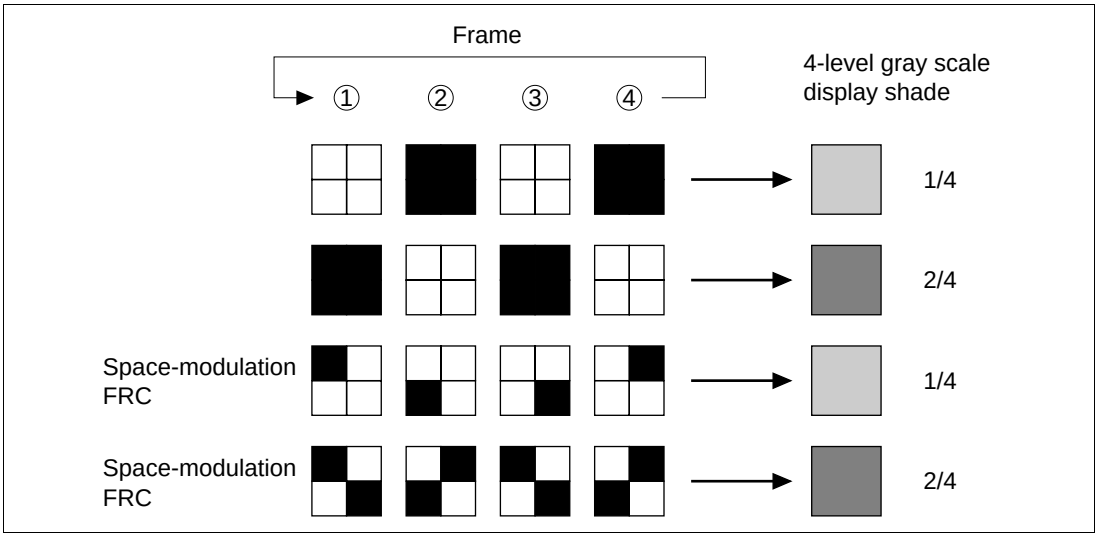


Figure 21.8 Principle of Space-Modulation FRC

21.3.2 Endian

The SH7707 allows data to be stored in memory in big-endian or little-endian form, and the LCDC offers a further choice of basic mode or linear byte access mode as the little-endian mode.

As the LCDC always fetches display data from memory 32 bits at a time, and processes the data sequentially, in little-endian basic mode a display in address order is obtained only when writes are performed in longword (32-bit) units. In little-endian linear byte address mode, on the other hand, a display that matches the address order is obtained when byte access is used for writes to memory. This is because the 32-bit data fetched by the LCDC is replaced byte by byte only when LEM =1 in little-endian mode. In big-endian mode, a display matching the address order is obtained regardless of whether the display data is written in byte, word, or longword units.

Figure 21.9 shows the correspondence between memory and the data sequence fetched by the LCDC in each endian mode, and figure 21.10 shows the correspondence between the memory data and display dots on the actual LCD panel.

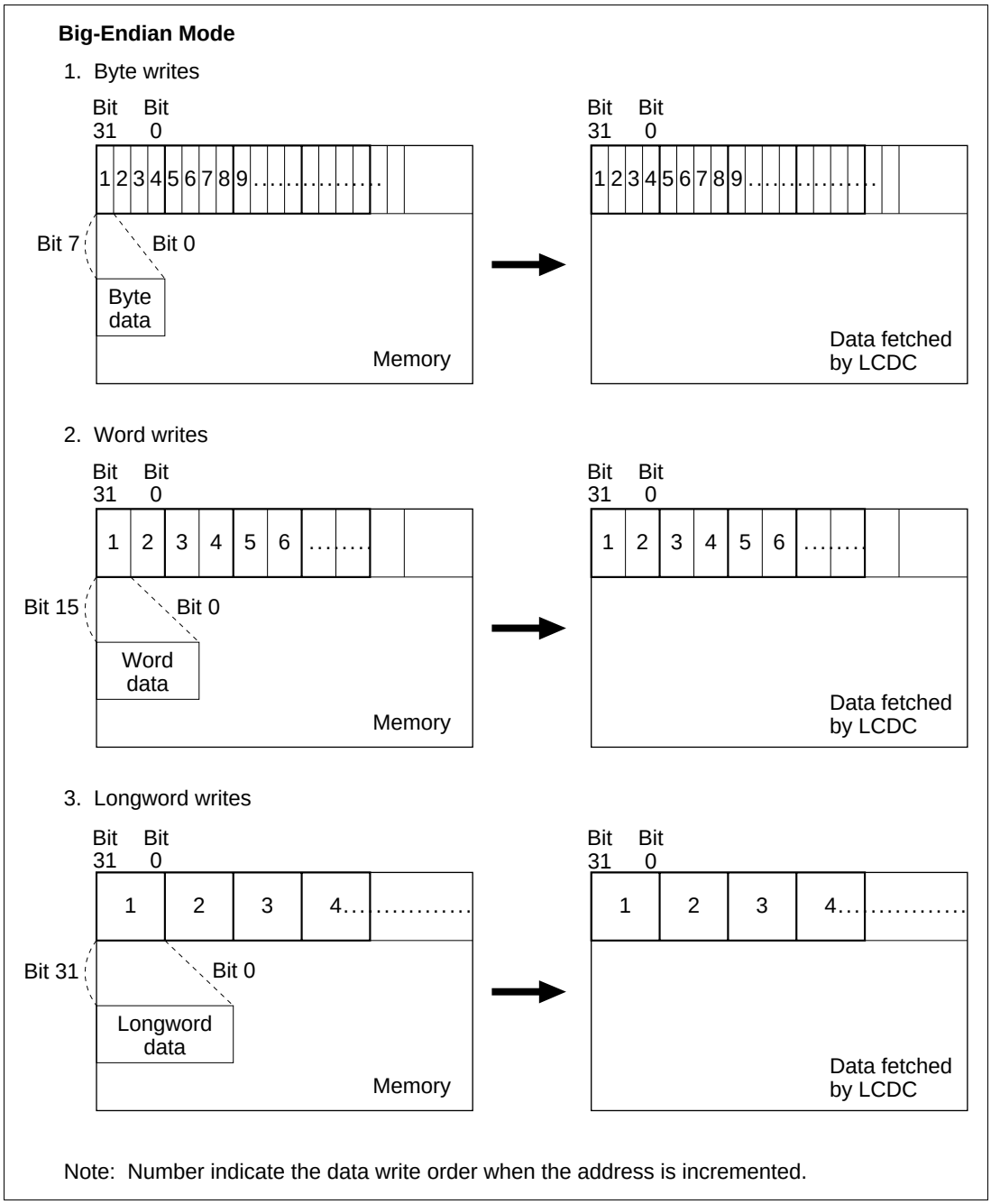
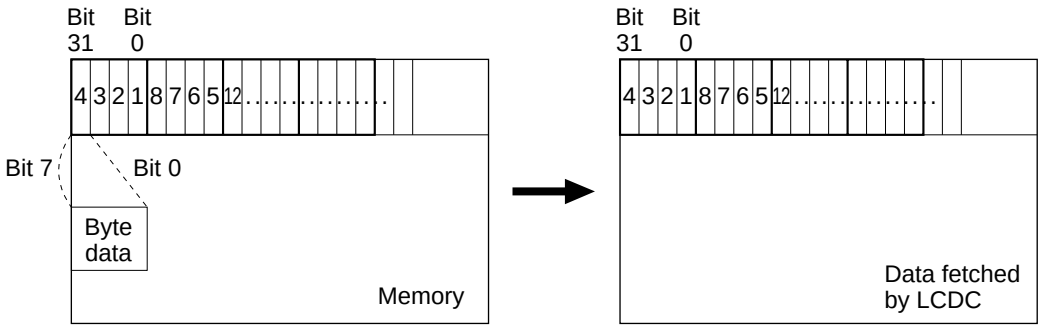


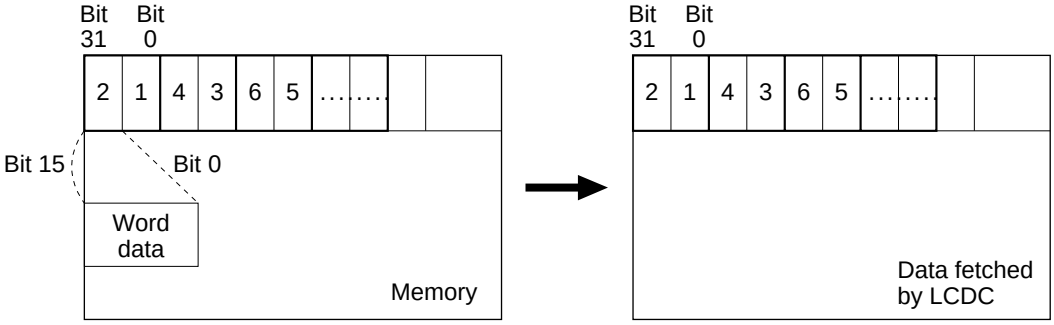
Figure 21.9 Data Correspondence in Each Endian Mode

Little-Endian Mode (LEM = 0)

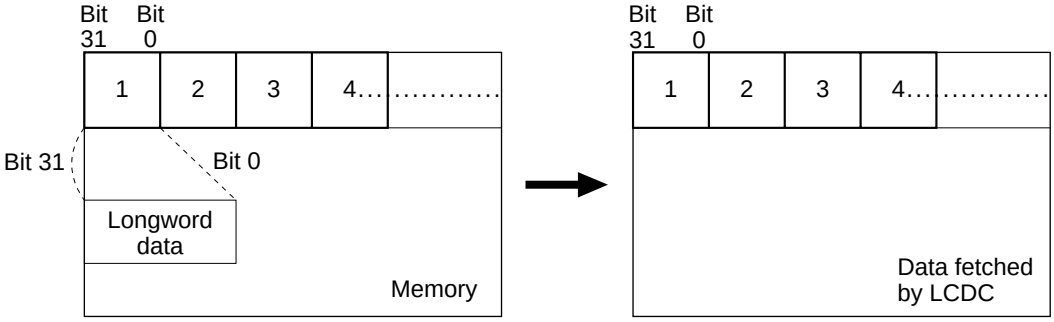
1. Byte writes



2. Word writes



3. Longword writes

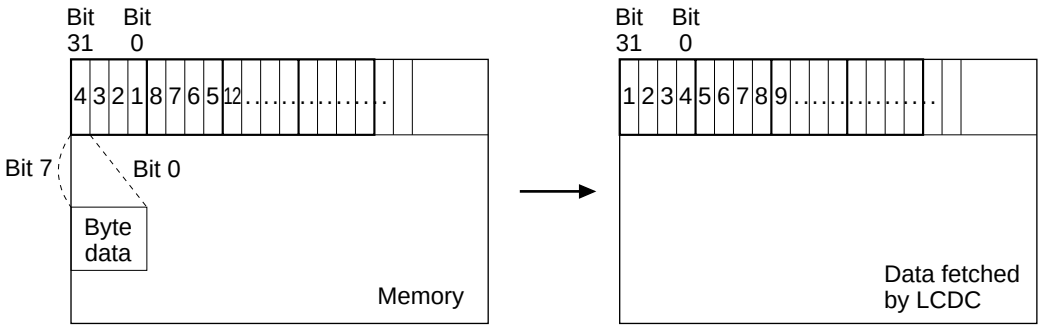


Note: Number indicate the data write order when the address is incremented.

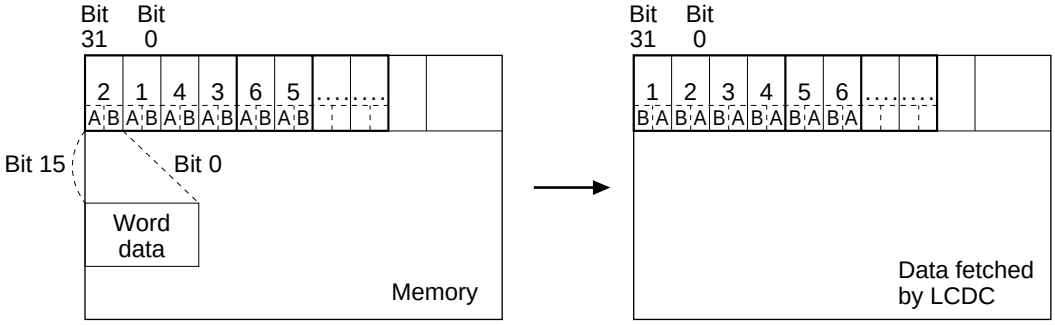
Figure 21.9 Data Correspondence in Each Endian Mode (cont)

Little-Endian Mode (LEM = 1)

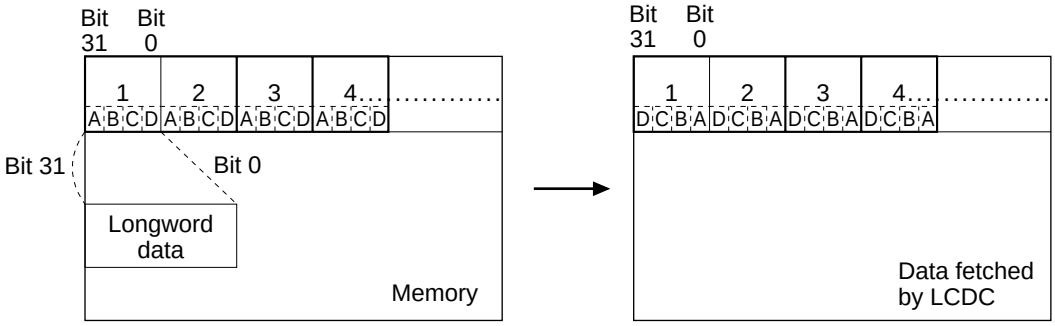
1. Byte writes



2. Word writes



3. Longword writes



Note: Number indicate the data write order when the address is incremented.
A, B, C and D in the figure indicate the byte data sequence and correspondence.

Figure 21.9 Data Correspondence in Each Endian Mode (cont)

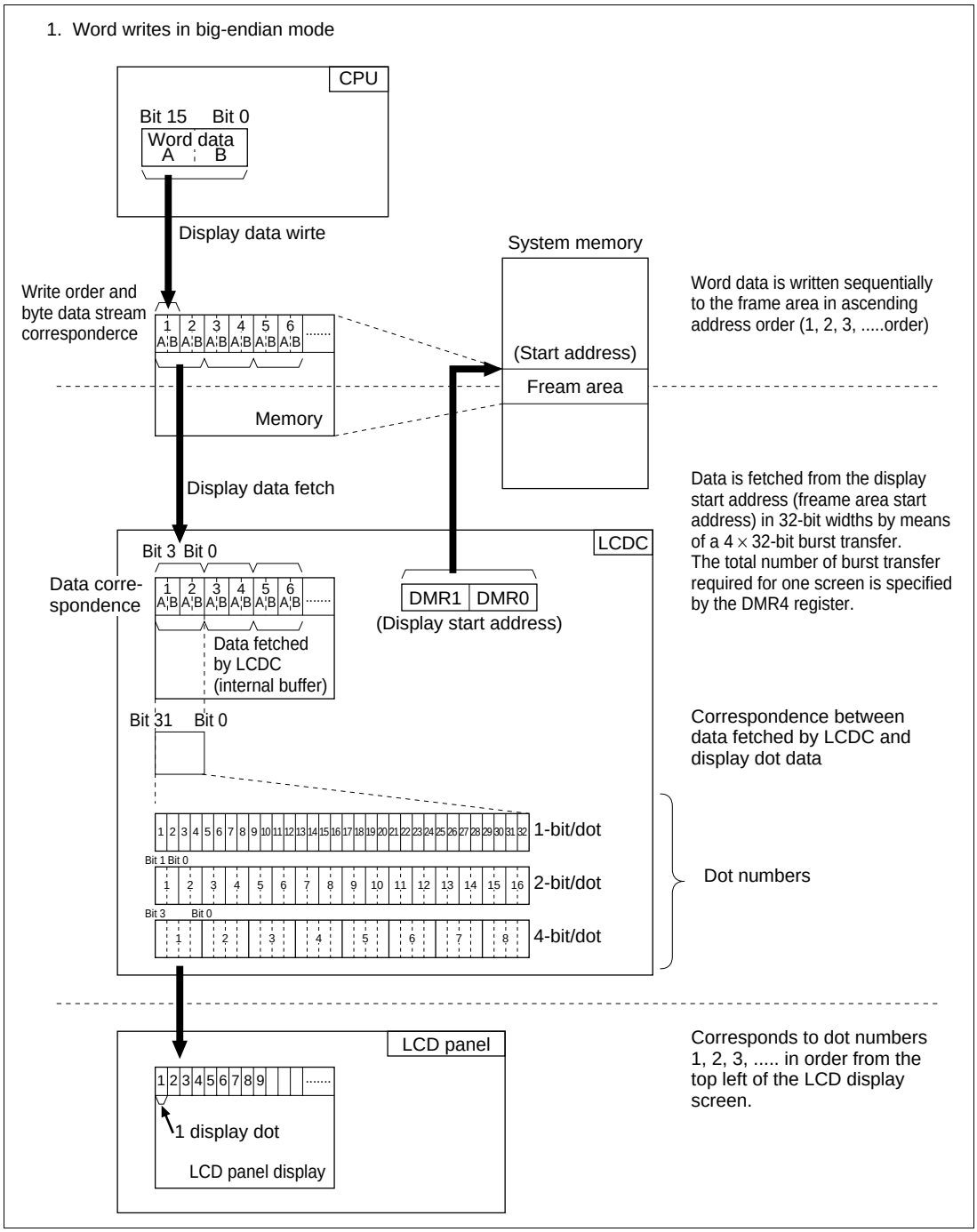


Figure 21.10 Example of Correspondence between Memory and Display

2. Longword writes in little-endian mode (LEM = 0), or longword write in big-endian mode

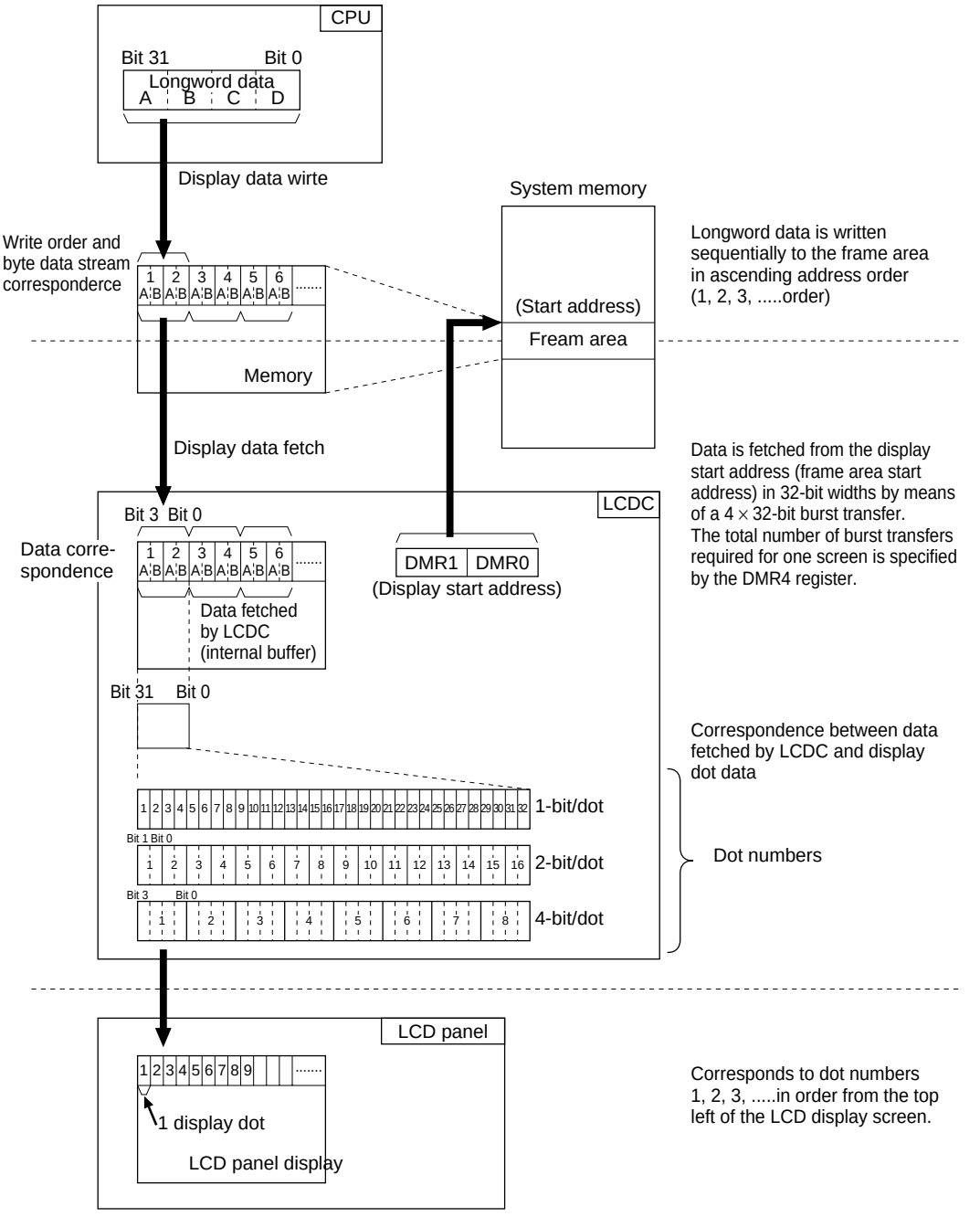


Figure 21.10 Example of Correspondence between Memory and Display (cont)

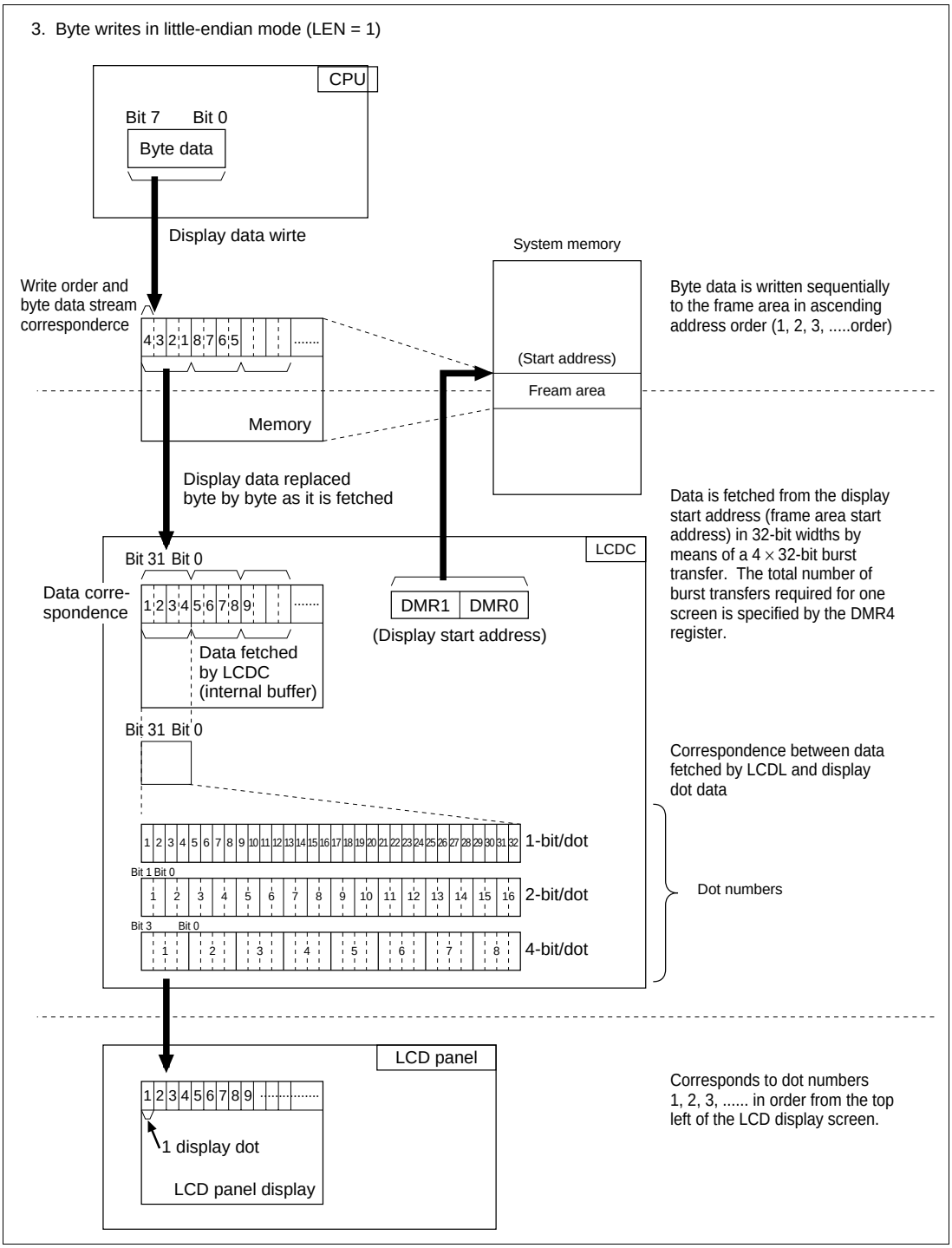


Figure 21.10 Example of Correspondence between Memory and Display (cont)

21.3.3 Palette Register Settings

See figure 21.5.

Monochrome Mode: Palette registers LCDPR0 and LCDPR1 must be set. LCDPR0 is selected when the dot data is 0, and LCDPR1 when the dot data is 1. The gradation number is set in the lower 4 bits of the register; a setting of H'F is treated as 0.

Monochrome 4-Gradation/Reflective Color 4-Color Mode: Palette registers LCDPR0–LCDPR3 should be set. The gradation number is set in the lower 4 bits of the register. An integer from 0 to 15 can be selected as the gradation number; the value should be determined on the basis of the characteristics of the panel to be used. LCDPR0, LCDPR1, LCDPR2, and LCDPR3, are selected for dot data 00, 01, 10, and 11, respectively, and the shade corresponding to the gradation number set in the respective register is displayed.

Figure 21.11 shows an example of display dot data and display shades.

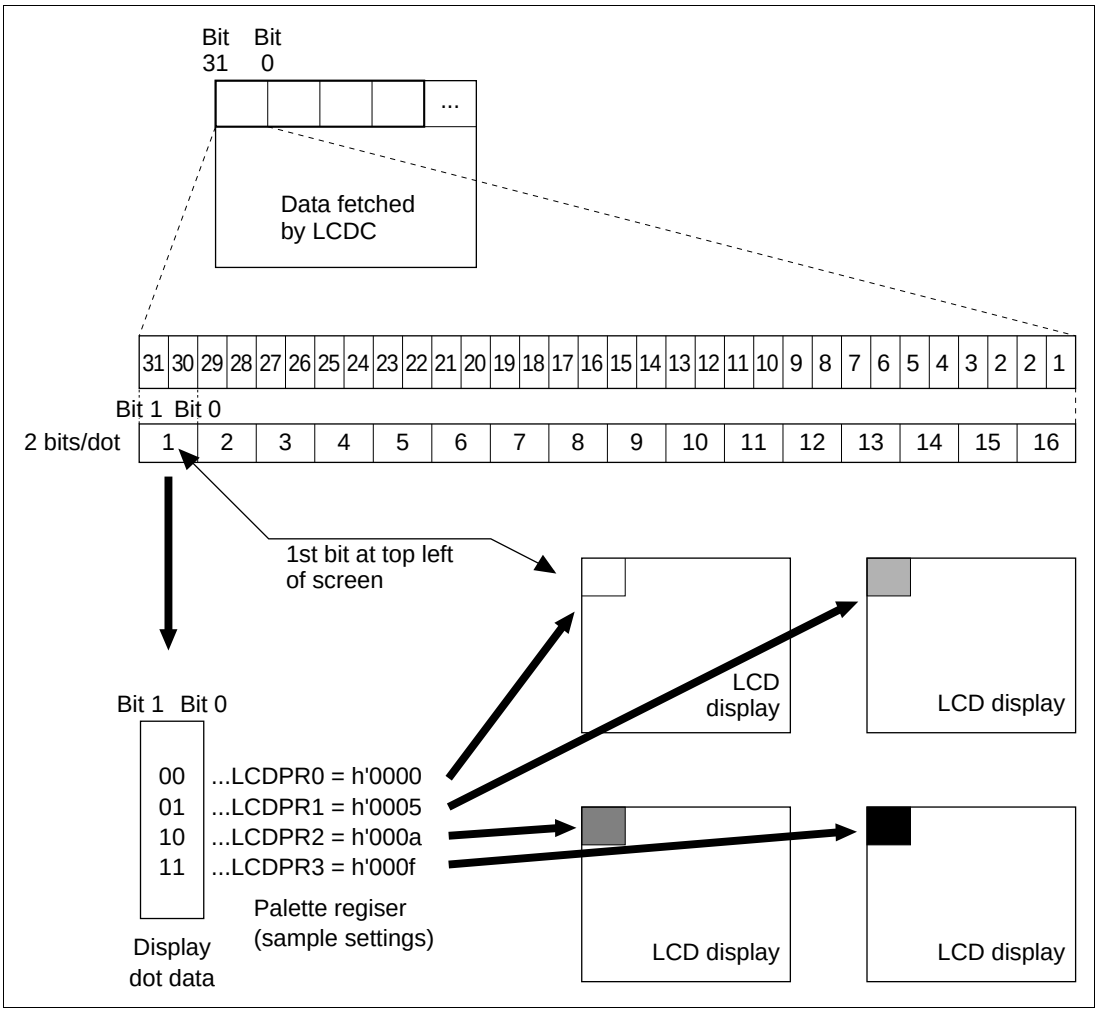


Figure 21.11 Relationship between Display Dot Data and Display

Monochrome 16-Level Gray Scale/Reflective Color 16-Color Mode: Palette registers LCDPR0– LCDPR15 are all selected for use. The gradation number is set in the lower 4 bits of the register. Set an integer from 0 to 15 as the gradation number. The register number corresponding to the dot data value is selected (dot data 0000 → LCDPR0, 0001 → LCDPR1, etc.). and the shade corresponding to the gradation number set in the respective register is displayed.

STN Color Mode: The procedure is basically the same as for monochrome 16-gradation mode, etc., but the gradation number corresponding to red should be set in the upper 4 bits (bits 11–8) of the palette register, that corresponding to green in the middle 4 bits (bits 7–4), and that corresponding to blue in the lower 4 bits (bits 3–0). 16 gradation numbers can be set for each of R, G, and B.

TFT/TFD Color Mode: The set value in the lower 6 bits (bits 5–0) of the palette register selected from the dot data is output directly without passing through the FRC (frame rate control) circuit. The upper 2 bits (bits 5 and 4) are defined as the red data area, the middle 2 bits (bits 3 and 2) as the green data area, and the lower 2 bits (bits 1 and 0) as the blue data area.

21.3.4 Determining the DIV Set Value (Clock Ratio)

The data transfer rate per display dot is determined by the screen size and frame frequency, and is known as the dot clock frequency. As the LCDC operates in synchronization with the CKIO clock, the dot clock frequency is also determined by the CKIO clock ratio. Thus, settings must be made so that both coincide.

A dot clock frequency/CKIO frequency ratio of ×2, ×1, ×1/2, ×1/4, or ×1/8 can be selected, the setting being made in bits 10–8 (DIV[2:0]) of display control register LCDDR1.

First, the clock ratio to be set is roughly determined based on the assumed CKIO clock frequency and the general frame frequency. The frame frequency can then be finely adjusted by changing the horizontal and vertical retrace line widths. The concept is illustrated by the following example.

Example: In a system with a 480 × 200 screen size and a CKIO frequency of 30 MHz, if the frame frequency is assumed to be 70 [Hz], the dot clock frequency will be:

480 [dots] × 200 [lines] × 70 [Hz] = 6.72 [MHz]

From this value, it can be seen that the appropriate clock ratio is x1/4. Thus, the setting for DIV[2:0] is 011, and the dot clock frequency determined by the ratio to CKIO is:

30 [MHz] × 1/4 = 7.5 [MHz]

To make the 6.72 [MHz] value obtained earlier match this value, the horizontal and vertical retrace line intervals can be set to 16 [dots] and 16 [lines], for example, which, taking the frame frequency as 70 [MHz], gives:

496 [dots] × 216 [lines] × 70 [Hz] = 7.5 [MHz]

The register settings will therefore be as follows:

DIV[2:0] = 011,
Nhd = 119, Nht = 123,
Nvd = 199, Nvt = 215

However, since the dot clock frequency should be 7.5 [MHz] in this example, it would also be possible to decrease the horizontal and vertical retrace line widths, and increase the frame frequency correspondingly.

These concepts are illustrated in figure 21.12.

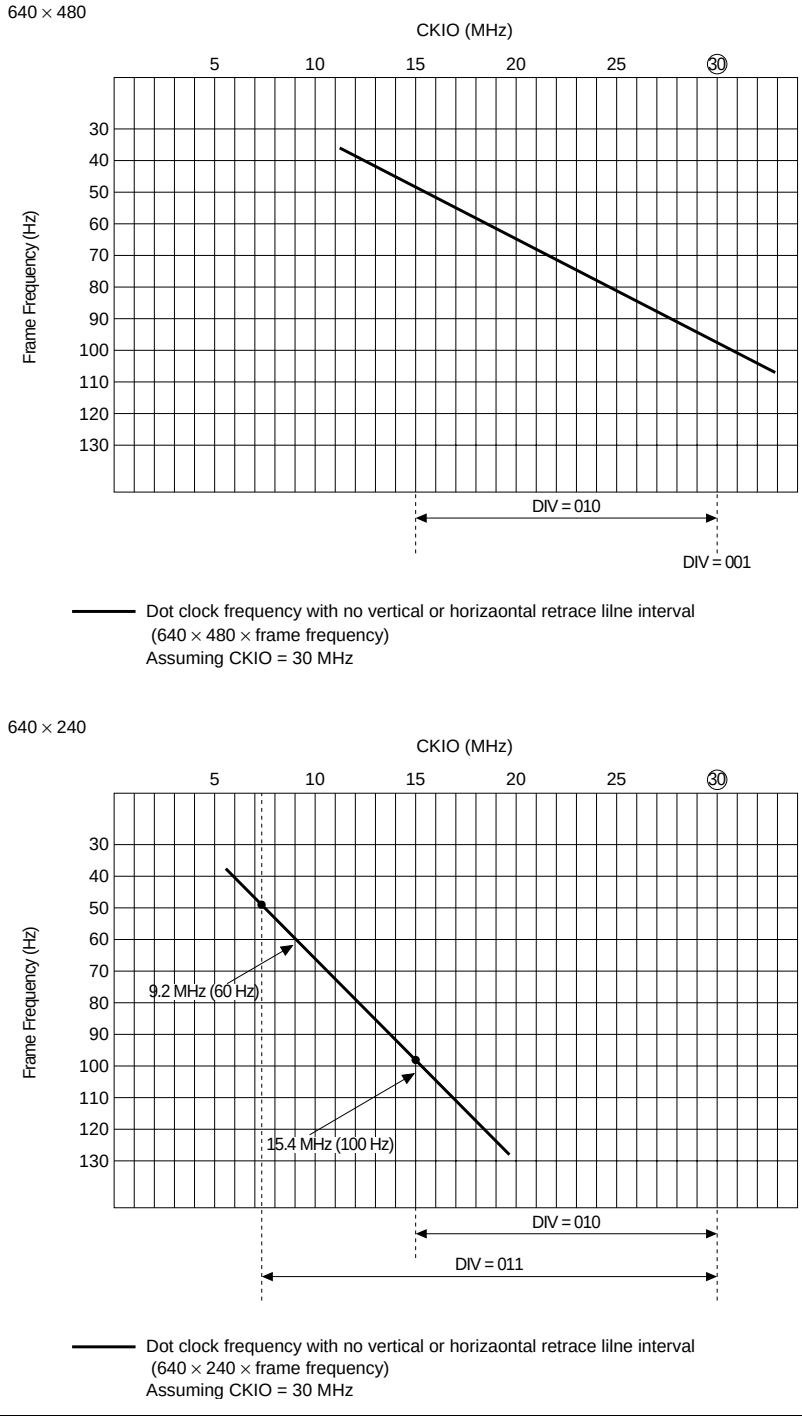
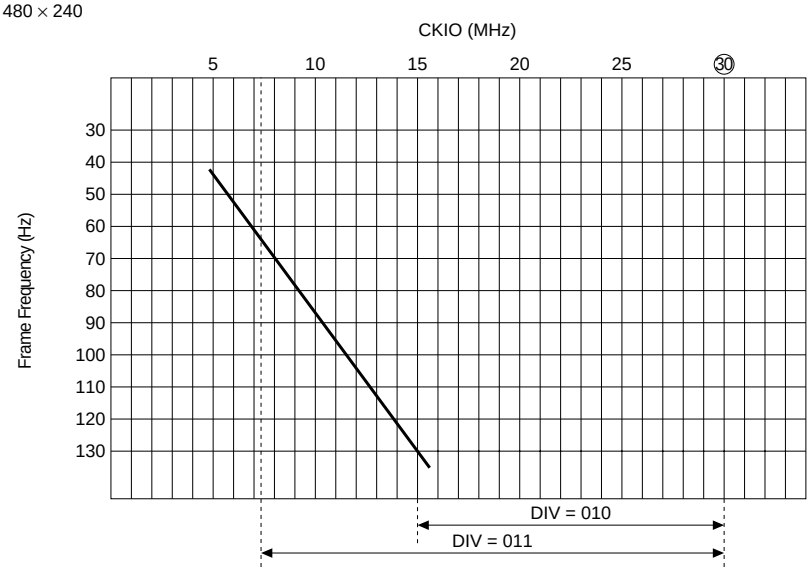


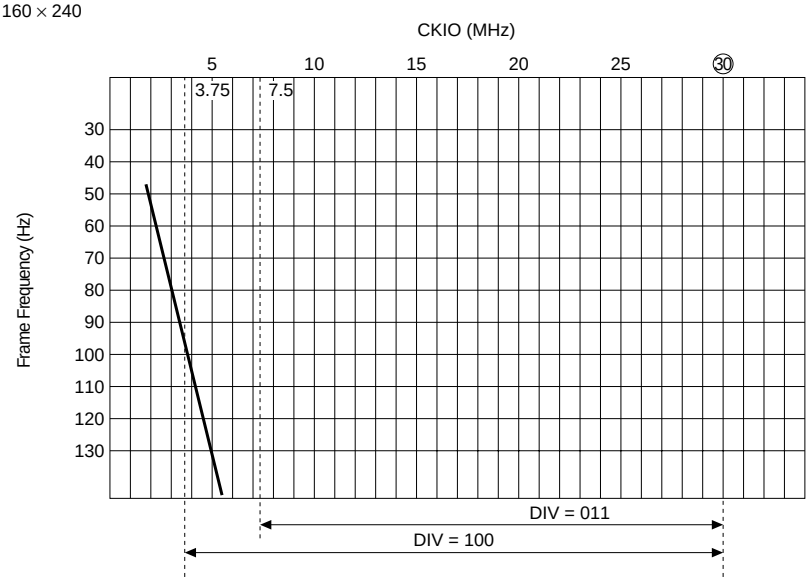
Figure 21.12 Relationship between CKIO and Frame Frequency

480 × 240



Dot clock frequency with no vertical or horizaontal retrace lilne interval
(480 × 240 × frame frequency)
Assuming CKIO = 30 MHz

160 × 240



Dot clock frequency with no vertical or horizaontal retrace lilne interval
(160 × 240 × frame frequency)
Assuming CKIO = 30 MHz

Figure 21.12 Relationship between CKIO and Frame Frequency (cont)

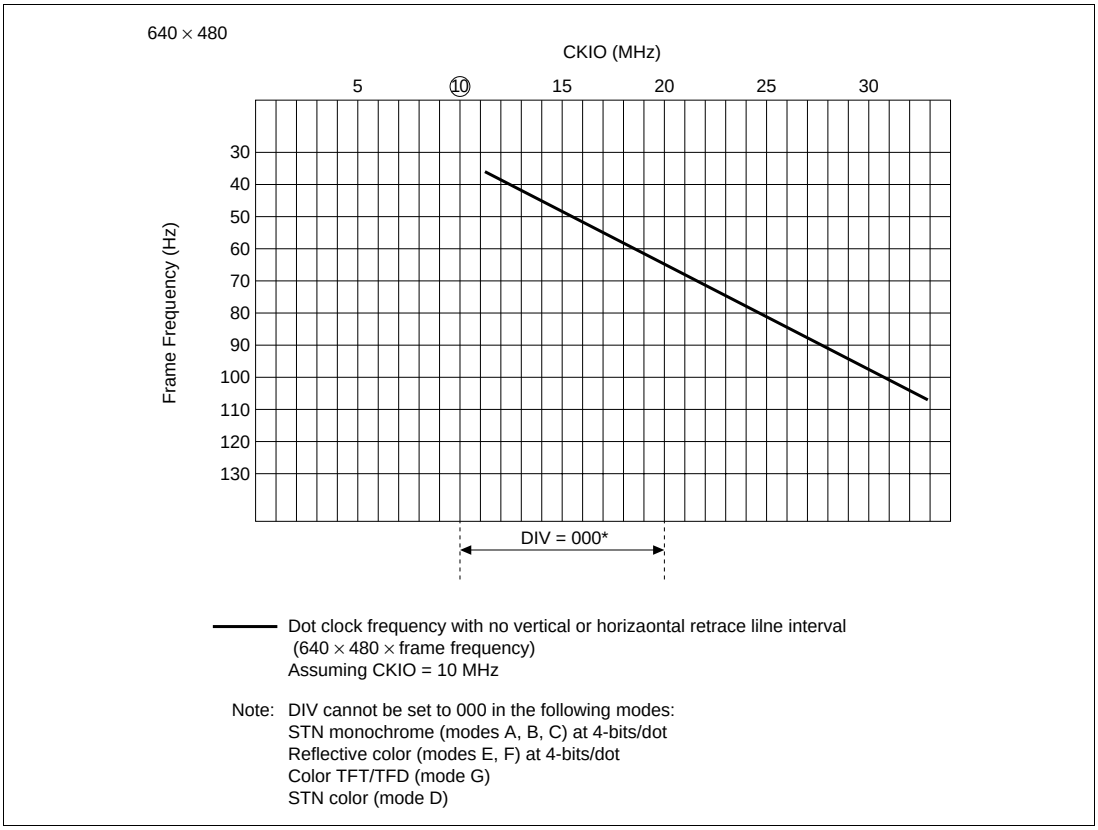
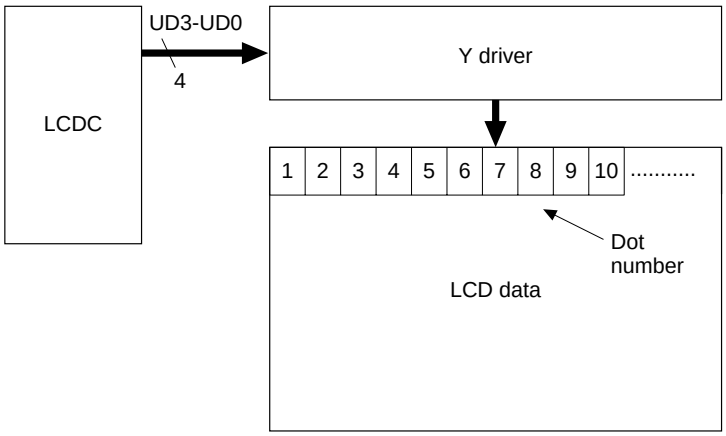


Figure 21.12 Relationship between CKIO and Frame Frequency (cont)

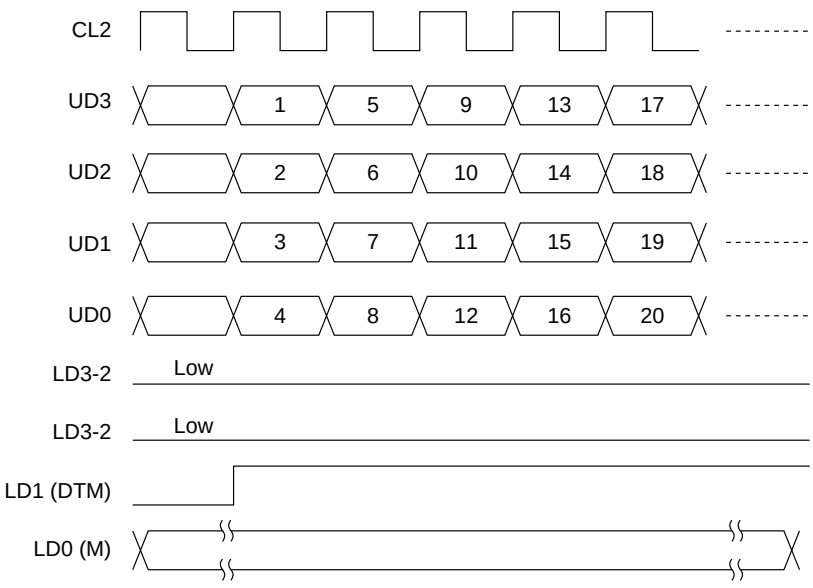
21.3.5 LCD Data Output

The configuration and output dot data in each display mode are shown in figure 21.13.

