To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

RENESAS
Renesas Technology Corp.

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

User's Manual

32

SH7708 Series
SH7708, SH7708S, SH7708R

Hardware Manual

Renesas SuperH™ RISC engine

Renesas Electronics
www.renesas.com

Rev.9.0 2000.09

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

Preface

The SH7708, SH7708S, and SH7708R(SH7708 Series) use a RISC (reduced instruction set computer) type CPU to achieve high-performance computational processing. Also incorporating the peripheral functions required for system configuration plus power-down features essential for microcontroller application systems, the SH7708 Series is a new-generation RISC microcontroller (SuperH™ RISC engine).

The SH7708 Series has a RISC type instruction set, with basic instructions executed in one state, offering a drastic improvement in instruction execution speed. It also has an on-chip 32-bit multiplier (producing a 64-bit result) capable of high-speed multiply-and-accumulate operations. The SH7708 Series's instructions are upward-compatible with those of the SH-1 and SH-2, facilitating migration from these series to the SH7708 Series.

SH7708R is completely pin compatible with the SH7708S. On-chip supporting modules that enable a user system to be configured with a minimum of components include oscillation circuits, an interrupt controller (INTC), timers, a realtime clock (RTC), and a serial communication interface (SCI). A user break controller (UBC) is provided as an on-chip module supporting program development, allowing easy configuration of a simple debugger.

On-chip cache memory improves CPU processing performance, and a built-in memory management unit (MMU) performs address translation between a 4-gigabyte virtual space and physical space. An on-chip bus state controller (BSC) provides more efficient external memory access, and enables direct connection to synchronous DRAM, DRAM, and pseudo-SRAM without the need for glue logic.

This hardware manual describes the hardware of the SH7708 Series. Details of instructions can be found in the programming manual.

Note: * SuperH is a trademark of Hitachi, Ltd.

Related Manuals

SH7708Series instructions
SH-3/SH-3E/SH3-DSP Programming Manual

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

Revisions and Additions in this Edition

Page	Item	Description
338, 339	12.4.3 Usage Note about Periodic Interrupt on RTC	New section added
494 554	Figure 16.65 Output Load Circuits Figure 17.65 Output Load Circuits	[Former Edition] The mark of VRER = ↓ [This Edition] The mark of VRER = V

Contents

Section 1 Overview and Pin Functions.. 1
1.1 SH7708 Series Features.. 1
1.2 Block Diagram.. 6
1.3 Pin Description.. 7
1.3.1 Pin Arrangement.. 7
1.3.2 SH7708 Series Pin Functions.. 8

Section 2 CPU.. 13
2.1 Register Configuration.. 13
2.1.1 Privileged Mode and Banks.. 13
2.1.2 General Registers.. 16
2.1.3 System Registers.. 17
2.1.4 Control Registers.. 17
2.2 Data Formats.. 19
2.2.1 Data Format in Registers.. 19
2.2.2 Data Format in Memory.. 19
2.3 Instruction Features.. 20
2.3.1 Execution Environment.. 20
2.3.2 Addressing Modes.. 22
2.3.3 Instruction Formats.. 26
2.4 Instruction Set.. 29
2.4.1 Instruction Set Classified by Function.. 29
2.4.2 Instruction Code Map.. 45
2.5 Processor States and Processor Modes.. 48
2.5.1 Processor States.. 48
2.5.2 Processor Modes.. 49
2.6 Usage Note.. 50

Section 3 Memory Management Unit (MMU).. 51
3.1 Overview.. 51
3.1.1 Features.. 51
3.1.2 Role of MMU.. 51
3.1.3 SH7708 Series MMU.. 54
3.1.4 Register Configuration.. 57
3.2 Register Description.. 58
3.3 TLB Functions.. 60
3.3.1 Configuration of the TLB.. 60
3.3.2 TLB Indexing.. 62
3.3.3 TLB Address Comparison.. 63

3.3.4 Page Management Information 65
3.4 MMU Functions 66
3.4.1 MMU Hardware Management 66
3.4.2 MMU Software Management 66
3.4.3 MMU Instruction (LDLTB) 67
3.4.4 Avoiding Synonym Problems 68
3.5 MMU Exceptions 70
3.5.1 TLB Miss Exception 70
3.5.2 TLB Protection Violation Exception 71
3.5.3 TLB Invalid Exception 72
3.5.4 Initial Page Write Exception 73
3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error) 75
3.6 Memory-Mapped TLB 77
3.6.1 Address Array 77
3.6.2 Data Array 78
3.6.3 Usage Examples 80
3.7 Usage Note 80

Section 4 Exception Handling 81
4.1 Overview 81
4.1.1 Features 81
4.1.2 Register Configuration 81
4.2 Exception Handling Function 81
4.2.1 Exception Handling Flow 81
4.2.2 Exception Handling Vector Addresses 82
4.2.3 Acceptance of Exceptions 84
4.2.4 Exception Codes 86
4.2.5 Exception Request Masks 87
4.2.6 Returning from Exception Handling 88
4.3 Register Description 88
4.4 Exception Handler Operation 89
4.4.1 Reset 89
4.4.2 Interrupts 89
4.4.3 General Exceptions 89
4.5 Individual Exception Operations 90
4.5.1 Resets 90
4.5.2 General Exceptions 91
4.5.3 Interrupts 94
4.6 Cautions 95

Section 5 Cache 97
5.1 Overview 97

5.1.1 Features ... 97
5.1.2 Cache Structure ... 97
5.1.3 Register Configuration ... 99
5.2 Register Description ... 99
5.2.1 Cache Control Register (CCR) ... 99
5.3 Cache Operation ... 100
5.3.1 Searching the Cache ... 100
5.3.2 Read Access ... 102
5.3.3 Write Access ... 102
5.3.4 Write-Back Buffer ... 102
5.3.5 Coherency of Cache and External Memory ... 103
5.3.6 RAM Mode ... 103
5.4 Memory-Mapped Cache ... 103
5.4.1 Address Array ... 103
5.4.2 Data Array ... 104
5.5 Usage Examples ... 106
5.5.1 Invalidating Specific Entries ... 106
5.5.2 Reading the Data of a Specific Entry ... 106

Section 6 Interrupt Controller (INTC) ... 107
6.1 Overview ... 107
6.1.1 Features ... 107
6.1.2 Block Diagram ... 108
6.1.3 Pin Configuration ... 109
6.1.4 Register Configuration ... 109
6.2 Interrupt Sources ... 109
6.2.1 NMI Interrupts ... 110
6.2.2 IRL Interrupts ... 110
6.2.3 On-Chip Supporting Module Interrupts ... 112
6.2.4 Interrupt Exception Handling and Priority ... 112
6.3 INTC Registers ... 115
6.3.1 Interrupt Priority Registers A and B (IPRA, IPRB) ... 115
6.3.2 Interrupt Control Register (ICR) ... 116
6.4 INTC Operation ... 117
6.4.1 Interrupt Sequence ... 117
6.4.2 Multiple Interrupts ... 119
6.5 Interrupt Response Time ... 120

Section 7 User Break Controller (UBC) ... 123
7.1 Overview ... 123
7.1.1 Features ... 123
7.1.2 Block Diagram ... 123
7.1.3 Register Configuration ... 125

7.1.4 Break Conditions and Register Settings 125
7.2 UBC Register Functions 126
7.2.1 Break Address Register A (BARA) 126
7.2.2 Break Address Register B (BARB) 126
7.2.3 Break ASID Register A (BASRA) 127
7.2.4 Break ASID Register B (BASRB) 127
7.2.5 Break Address Mask Register A (BAMRA) 127
7.2.6 Break Address Mask Register B (BAMRB) 128
7.2.7 Break Bus Cycle Register A (BBRA) 128
7.2.8 Break Bus Cycle Register B (BBRB) 129
7.2.9 Break Data Register B (BDRB) 130
7.2.10 Break Data Mask Register B (BDMRB) 131
7.2.11 Break Control Register (BRCR) 132
7.3 UBC Operation 134
7.3.1 User Break Operation Flow 134
7.3.2 Instruction Fetch Cycle Break 135
7.3.3 Data Access Cycle Break 136
7.3.4 Saved Program Counter (PC) Value 137
7.3.5 Examples of Use 138
7.3.6 Cautions 140

Section 8 Power-Down Modes 141
8.1 Overview 141
8.1.1 Power-Down Modes 141
8.1.2 Register Configuration 142
8.1.3 Pin Configuration 143
8.2 Register Description 143
8.2.1 Standby Control Register (STBCR) 143
8.3 Sleep Mode 145
8.3.1 Transition to Sleep Mode 145
8.3.2 Canceling Sleep Mode 145
8.4 Standby Mode 145
8.4.1 Transition to Standby Mode 145
8.4.2 Canceling Standby Mode 146
8.4.3 Clock Pause Function 147
8.5 Module Standby Function 147
8.5.1 Transition to Module Standby Function 147
8.5.2 Clearing the Module Standby Function 148
8.6 Timing of STATUS Pin Changes 148
8.6.1 Timing for Resets 148
8.6.2 Timing for Canceling Standbys 150
8.6.3 Timing for Canceling Sleep Mode 151
8.7 Hardware Standby Mode 153

8.7.1 Transition to Hardware Standby Mode 153
8.7.2 Canceling Hardware Standby Mode 154
8.7.3 Hardware Standby Mode Timing 154