1. Monochrome/dual screens/4-bit output (mode A, E)



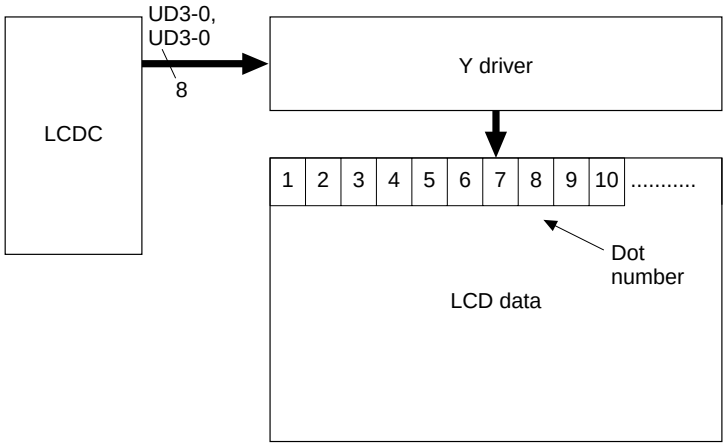
(a) Configuration Diagram



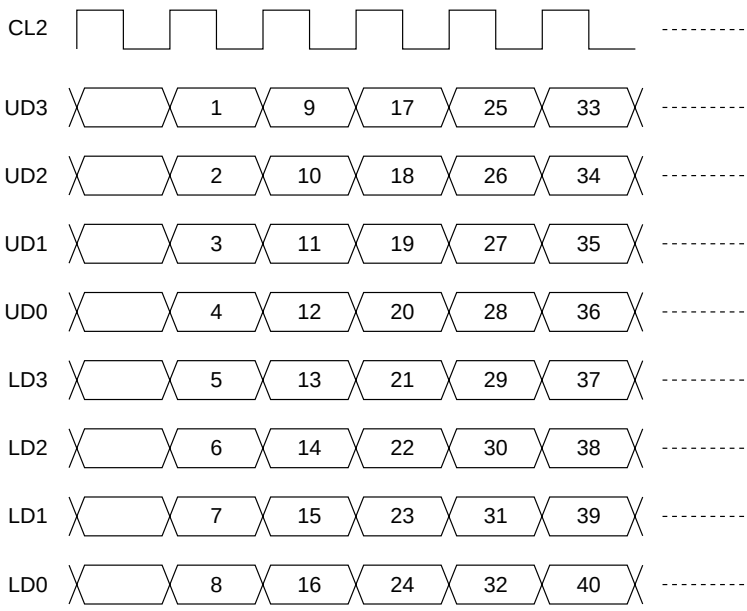
(b) Output from LCDC

Figure 21.13 LCD Configuration Diagram and Data Output

2. Monochrome/single screens/8-bit output (mode B, F)



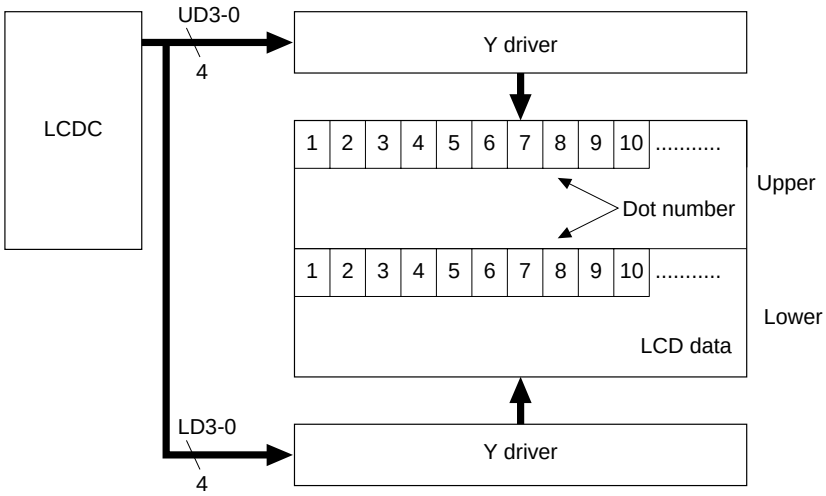
(a) Configuration Diagram



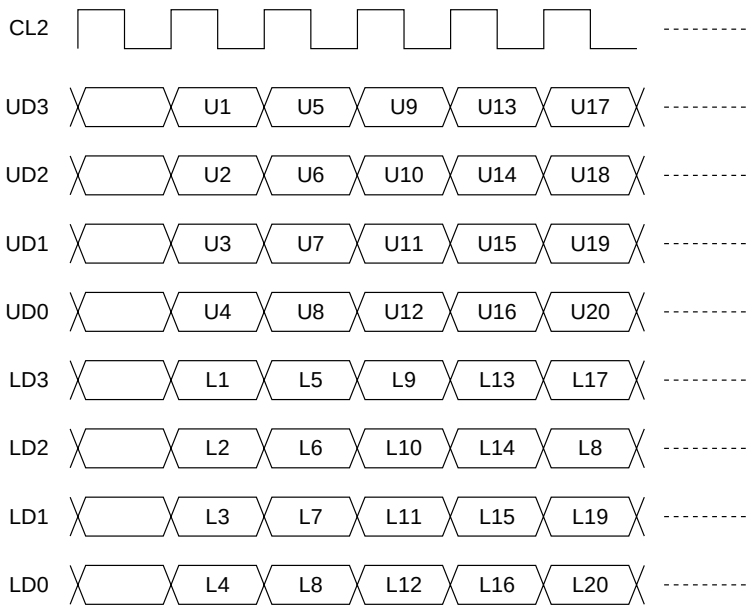
(b) Output from LCDC

Figure 21.13 LCD Configuration Diagram and Data Output (cont)

3. Monochrome/dual screens/4-bit output (mode C)



(a) Configuration Diagram

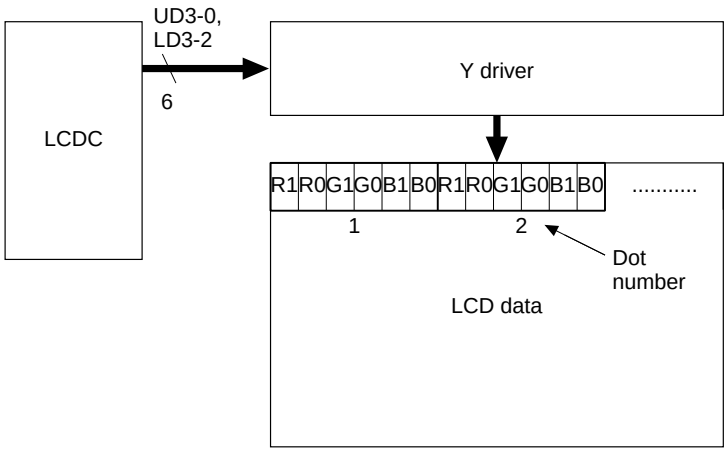


(b) Output from LCDC

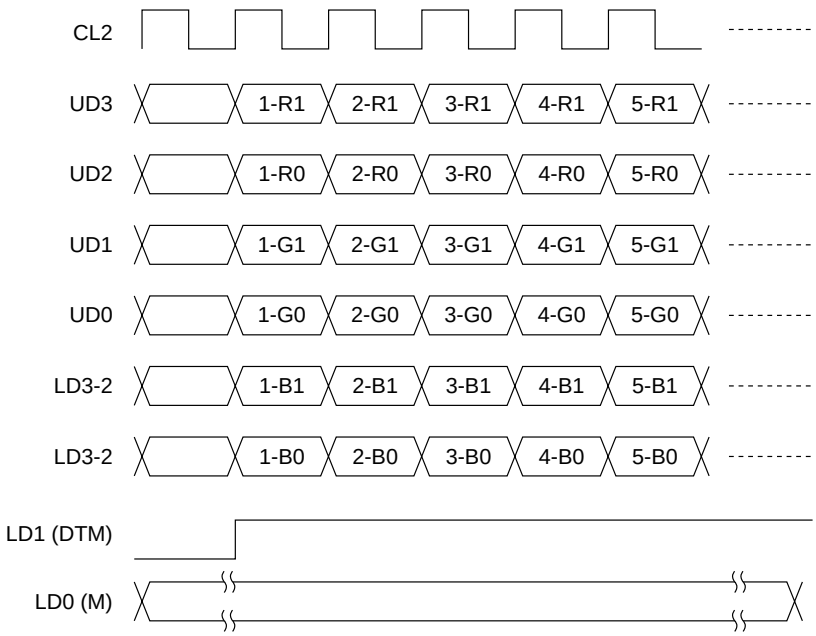
(b) Output from LCDC

Figure 21.13 LCD Configuration Diagram and Data Output (cont)

4. TFT/TFD Color (mode G)



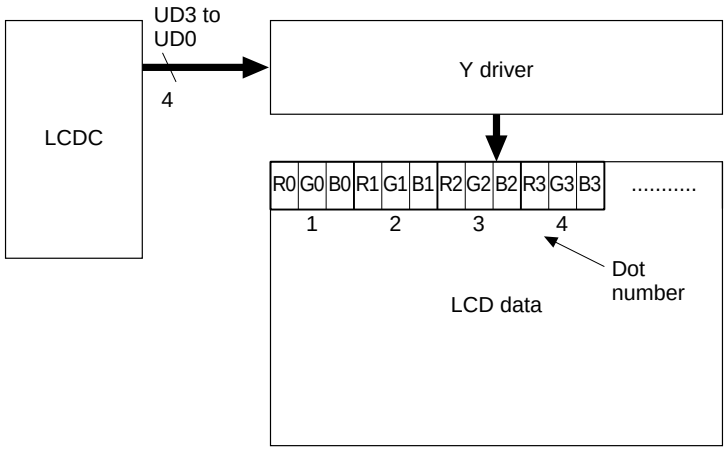
(a) Configuration Diagram



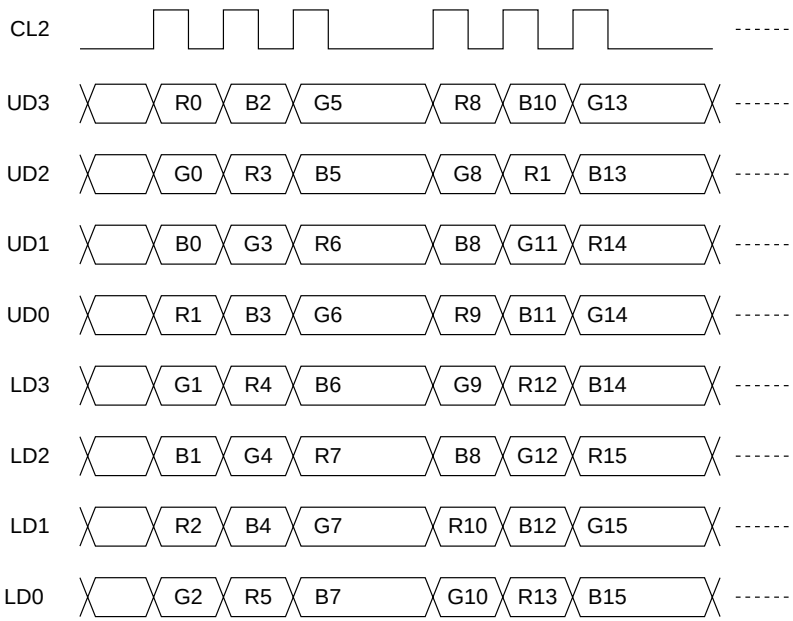
(b) Output from LCDC

Figure 21.13 LCD Configuration Diagram and Data Output (cont)

5. STN Color (mode D)



(a) Configuration Diagram



(b) Output from LCDC

Figure 21.13 LCD Configuration Diagram and Data Output (cont)

21.3.6 Module Stop Function

When 1 is written to bit 4 (MSTP7) or bit 3 (MSTP6) in standby control register STBCR2, the LCDC turns LCD display off, or halts display data transfer, respectively. The LCD pin output and circuit clock operating states during this period are shown in figure 21.8.

If both MSTP6 and MTSP7 are set to 1 at the same time, MSTP7 has priority regarding LCDC output.

Table 21.8 Operation in Module Stop Mode

		MSTP6 (DMA Stop)	MSTP7 (Display Stop)
LCDC output pins	DON	Continues normal output during module stop period	Goes low during module stop period After release, returns to normal output after a maximum of 2 frames
	CL1, CL2, FLM (DTM, M)	Continue normal output during module stop period	Stop output during module stop period, continuing fixed output of last output prior to transition After release, restart output continuously
	UD3–0, LD3–0 (except DTM, M)	Go low during module stop period	Stop output during module stop period, continuing fixed output of last output prior to transition After release, restart output continuously
LCDC circuit clocks	DMA circuit	Halted	Active (DMA transfer of display data halted)
	Display circuit	Active	Halted
	Register circuits	Active	Active

Section 22 A/D Converter

22.1 Overview

The SH7707 includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

22.1.1 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 on 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.

22.1.2 Block Diagram

Figure 22.1 shows a block diagram of the A/D converter.

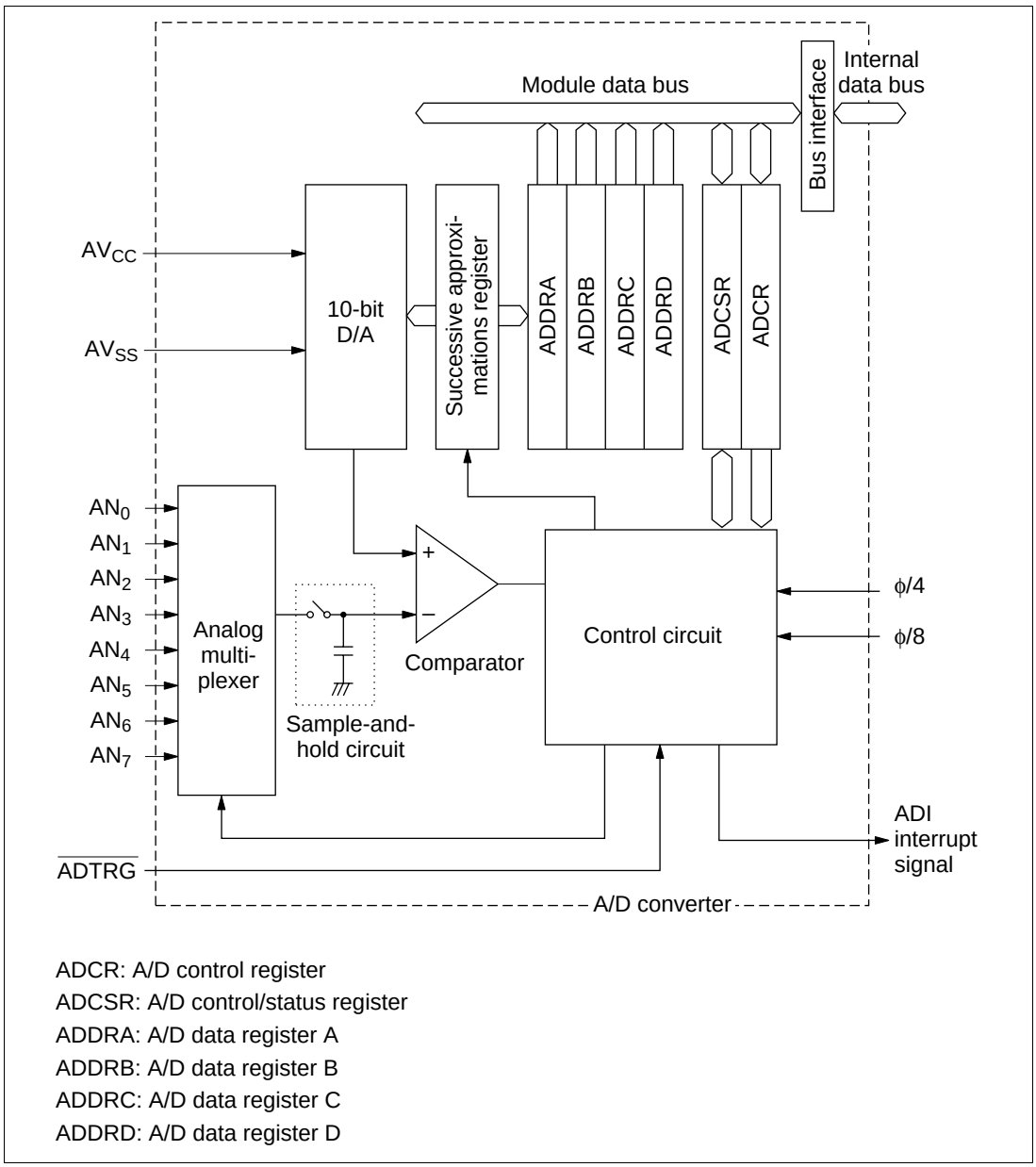


Figure 22.1 A/D Converter Block Diagram

22.1.3 Input Pins

Table 22.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN0–AN3), and group 1 (AN4–AN7). AVCC and AVSS are the power supply pins for the analog circuits in the A/D converter. AVcc also functions as the A/D converter reference voltage.

Table 22.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVCC	Input	Analog power supply
Analog ground pin	AVSS	Input	Analog ground
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin3	AN3	Input	
Analog input pin 4	AN4	Input	Group1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin6	AN6	Input	
Analog input pin7	AN7	Input	
Analog power supply pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

22.1.4 Register Configuration

Table 22.2 summarizes the A/D converter's registers.

Table 22.2 A/D Converter Registers

Address	Name	Abbreviation	R/W	Initial Value
H'4000080	A/D data register A (high)	ADDRAH	R	H'00
H'4000082	A/D data register A (low)	ADDRAL	R	H'00
H'4000084	A/D data register B (high)	ADDRBH	R	H'00
H'4000086	A/D data register B (low)	ADDRBL	R	H'00
H'4000088	A/D data register C (high)	ADDRCH	R	H'00
H'400008A	A/D data register C (low)	ADDRCL	R	H'00
H'400008C	A/D data register D (high)	ADDRDH	R	H'00
H'400008E	A/D data register D (low)	ADDRDL	R	H'00
H'4000090	A/D control/status register	ADCSR	R/(W)*	H'00
H'4000092	A/D control register	ADCR	R/W	H'3F

Note: Only 0 can be written in bit 7, to clear the flag.

22.2 Register Descriptions

22.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

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

The four A/D data registers (ADDRA–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 5–0 of an A/D data register are reserved bits that always read 0. Table 22.3 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 22.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 22.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

22.2.2 A/D Control/Status Register (ADCSR)

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: Only 0 can be written, to clear the flag.

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—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description	
0	[Clearing conditions]	(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	
	Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode	

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: MULTI	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–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–AN2
		1	AN3	AN0–AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4–AN6
		1	AN7	AN4–AN7

22.2.3 A/D Control Register (ADCR)

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	—	—	—	—	—	—

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.

Bits 7 and 6—Trigger Enable (TRGE1, TRGE0): These bits enable or disable external triggering of A/D conversion.

The TRGE1 and TRGE0 bits should only be set when conversion is not in progress.

Bit 7: TRGE1	Bit 6: TRGE0	Description
0	0	A/D conversion is not started by external trigger input (initial value)
	1	
1	0	
	1	A/D conversion starts at the falling edge of an input signal from the external trigger pin ($\overline{\text{ADTRG}}$)

22.3 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 22.2 shows the data flow for access to an A/D data register.

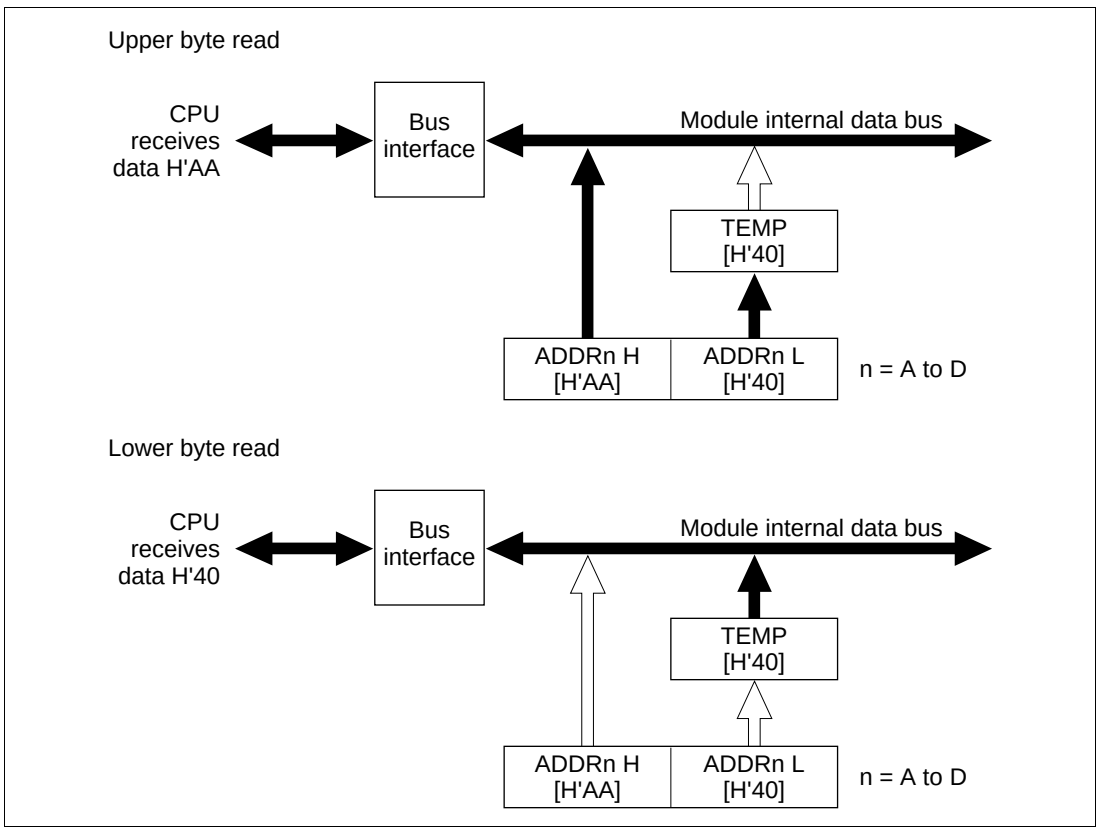


Figure 22.2 A/D Data Register Access Operation (Reading H'AA40)

22.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and multi mode.

22.4.1 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.

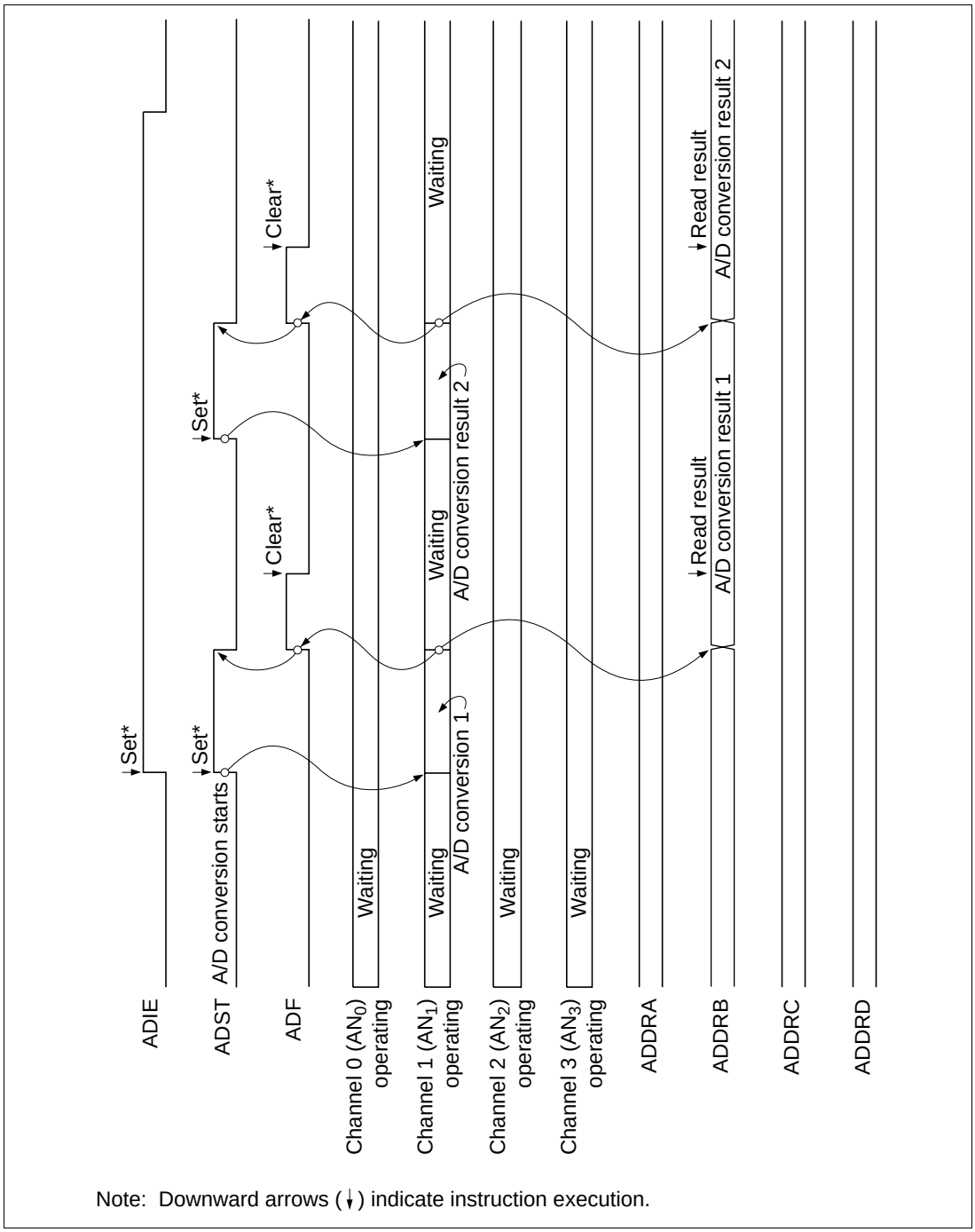
When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

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 22.3 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 22.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

22.4.2 Multi Mode (MULTI = 1)

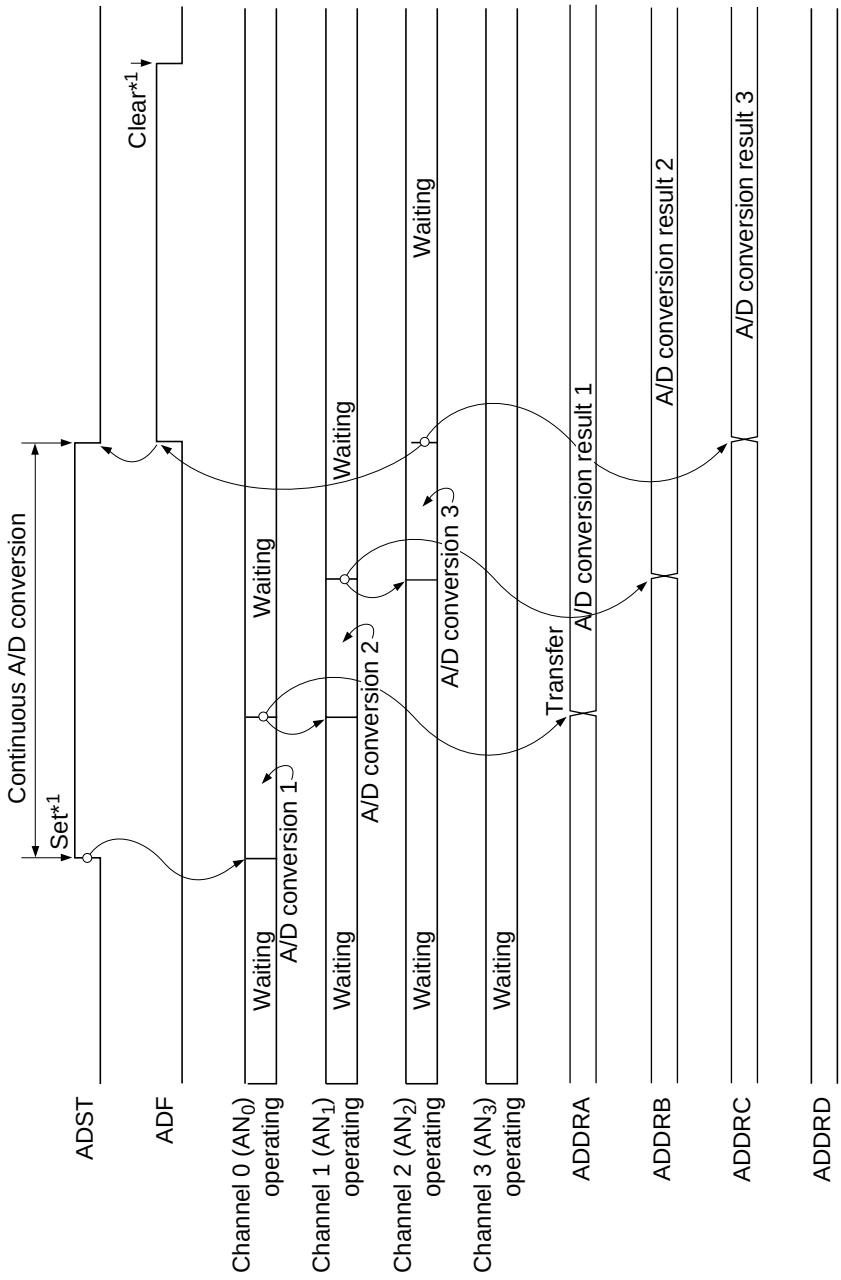
Multi mode should be selected for continuous A/D conversions on one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, AN4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1 or AN5) starts immediately. When A/D conversions end on the selected channels, the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed 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. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN0–AN2) are selected in scan mode are described next. Figure 22.4 shows a timing diagram for this example.

1. Multi mode is selected (MULTI = 1), channel group 0 is selected (CH2 = 0), analog input channels AN0–AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0–AN2) is completed, the ADF flag is set to 1 and the ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.

When A/D conversion stops the ADST bit is cleared to 0.



Notes: 1. Downward arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

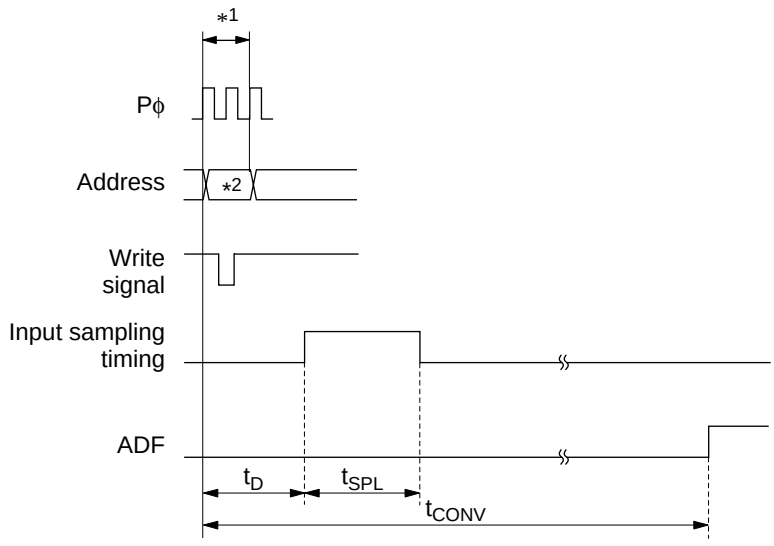
Figure 22.4 Example of A/D Converter Operation (Multi Mode, Channels AN0–AN2 Selected)

22.4.3 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 22.5 shows the A/D conversion timing. Table 22.4 indicates the A/D conversion time.

As indicated in figure 22.5, 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 22.4.

In multi mode, the values given in table 22.4 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 22.5 A/D Conversion Timing

Table 22.4 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.

22.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE1 and 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 multi modes, are the same as if the ADST bit had been set to 1 by software. Figure 22.6 shows the timing.

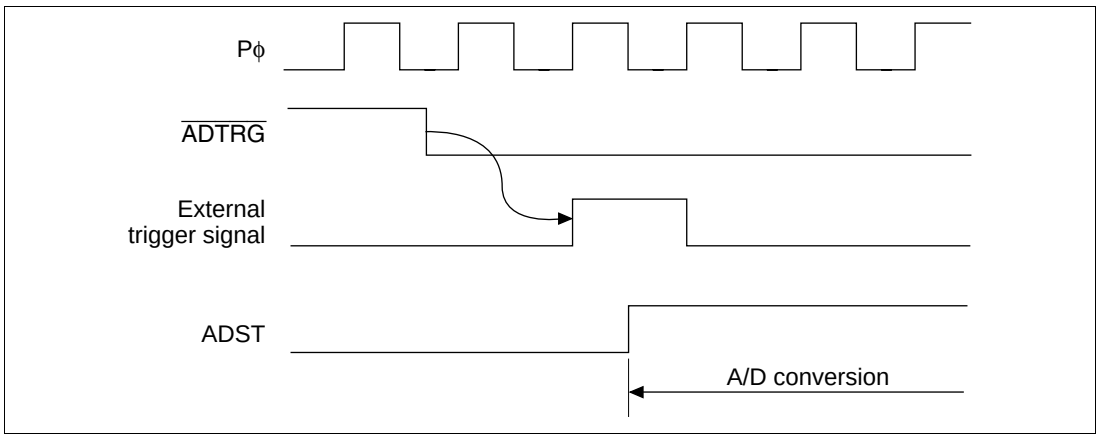


Figure 22.6 External Trigger Input Timing

22.5 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.

22.6 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 22.7. 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 22.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure) (figure 22.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 22.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 22.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

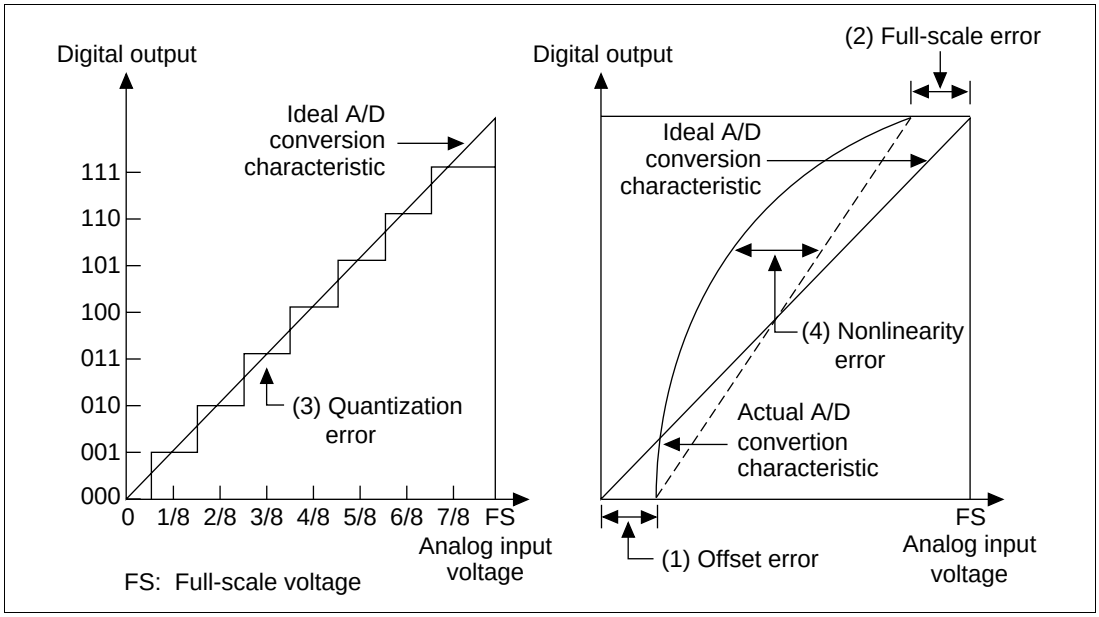


Figure 22.7 Definitions of A/D Conversion Accuracy

22.7 A/D Converter Usage Notes

When using the A/D converter, note the points listed in section 22.7.1 below.

22.7.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to analog input pins ANn should be in the range AV_{SS} ANn 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: $AV_{CC} = V_{CC} \pm 0.3$ V and $AV_{SS} = V_{SS}$.

22.7.2 Handling of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0–AN7), connect an input protection circuit such as that shown in figure 22.8. 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 22.5 list the analog input pin specifications and figure 22.9 shows an equivalent circuit diagram of the analog input ports.

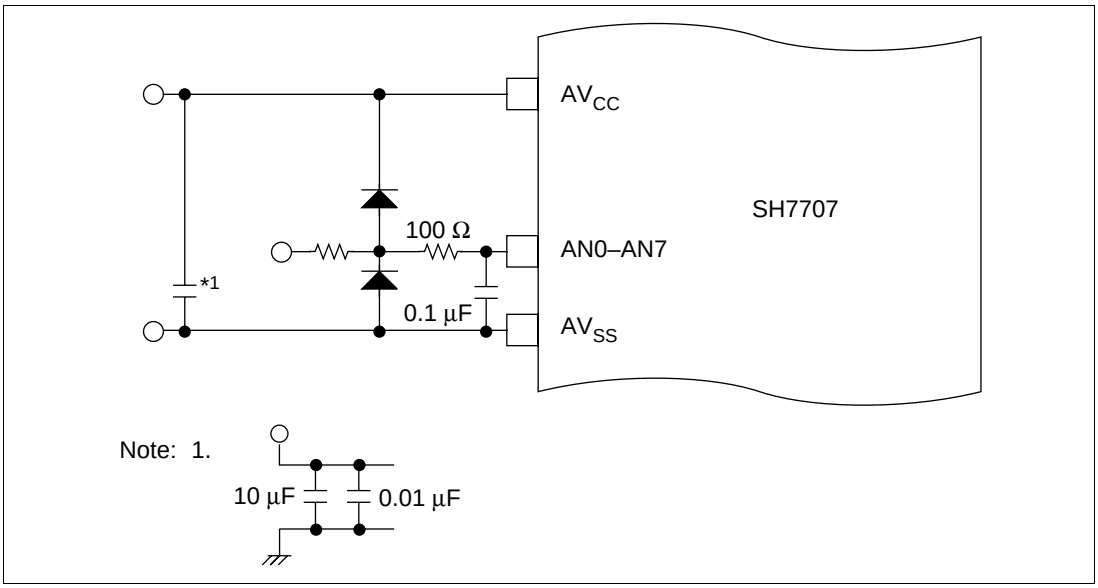


Figure 22.8 Example of Analog Input Protection Circuit

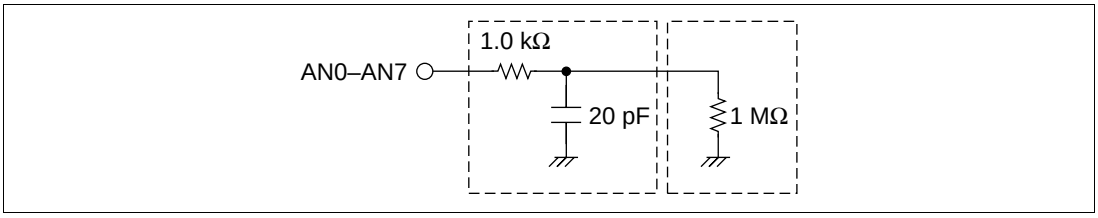


Figure 22.9 Analog Input Pin Equivalent Circuit

Table 22.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

Section 23 D/A Converter

23.1 Overview

The SH7707 includes a D/A converter with two channels.

23.1.1 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 to AVcc

23.1.2 Block Diagram

Figure 23.1 shows a block diagram of the D/A converter.

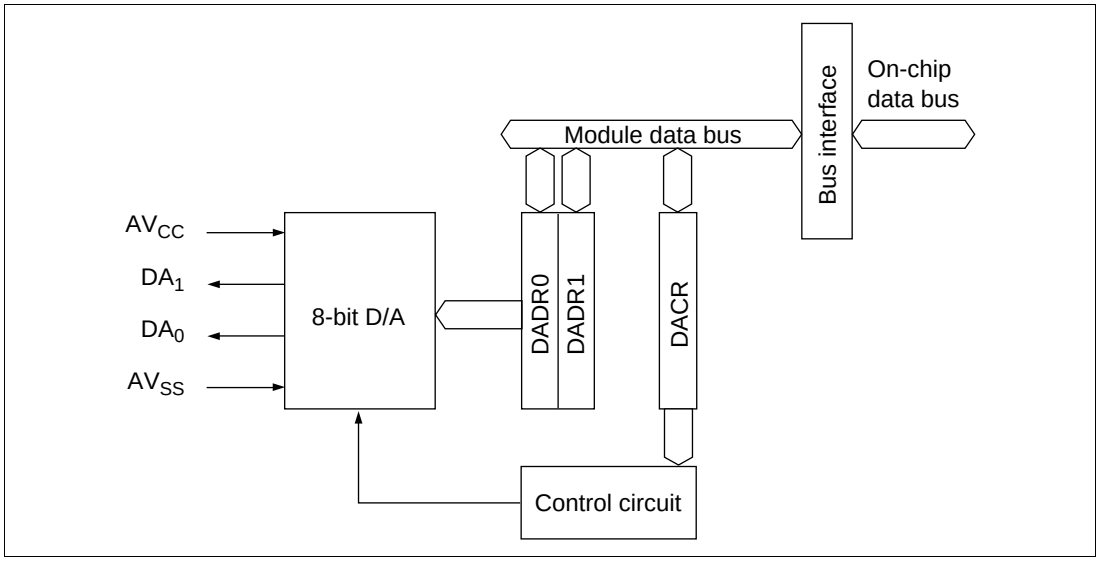


Figure 23.1 D/A Converter Block Diagram

23.1.3 Input/Output Pins

Table 23.1 summarizes the D/A converter's input and output pins.

Table 23.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVCC	Input	Analog power supply
Analog ground pin	AVSS	Input	Analog ground and reference voltage
Analog output pin 0	DA0	Output	Analog output, channel 0
Analog output pin 1	DA1	Output	Analog output, channel 1

23.1.4 Register Configuration

Table 23.2 summarizes the D/A converter's registers.

Table 23.2 D/A Converter Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'40000A0	D/A data register 0	DADR0	R/W	H'00
H'40000A2	D/A data register 1	DADR1	R/W	H'00
H'40000A4	D/A control register	DACR	R/W	H'1F

Note: Lower 16 bits of the address

23.2 Register Descriptions

23.2.1 D/A Data Registers 0 and 1 (DADR0/1)

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

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.

23.2.2 D/A Control Register (DACR)

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	—	—	—	—	—

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—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7: DAOE1	Description
0	DA1 analog output is disabled
1	Channel-1 D/A conversion and DA1 analog output are enabled

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6: DAOE0	Description
0	DA0 analog output is disabled
1	Channel-0 D/A conversion and DA0 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 the SH7707 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.

23.3 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.

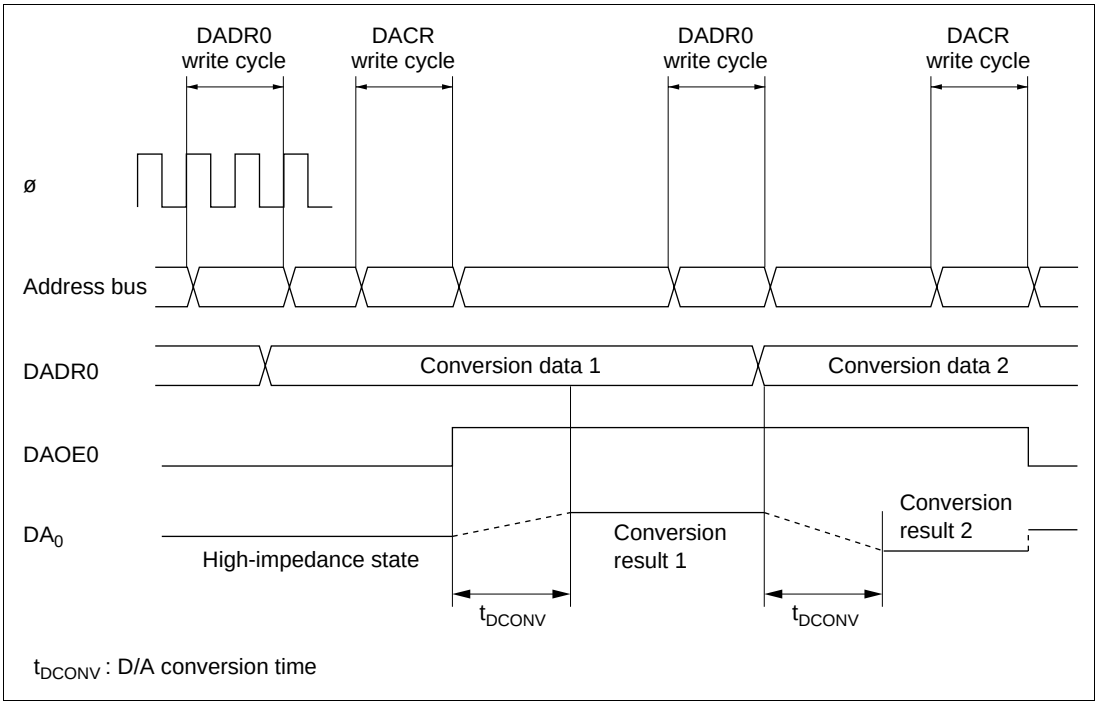


Figure 23.2 Example of D/A Converter Operation

Section 24 Electrical Characteristics

24.1 Absolute Maximum Ratings

Table 24.1 shows the absolute maximum ratings.

Table 24.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{CC}	–0.3 to 4.6	V
Input voltage (except port L)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port L)	V_{in}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to 4.6	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{str}	–55 to 125	°C

Caution: Operating the chip in excess of the absolute maximum rating may result in permanent damage.

24.2 DC Characteristics

Tables 24.2 and 24.3 list DC characteristics.

Table 24.2 DC Characteristics
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Power supply voltage		V_{CC}	3.0	3.3	3.6	V	During normal operation, sleep mode, standby mode
			2.0	3.3	3.6	V	RTC operating voltage in standby mode
Current dissipation	Normal operation	I_{CC}	—	—	195*[1]	mA	V_{CC} = 3.3 V
			—	—	100*[2]	mA	*1: 60 MHz (Pϕ: 30 MHz)
			—	—	50*[3]	mA	*2: 30 MHz
	In sleep mode		—	75*[1]	95*[1]	mA	*3: 15 MHz
			—	40*[2]	50*[2]	mA	
			—	20*[3]	25*[3]	mA	
	In standby mode		—	15	30	µA	Ta = 25°C (RTC on)
			—	—	400	µA	Ta > 50°C (RTC on)
			—	5	15	µA	Ta = 25°C (RTC off)
			—	—	300	µA	Ta > 50°C (RTCoff)
Input high voltage	$\overline{\text{RESET}}$, NMI	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	$\overline{\text{BREQ}}$, IRQ5–IRQ0, MD5–MD0		V_{CC} – 0.5	—	V_{CC} + 0.3	V	
	EXTAL, CKIO		V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	Port L		2.0	—	AV_{CC} + 0.3	V	
	Other input pins		2.0	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{\text{RESET}}$, NMI	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	$\overline{\text{BREQ}}$, IRQ5–IRQ0, MD5–MD0		–0.3	—	0.5	V	In standby mode
			–0.3	—	$V_{CC} \times 0.2$	V	Normal operation
	Port L		–0.3	—	$AV_{CC} \times 0.2$	V	
	Other input pins		–0.3	—	$V_{CC} \times 0.2$	V	

Table 24.2 DC Characteristics (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	All input pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Three-state leakage current	I/O, all output pins (off condition)	$\lvert I_{sti} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	V_{CC} = 3.0 V, I_{OH} = –200µA
			2.0	—	—	V	V_{CC} = 3.0 V, I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	V_{CC} = 3.6 V, I_{OL} = 1.6 mA
Pull-up resistance	Port pins	P_{pull}	30	60	120	KΩ	
Pin capacitance	All pins	C	—	—	20	PF	
Analog power supply voltage		AV_{CC}	3.0	3.3	3.6	V	
Analog power supply current	During A/D conversion	AI_{CC}	—	0.8	2	mA	
	During A/D and D/A conversion		—	2.4	6	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Regardless of whether the PLL or RTC is used, connect $PLLV_{CC}$ and $RTCV_{CC}$ to V_{CC}, and $PLLV_{SS}$ and $RTCV_{SS}$ to V_{SS}.
2. AV_{CC} conditions must be: $V_{CC} – 0.3V \le AV_{CC} \le V_{CC} + 0.3$ V. 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 AV_{SS} to V_{SS}.
3. Current dissipation values shown are the values at which all output pins are unloaded under the conditions $V_{IHmin} = V_{CC} – 0.5$ V, $V_{ILmax} = 0.5$ V.

Table 24.3 Permitted Output Current Values
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum(-I_{OH})$	—	—	40	mA

Caution: To ensure chip reliability, do not exceed the output current values given in table 24.3.

24.3 AC Characteristics

In general, input for the SH7707 should be synchronous. Observe the setup and hold times for each input signal unless otherwise specified.

Table 24.4 Maximum Operating Frequencies
(V_{CC} = 3.3V ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Notes
Operating frequency	CPU, cache, TLB	f	1	—	60	MHz	
	External bus		1	—	30		
	Supporting modules		0.25	—	30		

24.3.1 Clock Timing

Table 24.5 Clock Timing (1)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, Ta = –20 to 75°C, maximum external bus operating frequency: 15 MHz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	2	30	MHz	
EXTAL clock input cycle time	t_{EXcyc}	33.3	500	ns	
EXTAL clock input low pulse width	t_{EXL}	8*[1] or 12*[2]	—	ns	24.1
EXTAL clock input high pulse width	t_{EXH}	8*[1] or 12*[2]	—	ns	
EXTAL clock input rise time	t_{EXR}	—	4	ns	
EXTAL clock input fall time	t_{EXF}	—	4	ns	
CKIO clock input frequency	f_{CKI}	8	15	MHz	
CKIO clock input cycle time	t_{CKIcyc}	66.7	125	ns	
CKIO clock input low pulse width	t_{CKIL}	8	—	ns	24.2
CKIO clock input high pulse width	t_{CKIH}	8	—	ns	
CKIO clock input rise time	t_{CKIR}	—	4	ns	
CKIO clock input fall time	t_{CKIF}	—	4	ns	
CKIO clock output frequency	f_{OP}	1	15	MHz	
CKIO clock output cycle time	t_{cyc}	66.7	1000	ns	
CKIO clock output low pulse width	t_{CKOL}	20	—	ns	24.3
CKIO clock output high pulse width	t_{CKOH}	20	—	ns	
CKIO clock output rise time	t_{CKOR}	—	7	ns	
CKIO clock output fall time	t_{CKOF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	
$\overline{\text{BREQ}}$ setup time	$t_{RESBRQS}$	20	—	ns	
$\overline{\text{BREQ}}$ reset hold time 1	$t_{BREQRH1}$	0	—	ns	24.4
$\overline{\text{RESET}}$ setup time 1	t_{RESS1}	20	—	ns	
$\overline{\text{BREQ}}$ setup time 1	t_{BREQS1}	20	—	ns	
$\overline{\text{RESET}}$ assert time	t_{RESW}	20	—	tcyc	24.4, 24.5
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.7

Table 24.5 Clock Timing (1) (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, Ta = –20 to 75°C, maximum external bus operating frequency: 15 MHz)

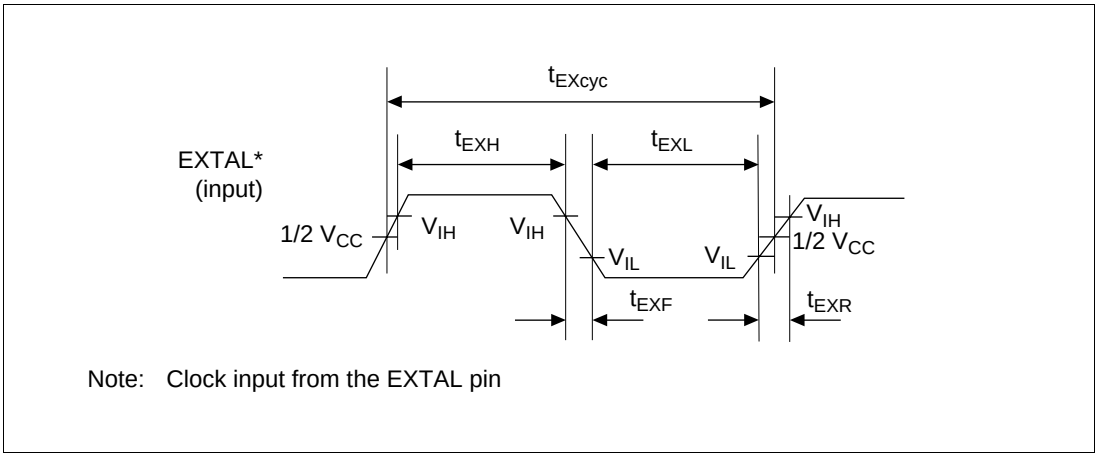
Item	Symbol	Min	Max	Unit	Figure
PLL synchronization settling time (PLL1, PLL2)	t_{PLL1}	100	—	us	24.8, 24.10
PLL synchronization settling time (PLL3)	t_{PLL2}	30	—	ms	24.9, 24.11
IRQ interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	us	24.10, 24.11

Notes: 1. With PLL circuit 1 operating.
2. With PLL circuit 2 not used.

Table 24.5 Clock Timing (2)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C, maximum external bus operating frequency: 30 MHz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	2	30	MHz	
EXTAL clock input cycle time	t_{EXcyc}	33.3	500	ns	
EXTAL clock input low pulse width	t_{EXL}	7*[1] or 10*[2]	—	ns	24.1
EXTAL clock input high pulse width	t_{EXH}	7*[1] or 10*[2]	—	ns	
EXTAL clock input rise time	t_{EXR}	—	4	ns	
EXTAL clock input fall time	t_{EXF}	—	4	ns	
CKIO clock input frequency	f_{CKI}	8	30	MHz	
CKIO clock input cycle time	t_{CKIcyc}	33.3	125	ns	
CKIO clock input low pulse width	t_{CKIL}	7	—	ns	24.2
CKIO clock input high pulse width	t_{CKIH}	7	—	ns	
CKIO clock input rise time	t_{CKIR}	—	3	ns	
CKIO clock input fall time	t_{CKIF}	—	3	ns	
CKIO clock output frequency	f_{OP}	1	30	MHz	
CKIO clock output cycle time	t_{cyc}	33.3	1000	ns	
CKIO clock output low pulse width	t_{CKOL}	8	—	ns	24.3
CKIO clock output high pulse width	t_{CKOH}	8	—	ns	
CKIO clock output rise time	t_{CKOR}	—	6	ns	
CKIO clock output fall time	t_{CKOF}	—	6	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	
$\overline{\text{BREQ}}$ setup time	$t_{RESBRQS}$	20	—	ns	
$\overline{\text{BREQ}}$ reset hold time 1	$t_{BREQRH1}$	0	—	ns	24.4
$\overline{\text{RESET}}$ setup time 1	t_{RESS1}	20	—	ns	
$\overline{\text{BREQ}}$ setup time 1	t_{BREQS1}	20	—	ns	
$\overline{\text{RESET}}$ assert time	t_{RESW}	20	—	tcyc	24.4, 24.5
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	24.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	24.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	24.7
PLL synchronization settling time 1	t_{PLL1}	100	—	us	24.8, 24.10
IRQ interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	us	24.10

Notes: 1. With PLL circuit 1 operating.
2. With PLL circuit 2 not used.



Note: Clock input from the EXTAL pin

Figure 24.1 EXTAL Clock Input Timing

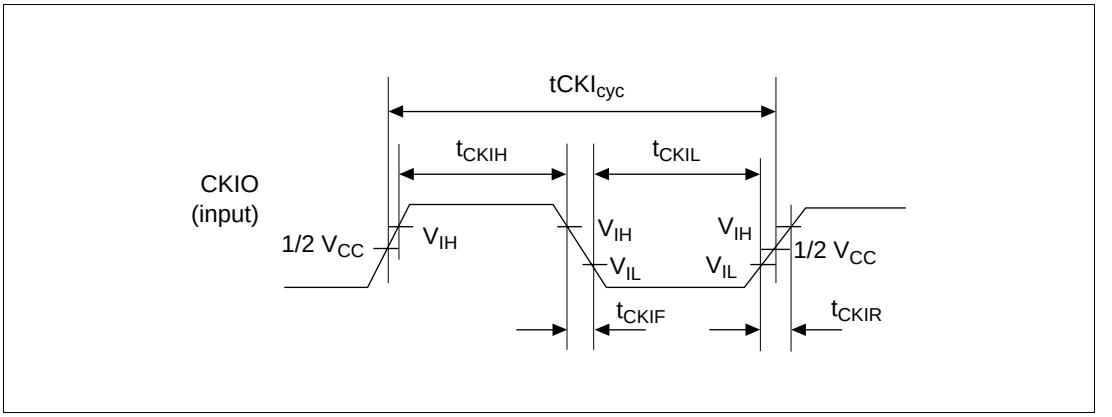


Figure 24.2 CKIO Clock Input Timing

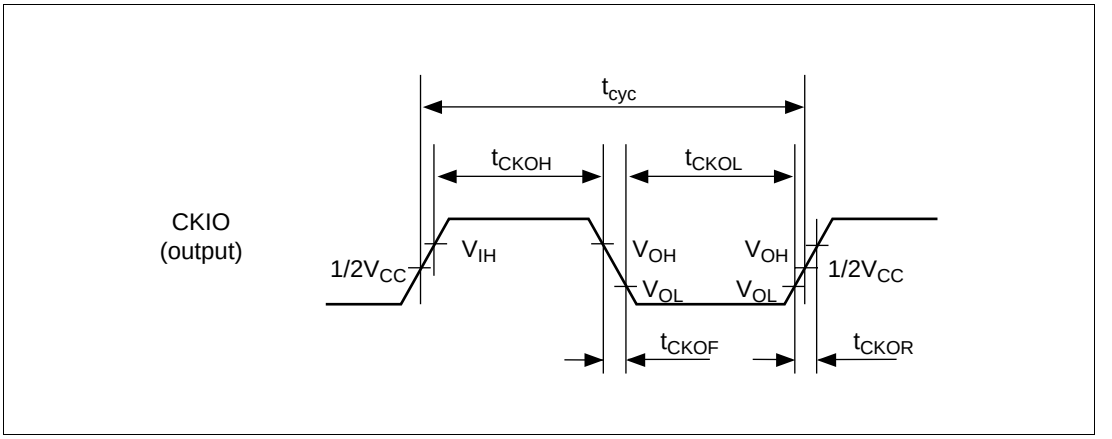
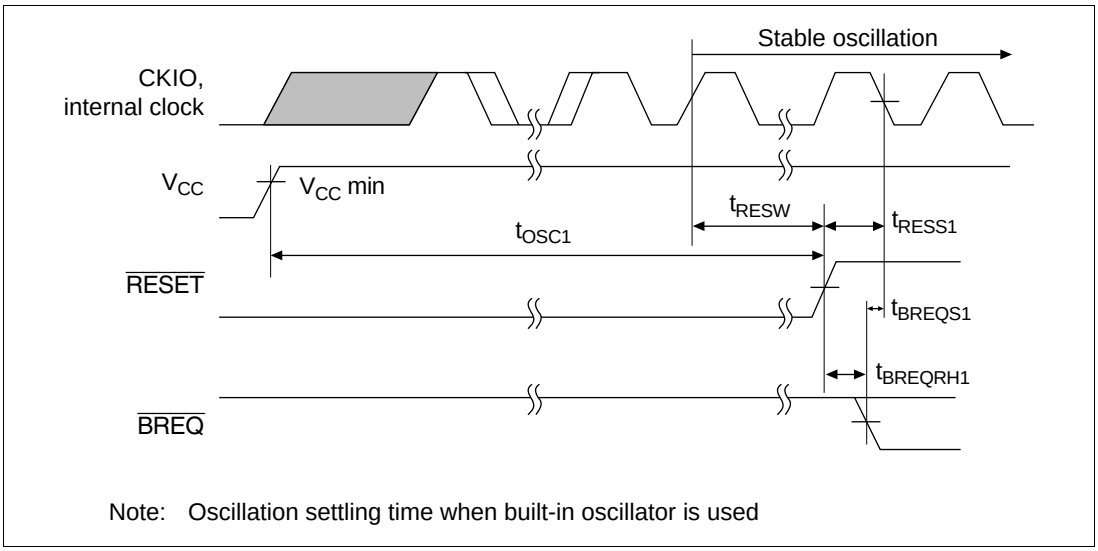
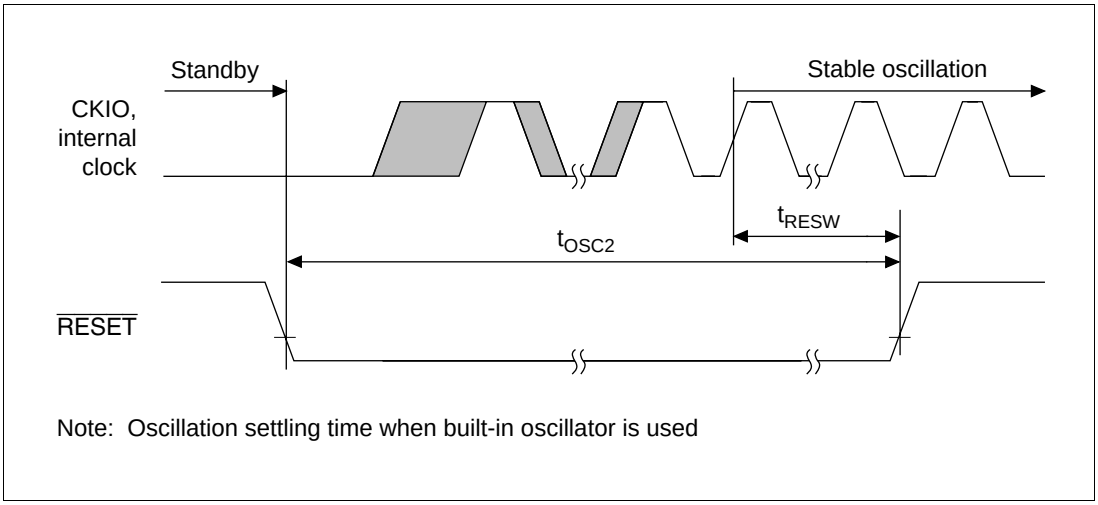


Figure 24.3 CKIO Clock Output Timing



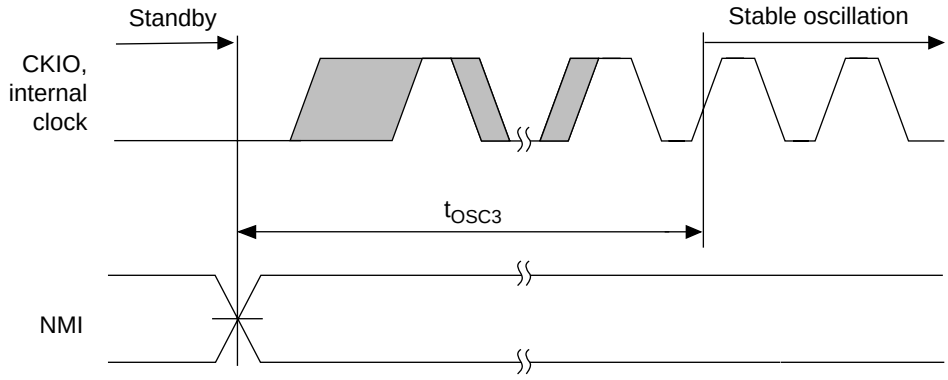
Note: Oscillation settling time when built-in oscillator is used

Figure 24.4 Power-on Oscillation Settling Time



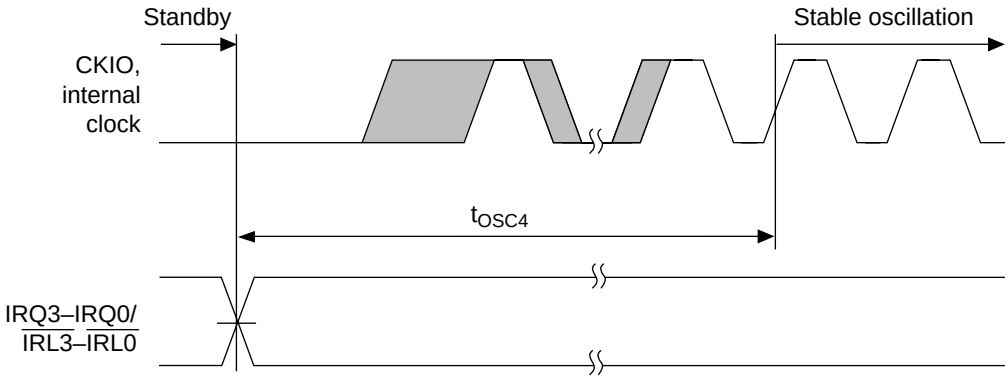
Note: Oscillation settling time when built-in oscillator is used

Figure 24.5 Oscillation Settling Time at Standby Return (Return by Reset)



Note: Oscillation settling time when built-in oscillator is used

Figure 24.6 Oscillation Settling Time at Standby Return (Return by NMI)



Note: Oscillation settling time when built-in oscillator is used

Figure 24.7 Oscillation Settling Time at Standby Return (Return by $\overline{IRQ3}$–$\overline{IRQ0}$)

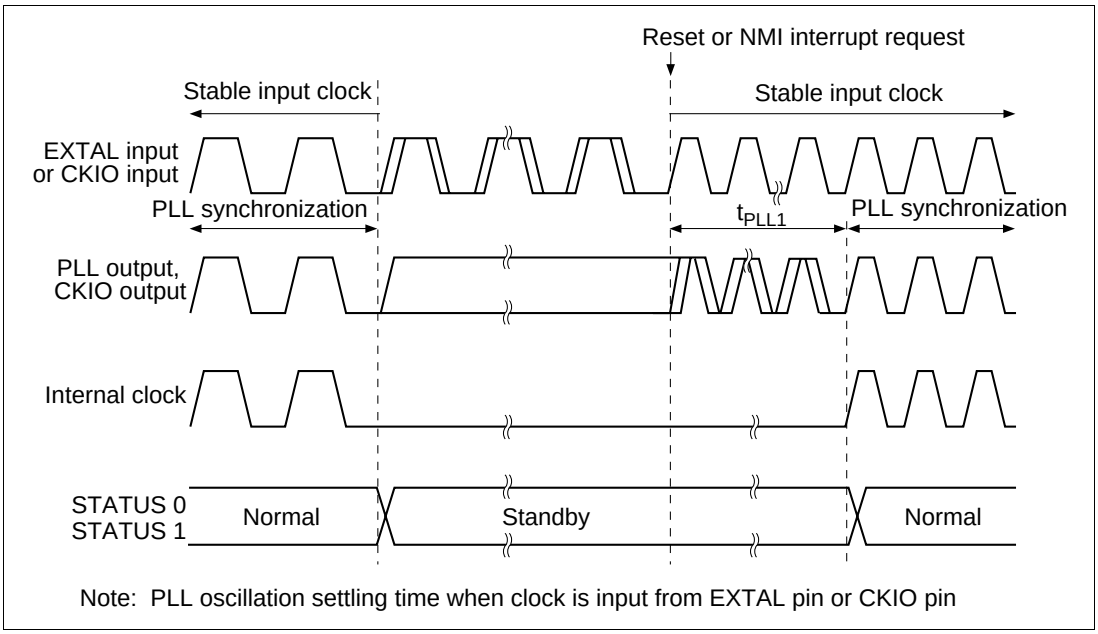


Figure 24.8 PLL Synchronization Settling Time by Reset or NMI

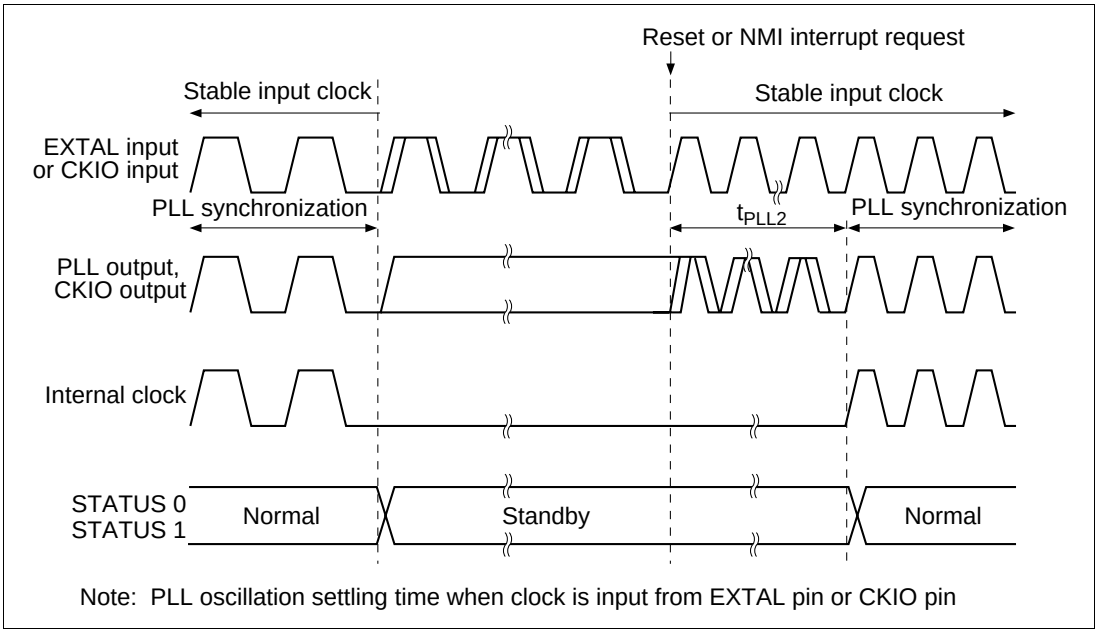


Figure 24.9 PLL Synchronization Settling Time by Reset or NMI

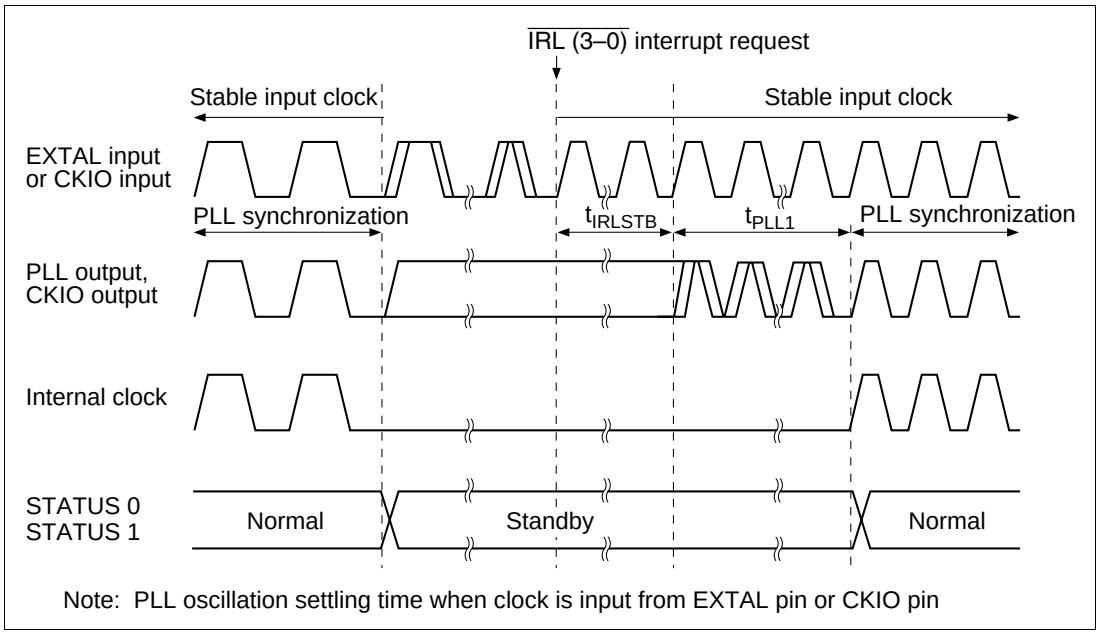


Figure 24.10 PLL Synchronization Settling Time by IRQ Interrupt

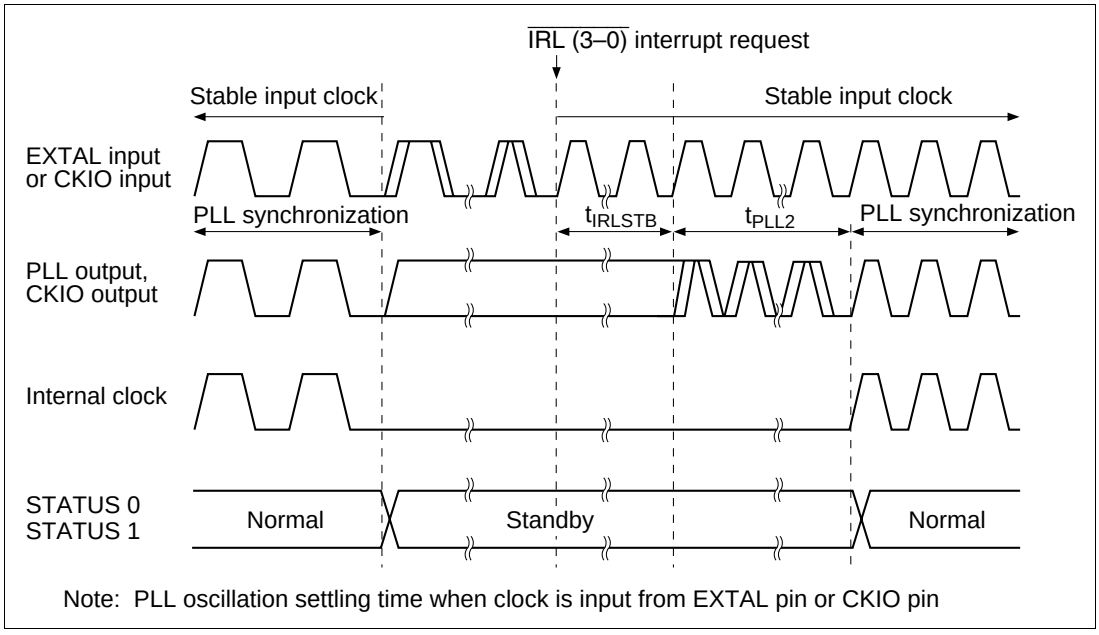


Figure 24.11 PLL Synchronization Settling Time by IRQ Interrupt

24.3.2 Control Signal Timing

Table 24.6 Control Signal Timing
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, Ta = –20 to 75°C)

		–15*[2]		–30*[3]			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RESET}}$ pulse width	t_{RESW}	20	—	20	—	tcyc	
$\overline{\text{RESET}}$ setup time 2*[1]	t_{RESS2}	23	—	23	—	ns	
$\overline{\text{RESET}}$ hold time	t_{RESH}	2	—	2	—	ns	24.12
$\overline{\text{BREQ}}$ setup time 2	t_{BREQS2}	12	—	12	—	ns	24.13
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—	3	—	ns	24.15
$\overline{\text{BREQ}}$ reset setup time	t_{BREQRS}	67	—	34	—	ns	
$\overline{\text{BREQ}}$ reset hold time 2	$t_{BREQRH2}$	16	—	16	—	ns	
NMI setup time*[1]	t_{NMIS}	12	—	12	—	ns	
IRQ5–IRQ0 setup time*[1]	t_{IRQS}	10	—	10	—	ns	24.13
NMI hold time	t_{NMIH}	4	—	4	—	ns	24.14
IRQ5–IRQ0 hold time	t_{IRQH}	4	—	4	—	ns	
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	17	—	14	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	17	—	14	ns	
STATUS1, STATUS0 delay time	t_{STD}	—	20	—	18	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	66	0	33	ns	24.15
Bus tri-state delay time 2	t_{BOFF2}	0	66	0	33	ns	24.16
Bus buffer on time 1	t_{BON1}	0	66	0	33	ns	
Bus buffer on time 2	t_{BON2}	0	66	0	33	ns	

Notes: 1. $\overline{\text{RESET}}$, NMI, and IRQ5–RQ0 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 15 MHz.
3. The upper limit of the external bus clock is 30 MHz.

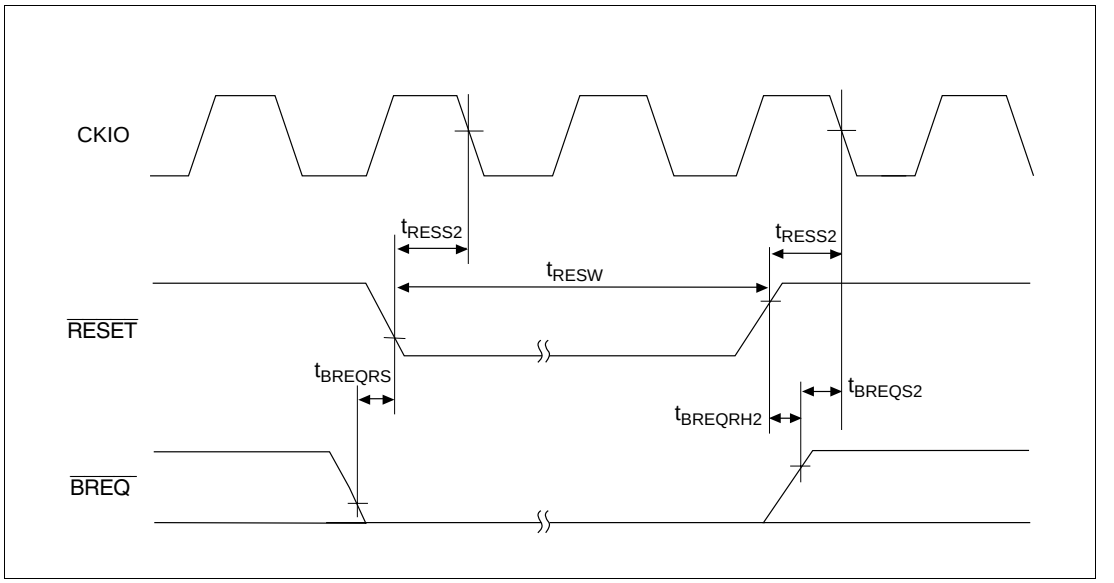


Figure 24.12 Manual Reset Input Timing

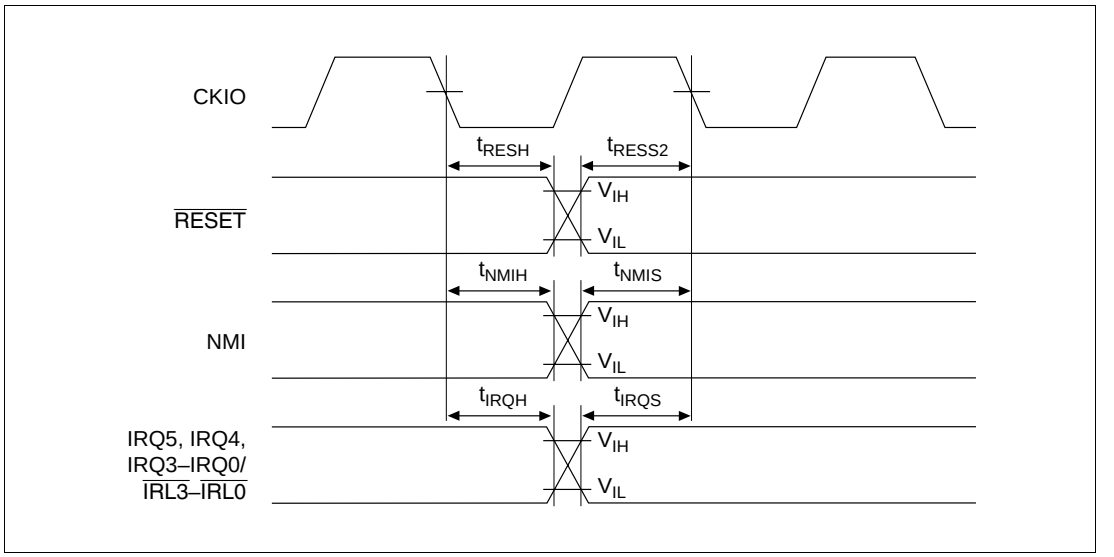


Figure 24.13 Interrupt Signal Input Timing

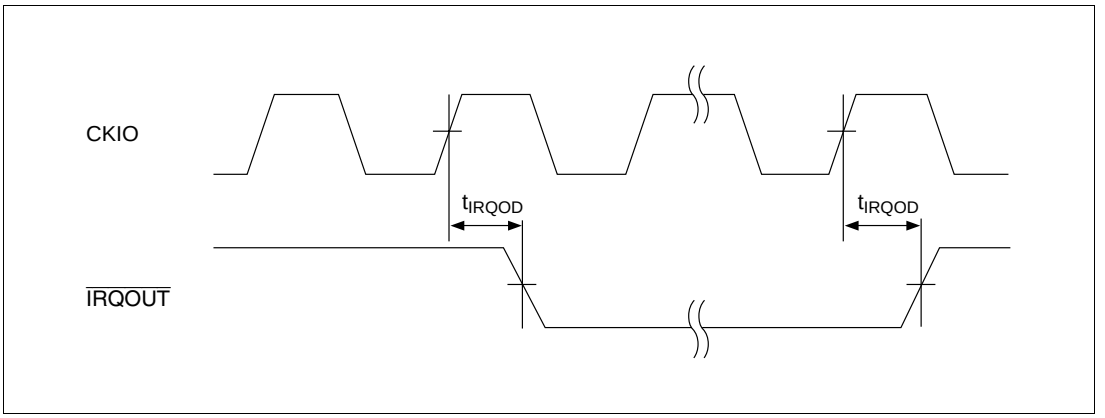


Figure 24.14 $\overline{\text{IRQOUT}}$ Timing

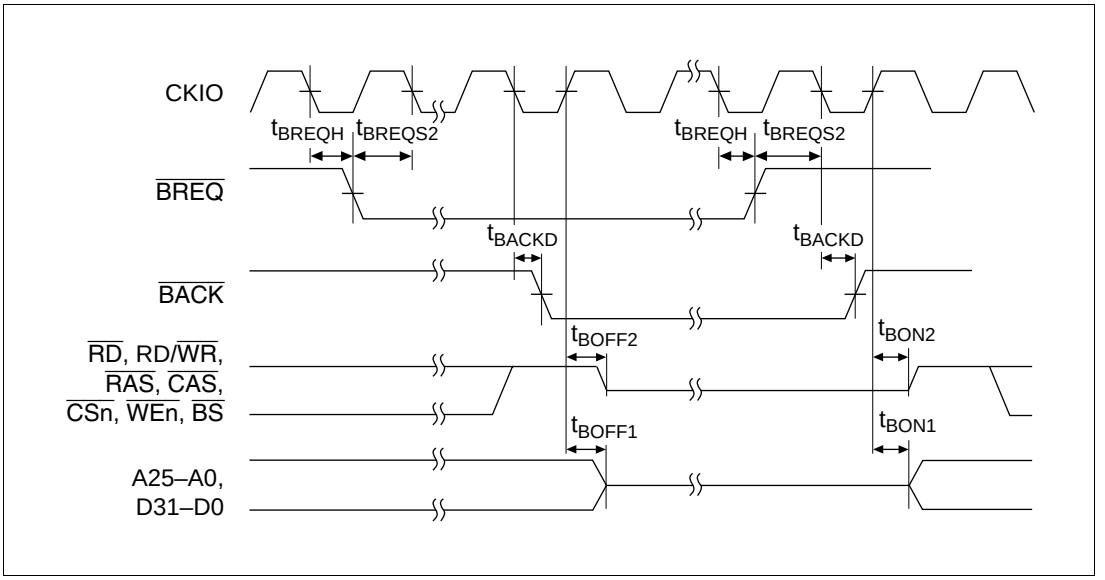


Figure 24.15 Bus Release Timing

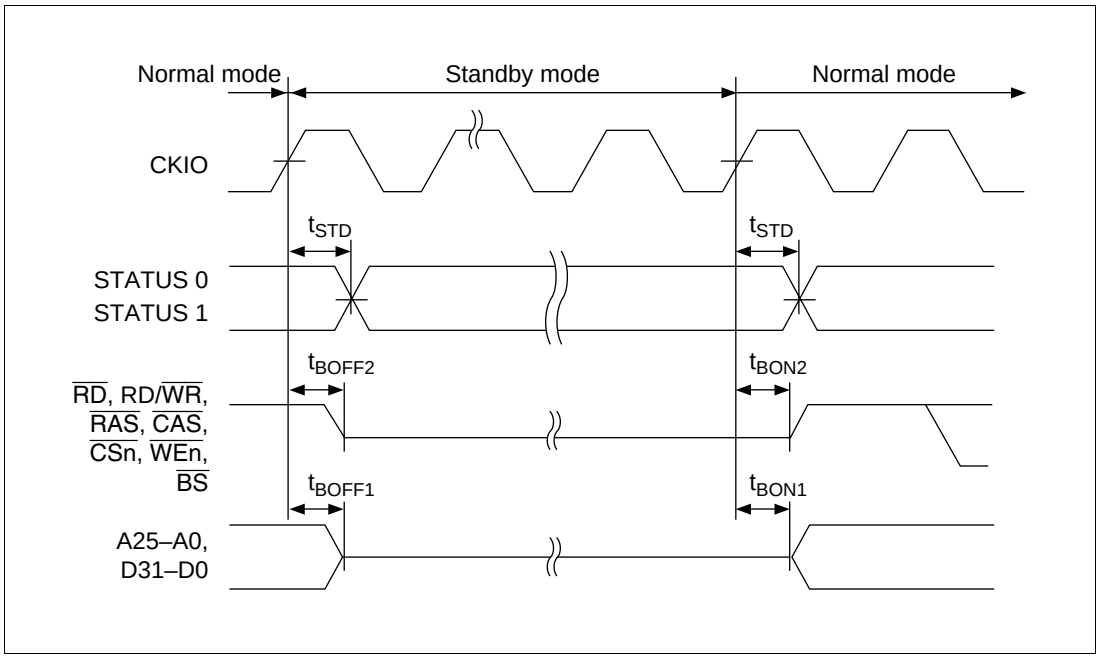


Figure 24.16 Pin Drive Timing at Standby

24.3.3 AC Bus Timing Specifications

Table 24.7 Bus Timing
(Clock Modes 0/1/2, V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3V, T_a = –20 to 75°C)

		–15*[1]		–30*[2]			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
Address delay time	t_{AD}	—	17	—	15	ns	24.17–24.34, 24.38–24.44
Address setup time	t_{AS}	0	—	0	—	ns	24.23–24.34
Address hold time	t_{AH}	20	—	10	—	ns	24.17–24.34
$\overline{BS}$ delay time	t_{BSD}	—	16	—	14	ns	24.17–24.34, 24.38–24.44
$\overline{CS}$ delay time 1	t_{CSD1}	—	16	—	14	ns	24.17–24.34, 24.38–24.44
$\overline{CS}$ delay time 1	t_{CSD2}	—	17	—	14	ns	24.17–24.22
Read write delay time	t_{RWD}	—	16	—	14	ns	24.17–24.34, 24.38–24.44
Read write hold time	t_{RWH}	0	—	0	—	ns	24.17–24.34
Read strobe delay time	t_{RSD}	—	17	—	14	ns	24.17–24.22, 24.38–24.41
Read data setup time 1	t_{RDS1}	12	—	12	—	ns	24.17–24.34, 24.38–24.44
Read data setup time 2	t_{RDS2}	12	—	12	—	ns	24.27–24.30
Read data setup time 3	t_{RDS3}	12	—	12	—	ns	24.31, 24.32
Read data setup time 4	t_{RDS4}	12	—	12	—	ns	24.33, 24.34
Read data hold time 1	t_{RDH1}	0	—	0	—	ns	24.17–24.26, 24.38–24.44
Read data hold time 2	t_{RDH2}	8	—	6	—	ns	24.27–24.30
Read data hold time 3	t_{RDH3}	0	—	0	—	ns	24.31, 24.32
Read data hold time 4	t_{RDH4}	8	—	6	—	ns	24.33, 24.34
Write enable delay time	t_{WED}	—	17	—	14	ns	24.17–24.19, 24.38, 24.39
Write data delay time 1	t_{WDD1}	—	18	—	17	ns	24.17–24.19, 24.38, 24.39, 24.42–24.44
Write data delay time 2	t_{WDD2}	—	16	—	16	ns	24.23–24.34, 24.44

Table 24.7 Bus Timing (cont)
(Clock Modes 0/1/2, V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3V, T_a = –20 to 75°C)

		–15*[1]		–30*[2]			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
Write data setup time	t_{WDS}	0	—	0	—	ns	24.23–24.34
Write data hold time 1	t_{WDH1}	0	—	0	—	ns	24.17–24.19, 24.23–24.34, 24.38–24.44
Write data hold time 3	t_{WDH3}	0	—	0	—	ns	24.17–24.19, 24.23–24.30, 24.44
Write data hold time 4	t_{WDH4}	0	—	0	—	ns	24.38, 24.39, 24.42–24.44
$\overline{\text{WAIT}}$ setup time	t_{WTS}	12	—	12	—	ns	24.18–24.22, 24.39, 24.41, 24.43, 24.44
$\overline{\text{WAIT}}$ hold time	t_{WTH}	8	—	6	—	ns	24.18–24.22, 24.39, 24.41, 24.43, 24.44
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	—	17	—	15	ns	24.23–24.37
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	—	16	—	15	ns	24.27–24.32
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	17	—	15	ns	24.23–24.30
$\overline{\text{CAS}}$ delay time 3	t_{CASD3}	—	17	—	15	ns	24.31–24.34
$\overline{\text{CAS}}$ delay time 4	t_{CASD4}	—	17	—	15	ns	24.31–24.34
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	17	—	14	ns	24.42–24.44
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	17	—	14	ns	24.42–24.44
PCC0WP setup time	t_{IO16S}	12	—	12	—	ns	24.43, 24.44
PCC0WP hold time	t_{IO16H}	8	—	6	—	ns	24.43, 24.44
$\overline{\text{DACK}}$ delay time 1	t_{DAKD1}	—	17	—	14	ns	24.17–24.34, 24.38–24.44
$\overline{\text{DACK}}$ delay time 2	t_{DAKD2}	—	16	—	15	ns	24.28, 24.31–24.34

Notes: 1. The upper limit of the external bus clock is 15 MHz.
2. The upper limit of the external bus clock is 30 MHz.