Section 9 On-Chip Oscillation Circuits 157
9.1 Overview 157
9.1.1 Features 157
9.2 Overview of the CPG 158
9.2.1 CPG Block Diagram 158
9.2.2 CPG Pin Configuration 161
9.2.3 CPG Register Configuration 161
9.3 Clock Operating Modes 162
9.4 Register Descriptions 170
9.4.1 Frequency Control Register (FRQCR) 170
9.5 Changing the Frequency 174
9.5.1 Changing the Multiplication Rate 174
9.5.2 Changing the Division Ratio 174
9.6 PLL Standby Function 175
9.6.1 Overview of the PLL Standby Function 175
9.6.2 Usage 175
9.7 Controlling Clock Output 176
9.7.1 Clock Modes 0–2 176
9.7.2 Clock Modes 3 and 4 176
9.8 Overview of the Watchdog Timer (WDT) 177
9.8.1 Block Diagram of the WDT 177
9.8.2 Register Configurations 177
9.9 WDT Registers 178
9.9.1 Watchdog Timer Counter (WTCNT) 178
9.9.2 Watchdog Timer Control/Status Register (WTCSR) 178
9.9.3 Notes on Register Access 180
9.10 Using the WDT 181
9.10.1 Canceling Standbys 181
9.10.2 Changing the Frequency 181
9.10.3 Using Watchdog Timer Mode 182
9.10.4 Using Interval Timer Mode 182
9.10.5 Usage Notes 183
9.11 Notes on Board Design 184

Section 10 Bus State Controller (BSC) 187
10.1 Overview 187
10.1.1 Features 187
10.1.2 Block Diagram 188
10.1.3 Pin Configuration 190

10.1.4 Register Configuration 192
10.1.5 Area Overview.. 193
10.1.6 PCMCIA Support 196
10.2 BSC Registers.. 200
10.2.1 Bus Control Register 1 (BCR1).. 200
10.2.2 Bus Control Register 2 (BCR2).. 203
10.2.3 Wait State Control Register 1 (WCR1) 204
10.2.4 Wait State Control Register 2 (WCR2) 206
10.2.5 Individual Memory Control Register (MCR) 209
10.2.6 DRAM Control Register (DCR) 214
10.2.7 PCMCIA Control Register (PCR)... 216
10.2.8 Synchronous DRAM Mode Register (SDMR) 217
10.2.9 Refresh Timer Control/Status Register (RTCSR).................................... 218
10.2.10 Refresh Timer Counter (RTCNT) 220
10.2.11 Refresh Time Constant Register (RTCOR).. 221
10.2.12 Refresh Count Register (RFCR) 221
10.2.13 Cautions on Accessing Refresh Control Related Registers.................... 222
10.3 BSC Operation... 223
10.3.1 Endian/Access Size and Data Alignment 223
10.3.2 Description of Areas.. 229
10.3.3 Basic Interface... 232
10.3.4 DRAM Interface 239
10.3.5 Synchronous DRAM Interface 255
10.3.6 Pseudo-SRAM Direct Connection 271
10.3.7 Burst ROM Interface... 280
10.3.8 PCMCIA Interface... 283
10.3.9 Waits between Access Cycles 295
10.3.10 Bus Arbitration 296
10.4 Usage Notes... 297
10.4.1 Area 2 and 3 Bus Width Setting 297
10.4.2 When Area 6 is Designated for PCMCIA, with a 16-Bit Bus Width........ 297
10.4.3 Self-Refreshing.. 298
10.4.4 PCMCIA Area Access... 299
10.4.5 Note on DRAM Connection for Areas 2 and 3.. 300

Section 11 Timer (TMU) 301
11.1 Overview.. 301
11.1.1 Features 301
11.1.2 Block Diagram... 301
11.1.3 Pin Configuration 303
11.1.4 Register Configuration 303
11.2 TMU Registers 304
11.2.1 Timer Output Control Register (TOCR) 304

11.2.2 Timer Start Register (TSTR) 305
11.2.3 Timer Control Register (TCR) 306
11.2.4 Timer Constant Register (TCOR) 309
11.2.5 Timer Counters (TCNT) 310
11.2.6 Input Capture Register (TCPR2) 311
11.3 TMU Operation 312
11.3.1 Overview 312
11.3.2 Basic Functions 312
11.4 Interrupts 316
11.4.1 Status Flag Set Timing 316
11.4.2 Status Flag Clear Timing 317
11.4.3 Interrupt Sources and Priorities 317
11.5 Usage Notes 318
11.5.1 Writing to Registers 318
11.5.2 Reading Registers 318
11.5.3 Clearing UNF in the TCR Register 318

Section 12 Realtime Clock (RTC) 319
12.1 Overview 319
12.1.1 Features 319
12.1.2 Block Diagram 319
12.1.3 Pin Configuration 321
12.1.4 RTC Register Configuration 322
12.2 RTC Registers 322
12.2.1 64-Hz Counter (R64CNT) 322
12.2.2 Second Counter (RSECCNT) 323
12.2.3 Minute Counter (RMINCNT) 323
12.2.4 Hour Counter (RHRCNT) 324
12.2.5 Day of the Week Counter (RWKCNT) 324
12.2.6 Date Counter (RDAYCNT) 325
12.2.7 Month Counter (RMONCNT) 326
12.2.8 Year Counter (RYRCNT) 326
12.2.9 Second Alarm Register (RSECAR) 327
12.2.10 Minute Alarm Register (RMINAR) 327
12.2.11 Hour Alarm Register (RHRAR) 328
12.2.12 Day of the Week Alarm Register (RWKAR) 328
12.2.13 Date Alarm Register (RDAYAR) 329
12.2.14 Month Alarm Register (RMONAR) 330
12.2.15 RTC Control Register 1 (RCR1) 330
12.2.16 RTC Control Register 2 (RCR2) 332
12.3 RTC Operation 333
12.3.1 Initial Settings of Registers after Power-On 333
12.3.2 Setting the Time 333

12.3.3 Reading the Time 335
12.3.4 Alarm Function 336
12.3.5 Crystal Oscillator Circuit 337
12.4 Usage Notes 338
12.4.1 Flag Clearing 338
12.4.2 Register Writes during RTC Count 338
12.4.3 Usage Note about Periodic Interrupt on RTC 338

Section 13 Serial Communication Interface (SCI) 341
13.1 Overview 341
13.1.1 Features 341
13.1.2 Block Diagram 342
13.1.3 Pin Configuration 343
13.1.4 Register Configuration 343
13.2 Register Descriptions 344
13.2.1 Receive Shift Register (SCRSR) 344
13.2.2 Receive Data Register (SCRDR) 344
13.2.3 Transmit Shift Register (SCTSR) 344
13.2.4 Transmit Data Register (SCTDR) 345
13.2.5 Serial Mode Register (SCSMR) 345
13.2.6 Serial Control Register (SCSCR) 348
13.2.7 Serial Status Register (SCSSR) 351
13.2.8 Serial Port Register (SCSPTR) 355
13.2.9 Bit Rate Register (SCBRR) 356
13.3 Operation 364
13.3.1 Overview 364
13.3.2 Operation in Asynchronous Mode 366
13.3.3 Multiprocessor Communication 376
13.3.4 Synchronous Operation 384
13.4 SCI Interrupt Sources 394
13.5 Usage Notes 394

Section 14 Smart Card Interface 399
14.1 Overview 399
14.1.1 Features 399
14.1.2 Block Diagram 400
14.1.3 Pin Configuration 401
14.1.4 Register Configuration 401
14.2 Register Descriptions 401
14.2.1 Smart Card Mode Register (SCSCMR) 402
14.2.2 Serial Status Register (SCSSR) 403
14.3 Operation 404
14.3.1 Overview 404

14.3.2 Pin Connections........ 405
14.3.3 Data Format........ 406
14.3.4 Register Settings........ 407
14.3.5 Clock........ 409
14.3.6 Data Transmission and Reception........ 412
14.4 Usage Notes........ 418
14.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode........ 418
14.4.2 Retransmission (Receive and Transmit Modes)........ 420

Section 15 I/O Ports........ 423
15.1 Overview........ 423
15.1.1 Features........ 423
15.1.2 Block Diagram........ 423
15.1.3 Pin Configuration........ 426
15.1.4 Register Configuration........ 427
15.2 Register Descriptions........ 427
15.2.1 Port Control Register (PCTR)........ 427
15.2.2 Port Data Register (PDTR)........ 428
15.2.3 Serial Port Register (SCSPTR)........ 429

Section 16 Electrical Characteristics (-SH7708, SH7708S-)........ 431
16.1 Absolute Maximum Ratings........ 431
16.2 DC Characteristics........ 432
16.3 AC Characteristics........ 433
16.3.1 Clock Timing........ 434
16.3.2 Control Signal Timing........ 444
16.3.3 AC Bus Timing Specifications........ 448
16.3.4 Basic Timing........ 452
16.3.5 Burst ROM Timing........ 455
16.3.6 DRAM Timing........ 458
16.3.7 Synchronous DRAM Timing........ 468
16.3.8 Pseudo-SRAM Timing........ 479
16.3.9 PCMCIA Timing........ 484
16.3.10 Peripheral Module Signal Timing........ 491
16.3.11 AC Characteristics Test Conditions........ 494