Table 24.8 Bus Timing
(Clock Modes 3/4/5/6/7, V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	Min	Max	Unit	Figures
Address delay time	t_{AD}	—	20	ns	24.17–24.34, 24.38–24.44
Address setup time	t_{AS}	0	—	ns	24.23–24.34
Address hold time	t_{AH}	20	—	ns	24.17–24.34
$\overline{BS}$ delay time	t_{BSD}	—	19	ns	24.17–24.34, 24.38–24.44
$\overline{CS}$ delay time 1	t_{CSD1}	—	19	ns	24.17–24.34, 24.38–24.44
$\overline{CS}$ delay time 2	t_{CSD2}	—	20	ns	24.17–24.22
Read write delay time	t_{RWD}	—	19	ns	24.17–24.34, 24.38–24.44
Read write hold time	t_{RWH}	0	—	ns	24.17–24.30
Read strobe delay time	t_{RSD}	—	20	ns	24.17–24.22, 24.38–24.41
Read data setup time 1	t_{RDS1}	12	—	ns	24.17–24.26, 24.38–24.44
Read data setup time 2	t_{RDS2}	12	—	ns	24.27–24.30
Read data setup time 3	t_{RDS3}	12	—	ns	24.31, 34.32
Read data setup time 4	t_{RDS4}	12	—	ns	24.33, 24.34
Read data hold time 1	t_{RDH1}	0	—	ns	24.17–24.26, 24.38–24.44
Read data hold time 2	t_{RDH2}	8	—	ns	24.27–24.30
Read data hold time 3	t_{RDH3}	0	—	ns	24.31, 34.32
Read data hold time 4	t_{RDH4}	8	—	ns	24.33, 24.34
Write enable delay time	t_{WED}	—	20	ns	24.17–24.19, 24.38, 24.39
Write data delay time 1	t_{WDD1}	—	25	ns	24.17–24.19, 24.38, 24.39, 24.42–24.44
Write data delay time 2	t_{WDD2}	—	19	ns	24.23–24.34, 24.44
Write data setup time	t_{WDS}	0	—	ns	24.23–24.30
Write data hold time 1	t_{WDH1}	0	—	ns	24.17–24.19, 24.23–24.34, 24.38–24.44
Write data hold time 3	t_{WDH3}	0	—	ns	24.17–24.19, 24.23–24.30, 24.44
Write data hold time 4	t_{WDH4}	0	—	ns	24.38, 24.39, 24.42–24.44
$\overline{WAIT}$ setup time	t_{WTS}	12	—	ns	24.18–24.22, 24.39, 24.41, 24.43, 24.44
$\overline{WAIT}$ hold time	t_{WTH}	8	—	ns	24.18–24.22, 24.39, 24.41, 24.43, 24.44

Table 24.7 Bus Timing (cont)
(Clock Modes 3/4/5/6/7, V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	Min	Max	Unit	Figures
$\overline{RAS}$ delay time 1	t_{RASD1}	—	20	ns	24.23–24.37
$\overline{RAS}$ delay time 2	t_{RASD2}	—	19	ns	24.27–24.32
$\overline{CAS}$ delay time 1	t_{CASD1}	—	20	ns	24.23–24.30
$\overline{CAS}$ delay time 3	t_{CASD3}	—	20	ns	24.31–24.34
$\overline{CAS}$ delay time 4	t_{CASD4}	—	20	ns	24.31–24.34
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	20	ns	24.42–24.44
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	20	ns	24.42–24.44
PCC0WP setup time	t_{IO16S}	12	—	ns	24.43, 24.44
PCC0WP hold time	t_{IO16H}	8	—	ns	24.43, 24.44
$\overline{DACK}$ delay time 1	t_{DAKD1}	—	20	ns	24.17–24.34, 24.38–24.44
$\overline{DACK}$ delay time 2	t_{DAKD2}	—	19	ns	24.28, 24.31–24.34

24.3.4 Basic Timing

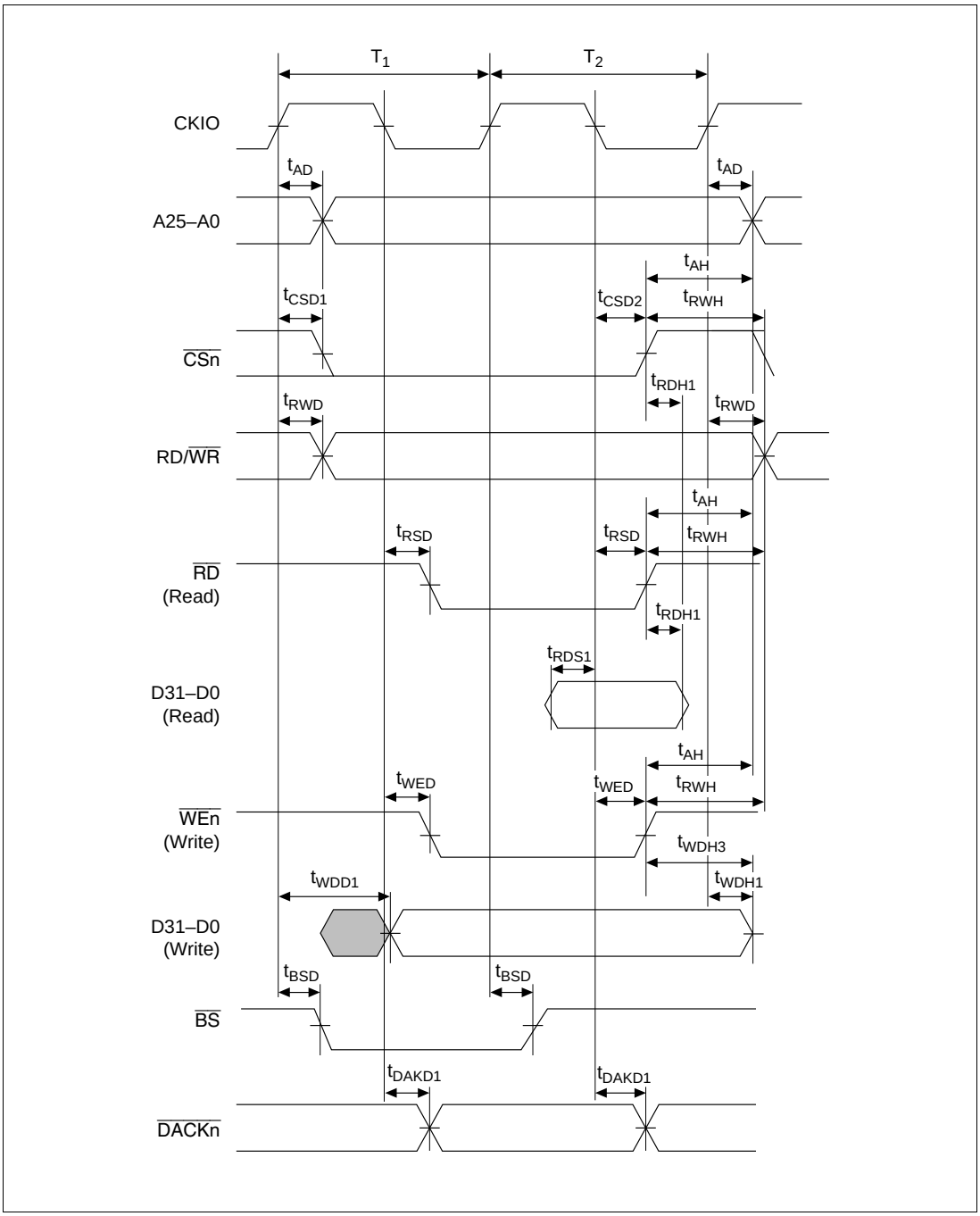


Figure 24.17 Basic Bus Cycle (No Wait)

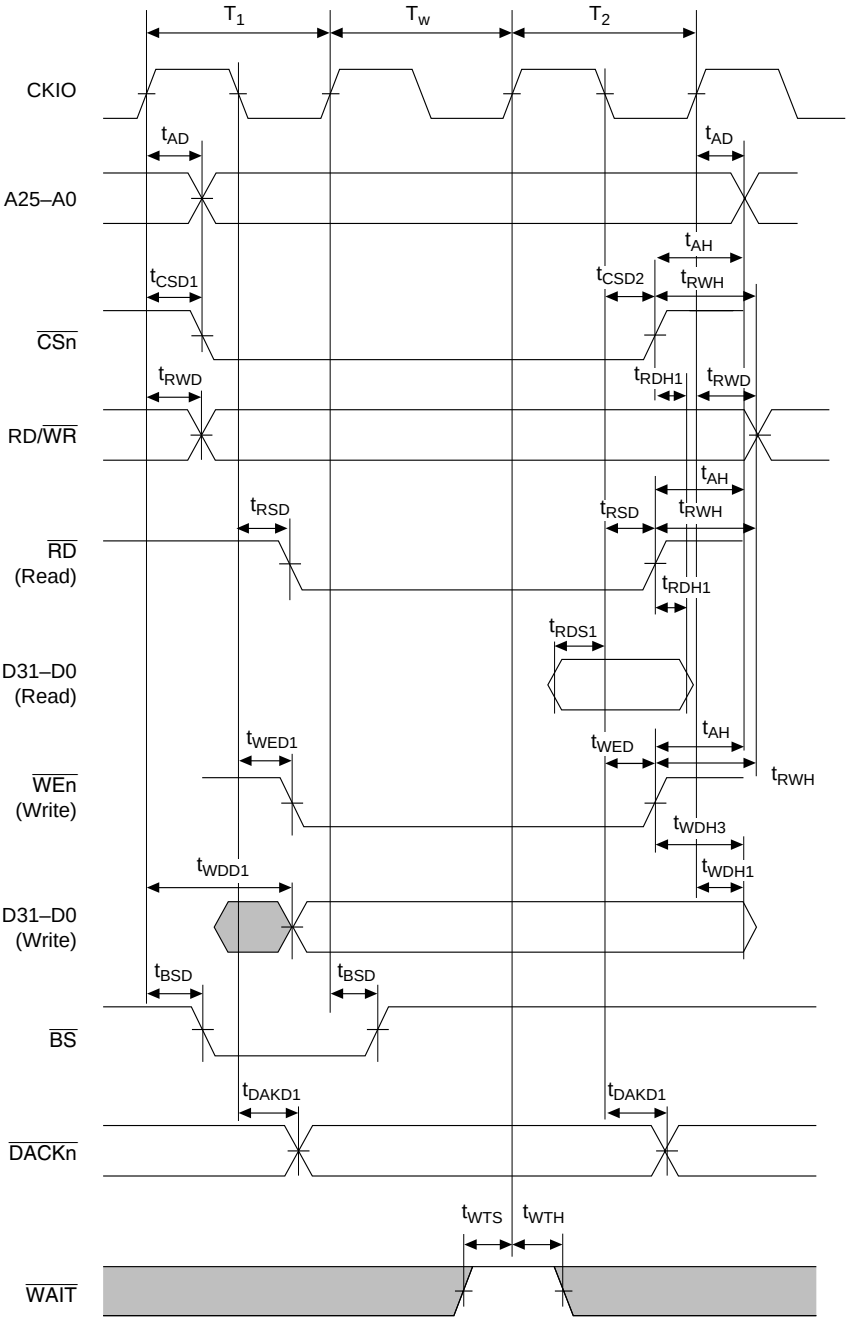


Figure 24.18 Basic Bus Cycle (One Wait)

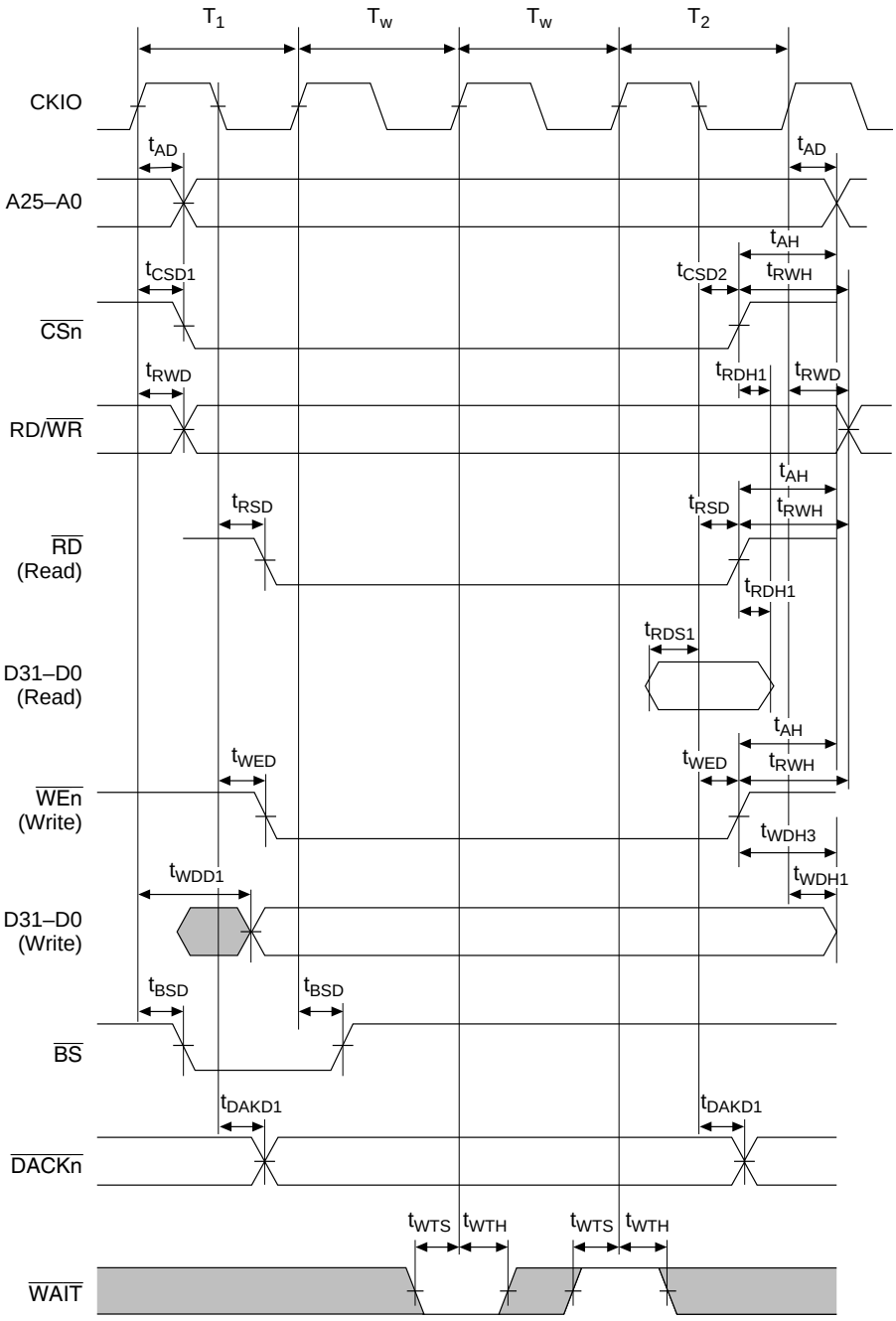
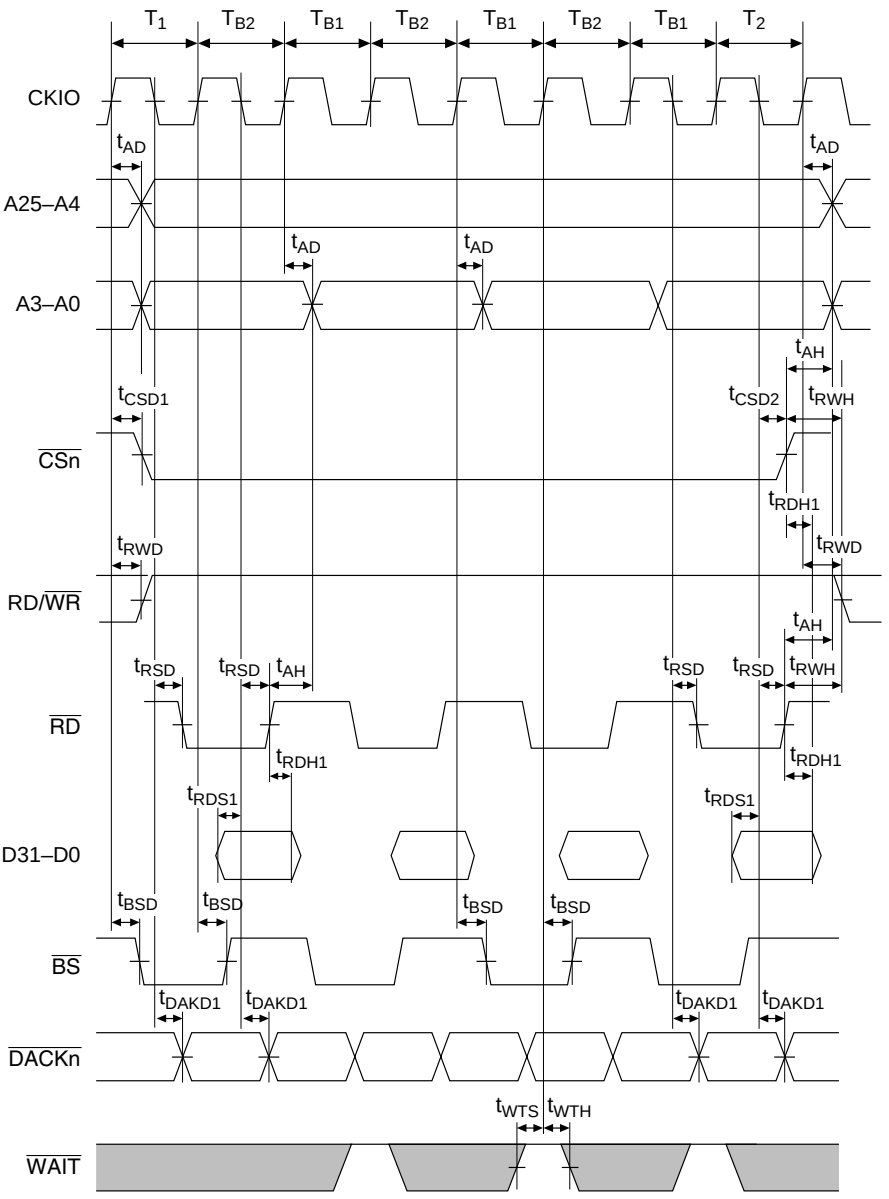


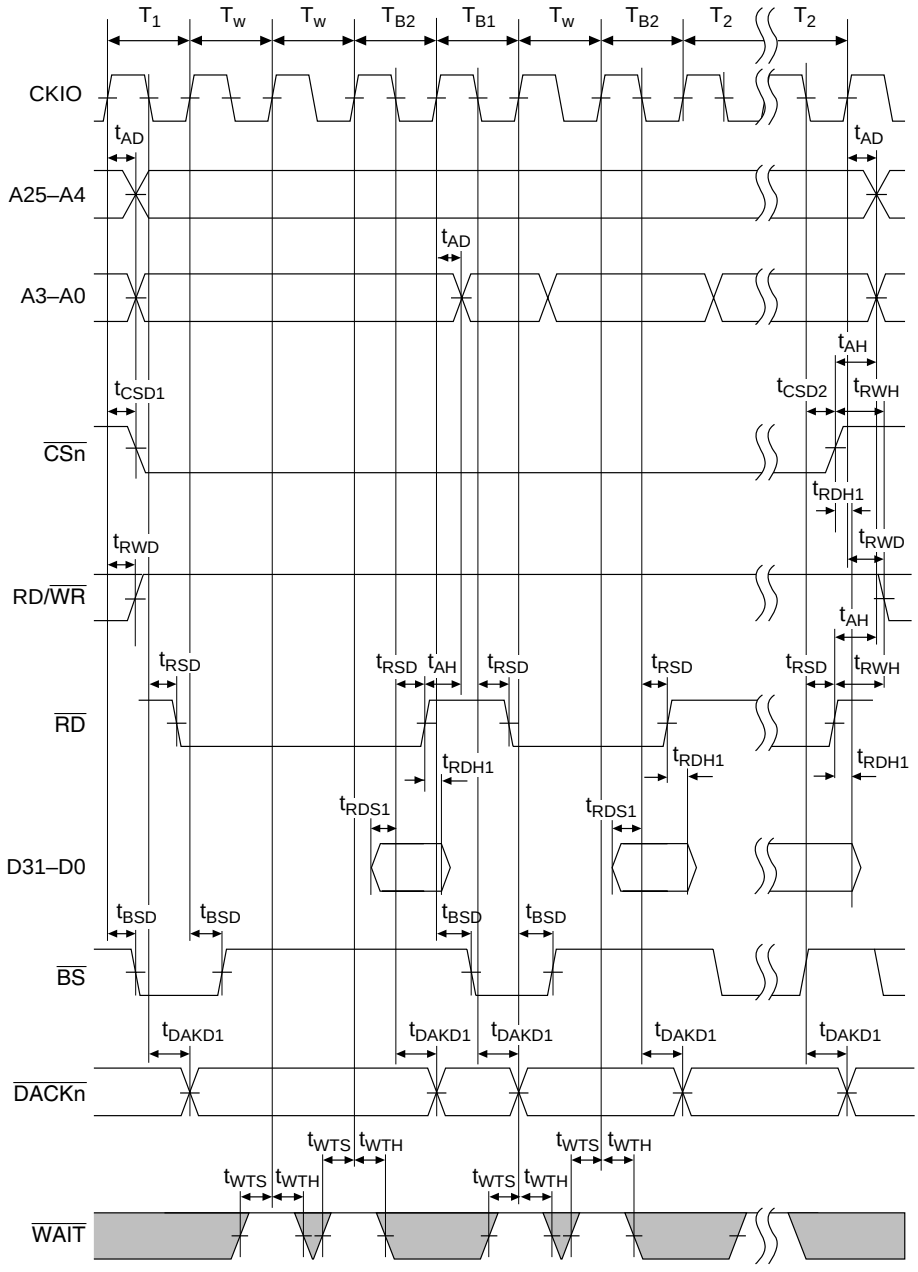
Figure 24.19 Basic Bus Cycle (External Wait)

24.3.5 Burst ROM Timing



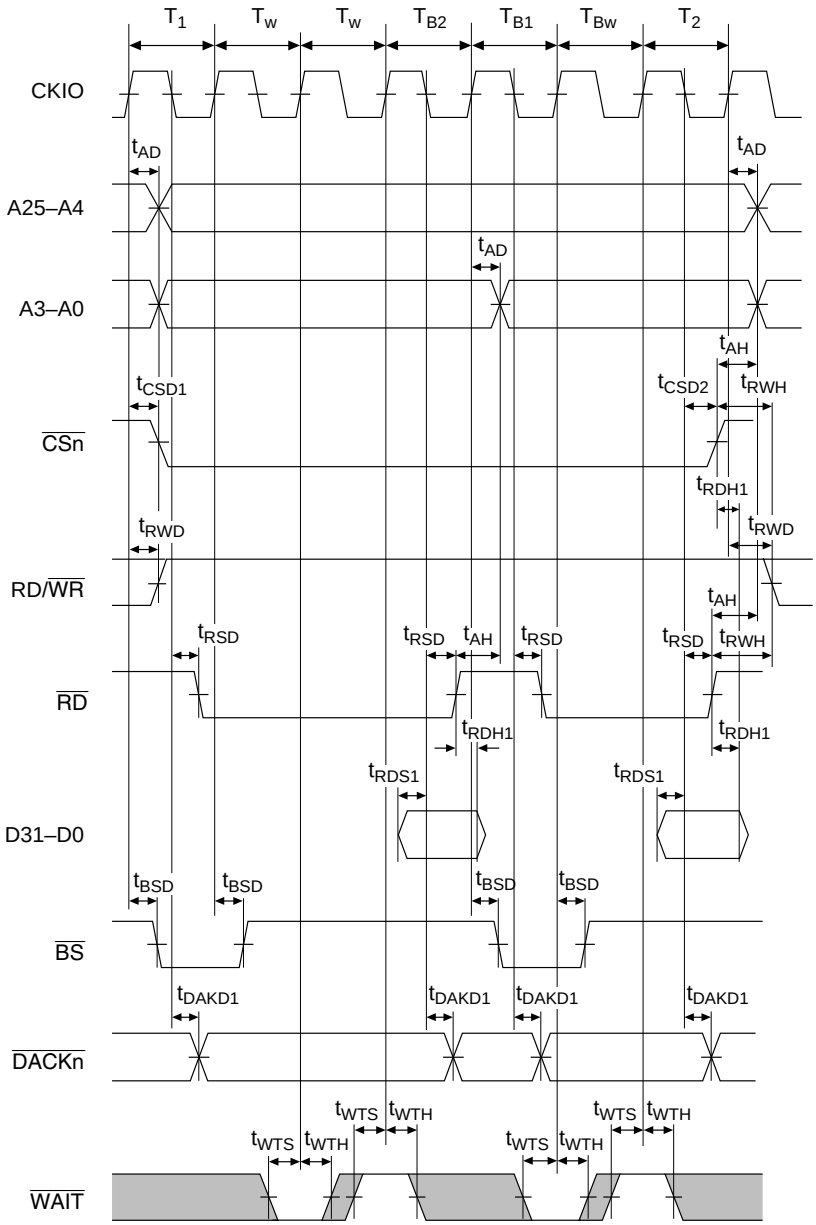
Note: In the write cycle, the basic bus cycle is performed.

Figure 24.20 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 24.21 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 24.22 Burst ROM Bus Cycle (External Wait)

24.3.6 DRAM Timing

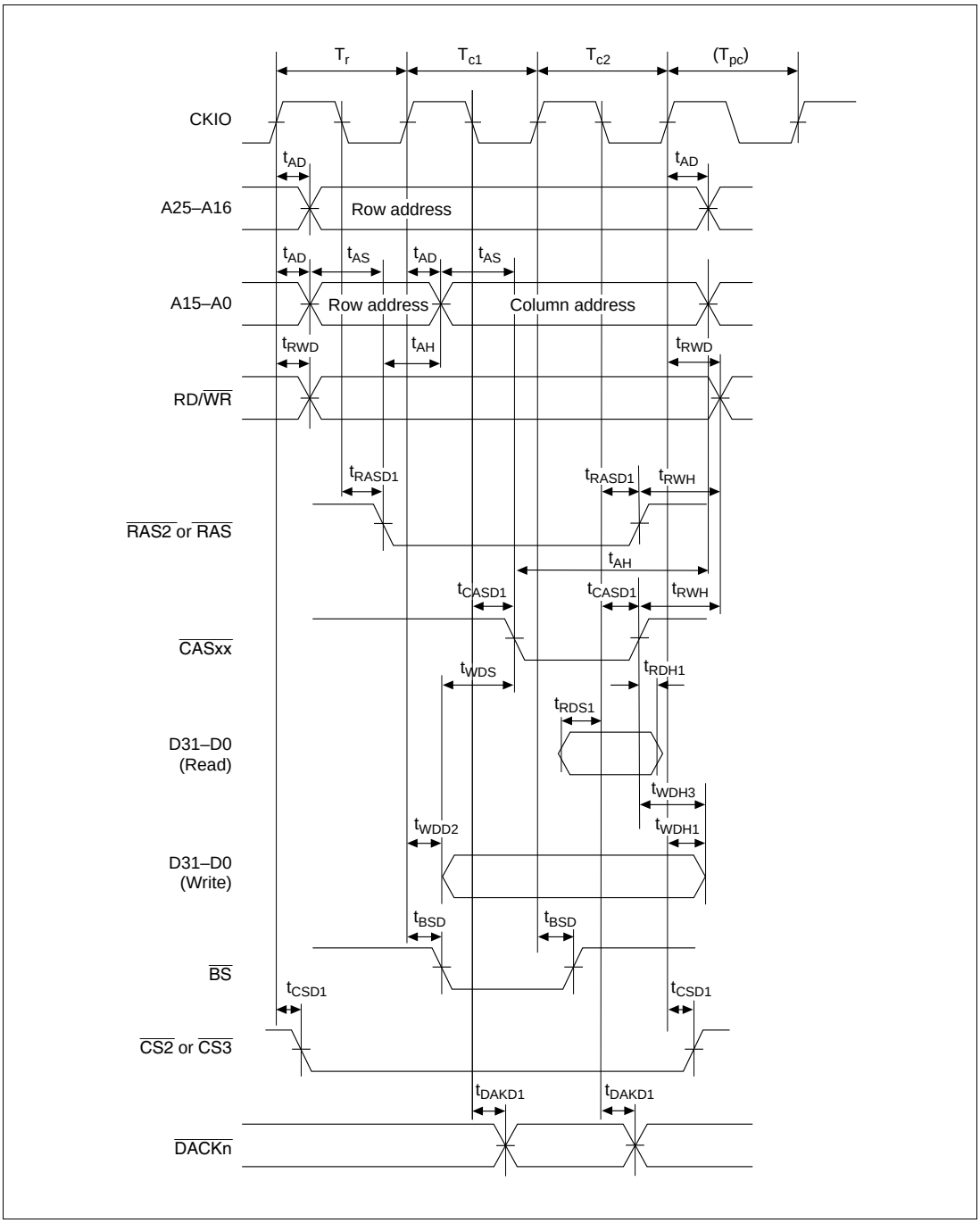


Figure 24.23 DRAM Bus Cycle (TRCD = 0, AnW = 1, TPC = 0)

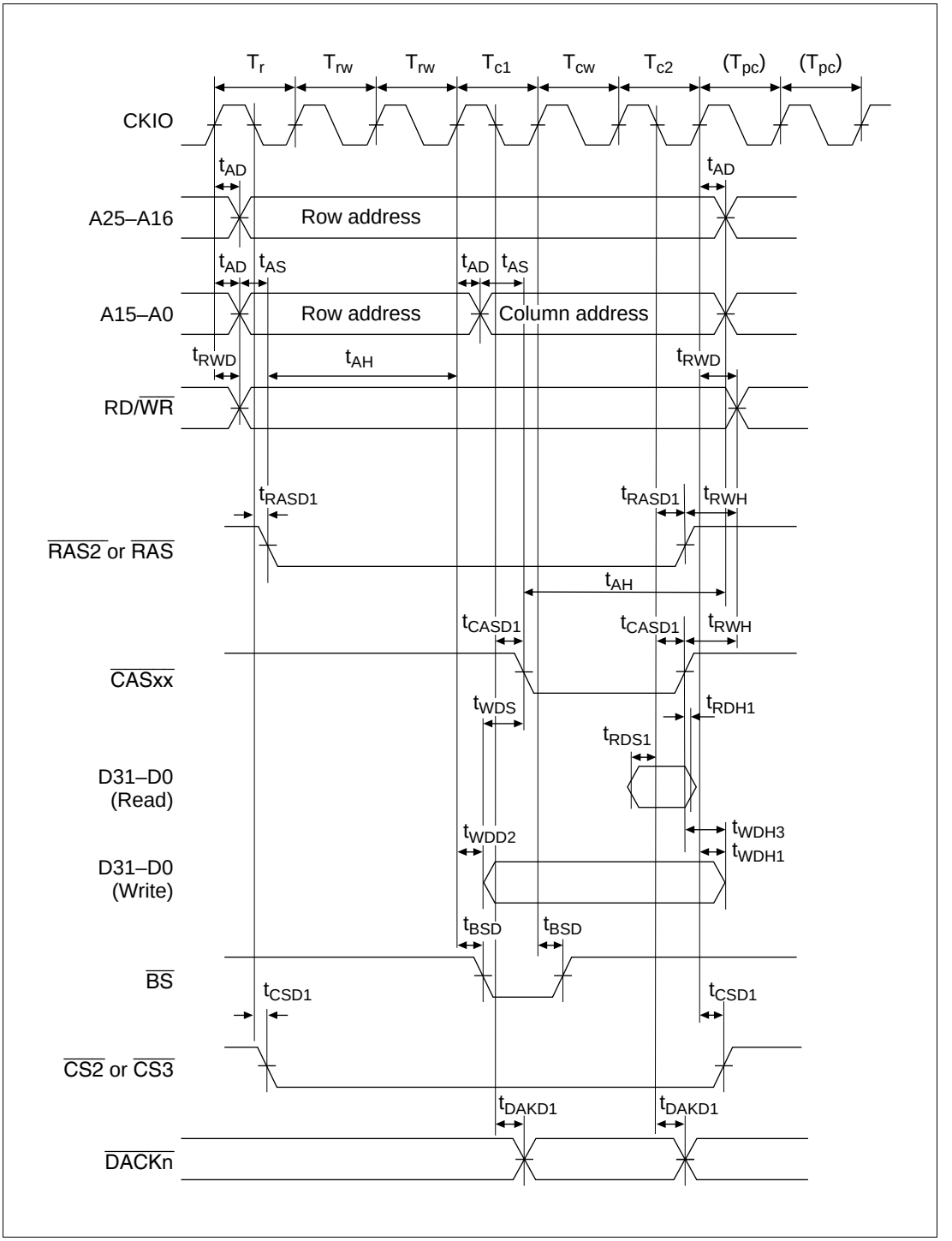


Figure 24.24 DRAM Bus Cycle (TRCD = 2, AnW = 2, TPC = 1)

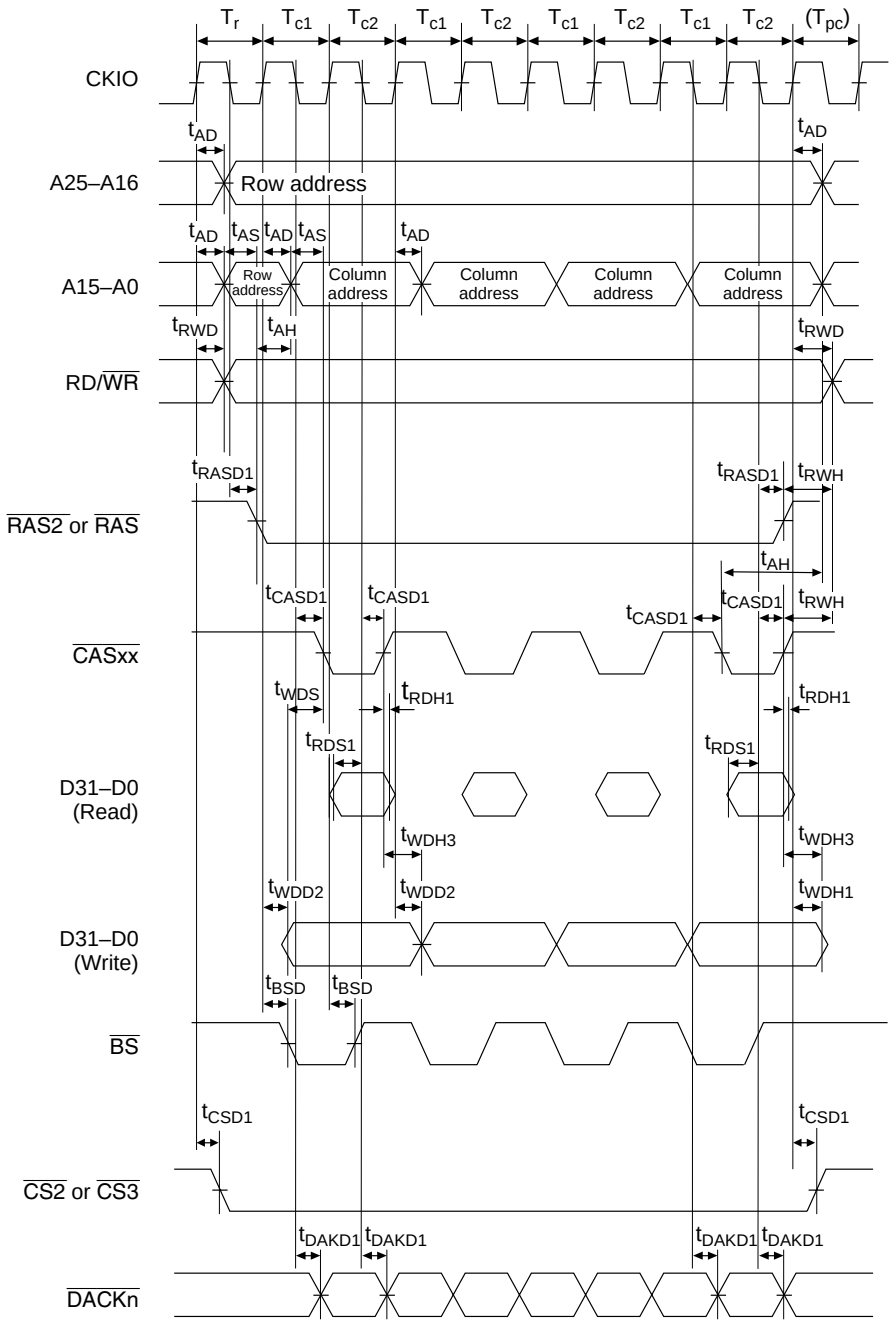


Figure 24.25 DRAM Burst Bus Cycle (TRCD = 0, AnW = 1, TPC = 0)

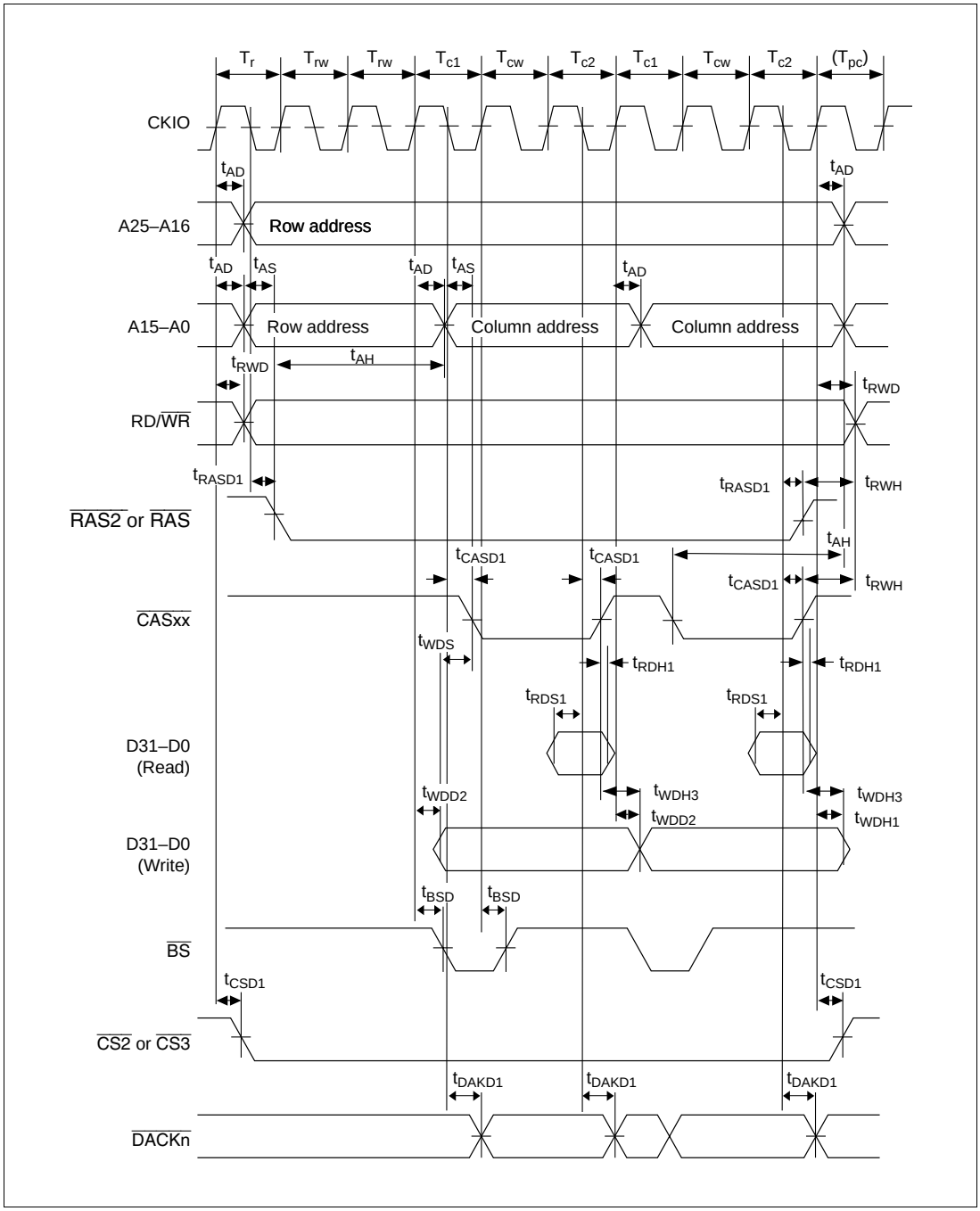


Figure 24.26 DRAM Burst Bus Cycle (TRCD = 2, AnW = 2, TPC = 0)