Section 17 Electrical Characteristics (-SH7708R-)........ 495
17.1 Absolute Maximum Ratings........ 495
17.2 DC Characteristics........ 496
17.3 AC Characteristics........ 497
17.3.1 Clock Timing........ 498
17.3.2 Control Signal Timing........ 504
17.3.3 AC Bus Timing Specifications........ 508

17.3.4 Basic Timing 512
17.3.5 Burst ROM Timing 515
17.3.6 DRAM Timing 518
17.3.7 Synchronous DRAM Timing... 528
17.3.8 Pseudo-SRAM Timing 539
17.3.9 PCMCIA Timing... 544
17.3.10 Peripheral Module Signal Timing 551
17.3.11 AC Characteristics Test Conditions 554

Appendix A Pin Functions 555
A.1 Pin States 555
A.2 Pin Specifications 558
A.3 Handling of Unused Pins 561
A.4 Pin States in Access to Each Address Space 562

Appendix B Control Registers 598
B.1 Register Address Map... 598
B.2 Register Bit List... 602
B.3 Register States in Reset and Power-Down States... 608

Appendix C Delay Time Variation Due to Load Capacitance 612

Appendix D Package Dimensions... 613

Section 1 Overview and Pin Functions

1.1 SH7708 Series Features

The SH7708, SH7708S, and SH7708R(SH7708 Series) are 32-bit RISC (reduced instruction set computer) microcomputers, featuring object code upward-compatibility with SH-1 and SH-2 microcomputers. The SH7708R is completely pin compatible with the SH7708S. It includes an 8-kbyte cache with a choice of write-back or write-through mode, and an MMU (memory management unit) with a 128-entry 4-way set associative TLB (translation lookaside buffer).

The SH7708 Series has an on-chip bus state controller (BSC) that allows direct connection to DRAM, synchronous DRAM (SDRAM), and pseudo-SRAM (PSRAM) without external circuitry. Its 16-bit fixed-length instruction set enables program code size to be reduced by almost 50% compared with 32-bit instructions.

The features of the SH7708 Series are summarized in table 1.1.

Table 1.1 SH7708 Series Features

Item	Features
CPU	• Original Hitachi SuperH RISC engine architecture • 32-bit internal data bus • General-register machine — Sixteen 32-bit general registers (eight 32-bit bank registers) — Five 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatibility with the SH-1 and SH-2 series) — Instruction length: 16-bit fixed length for improved code efficiency — Load-store architecture — Delayed branch instructions — C-oriented instruction set • Instruction execution time: one instruction/cycle for basic instructions • Logical address space: 4 Gbytes (448-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 by PLL synchronization • Processing states: — Power-on reset state — Manual reset state — Exception processing state — Program execution state — Power-down state — Bus-released state • Power-down modes: — Sleep mode — Standby mode — Hardware Standby mode(SH7708S, SH7708R only) • On-chip clock pulse generator • One watchdog timer channel

Table 1.1 SH7708 Series Features (cont)

Item	Features
Memory management unit (MMU)	• 4 Gbytes of address space, 256 address spaces (8-bit ASID) • Supports single virtual memory mode and multiple virtual memory mode • Paging system • 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	• Choice of operating mode — Normal mode (8-kbyte cache) — RAM mode (4-kbyte cache + 4-kbyte RAM) • Mixed instruction/data, 128 entries, 16-byte block length — 4-way set associative (8-kbyte cache) — 2-way set associative (4-kbyte cache) • Selectable write method (write-back/write-through), LRU (least recently used) replacement algorithm • Single-stage write-back buffer • Contents of TLB can be accessed directly by address mapping (can be used as on-chip memory)
Interrupt controller (INTC)	• 5 external interrupt pins (NMI, IRL0 to IRL3) • Encoded input of 15 external interrupt sources via pins IRL0 to IRL3 • On-chip peripheral interrupts: priority levels set for each module
User break controller (UBC)	• Supports debugging by user break interrupts • 2 break channels • Addresses, data values, type of access, and data size can all be set as break conditions • Supports a sequential break function

Table 1.1 SH7708 Series Features (cont)

Item	Features
Bus state controller (BSC)	• Supports external memory access — 32/16/8-bit external data bus • Physical address space divided into seven 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 DRAM, synchronous DRAM, pseudo-SRAM, and burst ROM — Supports fast page mode and EDO for DRAM — Supports PCMCIA interface — Outputs chip select signal (CS0–CS6) for corresponding area • DRAM/synchronous DRAM/pseudo-SRAM refresh function — Programmable refresh interval — Supports CAS-before-RAS refresh and self-refresh modes • DRAM/synchronous DRAM/pseudo-SRAM burst access function • Usable as either big- or little-endian machine
Timer	• 3-channel auto-reload type 32-bit timer • Input capture function • 6 types of counter input clock can be selected • Maximum resolution: 2 MHz
Realtime clock (RTC)	• On-chip clock and calendar functions • On-chip 32-kHz crystal oscillator circuit with a maximum resolution (interrupt cycle) of 1/256 second
Serial communication interface (SCI)	• Selection of asynchronous or synchronous mode • Full-duplex communication • Supports smart card interface
Package	• 144-pin plastic QFP(FP-144) • 144-pin plastic TQFP (TFP-144)*

Note: * SH7708S only

Table 1.1 SH7708 Series Features (cont)

Item	Features					
Product Line-up	**Product Number**	**On-chip Voltage**	**Operation Frequency**	**Mask Version**	**Model**	**Package**
	SH7708	3.3V±0.3V	60MHz	—	HD6417708F60	144-pin Plastic LQFP (FP-144F)
	SH7708S	3.3V±0.3V	60MHz	—	HD6417708SF60	
				—	HD6417708STF60	144-pin Plastic TQFP (TFP-144)
	SH7708R	3.15-3.6V (typ.)	100MHz	—	HD6417708RF100	144-pin Plastic L-QFP (FP-144F)
				A-mask	HD6417708RF100A	

1.2 Block Diagram

Figure 1.1 shows a block diagram of the SH7708 Series.