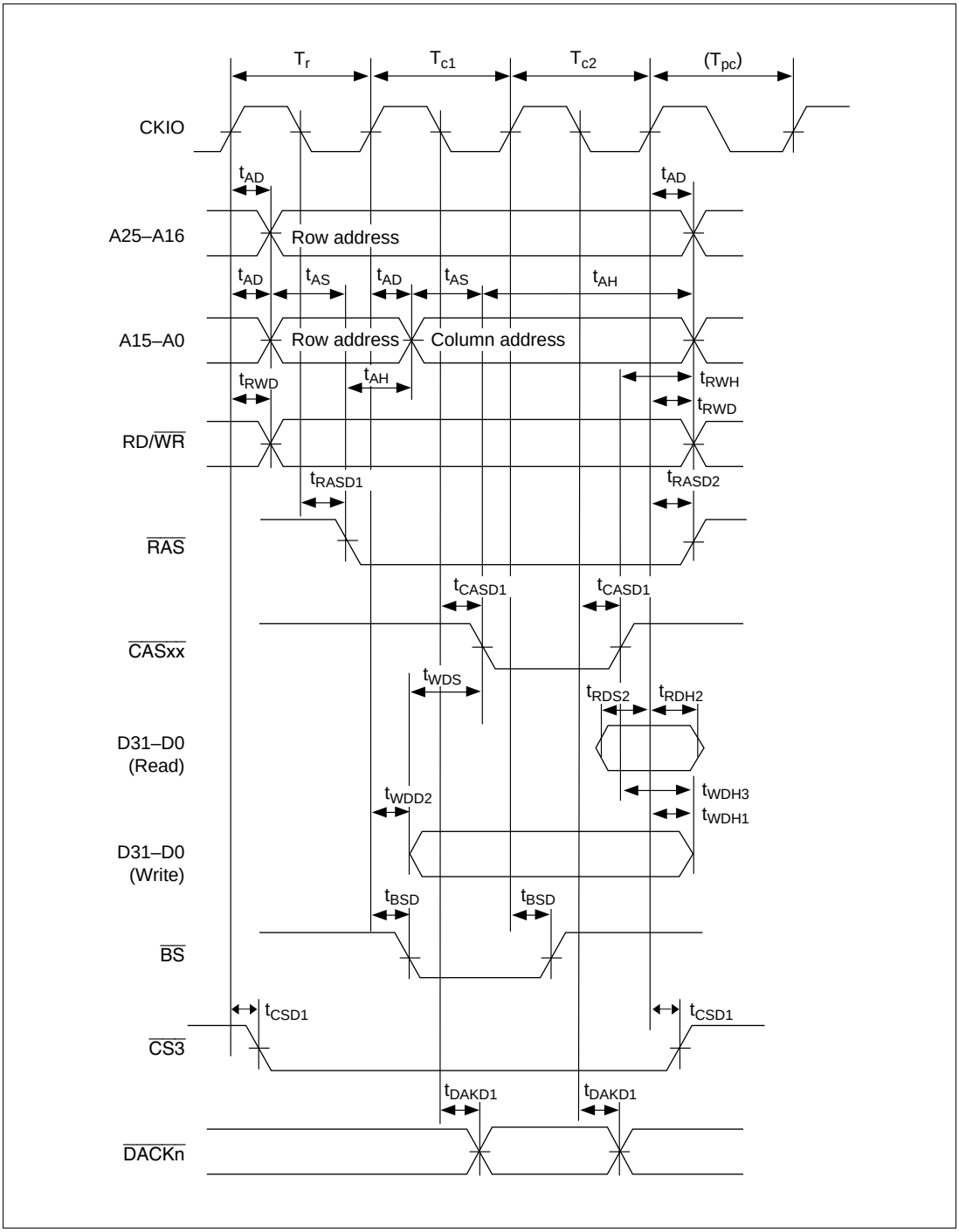


Figure 24.27 DRAM Bus Cycle (EDO mode, TRCD = 0, AnW = 1, TPC = 0)

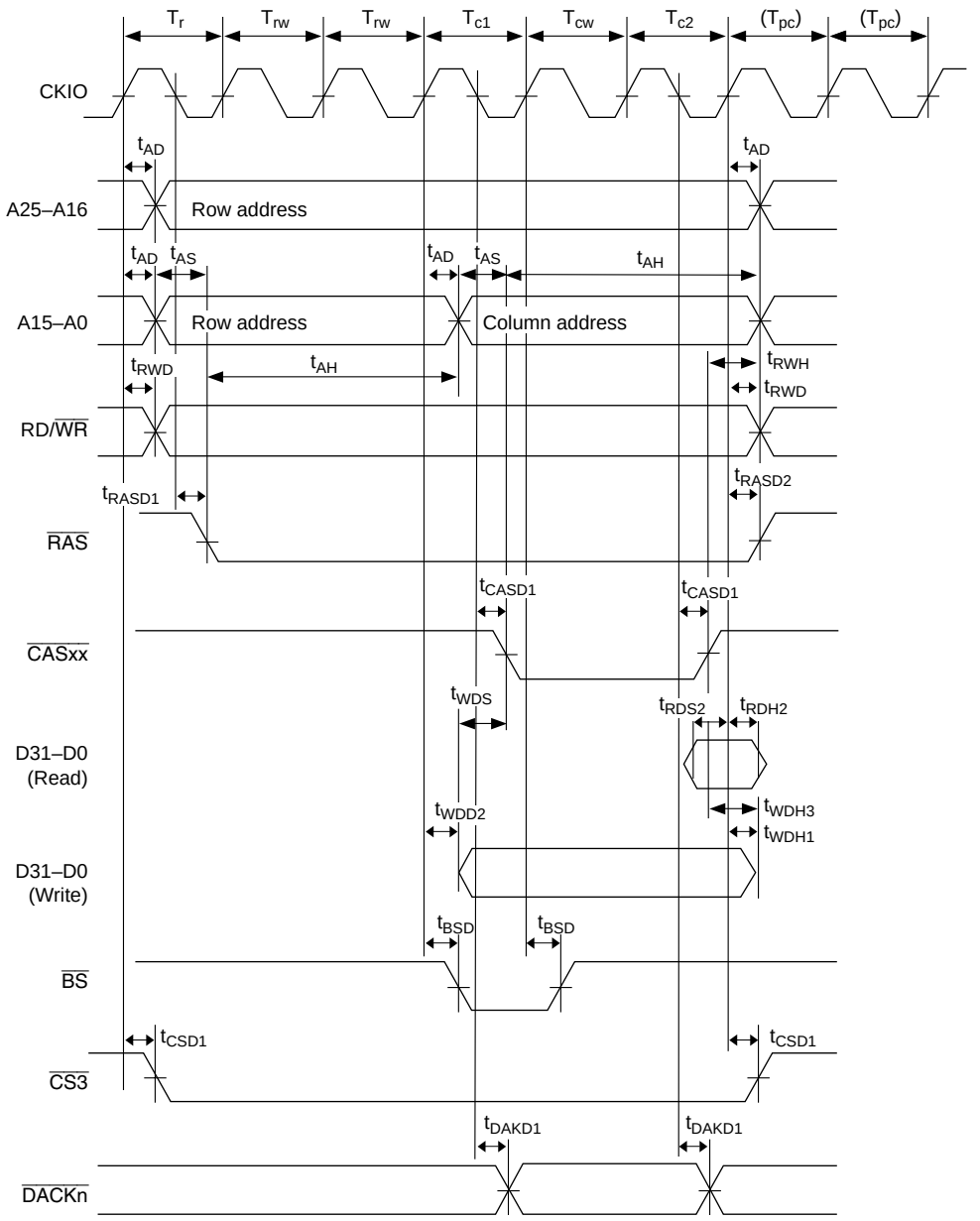


Figure 24.28 DRAM Bus Cycle (EDO mode, TRCD = 2, AnW = 2, TPC = 1)

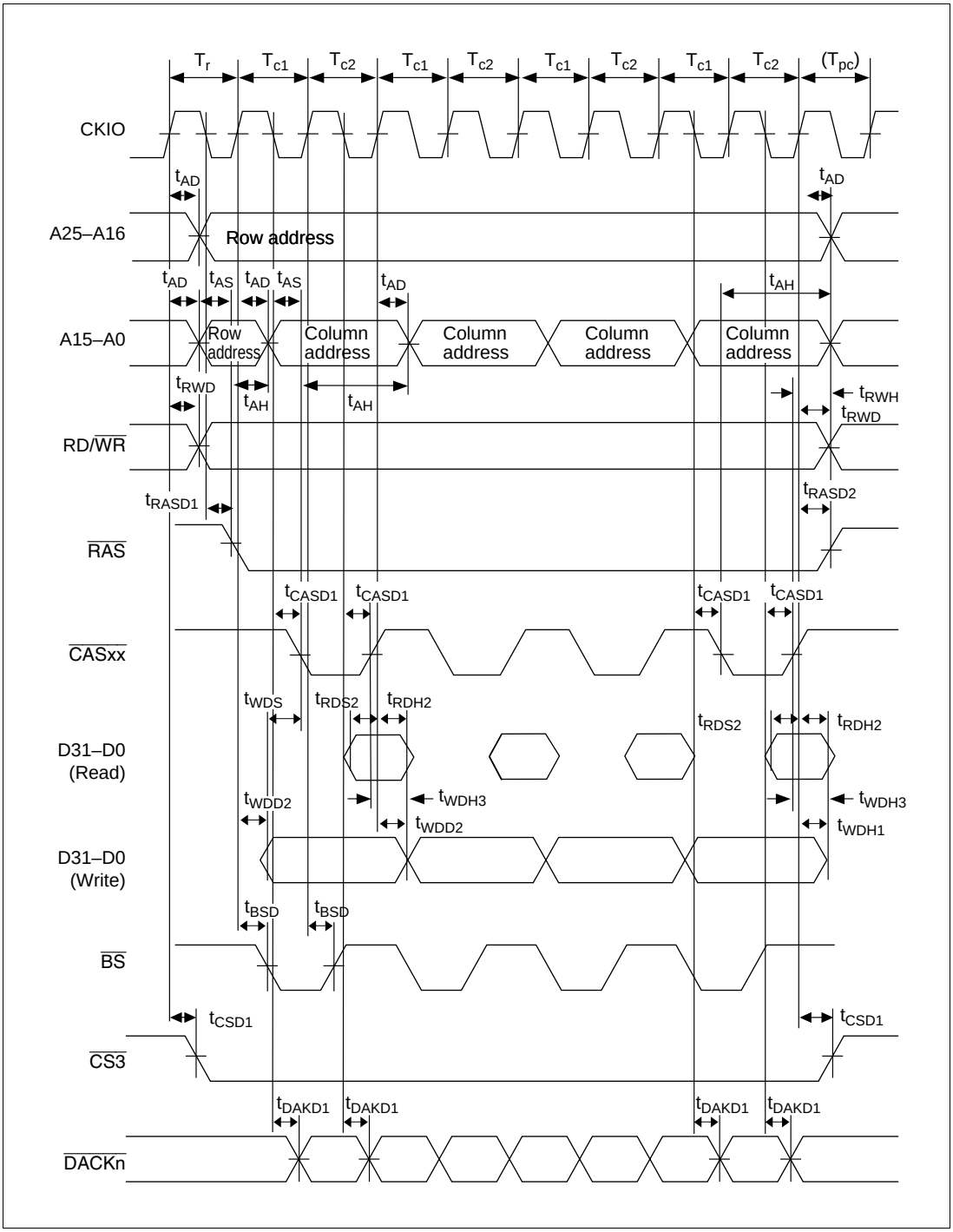


Figure 24.29 DRAM Burst Bus Cycle (EDO mode, TRCD = 0, AnW = 1, TPC = 0)

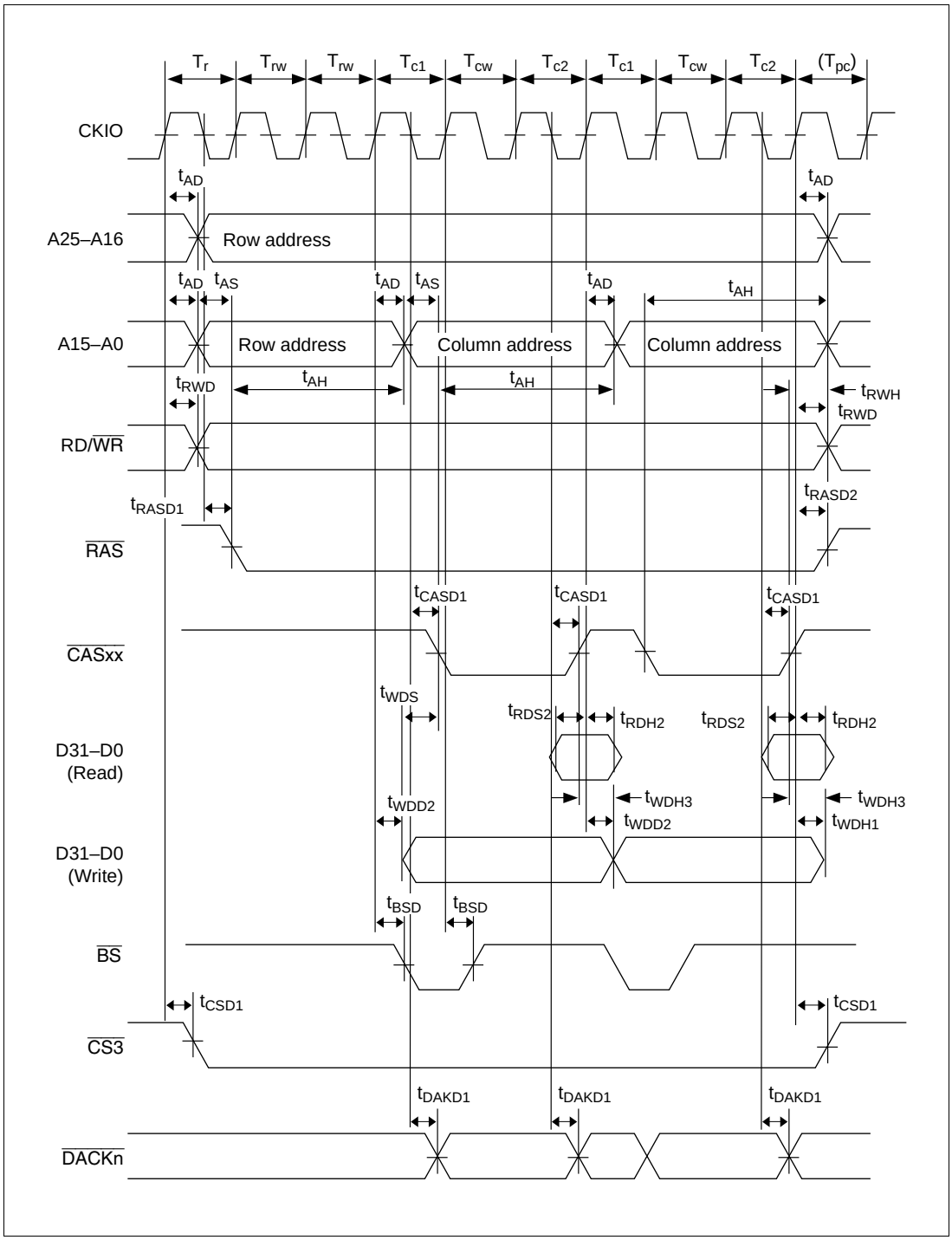


Figure 24.30 DRAM Burst Bus Cycle (EDO mode, TRCD = 2, AnW = 2, TPC = 0)

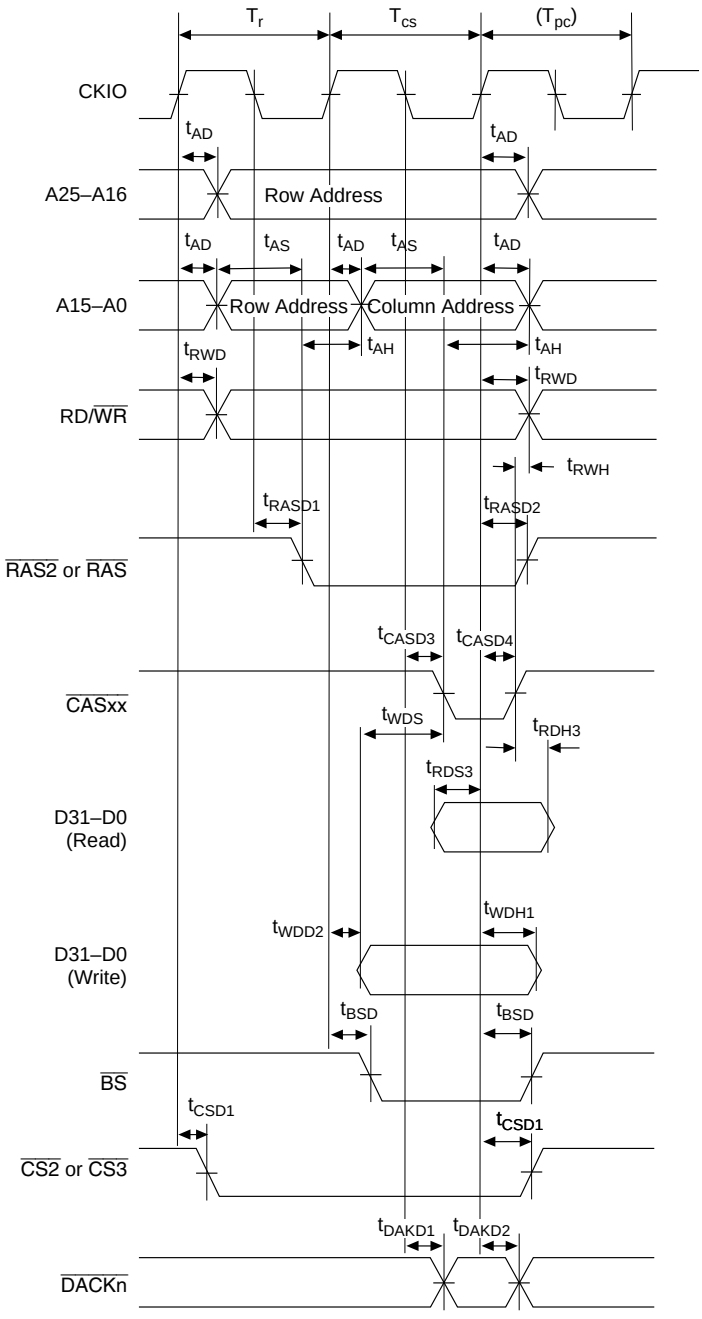


Figure 24.31 DRAM Short-Pitch Access Timing

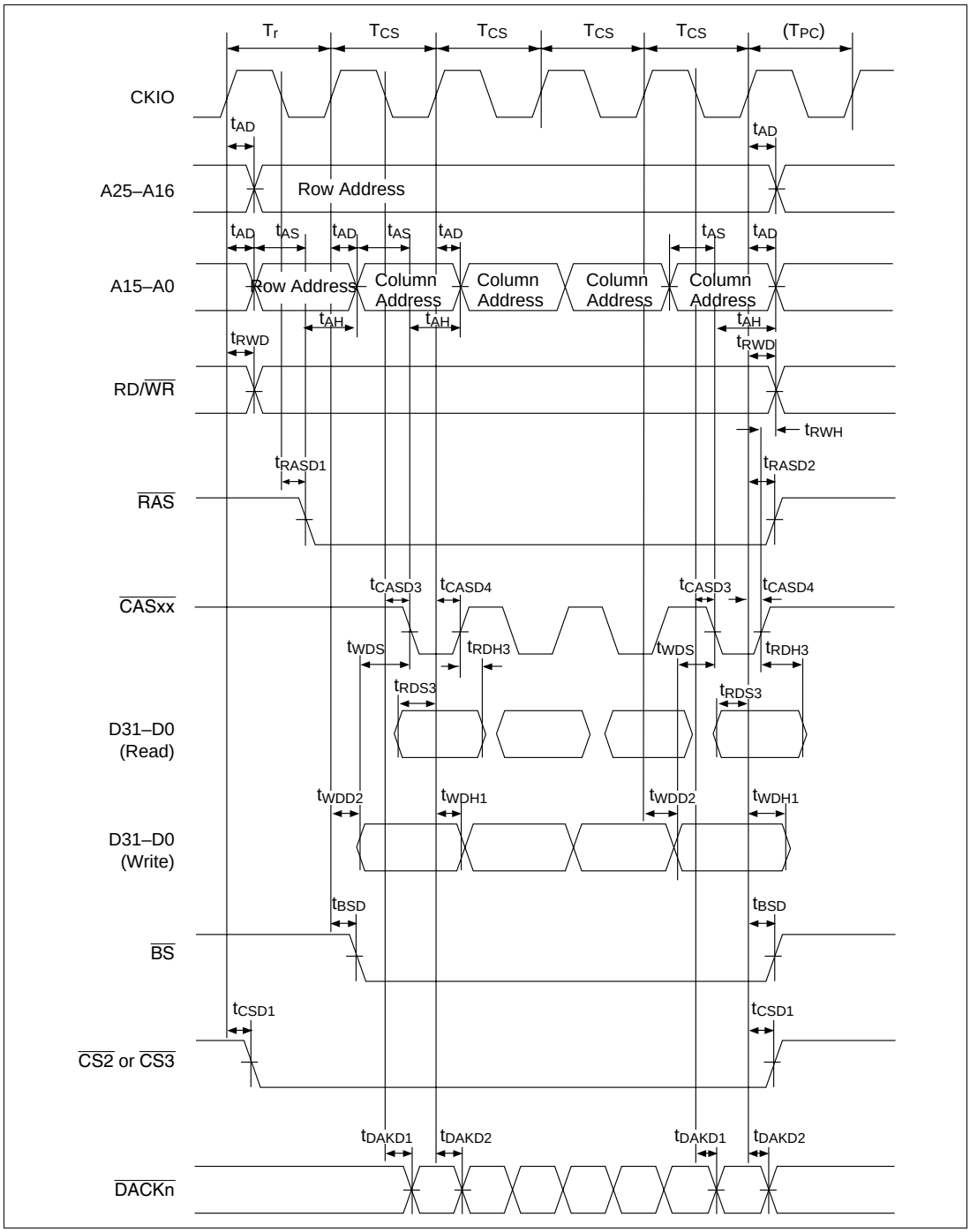


Figure 24.32 DRAM Short-Pitch Burst Access Timing

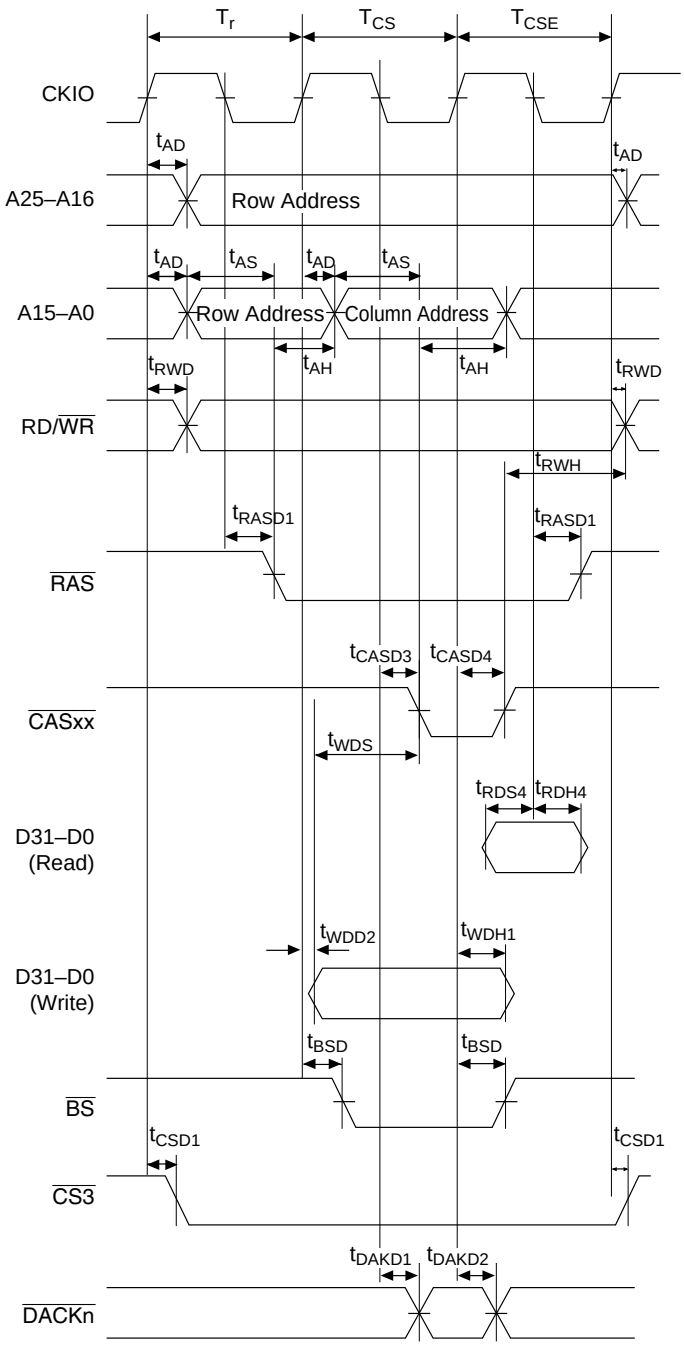


Figure 24.33 DRAM Short-Pitch Access Timing (EDO Mode)

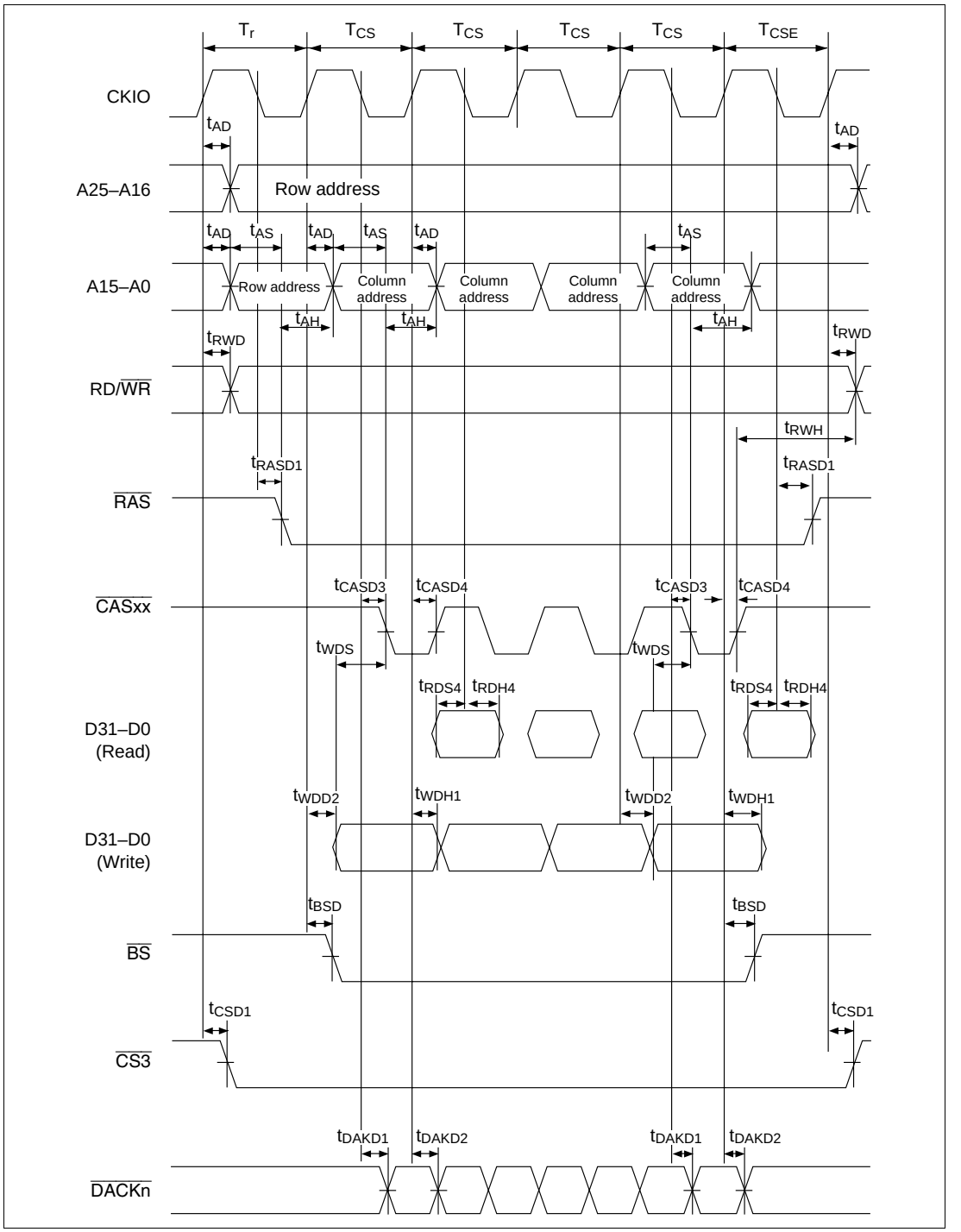


Figure 24.34 DRAM Short-Pitch Burst Access Timing (EDO Mode)

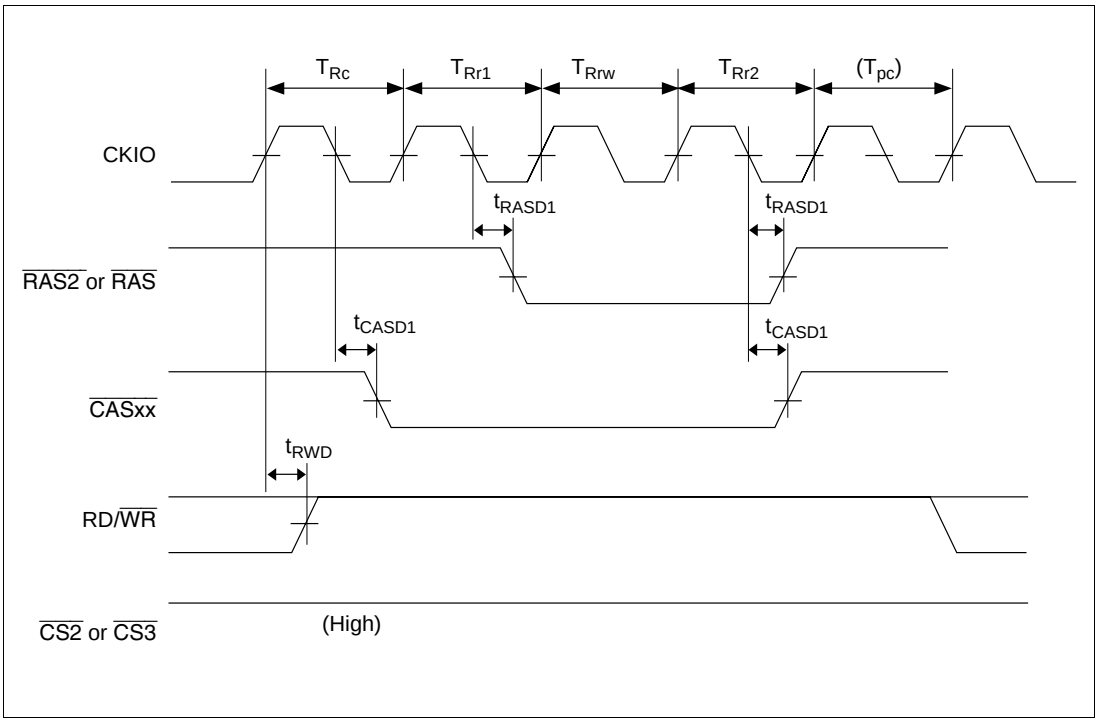


Figure 24.35 DRAM CAS-before-RAS Refresh Cycle (TRAS = 0, TPC = 0)

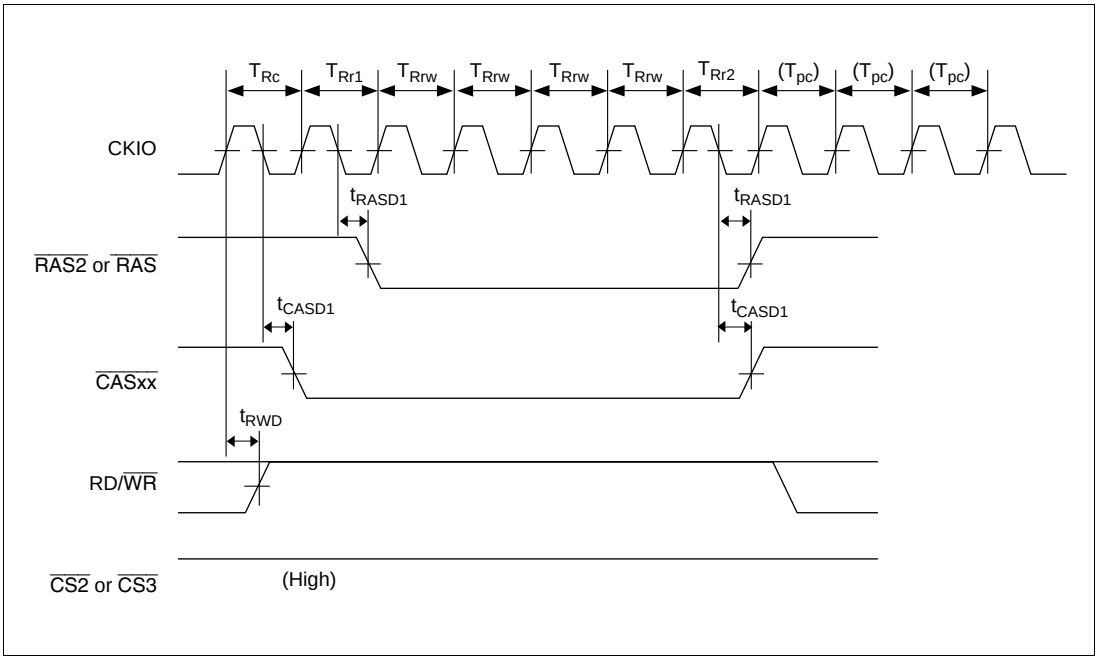


Figure 24.36 DRAM CAS-before-RAS Refresh Cycle (TRAS = 3, TPC = 2)

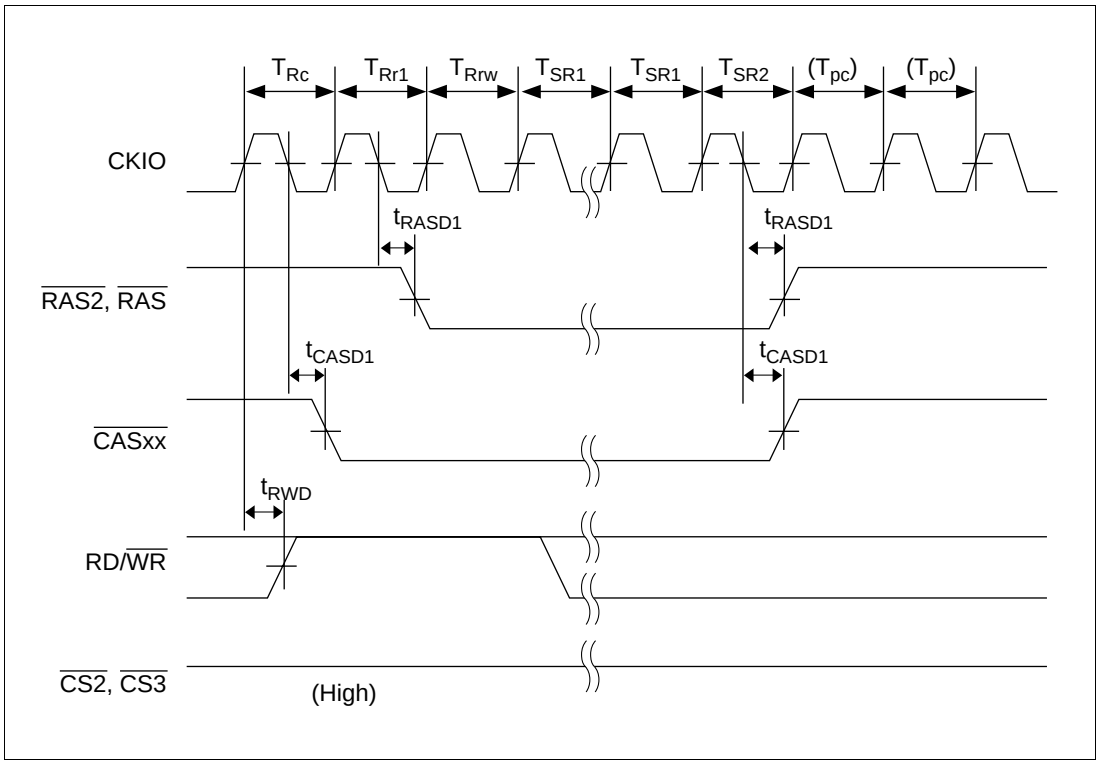


Figure 24.37 DRAM Self-Refresh Cycle (TPC = 0)

24.3.7 PCMCIA Timing

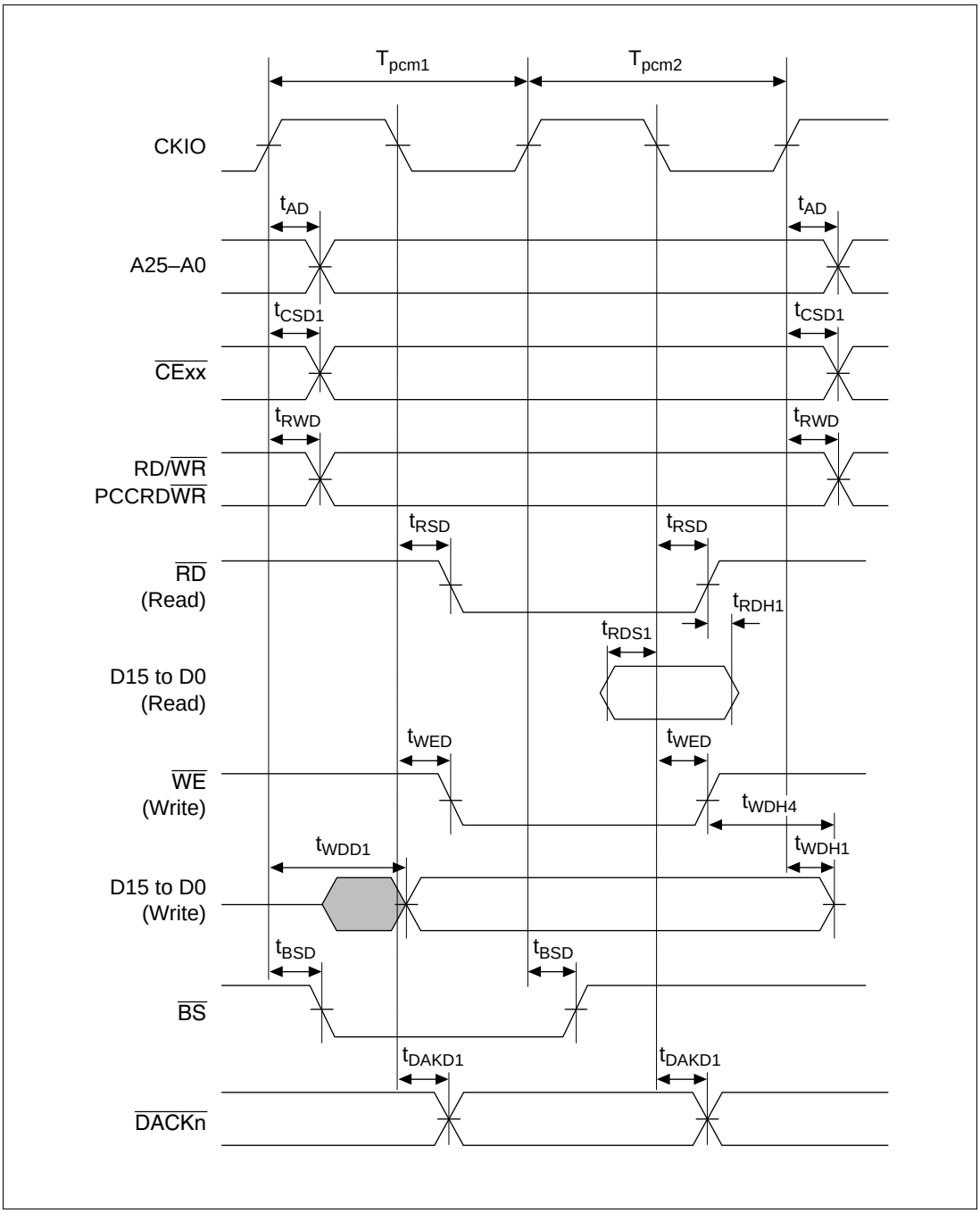


Figure 24.38 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

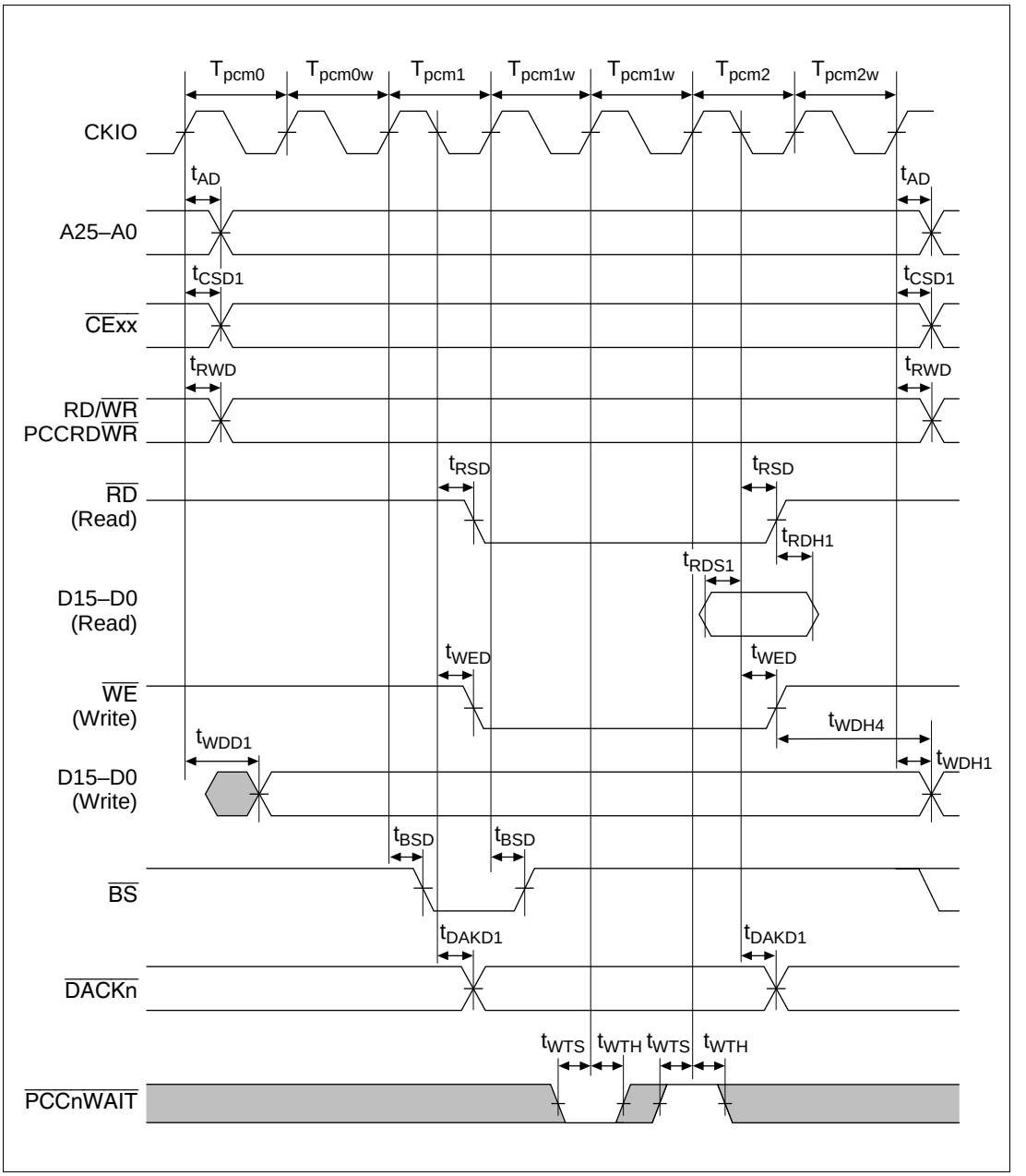
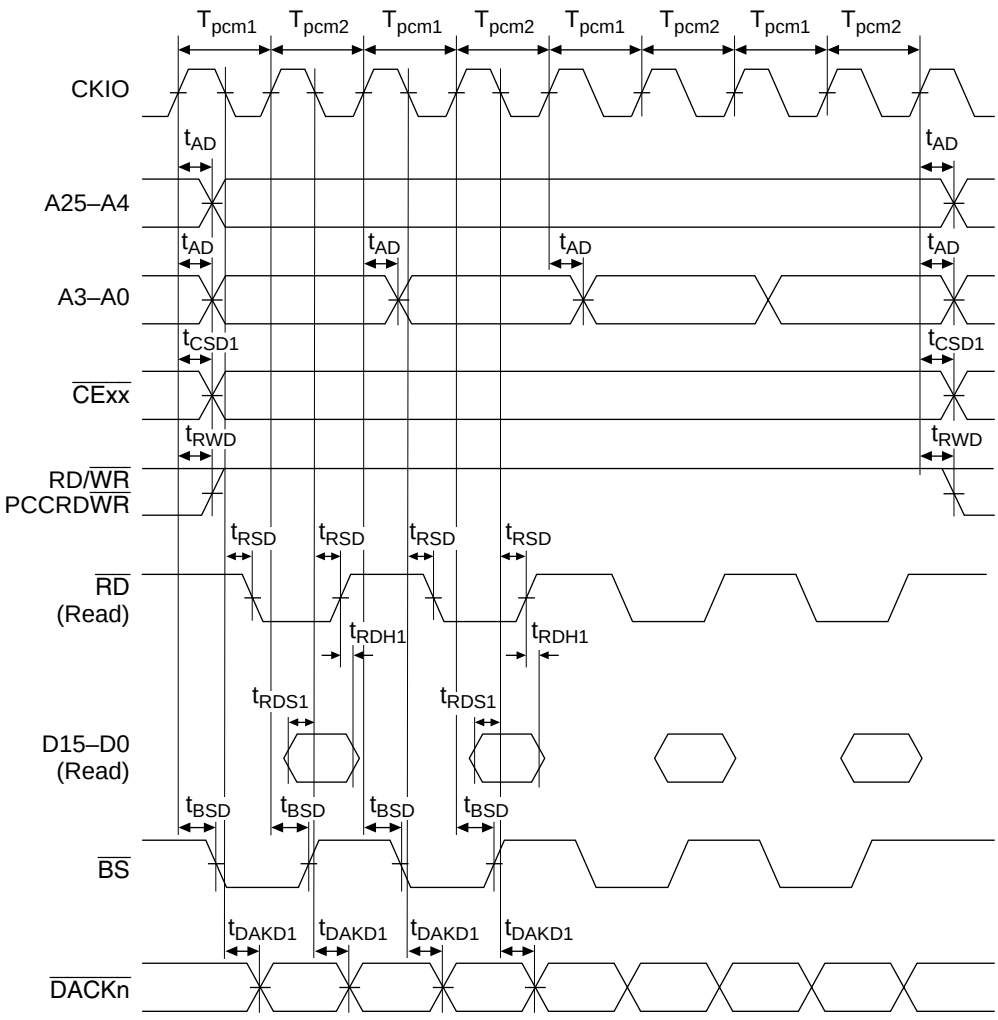
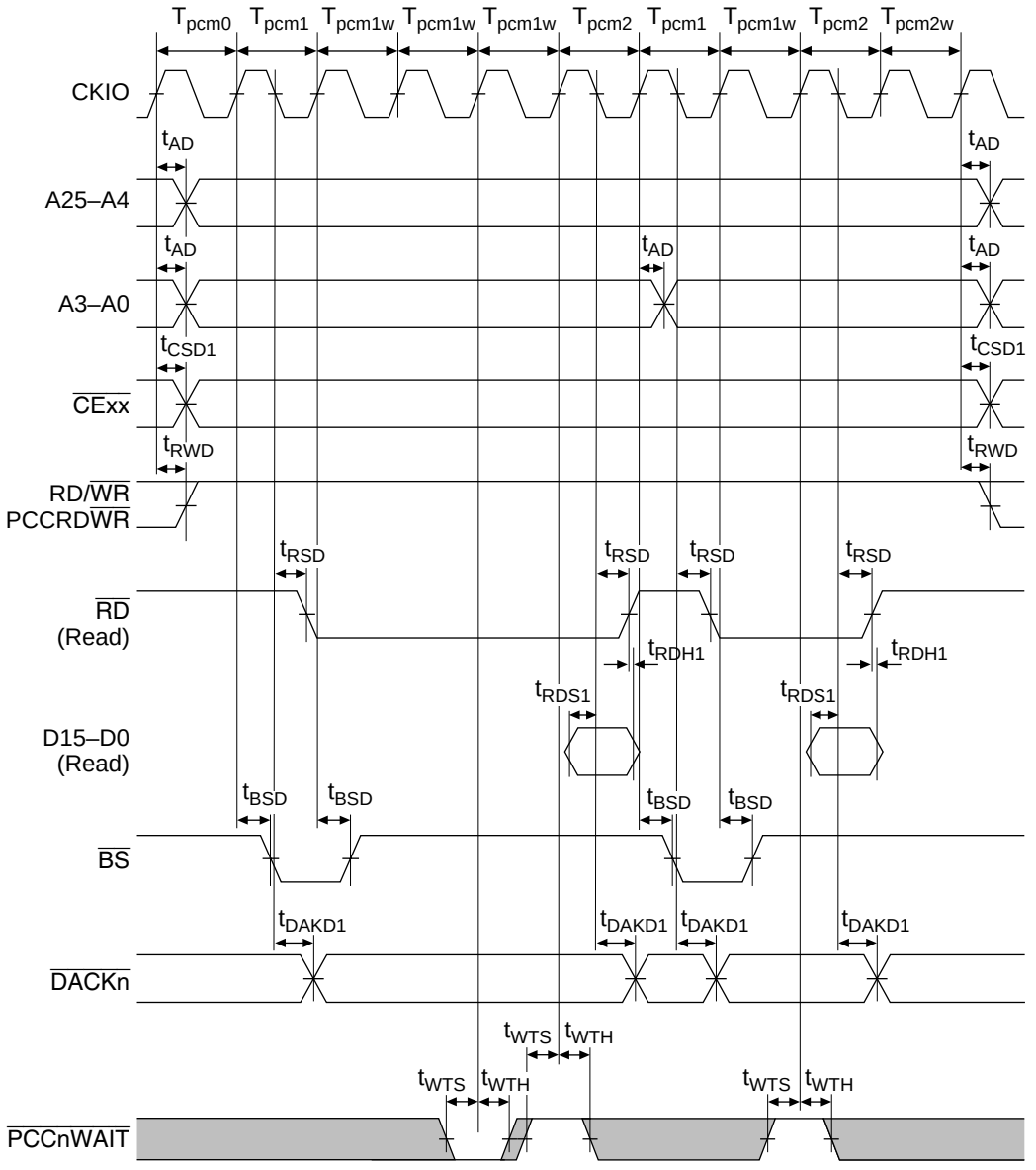


Figure 24.39 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait)



Note: Although burst mode is set, the write cycle operation is the same as in normal mode.

Figure 24.40 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Although burst mode is set, the write cycle operation is the same as in normal mode.

Figure 24.41 PCMCIA Memory Bus Cycle (Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3)

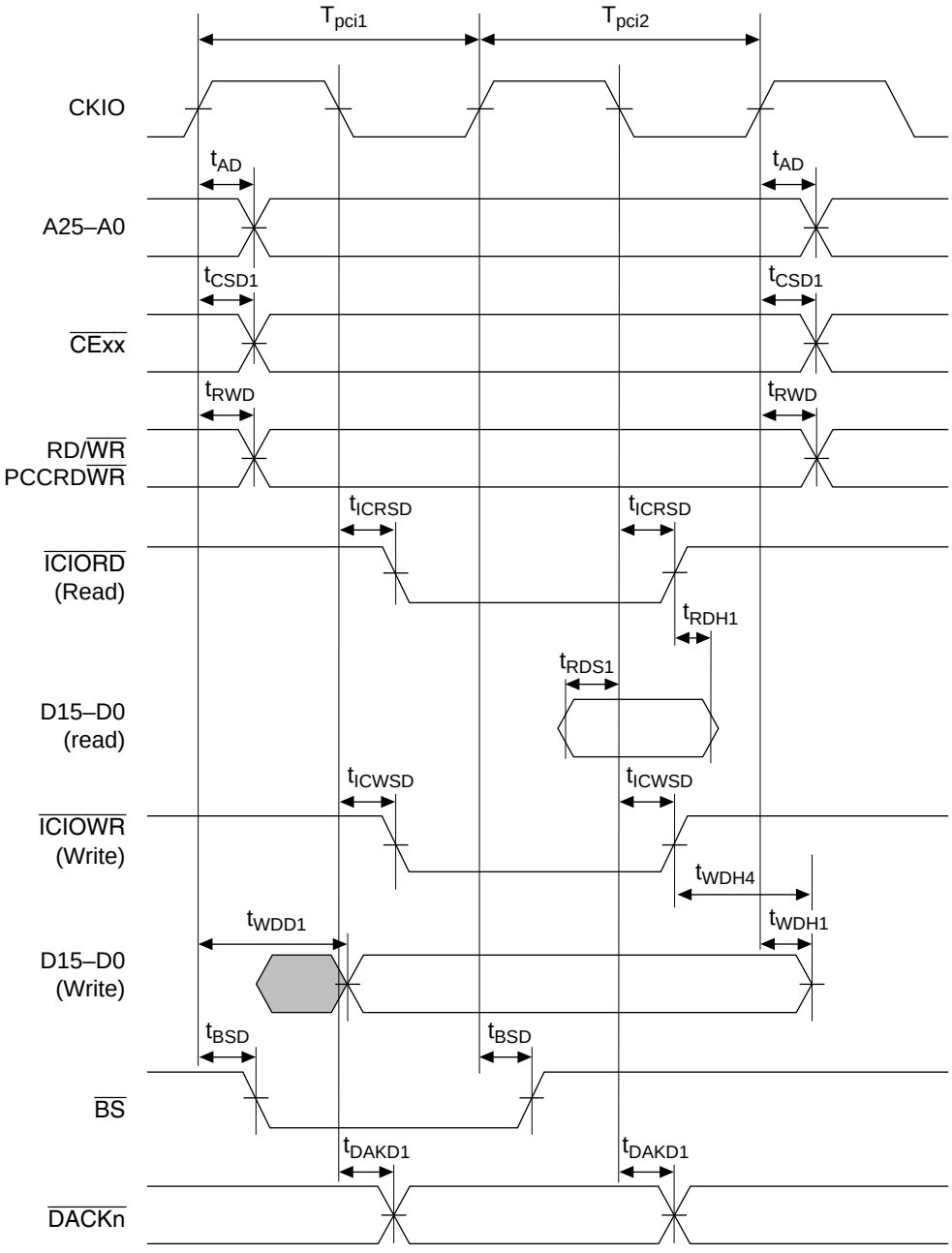


Figure 24.42 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

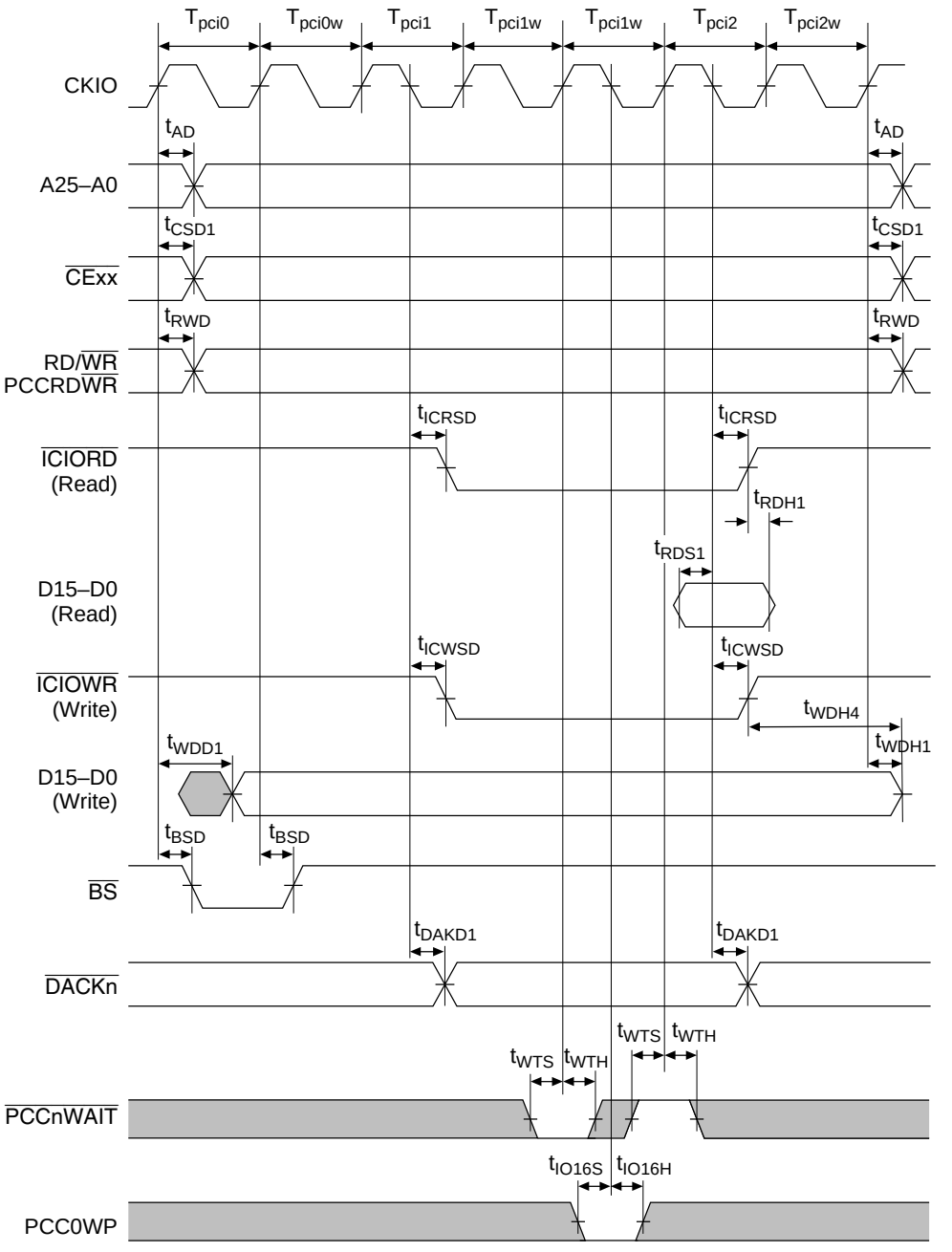


Figure 24.43 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait)

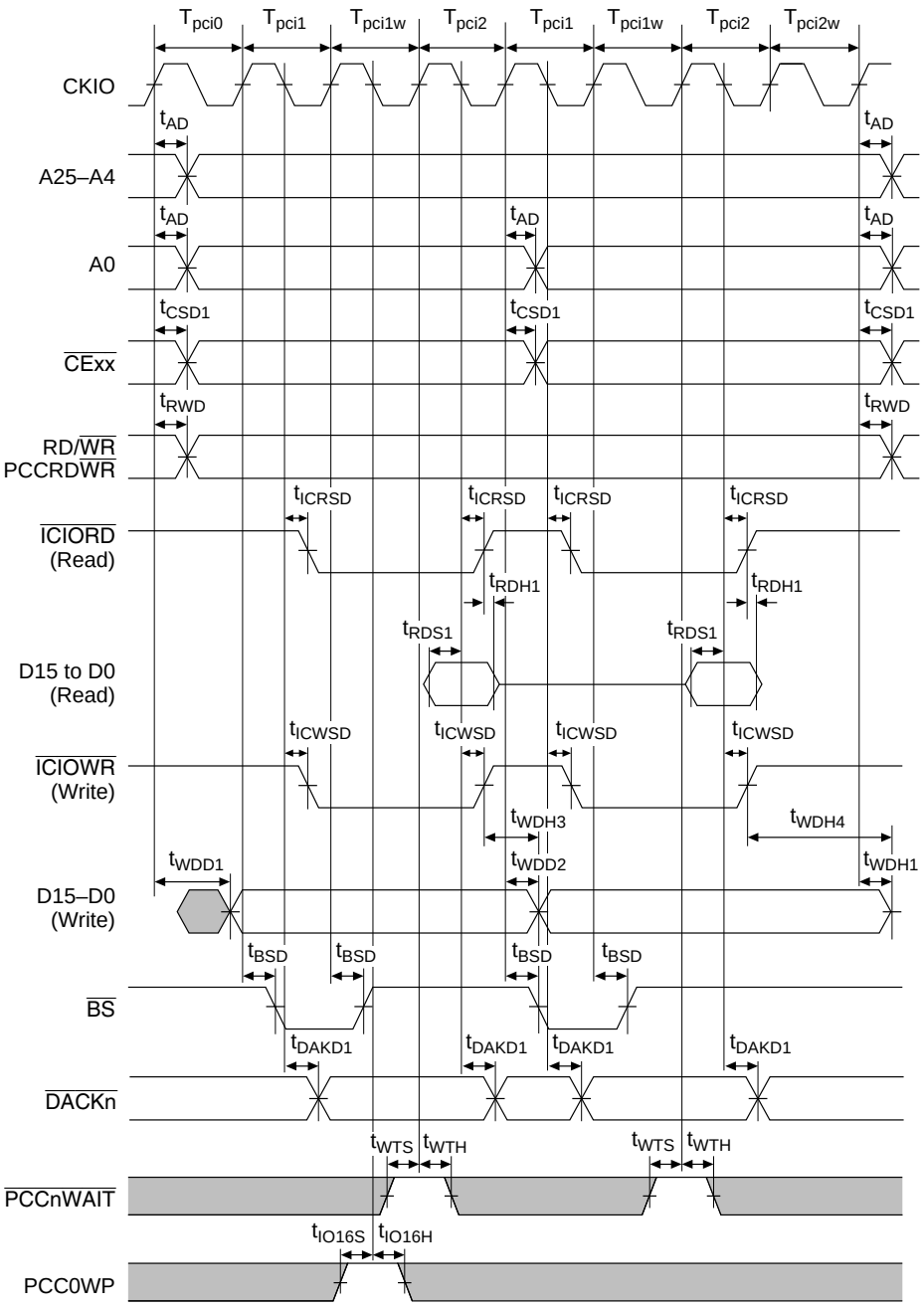


Figure 24.44 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, One Wait, Bus Sizing)

24.3.8 Supporting Module Signal Timing

Table 24.9 Supporting Module Signal Timing
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

				–15		–30			
Module	**Item**		**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
TMU, RTC	Timer input setup time		t_{TCLKS}	20	—	15	—	ns	24.45
	Timer clock input setup time		t_{TCKS}	20	—	15	—	ns	24.46
	Timer clock pulse width	Edge specification	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edge specification	t_{TCKWL}	2.5	—	2.5	—	t_{cyc}	
	Oscillation settling time		t_{ROSC}	—	3	—	3	S	24.47
SCI	Input clock cycle	Asynchronous	t_{SCYC}	4	—	4	—	t_{cyc}	24.48
		Synchronous		6	—	6	—	t_{cyc}	24.49
	Input clock rise time		t_{SCKR}	—	1.5	—	1.5	t_{cyc}	24.48
	Input clock fall time		t_{SCKF}	—	1.5	—	1.5	t_{cyc}	
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{scyc}	
	Transmission data delay time		t_{TXD}	—	100	—	100	ns	24.49
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	100	—	100	—	ns	
	RTS delay time		t_{RTSD}	—	100	—	100	ns	
	CTS setup time (synchronous)		t_{CTSS}	100	—	100	—	ns	
	CTS hold time (synchronous)		t_{CTSH}	100	—	100	—	ns	
Port	Output data delay time		t_{PORTD}	—	20	—	17	ns	24.50
	Input data setup time		t_{PORTS1}	20	—	15	—	ns	
	Input data hold time		t_{PORTH1}	10	—	8	—	ns	
	Input data setup time		t_{PORTS2}	22	—	17	—	ns	
	Input data hold time		t_{PORTH2}	12	—	10	—	ns	

Table 24.9 Peripheral Module Signal Timing (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

			–15		–30			
Module	**Item**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
PCC	Input data setup time	t_{PCCinS}	20	—	15	—	ns	24.51
	Input data hold time	t_{PCCinH}	10	—	8	—		
	PCCnWP setup time	t_{PCCWPS}	20	—	15	—		
	PCCnWP hold time	t_{PCCWPH}	10	—	8	—		
	PCCREG delay time	$t_{PCCREGD}$	—	20	—	17		
	PCCnDRV delay time	$t_{PCCDRVD}$	—	20	—	17		
	PCCnRESET delay time	$t_{PCCRESD}$	—	20	—	17		
DMAC	$\overline{\text{DREQ}}$ setup time	t_{DRQS}	12	—	12	—	ns	24.52
	$\overline{\text{DREQ}}$ hold time	t_{DRQH}	8	—	8	—		
LCDC	DON delay time	t_{DDON}	—	20	—	17	ns	24.53

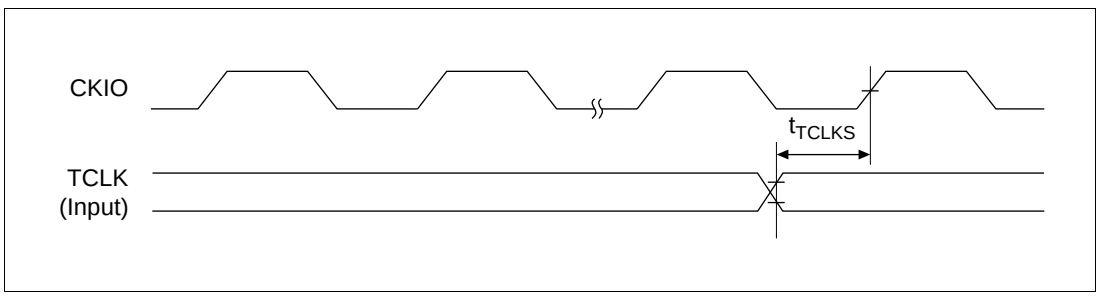


Figure 24.45 TCLK Input Timing

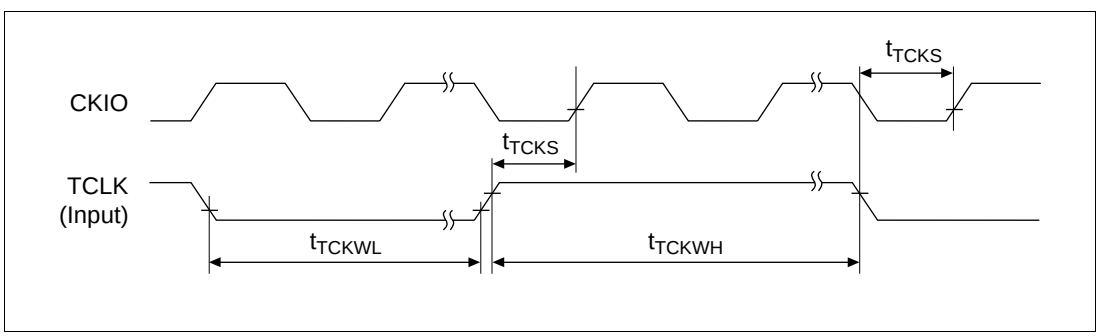


Figure 24.46 TCLK Clock Input Timing

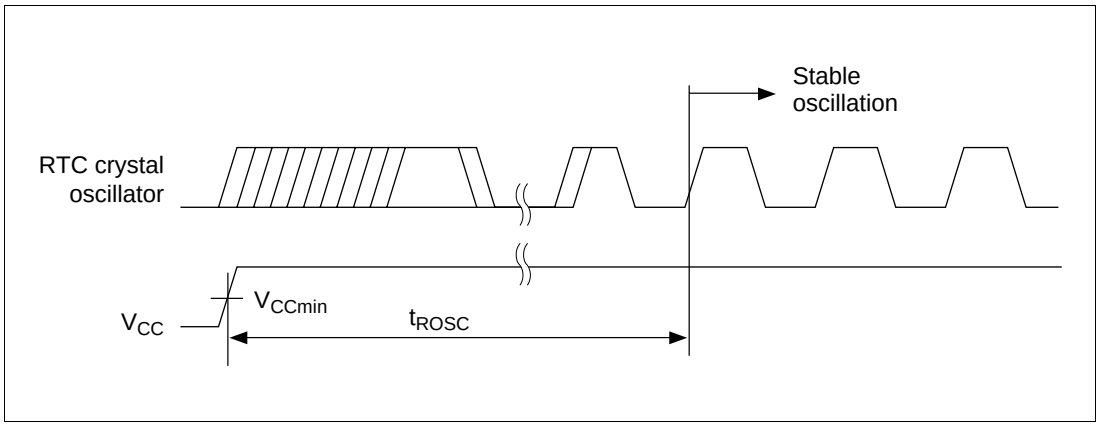


Figure 24.47 Oscillation Settling Time at RTC Crystal Oscillator Power-on

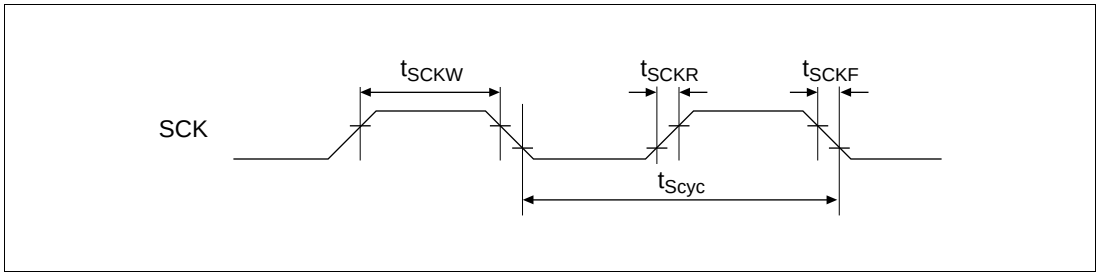


Figure 24.48 SCK Input Clock Timing

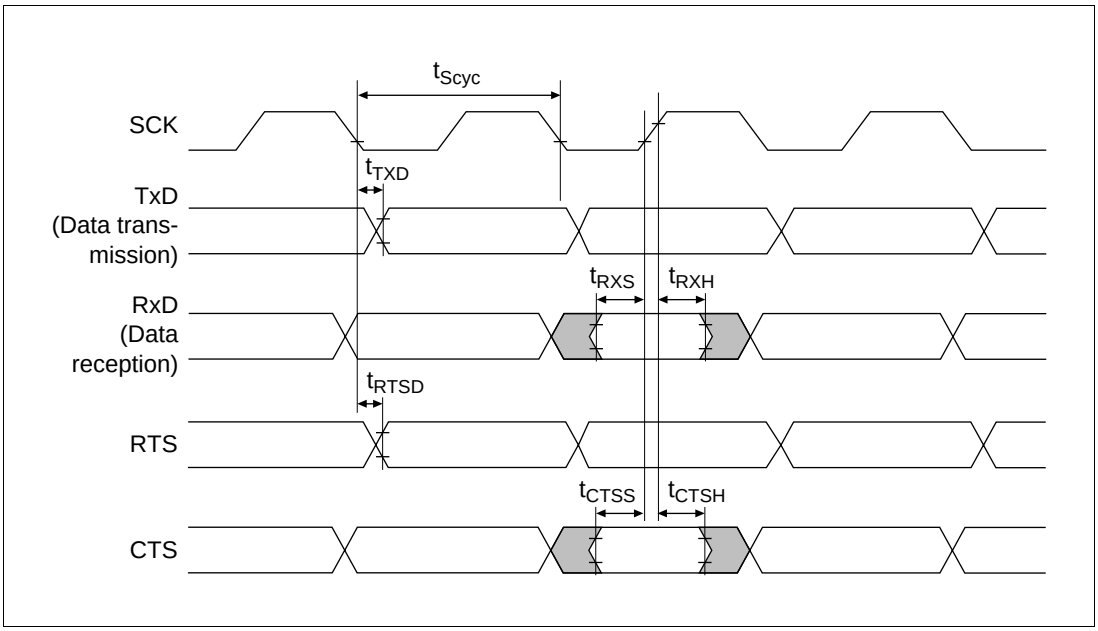


Figure 24.49 SCI I/O Timing in Synchronous Mode

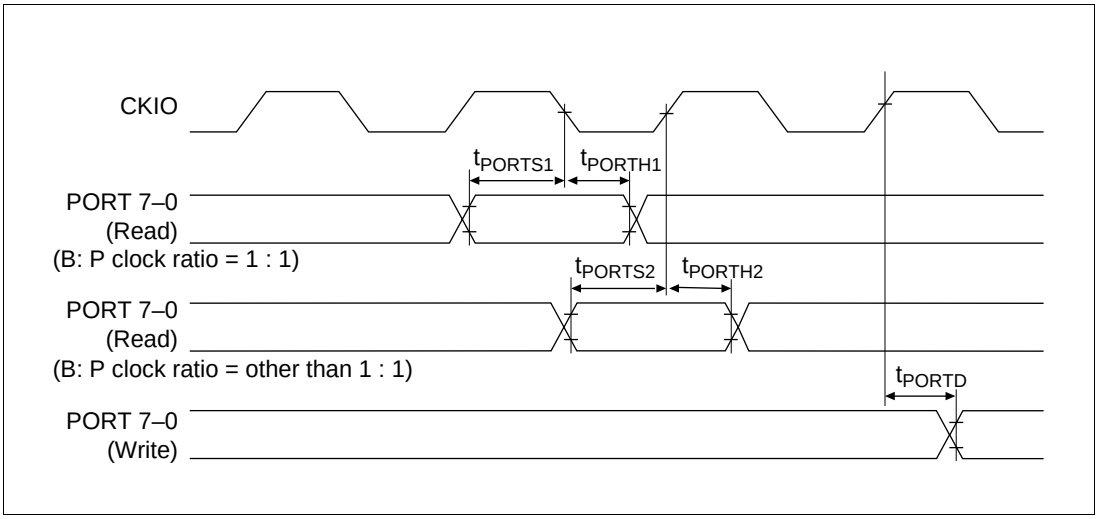


Figure 24.50 I/O Port Timing

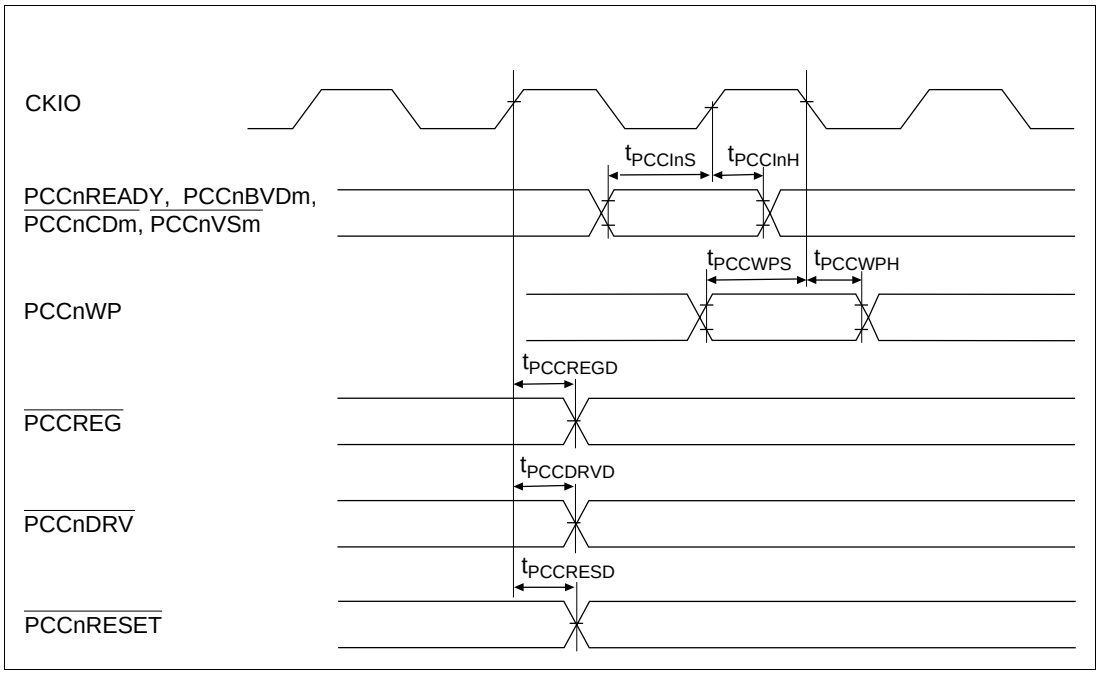


Figure 24.51 PCC Input/Output Timing

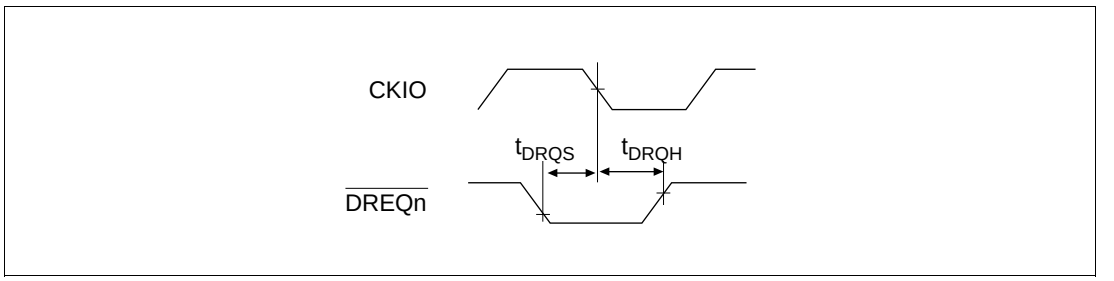


Figure 24.52 DREQ Input Timing

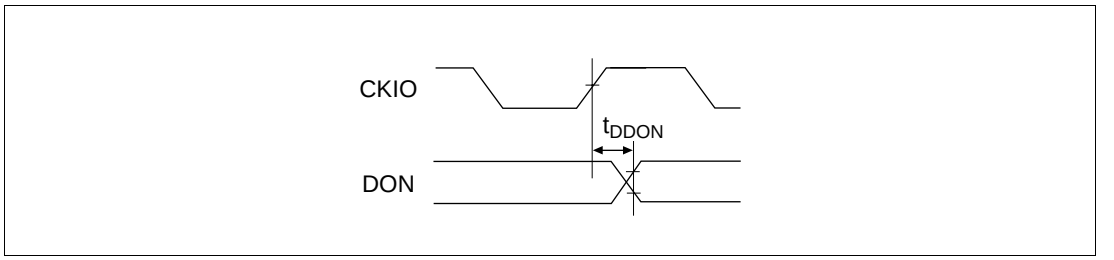


Figure 24.53 DON Output Timing

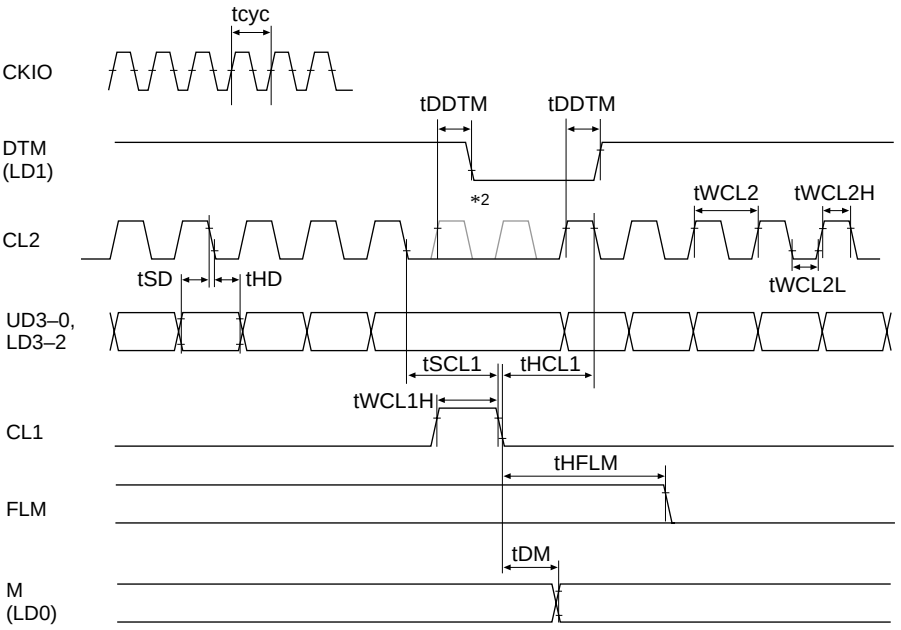
Table 24.10 Supporting Module Signal Timing (LCD Controller : Monochrome/Single Screen/4-Bit Output Mode)
(V_{CC} = 3.3 ± 0.3V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL2 cycle time	t_{WCL2}	000	$2t_{cyc}$-10	—	ns	24.54
		001	$4t_{cyc}$-10	—		
		010	$8t_{cyc}$-10	—		
		011	$16t_{cyc}$-10	—		
		100	$32t_{cyc}$-10	—		
CL2 high width	t_{WCL2H}	000	t_{cyc}-15	—		
		001	$2t_{cyc}$-15	—		
		010	$4t_{cyc}$-15	—		
		011	$8t_{cyc}$-15	—		
		100	$16t_{cyc}$-15	—		
CL2 low width	t_{WCL2L}	000	t_{cyc}-15	—		
		001	$2t_{cyc}$-15	—		
		010	$4t_{cyc}$-15	—		
		011	$8t_{cyc}$-15	—		
		100	$16t_{cyc}$-15	—		
Data setup time	t_{SD}	000	t_{cyc}-15	—		
		001	$2t_{cyc}$-15	—		
		010	$4t_{cyc}$-15	—		
		011	$8t_{cyc}$-15	—		
		100	$16t_{cyc}$-15	—		
Data hold time	t_{HD}	000	t_{cyc}-15	—		
		001	$2t_{cyc}$-15	—		
		010	$4t_{cyc}$-15	—		
		011	$8t_{cyc}$-15	—		
		100	$16t_{cyc}$-15	—		
CL1 setup time	t_{SCL1}	000	$3t_{cyc}$-15	—		
		001	$6t_{cyc}$-15	—		
		010	$12t_{cyc}$-15	—		
		011	$24t_{cyc}$-15	—		
		100	$48t_{cyc}$-15	—		

Table 24.10 Supporting Module Signal Timing (LCD Controller : Monochrome/Single Screen/4-Bit Output Mode) (cont)
(V_{CC} = 3.3 ± 0.3V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL1 hold time	t_{HCL1}	000	t_{cyc}-15	—	ns	
		001	$2t_{cyc}$-15	—		
		010	$4t_{cyc}$-15	—		
		011	$8t_{cyc}$-15	—		
		100	$16t_{cyc}$-15	—		
CL1 high width	t_{WCL1H}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
FLM hold time	t_{HFLM}	000	$3.5t_{cyc}$-15	—		
		001	$7t_{cyc}$-15	—		
		010	$14t_{cyc}$-15	—		
		011	$28t_{cyc}$-15	—		
		100	$56t_{cyc}$-15	—		
M delay time	t_{DM}		—	20		
DTM delay time	t_{DDTM}		—	20		

Notes: 1. In monochrome 16-level gray scale mode, DIV cannot be set to 000.
2. Specification when the output position and width are set so that the CL1 clock is entirely within the horizontal retrace line interval.
Fine tuning can be performed with a register setting to match the corresponding LCD driver.
3. CC1 = 1: CL1 masked during vertical retrace line interval
CC1 = 0: CL1 output
4. CC0 = 0: CL2 masked during horizontal retrace line interval
CC0 = 1: CL2 output



Notes: 1. When the horizontal retrace line interval is 8 dots, and the CL1 clock has the minimum settable width (4 dots).
2. Solid line: CC0= 0, Dotted line: CC0= 1
3. The CL1 clock is output during the horizontal retrace line interval.

Figure 24.54 Monochrome/Single Screen/4-Bit Output

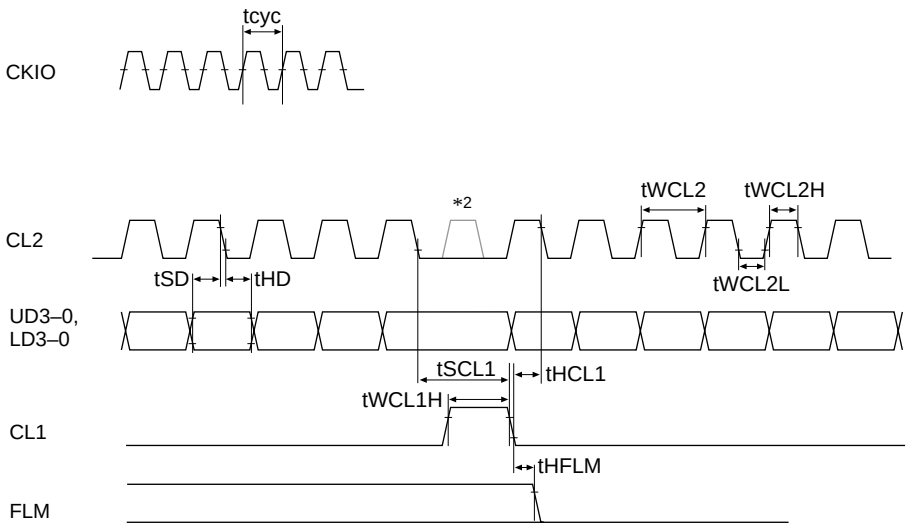
Table 24.11 Supporting Module Signal Timing (LCD Controller: Monochrome/Single Screen/8-Bit Output Mode)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL2 cycle time	t_{WCL2}	000	$4t_{cyc}$-10	—	ns	24.55
		001	$8t_{cyc}$-10	—		
		010	$16t_{cyc}$-10	—		
		011	$32t_{cyc}$-10	—		
		100	$64t_{cyc}$-10	—		
CL2 high width	t_{WCL2H}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
CL2 low width	t_{WCL2L}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
Data setup time	t_{SD}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
Data hold time	t_{HD}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
CL1 setup time	t_{SCL1}	000	$6t_{cyc}$-15	—		
		001	$12t_{cyc}$-15	—		
		010	$24t_{cyc}$-15	—		
		011	$48t_{cyc}$-15	—		
		100	$96t_{cyc}$-15	—		

Table 24.11 Supporting Module Signal Timing (LCD Controller: Monochrome/Single Screen/8-Bit Output Mode) (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL1 hold time	t_{HCL1}	000	$2t_{cyc}$-15	—		
		001	$4t_{cyc}$-15	—		
		010	$8t_{cyc}$-15	—		
		011	$16t_{cyc}$-15	—		
		100	$32t_{cyc}$-15	—		
CL1 high width	t_{WCL1H}	000	$4t_{cyc}$-15	—		
		001	$8t_{cyc}$-15	—		
		010	$16t_{cyc}$-15	—		
		011	$32t_{cyc}$-15	—		
		100	$64t_{cyc}$-15	—		
FLM hold time	t_{HFLM}	000	$0.5t_{cyc}$-15	—		
		001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		

Notes: 1. In monochrome 16-level gray scale mode, DIV cannot be set to 000.
2. Specification when the output position and width are set so that the CL1 clock is entirely within the horizontal retrace line interval.
Fine tuning can be performed with a register setting to match the corresponding LCD driver.
3. CC1 = 1: CL1 masked during vertical retrace line interval
CC1 = 0: CL1 output
4. CC0 = 0: CL2 masked during horizontal retrace line interval
CC0 = 1: CL2 output



Notes: 1. When the horizontal retrace line interval is 8 dots, and the CL1 clock has the minimum settable width (8 dots).
2. Solid line: CC0= 0, Dotted line: CC0= 1
3. The CL1 clock is output during the horizontal retrace line interval.