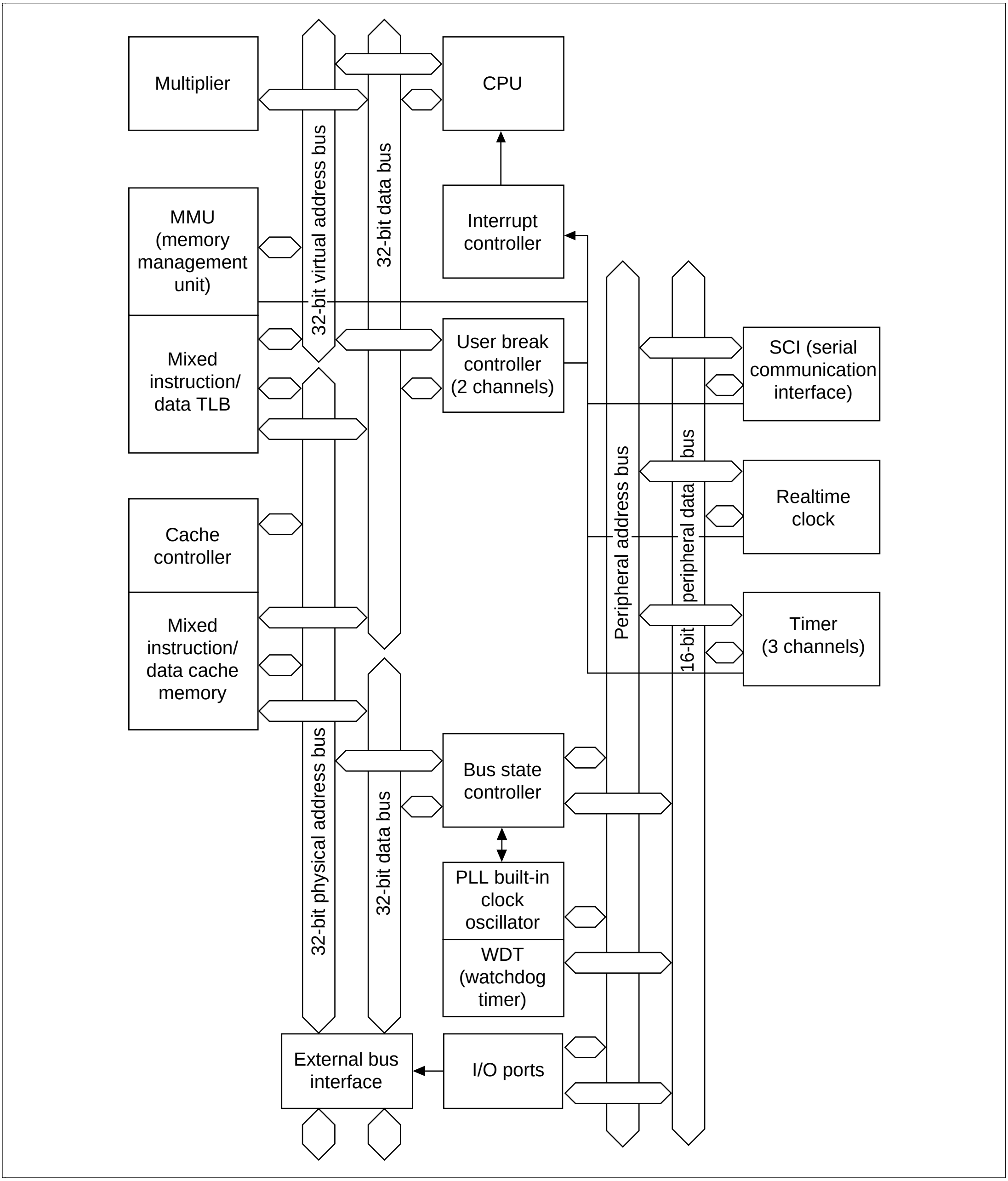
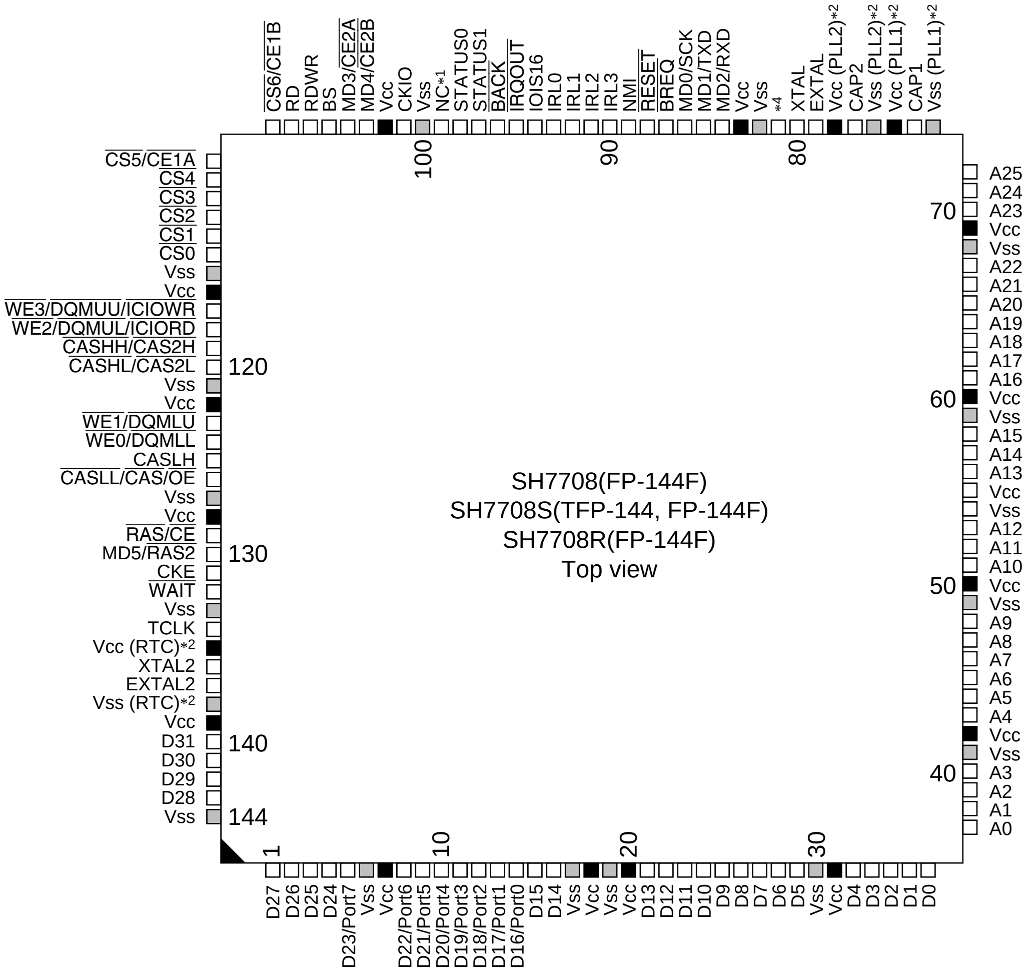

Multiplier
CPU
MMU (memory management unit)
Interrupt controller
Mixed instruction/ data TLB
User break controller (2 channels)
SCI (serial communication interface)
Cache controller
Realtime clock
Mixed instruction/ data cache memory
Timer (3 channels)
Bus state controller
PLL built-in clock oscillator
WDT (watchdog timer)
External bus interface
I/O ports
32-bit virtual address bus
32-bit data bus
32-bit physical address bus
32-bit data bus
Peripheral address bus
16-bit peripheral data bus

Figure 1.1 SH7708 Series Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement



Notes:
1. Make no connection.
2. Power supply pins for the on-chip RTC and on-chip PLL. These pins must be connected to the power supply even if the RTC or PLL are not used.
3. Power supply pins for the on-chip PLL. Except in hardware standby mode, these pins must be connected to the power supply even if the PLL is not used.
4. SH7708:Vcc
   SH7708S,SH7708R:CA

Figure 1.2 Pin Arrangement

1.3.2 SH7708 Series Pin Functions

Table 1.2 SH7708 Series Pin Functions

No.	Terminal	I/O	Description
1	D27	I/O	Data bus
2	D26	I/O	Data bus
3	D25	I/O	Data bus
4	D24	I/O	Data bus
5	D23/Port7	I/O	Data bus/port
6	V_{SS}	Power	Power (0 V)
7	V_{CC}	Power	Power (3.3 V)
8	D22/Port6	I/O	Data bus/port
9	D21/Port5	I/O	Data bus/port
10	D20/Port4	I/O	Data bus/port
11	D19/Port3	I/O	Data bus/port
12	D18/Port2	I/O	Data bus/port
13	D17/Port1	I/O	Data bus/port
14	D16/Port0	I/O	Data bus/port
15	D15	I/O	Data bus
16	D14	I/O	Data bus
17	V_{SS}	Power	Power (0 V)
18	V_{CC}	Power	Power (3.3 V)
19	V_{SS}	Power	Power (0 V)
20	V_{CC}	Power	Power (3.3 V)
21	D13	I/O	Data bus
22	D12	I/O	Data bus
23	D11	I/O	Data bus
24	D10	I/O	Data bus
25	D9	I/O	Data bus
26	D8	I/O	Data bus
27	D7	I/O	Data bus
28	D6	I/O	Data bus
29	D5	I/O	Data bus

Table 1.2 SH7708 Series Pin Functions (cont)

No.	Terminal	I/O	Description
30	V_{SS}	Power	Power (0 V)
31	V_{CC}	Power	Power (3.3 V)
32	D4	I/O	Data bus
33	D3	I/O	Data bus
34	D2	I/O	Data bus
35	D1	I/O	Data bus
36	D0	I/O	Data bus
37	A0	O	Address bus
38	A1	O	Address bus
39	A2	O	Address bus
40	A3	O	Address bus
41	V_{SS}	Power	Power (0 V)
42	V_{CC}	Power	Power (3.3 V)
43	A4	O	Address bus
44	A5	O	Address bus
45	A6	O	Address bus
46	A7	O	Address bus
47	A8	O	Address bus
48	A9	O	Address bus
49	V_{SS}	Power	Power (0 V)
50	V_{CC}	Power	Power (3.3 V)
51	A10	O	Address bus
52	A11	O	Address bus
53	A12	O	Address bus
54	V_{SS}	Power	Power (0 V)
55	V_{CC}	Power	Power (3.3 V)
56	A13	O	Address bus
57	A14	O	Address bus
58	A15	O	Address bus
59	V_{SS}	Power	Power (0 V)
60	V_{CC}	Power	Power (3.3 V)

Table 1.2 SH7708 Series Pin Functions (cont)

No.	Terminal	I/O	Description
61	A16	O	Address bus
62	A17	O	Address bus
63	A18	O	Address bus
64	A19	O	Address bus
65	A20	O	Address bus
66	A21	O	Address bus
67	A22	O	Address bus
68	V_{SS}	Power	Power (0 V)
69	V_{CC}	Power	Power (3.3 V)
70	A23	O	Address bus
71	A24	O	Address bus
72	A25	O	Address bus
73	V_{SS}(PLL1)*2	Power	Power (0 V) for PLL1
74	CAP1	O	External capacitance pin for PLL1
75	V_{CC}(PLL1)*2	Power	Power (3.3 V) for PLL1
76	V_{SS}(PLL2)*2	Power	Power (0 V) for PLL2
77	CAP2	O	External capacitance pin for PLL2
78	V_{CC}(PLL2)*2	Power	Power (3.3 V) for PLL2
79	EXTAL	I	External clock/crystal oscillator pin
80	XTAL	O	Crystal oscillator pin
81(SH7708)	V_{CC}	Power	Power(3.3V)
81(SH7708S, SH7708R)	CA	I	Chip active
82	V_{SS}	Power	Power (0 V)
83	V_{CC}	Power	Power (3.3 V)
84	MD2/RXD	I	Operating mode pin/serial data input
85	MD1/TXD	I/O	Operating mode pin/serial data output
86	MD0/SCK	I/O	Operating mode pin/serial clock
87	$\overline{\text{BREQ}}$	I	Bus request
88	$\overline{\text{RESET}}$	I	Reset
89	NMI	I	Nonmaskable interrupt request
90	$\overline{\text{IRL3}}$	I	External interrupt source input

Table 1.2 SH7708 Series Pin Functions (cont)

No.	Terminal	I/O	Description
91	$\overline{\text{IRL2}}$	I	External interrupt source input
92	$\overline{\text{IRL1}}$	I	External interrupt source input
93	$\overline{\text{IRL0}}$	I	External interrupt source input
94	$\overline{\text{IOIS16}}$	I	IO16-bit instruction
95	$\overline{\text{IRQOUT}}$	O	Bus request notification output
96	$\overline{\text{BACK}}$	O	Bus acknowledge
97	STATUS1	O	Processor status
98	STATUS0	O	Processor status
99	NC	O	Leave unconnected
100	V_{SS}	Power	Power (0 V)
101	CKIO	I/O	System clock I/O
102	V_{CC}	Power	Power (3.3 V)
103	MD4/$\overline{\text{CE2B}}$	I/O	Operating mode pin/PCMCIA CE pin
104	MD3/$\overline{\text{CE2A}}$	I/O	Operating mode pin/PCMCIA CE pin
105	$\overline{\text{BS}}$	O	Bus cycle start
106	RD/$\overline{\text{WR}}$	O	Read/write
107	$\overline{\text{RD}}$	O	Read pulse
108	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O	Chip select 6/PCMCIA CE pin
109	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	O	Chip select 5/PCMCIA CE pin
110	$\overline{\text{CS4}}$	O	Chip select 4
111	$\overline{\text{CS3}}$	O	Chip select 3
112	$\overline{\text{CS2}}$	O	Chip select 2
113	$\overline{\text{CS1}}$	O	Chip select 1
114	$\overline{\text{CS0}}$	O	Chip select 0
115	V_{SS}	Power	Power (0 V)
116	V_{CC}	Power	Power (3.3 V)
117	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$	O	D31–D24 selection signal/IO write
118	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$	O	D23–D16 selection signal/IO read
119	$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$	O	D31–D24/D15–D8 selection signal
120	$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$	O	D23–D16/D7–D0 selection signal
121	V_{SS}	Power	Power (0 V)

Table 1.2 SH7708 Series Pin Functions (cont)

No.	Terminal	I/O	Description
122	V_{CC}	Power	Power (3.3 V)
123	$\overline{\text{WE1}}$/DQMLU	O	D15–D8 selection signal
124	$\overline{\text{WE0}}$/DQMLL	O	D7–D0 selection signal
125	$\overline{\text{CASLH}}$	O	D15–D8 selection signal
126	$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/$\overline{\text{OE}}$	O	D7–D0 selection/memory selection signal
127	V_{SS}	Power	Power (0 V)
128	V_{CC}	Power	Power (3.3 V)
129	$\overline{\text{RAS}}$/$\overline{\text{CE}}$	O	RAS for DRAM, SDRAM/CE for PSRAM
130	MD5/$\overline{\text{RAS2}}$	I/O	Operating mode pin/RAS for DRAM
131	CKE	O	Clock enable control for SDRAM
132	$\overline{\text{WAIT}}$	I	Hardware wait request
133	V_{SS}	Power	Power (0 V)
134	TCLK	I/O	Clock I/O for TMU/RTC
135	V_{CC} (RTC)*3	Power	Power (3.3 V)
136	XTAL2	O	Crystal oscillator pin for on-chip RTC
137	EXTAL2	I	Crystal oscillator pin for on-chip RTC
138	V_{SS} (RTC)*3	Power	Power (0 V)
139	V_{CC}	Power	Power (3.3 V)
140	D31	I/O	Data bus
141	D30	I/O	Data bus
142	D29	I/O	Data bus
143	D28	I/O	Data bus
144	V_{SS}	Power	Power (0 V)

Notes: 1. With the SH7708S and SH7708R, except in hardware standby mode, all V_{CC} and V_{SS} pins should be connected to the system power supply (power should be supplied constantly). In hardware standby mode, power should be supplied at least to V_{CC} (RTC) and V_{SS} (RTC). If power is not supplied to V_{CC} and V_{SS} pins other than V_{CC} (RTC) and V_{SS} (RTC), hold the CA pin low. With the SH7708, it is not possible to supply power only to the RTC power supply.

2. Power should be supplied regardless of whether or not the on-chip PLL is used.

3. Power should be supplied regardless of whether or not the RTC is used.

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 SH7708 Series 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.

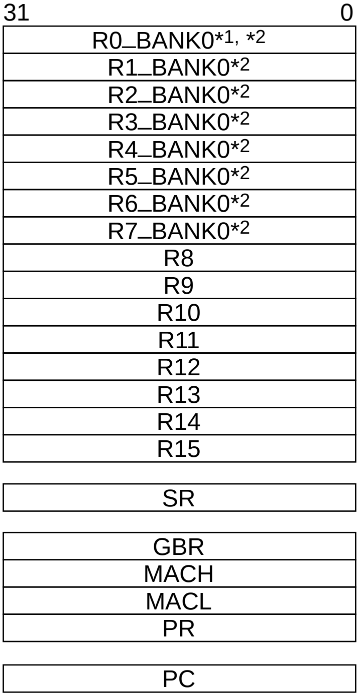
When the RB bit is 0, BANK0 general registers R0_BANK0–R7_BANK0 and nonbanked general registers R8–R15 function as the general register set, with BANK1 general registers R0_BANK1–R7_BANK1 accessed only by the LDC/STC instructions. In user mode, the 16 registers comprising bank 0 general registers R0_BANK0–R7_BANK0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

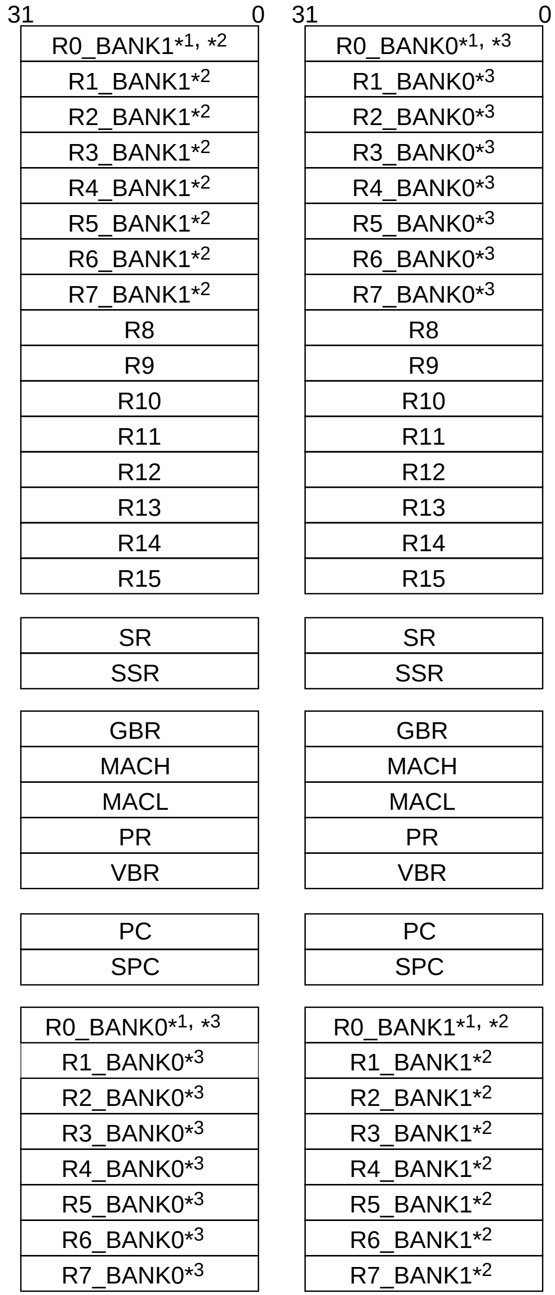


User mode register configuration

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

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

Figure 2.2 Privileged Mode Register Configuration

Register values after a reset are shown in table 2.1.

Table 2.1 Initial Register Values

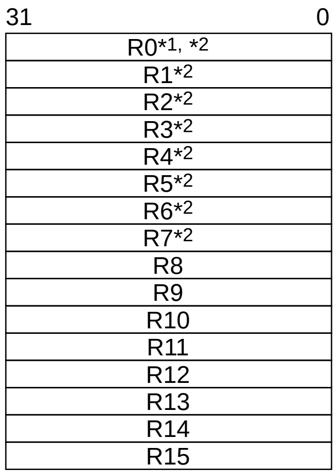
Type	Registers	Initial Value
General registers	R0 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3–I0 = 1111 (H'F), reserved bits = 0, others undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000

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

2.1.2 General Registers

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

General Registers



Notes:

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

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

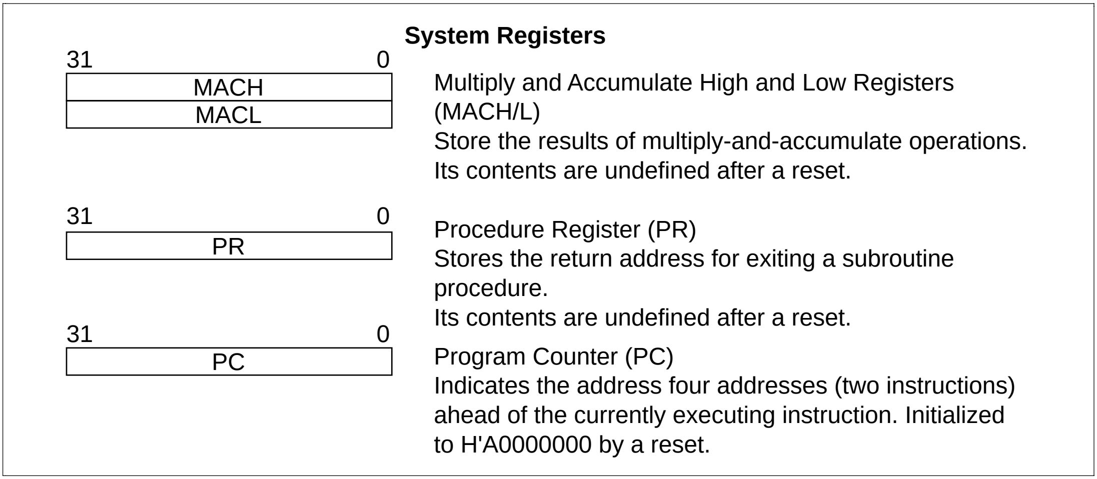
Figure 2.3 General Registers

2.1.3 System Registers

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

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

The system register configuration is shown in figure 2.4.