Figure 24.55 Monochrome/Single Screen/8-Bit Output

Table 24.12 Supporting Module Signal Timing (LCD Controller: TFT/TFD Color Mode) (V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL2 cycle time	t_{WCL2}	001	t_{cyc}-10	—	ns	24.56
		010	$2t_{cyc}$-10	—		
		011	$4t_{cyc}$-10	—		
		100	$8t_{cyc}$-10	—		
CL2 high width	t_{WCL2H}	001	$0.5t_{cyc}$-15	—		
		010	t_{cyc}-15	—		
		011	$2t_{cyc}$-15	—		
		100	$4t_{cyc}$-15	—		
CL2 low width	t_{WCL2L}	001	$0.5t_{cyc}$-15	—		
		010	t_{cyc}-15	—		
		011	$2t_{cyc}$-15	—		
		100	$4t_{cyc}$-15	—		
Data setup time	t_{SD}	001	$0.5t_{cyc}$-15	—		
		010	t_{cyc}-15	—		
		011	$2t_{cyc}$-15	—		
		100	$4t_{cyc}$-15	—		
Data hold time	t_{HD}	001	$0.5t_{cyc}$ -15	—		
		010	t_{cyc}-15	—		
		011	$2t_{cyc}$-15	—		
		100	$4t_{cyc}$-15	—		
CL1 setup time	t_{SCL1}	001	$8.5t_{cyc}$-15	—		
		010	$17t_{cyc}$-15	—		
		011	$34t_{cyc}$-15	—		
		100	$68t_{cyc}$-15	—		
CL1 hold time	t_{HCL1}	001	$0.5t_{cyc}$-15	—		
		010	t_{cyc}-15	—		
		011	$2t_{cyc}$-15	—		
		100	$4t_{cyc}$-15	—		

Table 24.12 Supporting Module Signal Timing (LCD Controller: TFT/TFD Color Mode) (cont)

(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL1 high width	t_{WCL1H}	001	$8t_{cyc}$-15	—		
		010	$16t_{cyc}$-15	—		
		011	$32t_{cyc}$-15	—		
		100	$64t_{cyc}$-15	—		
FLM hold time	t_{HFLM}	001	$11.5t_{cyc}$-15	—		
		010	$23t_{cyc}$-15	—		
		011	$46t_{cyc}$-15	—		
		100	$92t_{cyc}$-15	—		
M delay time	t_{DM}		—	20		
DTM delay time	t_{DDTM}		—	20		

Notes: 1. Specification when the output position and width are set so that the CL1 clock is entirely within the horizontal retrace line interval.

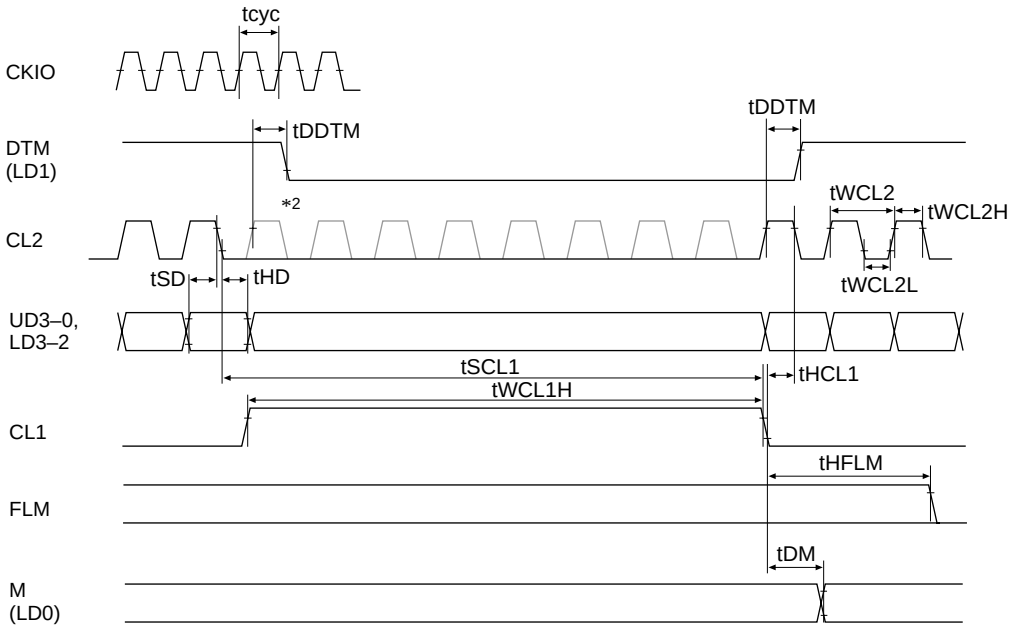
Fine tuning can be performed with a register setting to match the corresponding LCD driver.

2. CC1 = 1: CL1 masked during vertical retrace line interval

CC1 = 0: CL1 output

3. CC0 = 0: CL2 masked during horizontal retrace line interval

CC0 = 1: CL2 output



Notes: 1. When the horizontal retrace line interval is 8 dots, and the CL1 clock has the minimum settable width (8 dots).
2. Solid line: CC0= 0, Dotted line: CC0= 1
3. The CL1 clock is output during the horizontal retrace line interval.

Figure 24.56 TFT/TFD Color

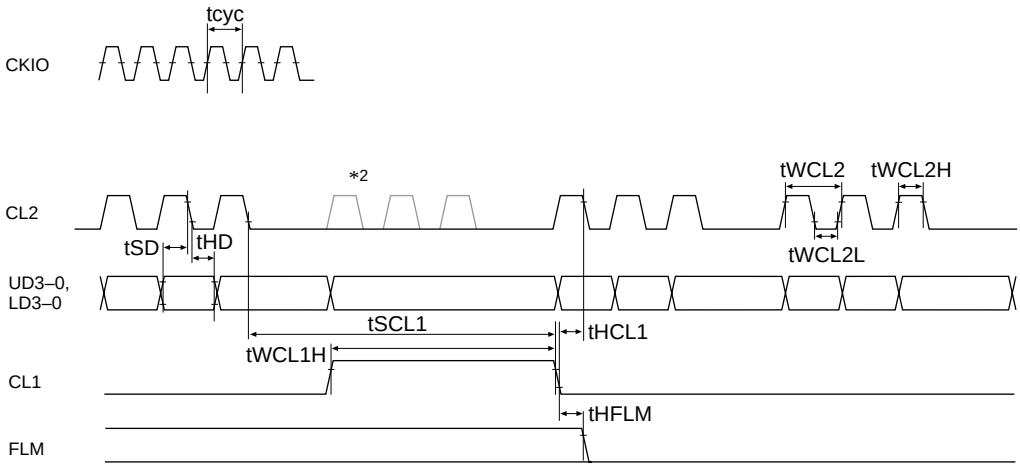
Table 24.13 Supporting Module Signal Timing (LCD Controller: STN Color Mode)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL2 cycle time	t_{WCL2}	001	$2t_{cyc}$-10	—	ns	24.57
		010	$4t_{cyc}$-10	—		
		011	$8t_{cyc}$-10	—		
		100	$16t_{cyc}$-10	—		
CL2 high width	t_{WCL2H}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		
CL2 low width	t_{WCL2L}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		
Data setup time	t_{SD}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		
Data hold time	t_{HD}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		
CL1 setup time	t_{SCL1}	001	$11t_{cyc}$-15	—		
		010	$22t_{cyc}$-15	—		
		011	$44t_{cyc}$-15	—		
		100	$88t_{cyc}$-15	—		
CL1 hold time	t_{HCL1}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		

Table 24.13 Supporting Module Signal Timing (LCD Controller: STN Color Mode) (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL1 high width	t_{WCL1H}	001	$8t_{cyc}$-15	—	ns	24.57
		010	$16t_{cyc}$-15	—		
		011	$32t_{cyc}$-15	—		
		100	$64t_{cyc}$-15	—		
FLM hold time	t_{HFLM}	001	t_{cyc}-15	—		
		010	$2t_{cyc}$-15	—		
		011	$4t_{cyc}$-15	—		
		100	$8t_{cyc}$-15	—		

Notes: 1. Specification when the output position and width are set so that the CL1 clock is entirely within the horizontal retrace line interval.
Fine tuning can be performed with a register setting to match the corresponding LCD driver.
2. CC1 = 1: CL1 masked during vertical retrace line interval
CC1 = 0: CL1 output
3. CC0 = 0: CL2 masked during horizontal retrace line interval
CC0 = 1: CL2 output



Notes: 1. When the horizontal retrace line interval is 8 dots, and the CL1 clock has the minimum settable width (8 dots).
2. Solid line: CC0= 0, Dotted line: CC0= 1
3. The CL1 clock is output during the horizontal retrace line interval.

Figure 24.57 STN Color

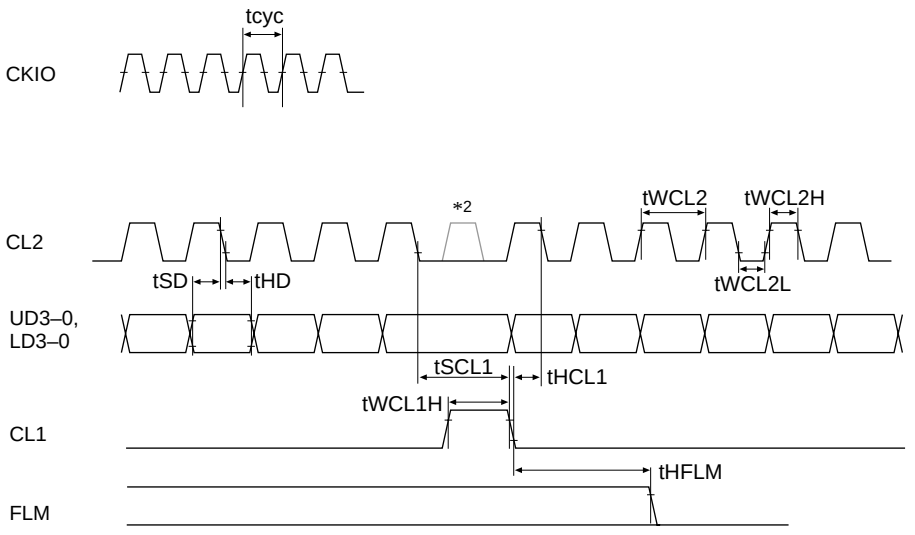
Table 24.14 Supporting Module Signal Timing (LCD Controller: Monochrome/Dual Screens/4-Bit Output Mode)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL2 cycle time	t_{WCL2}	000	$4t_{cyc}$-10		ns	24.58
		001	$8t_{cyc}$-10			
		010	$16t_{cyc}$-10			
		011	$32t_{cyc}$-10			
		100	$64t_{cyc}$-10			
CL2 high width	t_{WCL2H}	000	$2t_{cyc}$-15			
		001	$4t_{cyc}$-15			
		010	$8t_{cyc}$-15			
		011	$16t_{cyc}$-15			
		100	$32t_{cyc}$-15			
CL2 low width	t_{WCL2L}	000	$2t_{cyc}$-15			
		001	$4t_{cyc}$-15			
		010	$8t_{cyc}$-15			
		011	$16t_{cyc}$-15			
		100	$32t_{cyc}$-15			
Data setup time	t_{SD}	000	$2t_{cyc}$-15			
		001	$4t_{cyc}$-15			
		010	$8t_{cyc}$-15			
		011	$16t_{cyc}$-15			
		100	$32t_{cyc}$-15			
Data hold time	t_{HD}	000	$2t_{cyc}$-15			
		001	$4t_{cyc}$-15			
		010	$8t_{cyc}$-15			
		011	$16t_{cyc}$-15			
		100	$32t_{cyc}$-15			
CL1 setup time	t_{SCL1}	000	$6t_{cyc}$-15			
		001	$12t_{cyc}$-15			
		010	$24t_{cyc}$-15			
		011	$48t_{cyc}$-15			
		100	$96t_{cyc}$-15			

Table 24.14 Supporting Module Signal Timing (LCD Controller: Monochrome/Dual Screens/4-Bit Output Mode) (cont)
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.0 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –20 to 75°C)

Item	Symbol	DIV	Min	Max	Unit	Figure
CL1 hold time	t_{HCL1}	000	$2t_{cyc}$-15		ns	24.58
		001	$4t_{cyc}$-15			
		010	$8t_{cyc}$-15			
		011	$16t_{cyc}$-15			
		100	$32t_{cyc}$-15			
CL1 high width	t_{WCL1H}	000	$4t_{cyc}$-15			
		001	$8t_{cyc}$-15			
		010	$16t_{cyc}$-15			
		011	$32t_{cyc}$-15			
		100	$64t_{cyc}$-15			
FLM hold time	t_{HFLM}	000	$12.5t_{cyc}$-15			
		001	$25t_{cyc}$-15			
		010	$50t_{cyc}$-15			
		011	$100t_{cyc}$-15			
		100	$200t_{cyc}$-15			

Notes: 1. In monochrome 16-level gray scale mode, DIV cannot be set to 000.
2. Specification when the output position and width are set so that the CL1 clock is entirely within the horizontal retrace line interval.
Fine tuning can be performed with a register setting to match the corresponding LCD driver.
3. CC1 = 1: CL1 masked during vertical retrace line interval
CC1 = 0: CL1 output
4. CC0 = 0: CL2 masked during horizontal retrace line interval
CC0 = 1: CL2 output

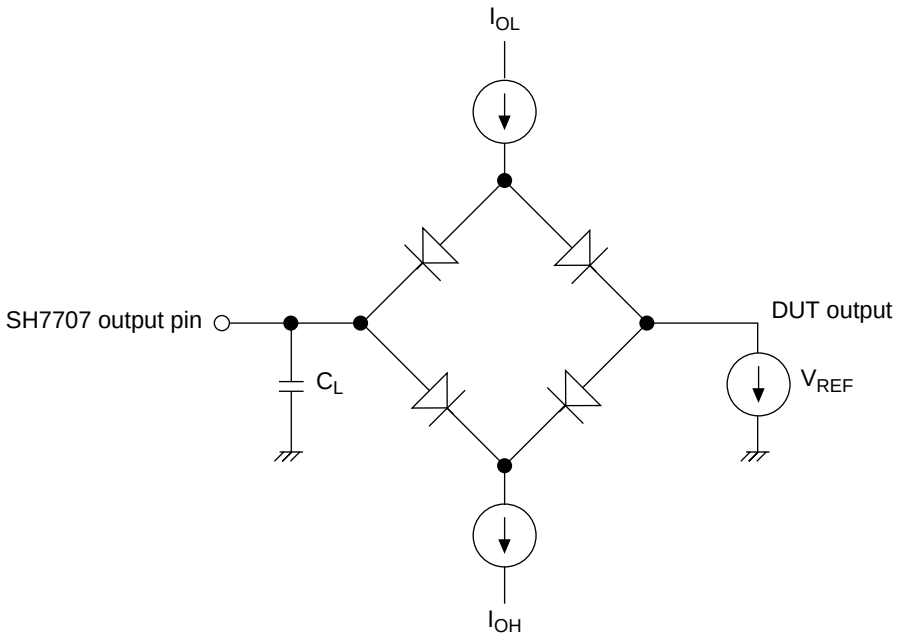


Notes: 1. When the horizontal retrace line interval is 8 dots, and the CL1 clock has the minimum settable width (8 dots).
2. Solid line: CC0= 0, Dotted line: CC0= 1
3. The CL1 clock is output during the horizontal retrace line interval.

Figure 24.58 Monochrome/Dual Screens/4-Bit Output

24.3.9 AC Characteristics Measurement Conditions

- I/O signal reference level: 1.5 V (V_{CC} = 3.3 ± 0.3 V)
- Input pulse level: V_{SS} to 3.0 V (where $\overline{\text{RESET}}$, $\overline{\text{BREQ}}$, NMI, $\overline{\text{IRQ5}}$–$\overline{\text{IRQ0}}$, CKIO, and MD5–MD0 are within V_{SS} to V_{CC})
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measuring instruments, etc., and is set as follows for each pin:
30 pF: CKIO, $\overline{\text{RASx}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$-$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
2. I_{OL} and I_{OH} are the values shown in table 3.2

Figure 24.59 Output Load Circuit

24.4 A/D Conversion Characteristics

Table 24.15 lists the A/D conversion characteristics.

Table 24.15 A/D Conversion Characteristics
(V_{CC} = 3.3 ± 0.3V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3 V, T_a = –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

24.5 D/A Conversion Characteristics

Table 24.16 lists the D/A conversion characteristics.

Table 24.16 D/A Conversion Characteristics
(V_{CC} = 3.3 ± 0.3 V, AV_{CC} = 3.3 ± 0.3 V, AV_{CC} = V_{CC} ± 0.3V, T_a = –20 to 75°C)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20 pF capacitive load
Absolute accuracy	—	±2.5	±4.0	LSB	2 MΩ resistive load

Appendix A Pin Functions

A.1 Pin States

Table A.1 shows pin states in resets, the power-down state, and the bus-released state.

Table A.1 Pin States in Resets, Power-Down State, 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	$\overline{\text{RESET}}$	I	I	I	I	I
	$\overline{\text{BREQ}}$	I	I	I	I	I
	$\overline{\text{BACK}}$	O	O	O	O	L
	MD[5:0]	I	I	I	I	I
	STATUS[1:0]/PTJ[7:6]	O	OP*[3]	OK*[3]	OP*[3]	OP*[3]
Interrupt	$\overline{\text{IRL}}$[3:0]/IRQ[3:0]/ PTH[3:0]	V*[2]	I	I	I	I
	IRQ4/$\overline{\text{ADTRG}}$/PTH[4]	V*[2]	I	I	I	I
	NMI	I	I	I	I	I
	$\overline{\text{IRQOUT}}$	O	O	O	O	O
Address bus	A[25:0]	O	O	Z	O	Z
Data bus	D[15:0]	Z	I	Z	IO	Z
	D[23:16]/PTA[7:0]/ PINT[7:0]	Z	IP*[3]	ZK*[3]	IOP*[3]	ZP*[3]
	D[31:24]/PTB[7;0]/ PINT[15:8]	Z	IP*[3]	ZK*[3]	IOP*[3]	ZP*[3]

Table A.1 Pin States in Resets, Power-Down State, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus-Released**
Bus control	$\overline{CS0}$	H	O	Z	O	Z
	$\overline{CS}$[2:4]/PTK[0:2]	H	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{CS5}$/$\overline{CE1A}$/PTK[3]	H	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{CS6}$/$\overline{CE1B}$	H	O	Z	O	Z
	$\overline{BS}$/PTK[4]	H	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{RAS}$/PTJ[0]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{RAS2}$/PTJ[1]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{CASLL}$/PTJ[2]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{CASLH}$/PTJ[3]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{CASHL}$/$\overline{CAS2L}$/PTJ[5]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{CASHH}$/$\overline{CAS2H}$/PTJ[5]	H	OP*[3]	ZOK*[4]	OP*[3]	ZOP*[4]
	$\overline{WE0}$	H	O	Z	O	Z
	$\overline{WE1}$/$\overline{WE}$	H	O	Z	O	Z
	$\overline{WE2}$/$\overline{ICIORD}$/PTK[6]	H	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{WE3}$/$\overline{ICIOWR}$/PTK[7]	H	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	RD/$\overline{WR}$	H	O	Z	O	Z
	$\overline{RD}$	H	O	Z	O	Z
	NC/PTK[5]	H	OP*[3]	OK*[3]	OP*[3]	OP*[3]
	$\overline{WAIT}$	Z	I	Z	I	Z
DMAC	$\overline{DREQ0}$/PTD[4]	V	ZI*[7]	Z	I	I
	$\overline{DACK0}$/PTD[5]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	$\overline{DREQ1}$/PTD[6]	V	ZI*[7]	Z	I	I
	$\overline{DACK1}$/PTD[7]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
Timer	TCLK/PTH[7]	V	ZP*[3]	ZK*[3]	IOP*[5]	IOP*[5]
SCI/smart card 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]
IRDA with FIFO	RxD1/SCPT[2]	I	I	I	I	I
	TxD1/SCPT[2]	Z	ZO*[7]	ZK*[3]	OZ*[6]	OZ*[6]
	SCK1/SCPT[3]	V	ZP*[3]	ZK*[3]	IOP*[5]	IOP*[5]

Table A.1 Pin States in Resets, Power-Down State, and Bus-Released State (cont)

		Reset		Power-Down		
Category	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus-Released**
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*[2]	I	I	I	I
PCMCIA control	PCCRD$\overline{WR}$/PTE[7]	V	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
PCC0: area 6	$\overline{PCCREG}$/PTE[6]	V	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
PCC1: area 5	$\overline{CE2B}$/PTE[5]	V	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{PCC0DRV}$/PTE[3]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	$\overline{PCC0WAIT}$/PTH[6]	V	I	Z	I	I
	PCC0WP/PTG[7]	V	I	Z	I	I
	PCC0READY/PTG[6]	V	I	Z	I	I
	PCC0BVD(1,2)/ PTG[5:4]	V	I	Z	I	I
	$\overline{PCC0CD(1,2)}$/PTG[3:2]	V	I	Z	I	I
	$\overline{PCC0VS(1,2)}$/PTG[1:0]	V	I	Z	I	I
	PCC0RESET/PTE[2]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	$\overline{CE2A}$/PTE[4]	V	OP*[3]	ZK*[3]	OP*[3]	ZP*[3]
	$\overline{PCC1DRV}$/PTE[1]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	$\overline{PCC1WAIT}$/PTH[5]	V	I	Z	I	I
	PCC1WP/PTF[7]	V	I	Z	I	I
	PCC1READY/PTF[6]	V	I	Z	I	I
	PCC1BVD(1,2)/ PTF[5:4]	V	I	Z	I	I
	$\overline{PCC1CD(1,2)}$/PTF[3:2]	V	I	Z	I	I
	$\overline{PCC1VS(1,2)}$/PTF[1:0]	V	I	Z	I	I
	PCC1RESET/PTE[0]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]

Table A.1 Pin States in Resets, Power-Down State, and Bus-Released State (cont)

		Reset		Power-Down		
Categoly	**Pin**	**Power-On Reset**	**Manual Reset**	**Standby**	**Sleep**	**Bus-Released**
LCD controller	UD[3:0]/PTC[7:4]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	LD[3:0]/PTC[3:0]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	CL(1,2)/PTD[3:2]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	FLM/PTD[1]	V	OP*[3]	ZK*[3]	OP*[3]	OP*[3]
	DON/PTD[0]	V	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]	OZ*[9]	IO*[8]	IO*[8]

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 pins are high-impedance, output pins retain their state
V: Input/output buffer off, pull-up MOS on

Notes:
1. Depends on the clock mode (MD2-MD0 setting).
2. The input Schmitt buffer is turned on for IRQ[5:0] and $\overline{\text{ADTRG}}$ when the pull-up MOS is on; for other inputs multiplexed on the same pins, the input buffer is turned off when the pull-up MOS is on.
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. Depends on register setting.
7. I or O when the port function is used.
8. When the port function is used, I. When the port function is not used, I or O depending on register setting.
9. When the DA function is selected as the pin function, O; otherwise, Z.

A.2 Pin Specifications

Table A.2 shows the pin specifications.

Table A.2 Pin Specifications

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
MD5	197	H02	I	Operating mode pin (endian mode)
MD4, MD3	196, 195	J01	I	Operating mode pin (area 0 bus width)
MD2–MD0	2, 1, 144	J02	I	Operating mode pin (clock mode)
$\overline{\text{RAS}}$/PTJ[0]	106	AJ28	I/O	RAS / I/O port
$\overline{\text{RAS2}}$/PTJ[1]	107	AH27	I/O	RAS / I/O port
$\overline{\text{CE2A}}$/PTE[4]	103	AG28	I/O	PCMCIA CE2A / I/O port
$\overline{\text{CE2B}}$/PTE[5]	104	AG29	I/O	PCMCIA CE2B / I/O port
RXD0/SCPT[0]	171	AA02	I	Serial port 0 data input / input port
RXD1/SCPT[2]	172	AA01	I	Serial port 1 data input / input port
RXD2/SCPT[4]	174	Y01	I	Serial port 2 data input / input port
TXD0/SCPT[0]	164	AE01	O	Serial port 0 data output / output port
TXD1/SCPT[2]	166	AD01	O	Serial port 1 data output / output port
TXD2/SCPT[4]	168	AC01	O	Serial port 2 data output / output port
SCK0/SCPT[1]	165	AD02	I/O	Serial port 0 clock input/output / I/O port
SCK1/SCPT[3]	167	AC02	I/O	Serial port 1 clock input/output / I/O port
SCK2/SCPT[5]	169	AB02	I/O	Serial port 2 clock input/output / I/O port
RTS2/SCPT[6]	170	AB01	I/O	Serial port 2 transmit request / I/O port
STATUS1/PTJ[7]	158	AH01	I/O	Processor status / I/O port
STATUS0/PTJ[6]	157	AH02	I/O	Processor status / I/O port
A25–A0	86, 84, 82, 80, 78–72, 70, 68–60, 58, 56–53	V29, U29, T29, R29, P29, P28, N29, N28, M29, M28, L29, K29, J29, J28, H29, H28, G29, G28, F29, F28, E29, D29, C29, C28, B29, B28	O	Address bus
D31–D24/ PINT[15]–PINT[8]/ PTB[7]–PTB[0]	13–18, 20, 22	B08, A08, B09, A09, B10, A10, A11, A12	I/O	Data bus / interrupt request pins / I/O port

Table A.2 Pin Specifications (cont)

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
D23–D16/ PINT[7]–PINT[0]/ PTA[7]–PTA[0]	23–26, 28, 30–32	B13, A13, B14, A14, A15, A16, B17, A17	I/O	Data bus / interrupt request pins / I/O port
D15–D0	34, 36–44, 46, 48–52	A18, A19, B20, A20, B21, A21, B22, A22, B23, A23, A24, A25, B26, A26, B27, A27	I/O	Data bus
UD[3:0]/PTC[7:4]	177–180	V02, V01, U02, U01	I/O	LCD data output / I/O port
CL1/PTD[3]	182	T01	I/O	LCD clock output / I/O port
CL2/PTD[2]	184	R01	I/O	LCD clock output / I/O port
LD[3:0]/PTC[3:0]	185–188	P02, P01, N02, N01	I/O	LCD data output / I/O port
FLM/PTD[1]	189	M02	I/O	LCD control pin / I/O port
DON/PTD[0]	190	M01	I/O	LCD control pin / I/O port
$\overline{\text{DREQ0}}$/PTD[4]	191	L02	I	DMA transfer request 0 / input port
$\overline{\text{DREQ1}}$/PTD[6]	192	L01	I	DMA transfer request 1 / input port
AN[5:0]/PTL[5:0]	204–199	E01, E02, F01, F02, G01, G02	I	Analog input pin / input port
AN[7:6]/DA[0:1]/ PTL[7:6]	207, 206	C02, D01	I/O	Analog input/output pin / input port
$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	102	AF29	O	Chip select 6 / PCMCIA CE1B
$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/ PTK[3]	101	AF28	I/O	Chip select 5 / PCMCIA CE2B / I/O port

Table A.2 Pin Specifications (cont)

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
$\overline{CS4}$/PTK[2]	100	AE29	I/O	Chip select 4 / I/O port
$\overline{CS3}$/PTK[1]	99	AE28	I/O	Chip select 3 / I/O port
$\overline{CS2}$/PTK[0]	98	AD29	I/O	Chip select 2 / I/O port
$\overline{CS0}$	96	AC29	O	Chip select 0
$\overline{BS}$/PTK[4]	87	W28	I/O	Bus cycle start / I/O port
$\overline{CASHH}$/$\overline{CAS2H}$/ PTJ[5]	113	AH24	I/O	D31–D24 selection CAS / D15–D8 selection CAS in area 2 / I/O port
$\overline{CASHL}$/$\overline{CAS2L}$/ PTJ[4]	112	AJ25	I/O	D23–D16 selection CAS / D7–D0 selection CAS in area 2 / I/O port
$\overline{CASLH}$/PTJ[3]	110	AJ26	I/O	D15–D8 selection CAS / I/O port
$\overline{CASLL}$/PTJ[2]	108	AJ27	I/O	D7–D0 selection CAS / I/O port
$\overline{DACK0}$/PTD[5]	114	AJ24	I/O	DMA transfer strobe 0 / I/O port
$\overline{DACK1}$/PTD[7]	115	AH23	I/O	DMA transfer strobe 1 / I/O port
$\overline{RD}$	88	W29	O	Read strobe pin
$\overline{WE0}$	89	Y28	O	D7–D0 select signal
$\overline{WE1}$/$\overline{WE}$	90	Y29	O	D15–D8 select signal / PCMCIA $\overline{WE}$ signal
$\overline{WE2}$/$\overline{ICIORD}$/ PTK[6]	91	AA28	I/O	D23–D16 select signal / PCMCIA $\overline{IORD}$ signal / I/O port
$\overline{WE3}$/$\overline{ICIOWR}$/ PTK[7]	92	AA29	I/O	D31–D24 select signal / PCMCIA $\overline{IOWR}$ signal / I/O port
RD/$\overline{WR}$	93	AB28	O	Read/write switchover signal
PCCRD$\overline{WR}$/ PTE[7]	94	AB29	I/O	PCMCIA read/write switchover signal
$\overline{PCCREG}$/PTE[6]	116	AJ23	I/O	PCMCIA REG pin / I/O port
$\overline{PCC0DRV}$/ PTE[3]	117	AH22	I/O	PCMCIA0 buffer control pin / I/O port
PCC0RESET/ PTE[2]	118	AJ22	I/O	PCMCIA0 reset output / I/O port
$\overline{PCC1DRV}$/ PTE[1]	119	AH21	I/O	PCMCIA1 buffer control pin / I/O port
PCC1RESET/ PTE[0]	120	AJ21	I/O	PCMCIA1 reset output / I/O port
$\overline{PCC0WAIT}$/ PTH[6]	124	AJ19	I	PCMCIA0 hardware wait request / input port

Table A.2 Pin Specifications (cont)

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
$\overline{\text{PCC1WAIT}}$/ PTH[5]	125	AH18	I	PCMCIA1 hardware wait request / input port
PCC0WP/ PTG[7]	126	AJ18	I	PCMCIA0 WP pin / input port
PCC0READY/ PTG[6]	127	AH17	I	PCMCIA0 BUSY/READY pin / input port
PCC0BVD1/ PTG[5]	128	AJ17	I	PCMCIA0 BVD1 pin / input port
PCC0BVD2/ PTG[4]	129	AH16	I	PCMCIA0 BVD2 pin / input port
$\overline{\text{PCC0CD1}}$/ PTG[3]	130	AJ16	I	PCMCIA0 CD1 pin / input port
$\overline{\text{PCC0CD2}}$/ PTG[2]	131	AH15	I	PCMCIA0 CD2 pin / input port
$\overline{\text{PCC0VS1}}$/ PTG[1]	133	AH14	I	PCMCIA0 VS1 pin / input port
$\overline{\text{PCC0VS2}}$/ PTG[0]	135	AH13	I	PCMCIA0 VS2 pin / input port
PCC1WP/ PTF[7]	136	AJ13	I	PCMCIA1 WP pin / input port
PCC1READY/ PTF[6]	137	AH12	I	PCMCIA1 BUSY/READY pin / input port
PCC1BVD1/ PTF[5]	138	AJ12	I	PCMCIA1 BVD1 pin / input port
PCC1BVD2/ PTF[4]	139	AH11	I	PCMCIA1 BVD2 pin / input port
$\overline{\text{PCC1CD1}}$/PTF[3]	140	AJ11	I	PCMCIA1 CD1 pin / input port
$\overline{\text{PCC1CD2}}$/PTF[2]	141	AH10	I	PCMCIA1 CD2 pin / input port
$\overline{\text{PCC1VS1}}$/PTF[1]	142	AJ10	I	PCMCIA1 VS1 pin / input port
$\overline{\text{PCC1VS2}}$/PTF[0]	143	AH09	I	PCMCIA1 VS2 pin / input port
$\overline{\text{WAIT}}$	123	AH19	I	Hardware wait request
$\overline{\text{BREQ}}$	122	AJ20	I	Bus request
$\overline{\text{BACK}}$	121	AH20	O	Bus acknowledge
$\overline{\text{IRQOUT}}$	160	AG01	O	Interrupt / refresh request notification

Table A.2 Pin Specifications (cont)

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
$\overline{\text{RESET}}$	193	K02	I	Reset
NMI	7	B05	I	Nonmaskable interrupt request
IRQ[3:0]/IRL[3:0]/ PTH[3:0]	11–8	B07, A06, B06, A05	I	External interrupt requests / external interrupt sources / input port
IRQ4/$\overline{\text{ADTRG}}$/ PTH[4]	12	A07	I	External interrupt request / ADC trigger request / input port
CTS2/IRQ5/ SCPT[7]	176	W01	I	Serial port 2 transmit enable / external interrupt request / input port
TCLK/PTH[7]	159	AG02	I/O	Clock input / output (for TMU/RTC) / I/O port
EXTAL	156	AJ03	I	External clock/crystal resonator pin
XTAL	155	AH03	O	Crystal resonator pin
CAP1	146	AJ08	—	External capacitance pin (for PLL1)
CAP2	149	AH06	—	External capacitance pin (for PLL2)
CKIO	162	AF01	I/O	System clock input/output
XTAL2	4	A03	O	Crystal resonator pin (for on-chip RTC)
EXTAL2	5	B04	I	Crystal resonator pin (for on-chip RTC)
NC/PTK[5]	105	AH28	I/O	I/O port
V_{CC}	21, 29, 35, 47, 59, 71, 81, 85, 97, 111, 134, 154, 163, 175, 183, 194	B12, B16, B19, B25, E28, L28, T28, V28, AD28, AH25, AJ14, AJ04, AE02, W02, R02, K01	Power supply	Power supply (3.3 V)
V_{CC} (RTC)	3	B03	Power supply	RTC oscillator power supply (3.3 V)
V_{CC} (PLL)	145, 150	AH08, AJ06	Power supply	PLL power supply (3.3 V)
AV_{CC}	205	D02	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	B11, B15, B18, B24, D28, K28, R28, U28, AC28, AH26, AJ15, AH04, AF02, Y02, T02	Power supply	Power supply (0 V)

Table A.2 Pin Specifications (cont)

Pin	Pin No. (FP-208A)	Pin No. (CSP-216)	I/O	Function
V_{SS} (RTC)	6	A04	Power supply	RTC oscillator power supply (0 V)
V_{SS} (PLL)	147, 148, 151, 152	AH07, AJ07, AH05, AJ05	Power supply	PLL power supply (0 V)
AV_{SS}	198, 208	H01, C01	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)

Pin		8-Bit Bus Width	16-Bit Bus Width		
		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	Low	Low	High	Low
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	High	High	Low	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Valid data	Invalid data	Valid data
D15–D8		High-Z[*2]	Invalid data	Valid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		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
$\overline{RAS}$		High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23–D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	Low	High	Low	Low
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	High	Low	High	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Invalid data	Valid data	Valid data
D15–D8		High-Z*2	Valid data	Invalid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		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
$\overline{RAS}$		High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23–D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31–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/Long-word Access**	**Byte Access (Address 2n)**	**Byte Access (Address 2n + 1)**	**Word/Longword Access**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Valid data	Invalid data	Valid data
D15–D8		High-Z[*2]	Invalid data	Valid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23–D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Invalid data	Valid data	Valid data
D15–D8		High-Z[*2]	Valid data	Invalid data	Valid data
D31–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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23–D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31–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)

Pin		16-Bit Bus Width (Area 3)			16-Bit Bus Width (Area 2)		
		Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		Low	Low	Low	High	High	High
$\overline{CASLL}$		Low	High	Low	High	High	High
$\overline{CASLH}$		High	Low	Low	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	Low	High	Low
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	Low	Low
$\overline{WE0}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{CE2A}$		High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	Low	Low	Low
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address
D7–D0		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data
D15–D8		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D31–D16		High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]

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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High	High
	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
$\overline{RAS}$		Low	Low	Low	Low	Low	Low	Low
$\overline{CASLL}$		Low	High	High	High	Low	High	Low
$\overline{CASLH}$		High	Low	High	High	Low	High	Low
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	Low	High	High	Low	Low
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	Low	High	Low	Low
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23–D16	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31–D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Note: 1. 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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		Low	Low	Low	High	High	High
$\overline{CASLL}$		High	Low	Low	High	High	High
$\overline{CASLH}$		Low	High	Low	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	Low	Low
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	Low	High	Low
$\overline{WE0}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
$\overline{CE2A}$		High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	Low	Low	Low
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address
D7–D0		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15–D8		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data
D31–D16		High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]	High-Z[*1]

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**
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High	High
	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
$\overline{RAS}$		Low	Low	Low	Low	Low	Low	Low
$\overline{CASLL}$		High	High	High	Low	High	Low	Low
$\overline{CASLH}$		High	High	Low	High	High	Low	Low
$\overline{CASHL}$/$\overline{CAS2L}$		High	Low	High	High	Low	High	Low
$\overline{CASHH}$/$\overline{CAS2H}$		Low	High	High	High	Low	High	Low
$\overline{WE0}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
$\overline{RAS2}$		High	High	High	High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$, $\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address
D7–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–D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23–D16	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31–D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Note: 1. Unused data pins should be switched to the port function, or pulled up or down.

Table A.9 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
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	Low	Low
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$		Disabled	Disabled	Disabled	Disabled
$\overline{PCC1WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Valid data	Invalid data	Valid data
D15–D8		High-Z*2	Invalid data	Valid data	Valid data
D31–D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.9 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
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	Low	Low	High	High	Low	Low
$\overline{RAS2}$		High	High	High	High	High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25–A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.9 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–D0	Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15–D8	High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data	Valid data
D31–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.10 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
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High
$\overline{CASLL}$		High	High	High	High
$\overline{CASLH}$		High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High
$\overline{WE0}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	Low	Low
$\overline{CE2B}$		High	High	High	High
$\overline{RAS2}$		High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$		Disabled	Disabled	Disabled	Disabled
$\overline{PCC1WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
PCC0WP		Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address
D7–D0		Valid data	Invalid data	Valid data	Valid data
D15–D8		High-Z*2	Valid data	Invalid data	Valid data
D31–D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.10 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
$\overline{CS6}$–$\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS}$		High	High	High	High	High	High	High	High
$\overline{CASLL}$		High	High	High	High	High	High	High	High
$\overline{CASLH}$		High	High	High	High	High	High	High	High
$\overline{CASHL}$/$\overline{CAS2L}$		High	High	High	High	High	High	High	High
$\overline{CASHH}$/$\overline{CAS2H}$		High	High	High	High	High	High	High	High
$\overline{WE0}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	Low	Low	High	High	Low	Low
$\overline{RAS2}$		High	High	High	High	High	High	High	High
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{PCC0WAIT}$		Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]	Enabled[*1]
$\overline{PCC1WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PCC0WP		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25–A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.10 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
D7–D0	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15–D8	High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data	Valid data
D31–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 Control Registers

B.1 Register Address Map

The address map of memory-mapped control registers is shown in Table B-1. The following module abbreviations are used.

MMU: Memory management unit
UBC: User break controller
CPG: Clock pulse generator
BSC: Bus state controller
RTC: Real-time clock
INTC: Interrupt controller
TMU: Timer unit
SCI: Serial communication interface controller
IRDA: Serial communication interface with IrDA
SCIF: Serial communication interface with FIFO
CCN: Cache controller
DMAC: Direct memory access controller
LCDC: LCD controller
ADC: Analog-to-digital converter
DAC: Digital-to-analog converter
PCC: PC card controller
PORT: Port controller

The Bus column shows the internal bus to which the control register is connected.

S: System bus, to which the CPU, cache, TLB, multiplier, and UBC are connected.
C: Cache bus, to which the BSC, LCDC, and DMAC cache are connected.
P: Peripheral bus, to which the BSC and supporting modules (RTC, INTC, TMU, SCI, IRDA, SCIF, ADC, DAC, PCC, PORT, DMAC, and LCDC) are connected.

The Size column shows the register size in bits.

The Access Size column shows the size used when the control register is accessed (read or written). If a size other than that indicated is used in an access, the result will be incorrect.

Registers located in the area from H'04000000 to H'0400015E are located in area 1 in physical space. Therefore, when the cache is turned on, either access these registers from the P2 area in virtual space, or make appropriate settings using the MMU to ensure that these registers are not cached.