System Registers

Multiply and Accumulate High and Low Registers (MACH/L)
Store the results of multiply-and-accumulate operations.
Its contents are undefined after a reset.

Procedure Register (PR)
Stores the return address for exiting a subroutine procedure.
Its contents are undefined after a reset.

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

Figure 2.4 System Registers

2.1.4 Control Registers

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

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

31	0
SSR	

Saved Status Register (SSR)
Stores current SR value at time of exception to indicate processor status in return to instruction stream from exception handler.
Its contents are undefined after a reset.

31	0
SPC	

Saved Program Counter (SPC)
Stores current PC value at time of exception to indicate return address at completion of exception handling.
Its contents are undefined after a reset.

31	0
GBR	

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.

31	0
VBR	

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

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

Status register (SR)

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.

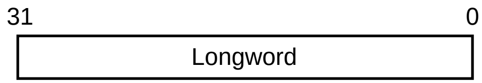


Figure 2.6 Longword

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.

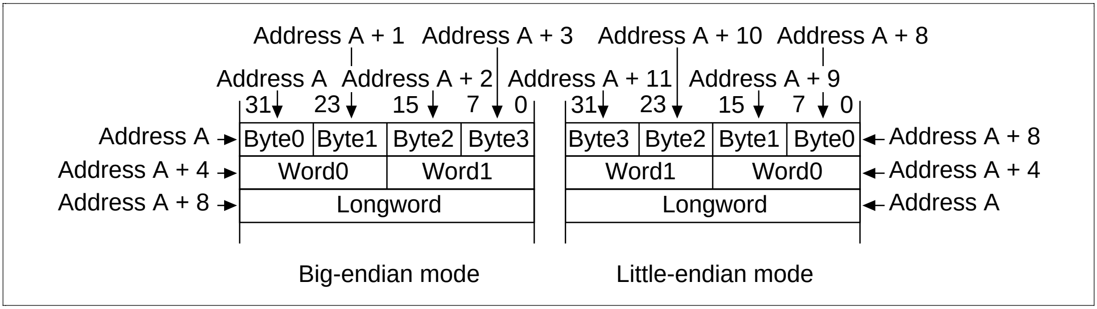


Figure 2.7 Byte, Word, and Longword Alignment

2.3 Instruction Features

2.3.1 Execution Environment

Data Length: The SH7708 Series 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 SH7708 Series features a load-store architecture in which basic operations are executed in registers. Operations requiring memory access are executed in registers following register loading, except for bit-manipulation operations such as logical AND functions, which are executed directly in memory.

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

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

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

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

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

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

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

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

2.3.2 Addressing Modes

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

Table 2.2 Addressing Modes and Effective Addresses

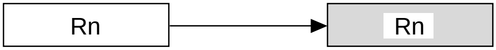
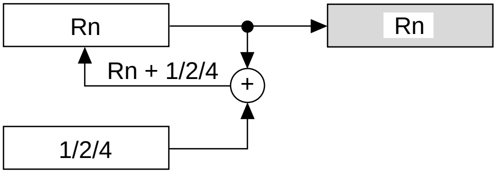
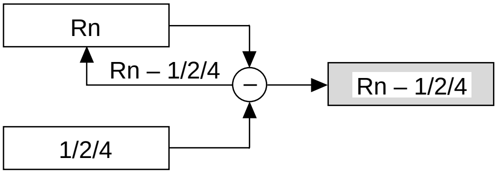
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents.  	Rn
Register indirect with post-increment	@Rn+	Effective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

Table 2.2 Addressing Modes and Effective Addresses (cont)

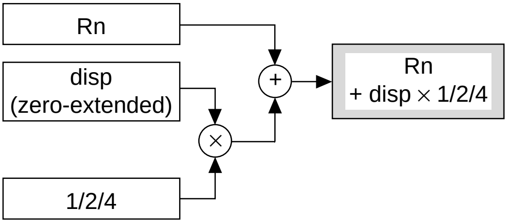
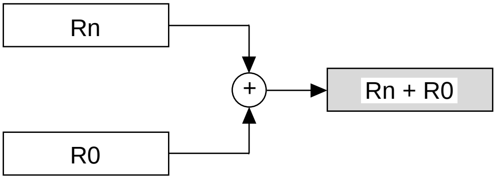
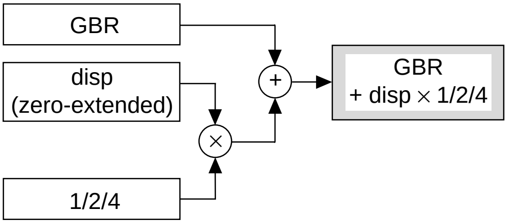
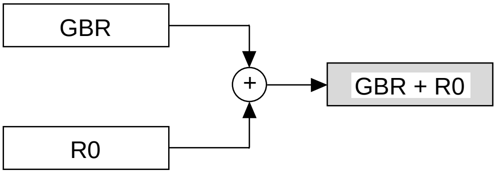
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword),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

Table 2.2 Addressing Modes and Effective Addresses (cont)

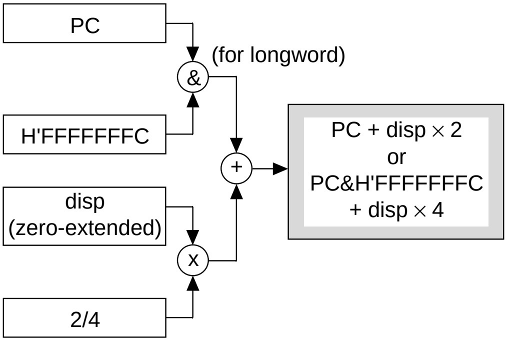
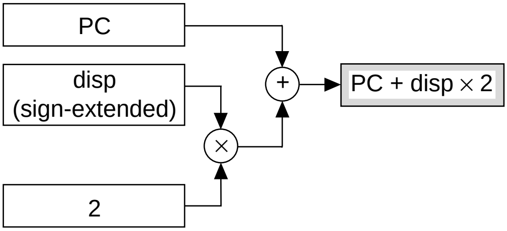
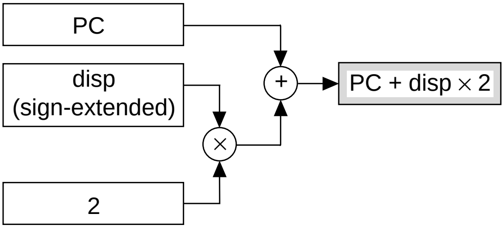
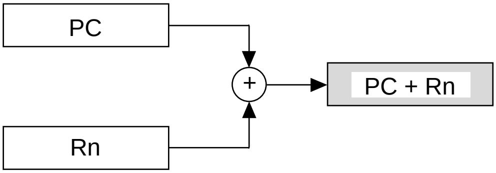
Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative with displacement	@(disp:8, PC)	Effective address is register PC contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word), or 4 (longword), according to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

Table 2.2 Addressing Modes and Effective Addresses (cont)

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

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

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

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

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

disp:8, disp:12; PC-relative

2.3.3 Instruction Formats

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

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

Table 2.3 Instruction Formats

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

Table 2.3 Instruction Formats (cont)

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

Table 2.3 Instruction Formats (cont)

Instruction Format		Source Operand	Destination Operand	Instruction Example
nmd format	15 0 xxxx nnnn mmmm dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
		mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn
d format	15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
		R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
		dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
		dddddddd: PC-relative	—	BF label
d12 format	15 0 xxxx dddd dddd dddd	dddddddddddd: PC-relative	—	BRA label (label = disp + PC)
nd8 format	15 0 xxxx nnnn dddd dddd	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i format	15 0 xxxx xxxx i i i i i i i i	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm, @(R0,GBR)
		iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
		iiiiiiii: immediate	—	TRAPA #imm
ni format	15 0 xxxx nnnn i i i i i i i i	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 SH7708 Series 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	33
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow check	
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
		SHAD	Dynamic arithmetic shift	
		SHLD	Dynamic logical shift	

Table 2.4 Classification of Instructions (cont)

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

Table 2.5 lists the SH7708 Series 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	→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Direction of transfer Memory operand Flag bits in SR Logical AND of each bit Logical OR of each bit Exclusive OR of each bit Logical NOT of each bit n-bit shift
Privileged mode		Indicates whether privileged mode applies
Execution cycles		Value when no wait states are inserted The execution cycles listed in the table are minimums. The actual number of cycles may be increased in cases such as the following: 1. When contention occurs between instruction fetches and data access 2. When the destination register of the load instruction (memory → register) and the register used by the next instruction are the same
T bit		Value of T bit after instruction is executed —: No change

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

Table 2.6 lists the SH7708 Series 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 → Rn	0110nnnnmmmm1000	—	1	—
SWAP.W	Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	—	1	—
XTRCT	Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	—	1	—

Table 2.7 lists the SH7708 Series 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 SH7708 Series 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 SH7708 Series 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 SH7708 Series branch instructions.

Table 2.10 Branch Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
`BF`	`label`	If T = 0, disp × 2 + PC Æ 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 SH7708 Series system control instructions.

Table 2.11 System Control Instructions

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

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

2.4.2 Instruction Code Map

Table 2.12 shows the instruction code map.

Table 2.12 Instruction Code Map

Instruction Code				Fx: 0000 MD: 00	Fx: 0001 MD: 01	Fx: 0010 MD: 10	Fx: 0011 to 1111 MD: 11
MSB			LSB				
0000	Rn	Fx	0000				
0000	Rn	Fx	0001				
0000	Rn	00MD	0010	STC SR,Rn	STC GBR,Rn	STC VBR,Rn	STC SSR,Rn
0000	Rn	01MD	0010	STC SPC,Rn			
0000	Rn	10MD	0010	STC R0_BANK,Rn	STC R1_BANK,Rn	STC R2_BANK,Rn	STC R3_BANK,Rn
0000	Rn	11MD	0010	STC R4_BANK,Rn	STC R5_BANK,Rn	STC R6_BANK,Rn	STC R7_BANK,Rn
0000	Rm	00MD	0011	BSRF Rm		BRAF Rm	
0000	Rm	10MD	0011	PREF @Rm			
0000	Rn	Rm	01MD	MOV.B Rm,@(R0,Rn)	MOV.W Rm,@(R0,Rn)	MOV.L Rm,@(R0,Rn)	MUL.L Rm,Rn
0000	0000	00MD	1000	CLRT	SETT	CLRMAC	LDTLB
0000	0000	01MD	1000	CLRS	SETS		
0000	0000	Fx	1001	NOP	DIV0U		
0000	0000	Fx	1010				
0000	0000	Fx	1011	RTS	SLEEP	RTE	
0000	Rn	Fx	1000				
0000	Rn	Fx	1001			MOVT Rn	
0000	Rn	Fx	1010	STS MACH,Rn	STS MACL,Rn	STS PR,Rn	
0000	Rn	Fx	1011				
0000	Rn	Rm	11MD	MOV.B @(R0,Rm),Rn	MOV.W @(R0,Rm),Rn	MOV.L @(R0,Rm),Rn	MAC.L @Rm+,@Rn+
0001	Rn	Rm	disp	MOV.L Rm,@(disp:4,Rn)			
0010	Rn	Rm	00MD	MOV.B Rm,@Rn	MOV.W Rm,@Rn	MOV.L Rm,@Rn	
0010	Rn	Rm	01MD	MOV.B Rm,@-Rn	MOV.W Rm,@-Rn	MOV.L Rm,@-Rn	DIV0S Rm,Rn
0010	Rn	Rm	10MD	TST Rm,Rn	AND Rm,Rn	XOR Rm,Rn	OR Rm,Rn
0010	Rn	Rm	11MD	CMP/STR Rm,Rn	XTRCT Rm,Rn	MULU.W Rm,Rn	MULSW Rm,Rn
0011	Rn	Rm	00MD	CMP/EQ Rm,Rn		CMP/HS Rm,Rn	CMP/GE Rm,Rn
0011	Rn	Rm	01MD	DIV1 Rm,Rn	DMULU.L Rm,Rn	CMP/HI Rm,Rn	CMP/GT Rm,Rn
0011	Rn	Rm	10MD	SUB Rm,Rn		SUBC Rm,Rn	SUBV Rm,Rn
0011	Rn	Rm	11MD	ADD Rm,Rn	DMULS.L Rm,Rn	ADDC Rm,Rn	ADDV Rm,Rn

Table 2.12 Instruction Code Map (cont)

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

Table 2.12 Instruction Code Map (cont)

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

Note: See the SH-3/SH-3E/SH3-DSP Programming Manual for details.

2.5 Processor States and Processor Modes

2.5.1 Processor States

The SH7708 Series 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 a reset, general exception, or interrupt exception handling.

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

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

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

Power-Down State: In the power-down state, CPU operation halts and power consumption is reduced. The power-down state is entered by means of the SLEEP instruction or the CA pin*. There are three modes in the power-down state: sleep mode, standby mode and hardware standby mode. See section 8, Power-Down Modes, for more information.
Note:SH7708S,SH7708R only

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

Transitions between the states are shown in figure 2.8.

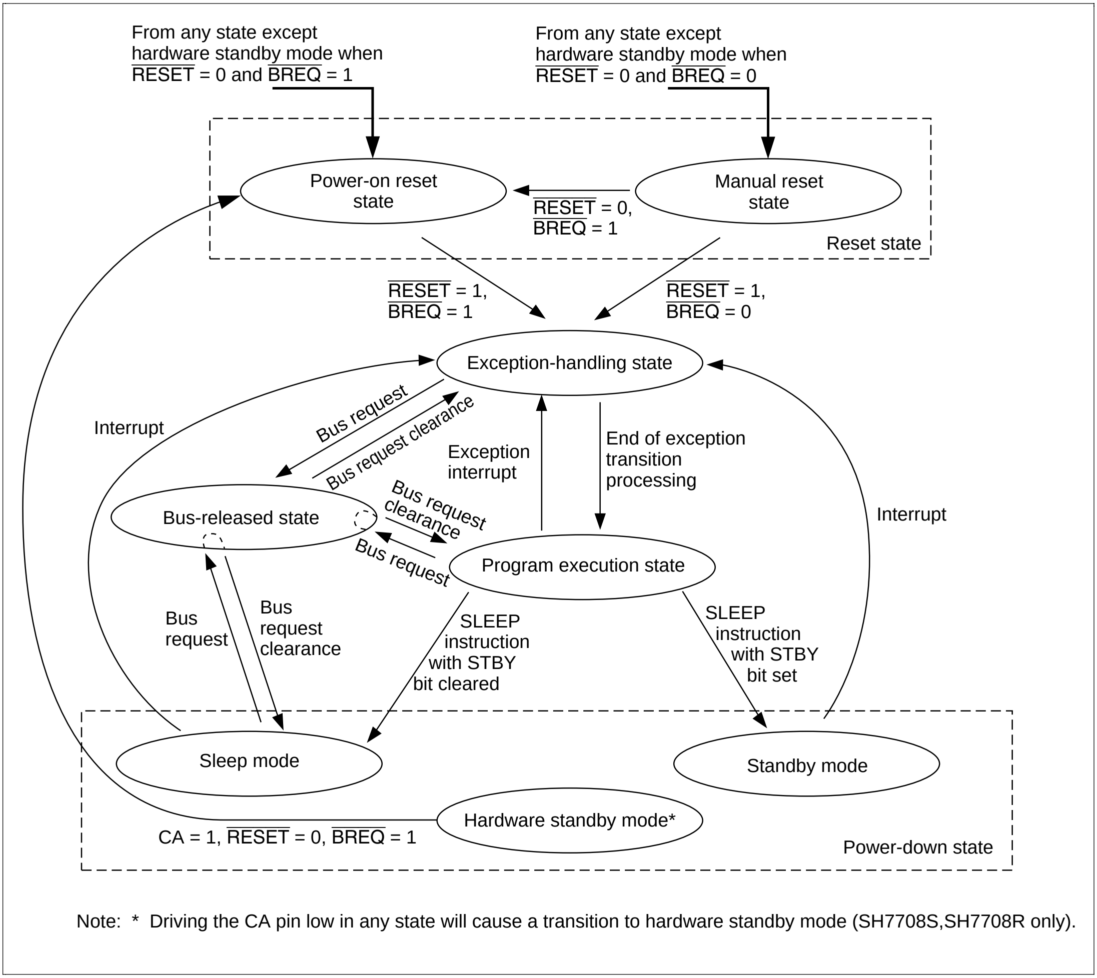

From any state except hardware standby mode when $\overline{\text{RESET}}$ = 0 and $\overline{\text{BREQ}}$ = 1
From any state except hardware standby mode when $\overline{\text{RESET}}$ = 0 and $\overline{\text{BREQ}}$ = 0
Power-on reset state
Manual reset state
$\overline{\text{RESET}}$ = 0, $\overline{\text{BREQ}}$ = 1
Reset state
$\overline{\text{RESET}}$ = 1, $\overline{\text{BREQ}}$ = 1
$\overline{\text{RESET}}$ = 1, $\overline{\text{BREQ}}$ = 0
Exception-handling state
Interrupt
Bus request
Bus request clearance
Exception interrupt
End of exception transition processing
Interrupt
Bus-released state
Bus request clearance
Bus request
Program execution state
Bus request
Bus request clearance
SLEEP instruction with STBY bit cleared
SLEEP instruction with STBY bit set
Sleep mode
Standby mode
Hardware standby mode*
CA = 1, $\overline{\text{RESET}}$ = 0, $\overline{\text{BREQ}}$ = 1
Power-down state
Note: * Driving the CA pin low in any state will cause a transition to hardware standby mode (SH7708S,SH7708R only).

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.

2.6 Usage Note

In the SH7708R, operations may not be performed correctly when the following sequences are executed with the cache on.