Table B.1 Memory-Mapped Control Register Address Map

Register	Abbreviation	Module	Bus	Address	Size	Access Size
Page table entry high register	PTEH	MMU	S	H'FFFFFFF0	32	32
Page table entry low register	PTEL	MMU	S	H'FFFFFFF4	32	32
Translation table page register	TTB	MMU	S	H'FFFFFFF8	32	32
TLB exception address register	TEA	MMU	S	H'FFFFFFFC	32	32
MMU control register	MMUCR	MMU	S	H'FFFFFFE0	32	32
Break ASID register A	BASRA	UBC	S	H'FFFFFFE4	8	8
Break ASID register B	BASRB	UBC	S	H'FFFFFFE8	8	8
Cache control register	CCR	CCN	S	H'FFFFFFEC	32	32
TRAPA exception register	TRA	INTC	S	H'FFFFFFD0	32	32
Exception event register	EXPEVT	INTC	S	H'FFFFFFD4	32	32
Interrupt event register	INTEVT	INTC	S	H'FFFFFFD8	32	32
Break address register A	BARA	UBC	S	H'FFFFFFB0	32	32
Break address mask register A	BAMRA	UBC	S	H'FFFFFFB4	8	8
Break bus cycle register A	BBRA	UBC	S	H'FFFFFFB8	16	16
Break address register B	BARB	UBC	S	H'FFFFFFA0	32	32
Break address mask register B	BAMRB	UBC	S	H'FFFFFFA4	8	8
Break bus cycle register B	BBRB	UBC	S	H'FFFFFFA8	16	16
Break data register B	BDRB	UBC	S	H'FFFFFF90	32	32
Break data mask register B	BDMRB	UBC	S	H'FFFFFF94	32	32
Break control register	BRCR	UBC	S	H'FFFFFF98	16	16
Frequency control register	FRQCR	CPG	S	H'FFFFFF80	16	16
Standby control register	STBCR	CPG	S	H'FFFFFF82	8	8
Watchdog timer counter	WTCNT	CPG	S	H'FFFFFF84	8	16 (W), 8 (R)
Watchdog timer control/status register	WTCSR	CPG	S	H'FFFFFF86	8	16 (W), 8 (R)
Bus control register 1	BCR1	BSC	C	H'FFFFFF60	16	16
Bus control register 2	BCR2	BSC	C	H'FFFFFF62	16	16

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
Wait state control register 1	WCR1	BSC	C	H'FFFFFF64	16	16
Wait state control register 2	WCR2	BSC	C	H'FFFFFF66	16	16
Individual memory control register	MCR	BSC	C	H'FFFFFF68	16	16
DRAM control register	DCR	BSC	C	H'FFFFFF6A	16	16
PCMCIA control register	PCR	BSC	C	H'FFFFFF6C	16	16
Refresh timer control/status register	RTCSR	BSC	C	H'FFFFFF6E	16	16
Refresh timer counter	RTCNT	BSC	C	H'FFFFFF70	16	16
Refresh timer constant register	RTCOR	BSC	C	H'FFFFFF72	16	16
Refresh count register	RFCR	BSC	C	H'FFFFFF74	16	16
Bus control register 3	BCR3	BSC	C	H'FFFFFF7E	16	16
64 Hz counter	R64CNT	RTC	P	H'FFFFFEC0	8	8
Second counter	RSECCNT	RTC	P	H'FFFFFEC2	8	8
Minute counter	RMINCNT	RTC	P	H'FFFFFEC4	8	8
Hour counter	RHRCNT	RTC	P	H'FFFFFEC6	8	8
Day of week counter	RWKCNT	RTC	P	H'FFFFFEC8	8	8
Day counter	RDAYCNT	RTC	P	H'FFFFFECA	8	8
Month counter	RMONCNT	RTC	P	H'FFFFFECC	8	8
Year counter	RYRCNT	RTC	P	H'FFFFFECE	8	8
Second alarm register	RSECAR	RTC	P	H'FFFFFED0	8	8
Minute alarm register	RMINAR	RTC	P	H'FFFFFED2	8	8
Hour alarm register	RHRAR	RTC	P	H'FFFFFED4	8	8
Day of week alarm register	RWKAR	RTC	P	H'FFFFFED6	8	8
Day alarm register	RDAYAR	RTC	P	H'FFFFFED8	8	8
Month alarm register	RMONAR	RTC	P	H'FFFFFEDA	8	8
RTC control register 1	RCR1	RTC	P	H'FFFFFEDC	8	8
RTC control register 2	RCR2	RTC	P	H'FFFFFEDE	8	8

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
Interrupt control register	ICR	INTC	P	H'FFFFFEE0	16	16
Interrupt priority level setting register A	IPRA	INTC	P	H'FFFFFEE2	16	16
Interrupt priority level setting register B	IPRB	INTC	P	H'FFFFFEE4	16	16
Timer output control register	TOCR	TMU	P	H'FFFFFE90	8	8
Timer start register	TSTR	TMU	P	H'FFFFFE92	8	8
Timer constant register 0	TCOR0	TMU	P	H'FFFFFE94	32	32
Timer counter 0	TCNT0	TMU	P	H'FFFFFE98	32	32
Timer control register 0	TCR0	TMU	P	H'FFFFFE9C	16	16
Timer constant register 1	TCOR1	TMU	P	H'FFFFFEA0	32	32
Timer counter 1	TCNT1	TMU	P	H'FFFFFEA4	32	32
Timer control register 1	TCR1	TMU	P	H'FFFFFEA8	16	16
Timer constant register 2	TCOR2	TMU	P	H'FFFFFEAC	32	32
Timer counter 2	TCNT2	TMU	P	H'FFFFFEB0	32	32
Timer control register 2	TCR2	TMU	P	H'FFFFFEB4	16	16
Input capture register 2	TCPR2	TMU	P	H'FFFFFEB8	32	32
Serial mode register	SCSMR	SCI	P	H'FFFFFE80	8	8
Bit rate register	SCBRR	SCI	P	H'FFFFFE82	8	8
Serial control register	SCSCR	SCI	P	H'FFFFFE84	8	8
Transmit data register	SCTDR	SCI	P	H'FFFFFE86	8	8
Serial status register	SCSSR	SCI	P	H'FFFFFE88	8	8
Receive data register	SCRDR	SCI	P	H'FFFFFE8A	8	8
Smart card mode register	SCSCMR	SCI	P	H'FFFFFE8C	8	8

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
Interrupt event register 2	INTEVT2	INTC	P	H'04000000*	32	16, 32
Interrupt request register 0	IRR0	INTC	P	H'04000004*	8	8
Interrupt request register 1	IRR1	INTC	P	H'04000006*	8	8
Interrupt request register 2	IRR2	INTC	P	H'04000008*	8	8
Interrupt request register 3	IRR3	INTC	P	H'0400000A*	8	8
Interrupt request register 4	IRR4	INTC	P	H'0400000C*	8	8
Interrupt control register 1	ICR1	INTC	P	H'04000010*	16	16
Interrupt control register 2	ICR2	INTC	P	H'04000012*	16	16
PINT interrrupt enable register	PINTER	INTC	P	H'04000014*	16	16
Interrupt priority level setting register C	IPRC	INTC	P	H'04000016*	16	16
Interrupt priority level setting register D	IPRD	INTC	P	H'04000018*	16	16
Interrupt priority level setting register E	IPRE	INTC	P	H'0400001A*	16	16
Interrupt priority level setting register F	IPRF	INTC	P	H'0400001C*	16	16
DMA source address register 0	SAR0	DMAC	P	H'04000020*	32	16, 32
DMA destination address register 0	DAR0	DMAC	P	H'04000024*	32	16, 32
DMA transfer count register 0	DMATCR0	DMAC	P	H'04000028*	32	16, 32
DMA channel control register 0	CHCR0	DMAC	P	H'0400002C*	32	8, 16, 32
DMA source address register 1	SAR1	DMAC	P	H'04000030*	32	16, 32
DMA destination address register 1	DAR1	DMAC	P	H'04000034*	32	16, 32
DMA transfer count register 1	DMATCR1	DMAC	P	H'04000038*	32	16, 32
DMA channel control register 1	CHCR1	DMAC	P	H'0400003C*	32	8, 16, 32

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
DMA source address register 2	SAR2	DMAC	P	H'04000040*	32	16, 32
DMA destination address register 2	DAR2	DMAC	P	H'04000044*	32	16, 32
DMA transfer count register 2	DMATCR2	DMAC	P	H'04000048*	32	16, 32
DMA channel control register 2	CHCR2	DMAC	P	H'0400004C*	32	8, 16, 32
DMA source address register 3	SAR3	DMAC	P	H'04000050*	32	16, 32
DMA destination address register 3	DAR3	DMAC	P	H'04000054*	32	16, 32
DMA transfer count register 3	DMATCR3	DMAC	P	H'04000058*	32	16, 32
DMA channel control register 3	CHCR3	DMAC	P	H'0400005C*	32	8, 16, 32
DMA operation register	DMAOR	DMAC	P	H'04000060*	16	8, 16
A/D data register A (high)	ADDRAH	ADC	P	H'04000080*	8	8
A/D data register A (low)	ADDRAL	ADC	P	H'04000082*	8	8
A/D data register B (high)	ADDRBH	ADC	P	H'04000084*	8	8
A/D data register B (low)	ADDRBL	ADC	P	H'04000086*	8	8
A/D data register C (high)	ADDRCH	ADC	P	H'04000088*	8	8
A/D data register C (low)	ADDRCL	ADC	P	H'0400008A*	8	8
A/D data register D (high)	ADDRDH	ADC	P	H'0400008C*	8	8
A/D data register D (low)	ADDRDL	ADC	P	H'0400008E*	8	8
A/D control/status register	ADCSR	ADC	P	H'04000090*	8	8
A/D control register	ADCR	ADC	P	H'04000092*	8	8
D/A data register 0	DADR0	DAC	P	H'040000A0*	8	8
D/A data register 1	DADR1	DAC	P	H'040000A2*	8	8
D/A control register	DACR	DAC	P	H'040000A4*	8	8
Address register	LCDAR	LCDC	P	H'040000C0*	16	16
Display control register	LCDDR	LCDC	P	H'040000C2*	16	16
Palette register	LCDPR	LCDC	P	H'040000C6*	16	16

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
DMA control register	LCDDMR	LCDC	P	H'040000CE*	16	16
Area 6 interface status register	PCC0ISR	PCC	P	H'040000E0*	8	8
Area 6 general control register	PCC0GCR	PCC	P	H'040000E2*	8	8
Area 6 card status change register	PCC0CSCR	PCC	P	H'040000E4*	8	8
Area 6 card status change interrupt enable register	PCC0CSCIER	PCC	P	H'040000E6*	8	8
Area 5 interface status register	PCC1ISR	PCC	P	H'040000F0*	8	8
Area 5 general control register	PCC1GCR	PCC	P	H'040000F2*	8	8
Area 5 card status change register	PCC1CSCR	PCC	P	H'040000F4*	8	8
Area 5 card status change interrupt enable register	PCC1CSCIER	PCC	P	H'040000F6*	8	8
Port A control register	PACR	PORT	P	H'04000100*	16	16
Port B control register	PBCR	PORT	P	H'04000102*	16	16
Port C control register	PCCR	PORT	P	H'04000104*	16	16
Port D control register	PDCR	PORT	P	H'04000106*	16	16
Port E control register	PECR	PORT	P	H'04000108*	16	16
Port F control register	PFCR	PORT	P	H'0400010A*	16	16
Port G control register	PGCR	PORT	P	H'0400010C*	16	16
Port H control register	PHCR	PORT	P	H'0400010E*	16	16
Port J control register	PJCR	PORT	P	H'04000110*	16	16
Port K control register	PKCR	PORT	P	H'04000112*	16	16
Port L control register	PLCR	PORT	P	H'04000114*	16	16
Port SC control register	SCPCR	PORT	P	H'04000116*	16	16
Port A data register	PADR	PORT	P	H'04000120*	8	8
Port B data register	PBDR	PORT	P	H'04000122*	8	8
Port C data register	PCDR	PORT	P	H'04000124*	8	8
Port D data register	PDDR	PORT	P	H'04000126*	8	8

Table B.1 Memory-Mapped Control Register Address Map (cont)

Register	Abbreviation	Module	Bus	Address	Size	Access Size
Port E data register	PEDR	PORT	P	H"04000128*	8	8
Port F data register	PFDR	PORT	P	H'0400012A*	8	8
Port G data register	PGDR	PORT	P	H"0400012C*	8	8
Port H data register	PHDR	PORT	P	H'0400012E*	8	8
Port J data register	PJDR	PORT	P	H"04000130*	8	8
Port K data register	PKDR	PORT	P	H'04000132*	8	8
Port L data register	PLDR	PORT	P	H'04000134*	8	8
Port SC data register	SCPDR	PORT	P	H'04000136*	8	8
Serial mode register 1	SCSMR1	IRDA	P	H'04000140*	8	8
Bit rate register 1	SCBRR1	IRDA	P	H'04000142*	8	8
Serial control register 1	SCSCR1	IRDA	P	H'04000144*	8	8
Transmit FIFO data register 1	SCFTDR1	IRDA	P	H'04000146*	8	8
Serial status register 1	SCSSR1	IRDA	P	H'04000148*	16	16
Receive data FIFO register 1	SCFRDR1	IRDA	P	H'0400014A*	8	8
FIFO control register 1	SCFCR1	IRDA	P	H'0400014C*	8	8
FIFO data count register 1	SCFDR1	IRDA	P	H'0400014E*	16	16
Serial mode register 2	SCSMR2	SCIF	P	H"04000150*	8	8
Bit rate register 2	SCBRR2	SCIF	P	H'04000152*	8	8
Serial control register 2	SCSCR2	SCIF	P	H'04000154*	8	8
Transmit FIFO data register 2	SCFTDR2	SCIF	P	H'04000156*	8	8
Serial status register 2	SCSSR2	SCIF	P	H"04000158*	16	16
Receive data FIFO register 2	SCFRDR2	SCIF	P	H'0400015A*	8	8
FIFO control register 2	SCFCR2	SCIF	P	H0400015C*	8	8
FIFO data count register 2	SCFDR2	SCIF	P	H'0400015E*	16	16

Note: * These registers are located in area 1 in physical space. Therefore, when the cache is turned on, either access these registers from the P2 area in virtual space, or make appropriate settings using the MMU to ensure that these registers are not cached.

Table B.2 Registers Bits

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SCSMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	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
TCR1	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR2									TMU

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCNT2									TMU
TCR2	—	—	—	—	—	—	ICPF	UNF	TMU
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCPR2									TMU
R64CNT	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz	RTC
RSECCNT	—	10 sec	10 sec	10 sec	1 sec	1 sec	1 sec	1 sec	RTC
RMINCNT	—	10 min	10 min	10 min	1 min	1 min	1 min	1 min	RTC
RHRCNT	—	—	10 hour	10 hour	1 hour	1 hour	1 hour	1 hour	RTC
RWKCNT	—	—	—	—	—	day of week	day of week	day of week	RTC
RDAYCNT	—	—	10 days	10 days	1 day	1 day	1 day	1 day	RTC
RMONCNT	—	—	—	10 months	1 month	1 month	1 month	1 month	RTC
RYRCNT	10 years	10 years	10 years	10 years	1 year	1 year	1 year	1 year	RTC
RSECAR	ENB	10 sec	10 sec	10 sec	1 sec	1 sec	1 sec	1 sec	RTC
RMINAR	ENB	10 min	10 min	10 min	1 min	1 min	1 min	1 min	RTC
RHRAR	ENB	—	10 hour	10 hour	1 hour	1 hour	1 hour	1 hour	RTC
RWKAR	ENB	—	—	—	—	day of week	day of week	day of week	RTC
RDAYAR	ENB	—	10 days	10 days	1 day	1 day	1 day	1 day	RTC
RMONAR	ENB	—	—	10 months	1 month	1 month	1 month	1 month	RTC
RCR1	CF	—	—	CIE	AIE	—	—	AF	RTC
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	RTC
ICR0	NML	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IPRA	TMU0	TMU0	TMU0	TMU0	TMU1	TMU1	TMU1	TMU1	INTC
	TMU2	TMU2	TMU2	TMU2	RTC	RTC	RTC	RTC	
IPRB	WDT	WDT	WDT	WDT	REF	REF	REF	REF	INTC
	SCI	SCI	SCI	SCI	—	—	—	—	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BCR1	—	—	—	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1	BSC
	A5BST0	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	A5PCM	A6PCM	
BCR2	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0	BSC
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—	
BCR3	EXTEND	—	TPC31	TPC30	RCD31	RCD30	TRAS31	TRAS30	BSC
	—	—	TPC21	TPC20	RCD21	RCD20	TRAS21	TRAS20	
WCR1	—	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0	BSC
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0	
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	BSC
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0	
MCR	TPC1	TPC0	RCD1	RCD0	—	—	TRAS1	TRAS0	BSC
	—	BE	SZ	AMX1	AMX0	RFSH	RMODE	EDOMODE	
DCR	TPC1	TPC0	RCD1	RCD0	—	—	TRAS1	TRAS0	BSC
	—	BE	—	AMX1	AMX0	RFSH	RMODE	—	
PCR	—	—	—	—	—	—	—	—	BSC
	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH1	A6TEH0	
RTCSR	—	—	—	—	—	—	—	—	BSC
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS	
RTCNT	—	—	—	—	—	—	—	—	BSC
RTCOR	—	—	—	—	—	—	—	—	BSC
RFCR	—	—	—	—	—	—			BSC
FRQCR	—	—	—	—	—	—	SLPFRQ	CKOEN	CPG
	PLLEN	PSTBY	STC1	STC0	IFC1	IFC0	PFC1	PFC0	
STBCR	STBY	—	—	—	—	MSTP2	MSTP1	MSTP0	CPG
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	CPG
STBCR3	MSTPSLP0	—	—	—	—	—	—	CKSS	CPG
WTCNT									CPG
WTCSR	TME	WT/$\overline{IT}$	RSTS	WOFV	IOVF	CKS2	CKS1	CKS0	CPG

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BDRB									UBC
BDMRB									UBC
BRCR	CMFA	CMFB	—	—	—	PCBA	—	—	UBC
	DBEB	PCBB	—	—	SEQ	—	—	—	
BARB									UBC
BAMRB	—	—	—	—	—	BASMB	BAMB1	BAMB0	UBC
BBRB	—	—	—	—	—	—	—	—	UBC
	—	—	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BARA									UBC
BAMRA	—	—	—	—	—	BASMA	BAMA1	BAMA0	UBC
BBRA	—	—	—	—	—	—	—	—	UBC
	—	—	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
TRA	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—			
							—	—	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
EXPEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
INTEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
MMUCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
BASRA									UBC
BASRB									UBC
CCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	RA	0	CF	—	WT	CE	
PTEH									CCN
							—	—	
PTEL									CCN
							—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									CCN
TEA									CCN

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
INTEVT2	—	—	—	—	—	—	—	—	INTC
	—	—	—	—	—	—	—	—	
	—	—	—	—					
IRR0	PINT0R	PINT1R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R	INTC
IRR1	TXI1R	BRI1R	RXI1R	ERI1R	DEI3R	DEI2R	DEI1R	DEI0R	INTC
IRR2	—	—	LCDIR	ADIR	TXI2R	BRI2R	RXI2R	ERI2R	INTC
IRR3	—	PC0 SWIR	PC0 IRIR	PC0 SCIR	PC0 CDIR	PC0 RCIR	PC0 BWIR	PC0 BDIR	INTC
IRR4	—	—	—	PC1 SWIR	PC1 CDIR	PC1 RCIR	PC1 BWIR	PC1 BDIR	INTC
ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S	INTC
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
ICR2	INT15S	INT14S	INT13S	INT12S	INT11S	INT10S	INT9S	INT8S	INTC
	INT7S	INT6S	INT5S	INT4S	INT3S	INT2S	INT1S	INT0S	
INTER	INT15E	INT14E	INT13E	INT12E	INT11E	INT10E	INT9E	INT8E	INTC
	INT7E	INT6E	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E	
IPRC									INTC
IPRD									INTC
IPRE									INTC
IPRF									INTC
SAR0									DMAC

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DAR0									DMAC
DMATCR0	—	—	—	—	—	—	—	—	DMAC
CHCR0	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR1									DMAC
DAR1									DMAC
DMATCR1	—	—	—	—	—	—	—	—	DMAC
CHCR1	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR2									DMAC

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DAR2									DMAC
DMATCR2	—	—	—	—	—	—	—	—	DMAC
CHCR2	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	RO	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
SAR3									DMAC
DAR3									DMAC
DMATCR3	—	—	—	—	—	—	—	—	DMAC
CHCR3	—	—	—	—	—	—	—	—	DMAC
	—	—	—	DI	—	—	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	—	DS	TM	TS1	TS0	IE	TE	DE	
DMAOR	—	—	—	—	—	—	PR1	PR0	DMAC
	—	—	—	—	—		NMIF	DME	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	ADC
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADE	ADST	MULTI	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
DADR0									DAC
DADR1									DAC
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	DAC
LCDAR	—	—	—	—	—	—	—	—	LCDC
	—	—	—	A4	A3	A2	A1	A0	
LCDDR0	—	—	—	—	—	—	—	Don	
	—	—	—	—	—	—	—		
LCDDR1	—	—	—	—	—	DIV2	DIV1	DIV0	
	CC1	CC0	REN	GR1	GR0	LM2	LM1	LM0	
LCDDR2	Nhd7	Nhd6	Nhd5	Nhd4	Nhd3	Nhd2	Nhd1	Nhd0	
	Nht7	Nht6	Nht5	Nht4	Nht3	Nht2	Nht1	Nht0	
LCDDR3	—	—	—	—	Nhsw3	Nhsw2	Nhsw1	Nhsw0	
	Nhsp7	Nhsp6	Nhsp5	Nhsp4	Nhsp3	Nhsp2	Nhsp1	Nhsp0	
LCDDR4	—	—	—	—	—	—	Nvt9	Nvt8	
	Nvt7	Nvt6	Nvt5	Nvt4	Nvt3	Nvt2	Nvt1	Nvt0	
LCDDR5	—	—	—	—	—	—	Nvd9	Nvd8	
	Nvd7	Nvd6	Nvd5	Nvd4	Nvd3	Nvd2	Nvd1	Nvd0	
LCDDR6	Mw4	Mw3	Mw2	Mw1	Mw0	—	Nvsp9	Nvsp8	
	Nvsp7	Nvsp6	Nvsp5	Nvsp4	Nvsp3	Nvsp2	Nvsp1	Nvsp0	
LCDDR13	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	VI	VIE	
LCDPR0– LCDPR15	—	—	—	—	P11	P10	P9	P8	
	P7	P6	P5	P4	P3	P2	P1	P0	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
LCDDMR0	UMA15	UMA14	UMA13	UMA12	UMA11	UMA10	UMA9	UMA8	LCDC
	UMA7	UMA6	UMA5	UMA4	UMA3	UMA2	UMA1	UMA0	
LCDDMR1	UMA31	UMA30	UMA29	UMA28	UMA27	UMA26	UMA25	UMA24	
	UMA23	UMA22	UMA21	UMA20	UMA19	UMA18	UMA17	UMA16	
LCDDMR2	LMA15	LMA14	LMA13	LMA12	LMA11	LMA10	LMA9	LMA8	
	LMA7	LMA6	LMA5	LMA4	LMA3	LMA2	LMA1	LMA0	
LCDMR3	LMA31	LMA30	LMA29	LMA28	LMA27	LMA26	LMA25	LMA24	
	LMA23	LMA22	LMA21	LMA20	LMA19	LMA18	LMA17	LMA16	
LCDDMR4	LEM	—	TM13	TM12	TM11	TM10	TM9	TM8	
	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
PCC0ISR	P0RDY /IREQ	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2 SPKR	P0BVD1 STSCHG	PCMCIA
PCC0GCR	P0DRVE	P0PCCR	P0PCCT	—	P0MMOD	P0PA25	P0PA24	P0REG	PCMCIA
PCC0CSCR	P0SCDI	—	IREQ	SC	P0CDC	P0RC	P0BW	P0BD	PCMCIA
PCC0CSCIER	P0CRE	IREQE1	IREQE0	SCE	P0CDE	P0RE	P0BWE	P0BDE	PCMCIA
PCC1ISR	P1RDY	P1MWP	P1VS2	P1VS1	P1CD2	P1CD1	P1BVD2	P1BVD1	PCMCIA
PCC1GCR	P1DRVE	P1PCCR	—	—	P1MMOD	P1PA25	P1PA24	P1REG	PCMCIA
PCC1CSCR	P1SCDI	—	—	—	P1CDC	P1RC	P1BW	P1BD	PCMCIA
PCC1CSCIER	P1CRE	—	—	—	P1CDE	P1RE	P1BWE	P1BDE	PCMCIA
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	
PCCR	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
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
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	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PHCR	PH7MD1	PH7MD0	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0	PORT
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0	
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
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	PK1MD1	PK1MD0	PK0MD1	PK0MD0	
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
SCSMR1	IRM0D	ICK3	ICK2	ICK1	ICK0	PSEL	CKS1	CKS0	IRDA
SCBRR1									IRDA
SCSCR1	TIE	RIE	TE	RE	0	0	CKE1	CKE0	IRDA
SCFTDR1									IRDA
SCSSR1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	IRDA
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR1									IRDA
SCFCR1	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	IRDA
SCFDR1	0	0	0	T4	T3	T2	T1	T0	IRDA
	0	0	0	R4	R3	R2	R1	R0	

Table B.2 Registers Bits (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SCSMR2	0	CHR	PE	O/E	STOP	0	CKS1	CKS0	SCIF
SCBRR2									SCIF
SCSCR2	TIE	RIE	TE	RE	0	0	CKE1	CKE0	SCIF
SCFTDR2									SCIF
SCSSR2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	SCIF
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR2									SCIF
SCFCR2	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	SCIF
SCFDR2	0	0	0	T4	T3	T2	T1	T0	SCIF
	0	0	0	R4	R3	R2	R1	R0	

Legend

SCI: Serial communication interface
TMU: Timer unit
RTC: Real-time clock
INTC: Interrupt controller
BSC: Bus state controller
CPG: Clock pulse generator
UBC: User break controller
CCN: Cache controller unit

Table B.3 Register States in Reset and Power-Down States

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
CPU	R0–R15	Undefined	Undefined	Held	Held
	MACH, MACL	Undefined	Undefined	Held	Held
	PR	Undefined	Undefined	Held	Held
	PC	H'A0000000	H'A0000000	Held	Held
	SR	Initialized*1	Initialized*1	Held	Held
	SSR	Undefined	Undefined	Held	Held
	SPC	Undefined	Undefined	Held	Held
	GBR	Undefined	Undefined	Held	Held
	VBR	H'00000000	H'00000000	Held	Held
MMU	PTEH	Undefined	Undefined	Held	Held
	PTEL	Undefined	Undefined	Held	Held
	TTB	Undefined	Undefined	Held	Held
	TEA	Undefined	Undefined	Held	Held
	MMUCR	Initialized*2	Initialized*2	Held	Held
Cache	CCR	H'00000000	H'00000000	Held	Held
INTC	ICR0	H'8000/H'0000*3	H'8000/H'0000*3	Held	Held
	ICR1	H'4000	H'4000	Held	Held
	ICR2	H'0000	H'0000	Held	Held
	INTER	H'0000	H'0000	Held	Held
	IPRA	H'0000	H'0000	Held	Held
	IPRB	H'0000	H'0000	Held	Held
	IPRC	H'0000	H'0000	Held	Held
	IPRD	H'0000	H'0000	Held	Held
	IPRE	H'0000	H'0000	Held	Held
	IPRF	H'0000	H'0000	Held	Held
	IRR0	H'0000	H'0000	Held	Held
	IRR1	H'0000	H'0000	Held	Held
	IRR2	H'0000	H'0000	Held	Held
	IRR3	H'0000	H'0000	Held	Held
	IRR4	H'0000	H'0000	Held	Held

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
INTC	TRA	Undefined	Undefined	Held	Held
	EXPEVT	H'00000000	H'00000020	Held	Held
	INTEVT	Undefined	Undefined	Held	Held
	INTEVT2	Undefined	Undefined	Held	Held
UBC	BARA	Undefined	Held	Held	Held
	BASRA	Undefined	Held	Held	Held
	BAMRA	Undefined	Held	Held	Held
	BBRA	H'0000	H'0000	Held	Held
	BARB	Undefined	Held	Held	Held
	BAMRB	Undefined	Held	Held	Held
	BASRB	Undefined	Held	Held	Held
	BBRB	H'0000	H'0000	Held	Held
	BDMRB	Undefined	Held	Held	Held
	BDRB	Undefined	Held	Held	Held
	BRCR	H'0000	H'0000	Held	Held
CPG	STBCR	H'00	Held	Held	Held
	STBCR2	H'00	Held	Held	Held
	STBCR3	H'00	Held	Held	Held
	FRQCR	H'0102*4	Held	Held	Held
	WTCNT	H'00*4	Runs	Runs	Runs
	WTCSR	H'00*4	Runs	Runs	Runs
BSC	BCR1	H'0000	Held	Held	Held
	BCR2	H'3FF0	Held	Held	Held
	WCR1	H'3FF3	Held	Held	Held
	WCR2	H'FFFF	Held	Held	Held
	MCR	H'0000	Held	Held	Held
	DCR	H'0000	Held	Held	Held
	PCR	H'0000	Held	Held	Held
	RTCSR	H'0000	Runs	Held	Runs

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
BSC	RTCNT	H'0000	Runs	Held	Runs
	RTCOR	H'0000	Held	Held	Held
	RFCR	H'0000	Runs	Held	Runs
	BCR3	H'0000	Held	Held	Held
PCC	PCC0ISR	*9	*9	*9	*9
	PCC0GCR	H'00	Held	Held	Held
	PCC0CSCR	H'00	Held	Held	Held
	PCC0CSCIER	H'00	Held	Held	Held
	PCC1ISR	*9	*9	*9	*9
	PCC1GCR	H'00	Held	Held	Held
	PCC1CSCR	H'00	Held	Held	Held
	PCC1CSCIER	H'00	Held	Held	Held
DMAC	SAR0	Undefined	Undefined	Held	Held
	DAR0	Undefined	Undefined	Held	Held
	DMATCR0	Undefined	Undefined	Held	Held
	CHCR0	H'00000000	H'00000000	Held	Held
	SAR1	Undefined	Undefined	Held	Held
	DAR1	Undefined	Undefined	Held	Held
	DMATCR1	Undefined	Undefined	Held	Held
	CHCR1	H'00000000	H'00000000	Held	Held
	SAR2	Undefined	Undefined	Held	Held
	DAR2	Undefined	Undefined	Held	Held
	DMATCR2	Undefined	Undefined	Held	Held
	CHCR2	H'00000000	H'00000000	Held	Held
	SAR3	Undefined	Undefined	Held	Held
	DAR3	Undefined	Undefined	Held	Held
	DMATCR3	Undefined	Undefined	Held	Held
	CHCR3	H'00000000	H'00000000	Held	Held
	DMAOR	H'0000	H'0000	Held	Held

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
TMU	TOCR	H'00	H'00	Held	Held
	TSTR	H'00	H'00	Initialized/ Held*5	Held
	TCOR0	H'FFFFFFFF	H'FFFFFFFF	Held	Held
	TCNT0	H'FFFFFFFF	H'FFFFFFFF	Held/Runs*5	Runs
	TCR0	H'0000	H'0000	Held/Runs*5	Runs
	TCOR1	H'FFFFFFFF	H'FFFFFFFF	Held	Held
	TCNT1	H'FFFFFFFF	H'FFFFFFFF	Held/Runs*5	Runs
	TCR1	H'0000	H'0000	Held/Runs*5	Runs
	TCOR2	H'FFFFFFFF	H'FFFFFFFF	Held	Held
	TCNT2	H'FFFFFFFF	H'FFFFFFFF	Held/Runs*5	Runs
	TCR2	H'0000	H'0000	Held/Runs*5	Runs
	TCPR2	Undefined	Undefined	Held	Held
RTC	R64CNT	Undefined	Runs	Runs	Runs
	RSECCNT	Runs	Runs	Runs	Runs
	RMINCNT	Runs	Runs	Runs	Runs
	RHRCNT	Runs	Runs	Runs	Runs
	RWKCNT	Runs	Runs	Runs	Runs
	RDAYCNT	Runs	Runs	Runs	Runs
	RMONCNT	Runs	Runs	Runs	Runs
	RYRCNT	Runs	Runs	Runs	Runs
	RSECAR	Held*6	Held	Held	Held
	RMINAR	Held*6	Held	Held	Held
	RHRAR	Held*6	Held	Held	Held
	RWKAR	Held*6	Held	Held	Held
	RDAYAR	Held*6	Held	Held	Held
	RMONAR	Held*6	Held	Held	Held
	RCR1	H'00	Initialized*7	Held	Held
	RCR2	H'09	Initialized*8	Held	Held

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
SCI	SCSMR	H'00	H'00	H'00	H'00*10
	SCBRR	H'FF	H'FF	H'FF	H'FF*10
	SCSCR	H'00	H'00	H'00	H'00*10
	SCTDR	H'FF	H'FF	H'FF	H'FF*10
	SCSSR	H'84	H'84	H'84	H'84*10
	SCRDR	H'00	H'00	H'00	H'00*10
	SCSCMR	Initialized*11	Initialized*11	Initialized*11	Initialized*11
SCIF	SCSMR2	H'00	H'00	H'00	H'00*10
	SCBRR2	H'FF	H'FF	H'FF	H'FF*10
	SCSCR2	H'00	H'00	H'00	H'00*10
	SCFTDR2	—	—	—	—*10
	SCSSR2	H'0060	H'0060	H'0060	H'0060*10
	SCFRDR2	Undefined	Undefined	Undefined	Undefined*10
	SCFCR2	H'00	H'00	H'00	H'00*10
	SCFDR2	H'0000	H'0000	H'0000	H'0000*10
IRDA	SCSMR1	H'00	H'00	H'00	H'00*10
	SCBRR1	H'FF	H'FF	H'FF	H'FF*10
	SCSCR1	H'00	H'00	H'00	H'00*10
	SCFTDR1	—	—	—	—*10
	SCSSR1	H'0060	H'0060	H'0060	H'0060*10
	SCFRDR1	Undefined	Undefined	Undefined	Undefined*10
	SCFCR1	H'00	H'00	H'00	H'00*10
	SCFDR1	H'0000	H'0000	H'0000	H'0000*10
PORT	PACR	H'0000	Held	Held	Held
	PBCR	H'0000	Held	Held	Held
	PCCCR	H'AAAA	Held	Held	Held
	PDCR	H'AAAA	Held	Held	Held
	PECR	H'AAAA	Held	Held	Held
	PFCR	H'AAAA	Held	Held	Held

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
PORT	PGCR	H'AAAA	Held	Held	Held
	PHCR	H'AAAA	Held	Held	Held
	PJCR	H'0000	Held	Held	Held
	PKCR	H'0000	Held	Held	Held
	PLCR	H'0000	Held	Held	Held
	SCPCR	H'A888	Held	Held	Held
	PADR	H'00	Held	Held	Held
	PBDR	H'00	Held	Held	Held
	PCDR	H'00	Held	Held	Held
	PDDR	H'00	Held	Held	Held
	PEDR	H'00	Held	Held	Held
	PFDR	H'00	Held	Held	Held
	PGDR	H'00	Held	Held	Held
	PHDR	H'00	Held	Held	Held
	PJDR	H'00	Held	Held	Held
	PKDR	H'00	Held	Held	Held
	PLDR	H'00	Held	Held	Held
	SCPDR	H'00	Held	Held	Held
LCDC	LCDAR	H'0000	Held	Held	Runs
	LCDDR0	H'0000	Initialized	Held[*12]	Runs
	LCDDR1	H'0000	Held	Held	Runs
	LCDDR2	H'0000	Held	Held	Runs
	LCDDR3	H'0000	Held	Held	Runs
	LCDDR4	H'0000	Held	Held	Runs
	LCDDR5	H'0000	Held	Held	Runs
	LCDDR6	H'0000	Held	Held	Runs
	LCDDR13	H'0000	Held (bit 0) Initialized (bit 1)	Held	Runs
	LCDPR0– LCDPR15	H'0000	Held	Held	Runs
	LCDDMR0	H'0000	Held	Held	Runs
	LCDDMR1	H'0000	Held	Held	Runs
	LCDDMR2	H'0000	Held	Held	Runs
	LCDDMR3	H'0000	Held	Held	Runs

Table B.3 Register States in Reset and Power-Down States (cont)

		Reset States		Power-Down States	
Module	**Register**	**Power-On**	**Manual**	**Standby**	**Sleep**
ADC	ADDRAH	H'00	H'00	H'00	Held
	ADDRAL	H'00	H'00	H'00	Held
	ADDRBH	H'00	H'00	H'00	Held
	ADDRBL	H'00	H'00	H'00	Held
	ADDRCH	H'00	H'00	H'00	Held
	ADDRCL	H'00	H'00	H'00	Held
	ADDRDH	H'00	H'00	H'00	Held
	ADDRDL	H'00	H'00	H'00	Held
	ADCSR	H'00	H'00	H'00	Held
	ADCR	H'3F	H'3F	H'3F	Held
DAC	DADR0	H'00	H'00	Held	Held
	DADR1	H'00	H'00	Held	Held
	DACR	H'1F	H'1F	Held	Held

Notes: 1. MD = 1, RB = 1, BL = 1, I3–I0 = B'1111
M, Q, S, T are undefined.
2. The SV bit is undefined, other bits = 0.
3. H'8000: NMI pin is high / H'0000: NMI pin is low.
4. Initialized in a power-on reset via the $\overline{\text{RESET}}$ pin.
Held in a power-on reset via the WDT.
5. Depends on the count clock mode.
6. Only the ENB bit is cleared.
7. CF bit is undefined, other bits = 0.
8. RTCEN and START are held, other bits = 0.
9. Depends on the PC card status.
10. Held when SCI is operating, initialized when SCI is not used.
11. Bits 0, 2, and 3 are cleared, other bits are undefined.
12. The register set value is retained during the software standby interval, but the output from the DON pin becomes 0 regardless of the set value and is restored after standby is cleared.

Appendix C Load Time Variation Due to Load Capacitance

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated is connected to the SH7707's pins is shown below. The graph shown in figure C.1 should be taken into consideration if the stipulated capacitance is exceeded in connecting an external device.

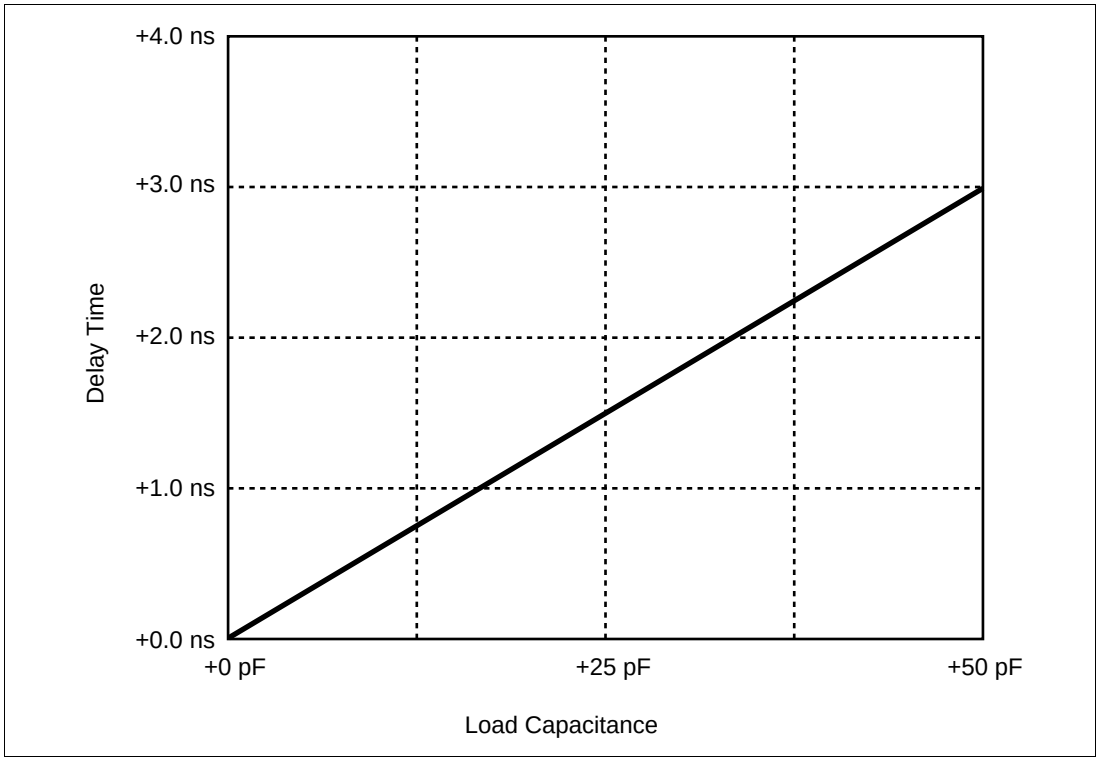


Figure C.1 Load Capacitance vs. Delay Time

Appendix D Package Dimensions

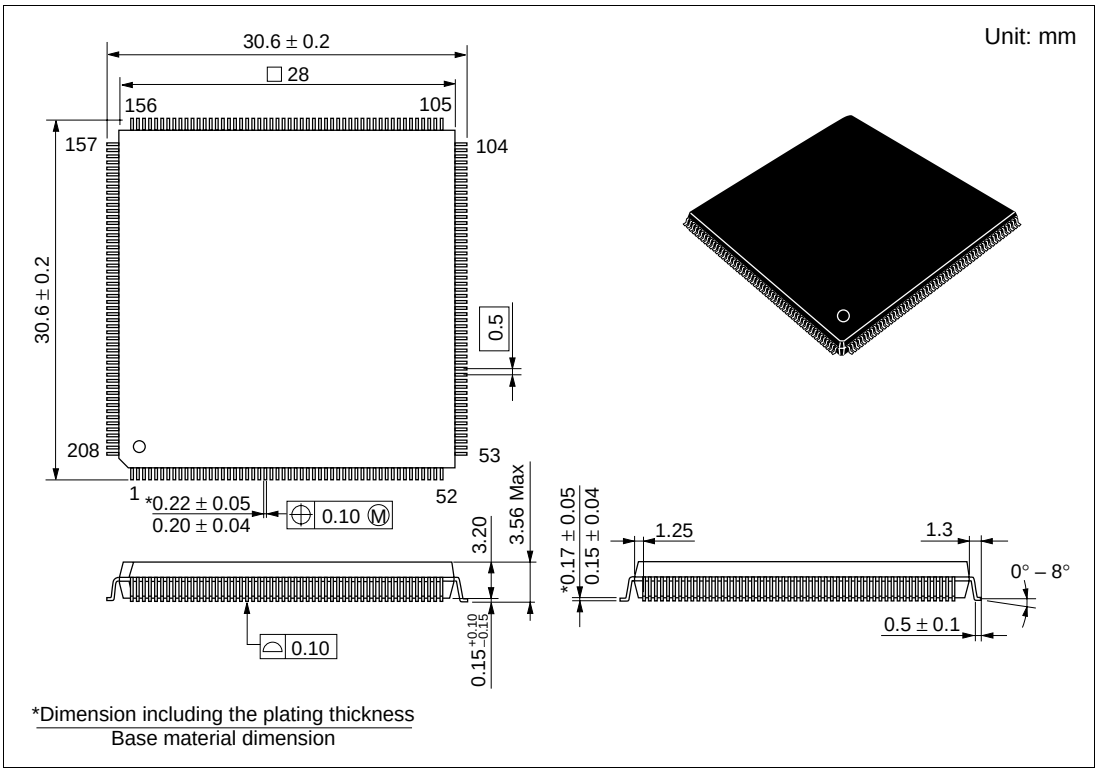


Figure D.1 Package Dimensions (FP-208A)

Figure D.2 Package Dimensions (CSP-216)

SH7707 Hardware Manual

Publication Date: 1st Edition, March 1998
Published by: Electronic Devices Business Group
Hitachi, Ltd.
Edited by: Technical Documentation Center
Hitachi ULSI Systems Co., Ltd.

HITACHI SEMICONDUCTOR TECHNICAL UPDATE

DATE	18 February 2000	No.	TN -SH7-212A /E
THEME	SH3 / SH3DSP have additional information on RTC		
CLASSIFICATION	☐ Spec. change ☐ Limitation on Use ■ Supplement of Documents		
PRODUCT NAME	HD6417708,HD6417708S,HD6417708R,HD6417718R,HD6417707,HD6417709,HD6417709A,HD6417729		
REFERENCE DOCUMENTS	SH7708 Series Hardware manual SH7709 Hardware manual Sh7729 Hardware manual SH7707 Hardware manual SH7709A Hardware manual	Effective Date	Eternity
		From	

HD6417708,HD6417708S,HD6417708R,HD6417718R,HD6417707,HD6417709,HD6417709A,
HD6417729 have followed additional information.

1. Additional information about Periodic interrupt on Real Time Clock (RTC)

Fig. 1 shows how to use the periodic interrupt function.
The periodic interrupt function can occur the interrupt periodically which designated by RCR2.PES bits. RCR2.PEF bit is set to 1 after the period designated by the RCR2.PES bits or written 1. When the PEF bit become 1, the periodic interrupt occurs. PEF bit should be set to 0 when setting PES bits or when the interrupt occurs.

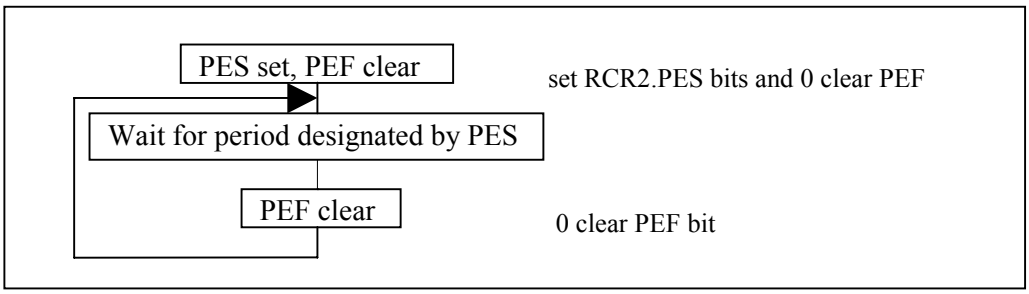


Fig.1 Using the Periodic Interrupt Function

HITACHI SEMICONDUCTOR TECHNICAL UPDATE

DATE	15 November 2000	No.	TN-SH7-278A/E
THEME	SH3, SH3-DSP An error in writing of "Mask All Interrupts (MAI) bit"		
CLASSIFICATION	☐ Spec. change ☐ Limitation on Use ■ Supplement of Documents		
PRODUCT NAME	SH7707, SH7709, SH7709A, SH7729		
REFERENCE DOCUMENTS	Hardware Manual s SH7707: ADE-602-137 Rev.1.0, SH7709: ADE-602-123B Rev.2.0 , SH7709A: ADE-602-187B Rev.3.0, SH7729: ADE-602-157A Rev.2.0	Effective Date From	Eternity Now

Hardware Manuals for SH7707,SH7709,SH7709A and SH7729 have following error of description.

SH7707 Manual(Rev1.0 or older): 6.3.3 Interrupt Control Register1 (ICR1)
SH7709 Manual(Rev2.0 or older): 6.3.3 Interrupt Control Register1 (ICR1)
SH7709A Manual(Rev3.0 or older): 6.3.3 Interrupt Control Register1 (ICR1)
SH7729 Manual(Rev2.0 or older): 6.3.3 Interrupt Control Register1 (ICR1)

Error:

Bit 15-Mask All Interrupts (MAI): Mask 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	

Correction:

Bit 15-Mask All Interrupts (MAI): When it is set to 1, all interrupt requests are masked when a low level is being input to the NMI pin. And it also mask NMI interrupt in standby mode.

Bit 15: MAI	Description	
0	When NMI pin is low level, all interrupt requests are not masked	(Initial value)
1	When NMI pin is low level, all interrupt requests are masked	