Address	Instruction
4n	MAC.L (or MAC.W, DMULS.L, DMULU.L, MUL.L)
4n+2	MAC.L (or MAC.W, DMULS.L, DMULU.L, MUL.L)
4n+4	STS(.L)/LDS(.L) MACH/MACL
or	
4n+2	MAC.L (or MAC.W, DMULS.L, DMULU.L, MUL.L)
4n+4	NOP (or one-cycle instruction except MAC instruction*[1] or branch instruction*[2])
4n+6	MAC.L (or MAC.W, DMULS.L, DMULU.L, MUL.L)
4n+8	STS(.L)/LDS(.L) MACH/MACL

Notes: 1. MULS.W, MULU.W, MAC.L, MAC.W, DMULS.L, DMULU.L, MUL.L, MULS.W, MULU.W

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

Software Remedy: This problem can be avoided in either of the following ways.

1. Do not use the above sequences.
2. When performing STS/LDS on the MACL/MACH register after consecutive execution of MAC instructions, insert an NOP instruction between the MAC instructions and STS/LDS instruction.

Section 3 Memory Management Unit (MMU)

3.1 Overview

3.1.1 Features

The SH7708 Series has an on-chip memory management unit (MMU) that implements address translation. The SH7708 Series features a resident translation lookaside buffer (TLB) that caches information for user-created address translation tables located in external memory. It enables high-speed translation of virtual addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 kbyte and 4 kbytes). The access right to virtual address space can be set for privileged and user modes to provide memory protection.

3.1.2 Role of MMU

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

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

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

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

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

In the following text, SH7708 Series address space in virtual memory is referred to as virtual address space, and address space in physical memory as physical memory space.

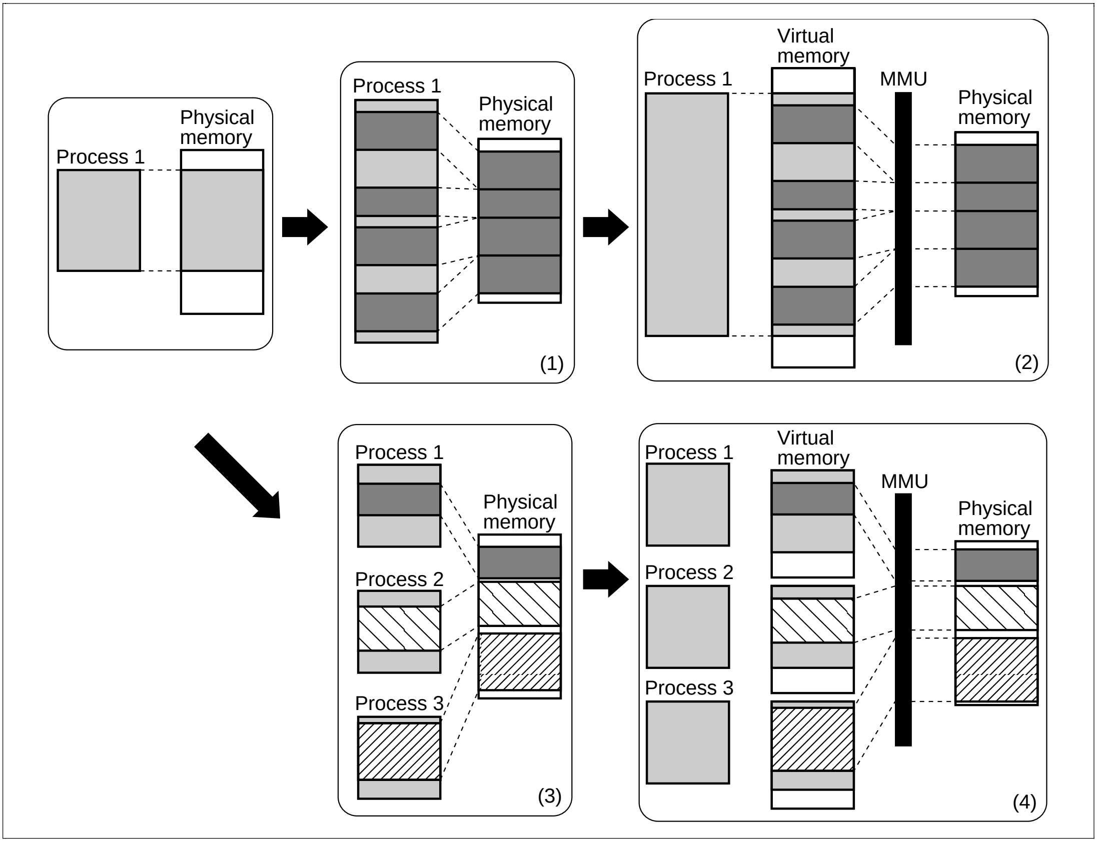


Figure 3.1 MMU Functions

3.1.3 SH7708 Series MMU

Virtual Address Map: The SH7708 Series 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. Write-back or write-through can be selected for write access by means of a CCR setting.

Mapping of the P1 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P1 area, setting a virtual address MSBs (bit 31) to 0 generates the corresponding physical address. P1 area access can be cached, and the cache control register (CCR) is set to indicate whether to cache or not. Write access is processed as write-through (SH7708). A CCR setting can be made to select write-back or write-through.

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

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

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. As with the P0 area, addresses H'7000000–H'7FFFFFFF can be used as on-chip RAM space by making a setting in the cache control register (CCR). In this case, mapping by means of the address translation table is not performed for the on-chip RAM space. The 2 Gbytes of virtual address space from H'80000000 to H'FFFFFFFF cannot be accessed in user mode. Attempting to do so creates an address error. Write-back or write-through mode can be selected for write accesses by means of a CCR setting.

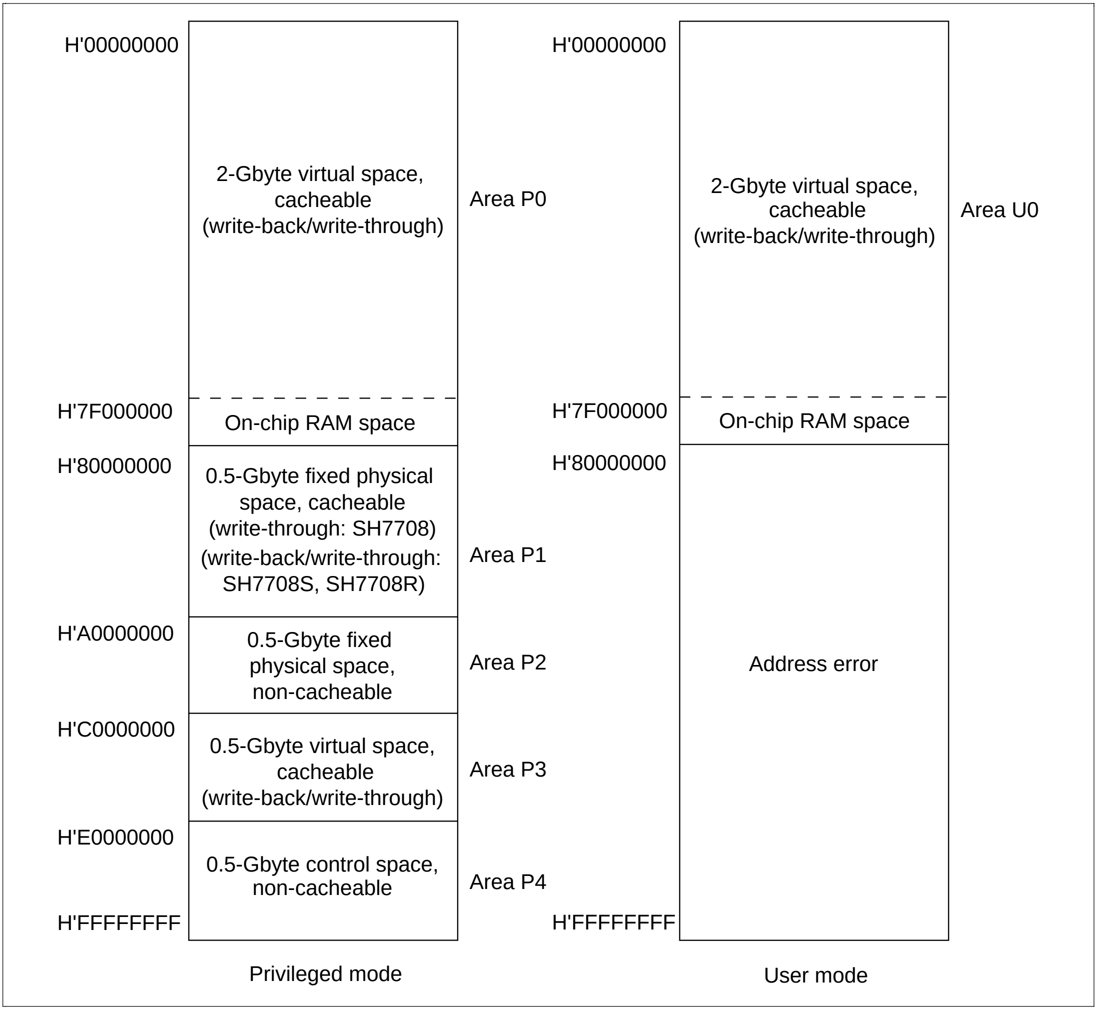

H'00000000
2-Gbyte virtual space, cacheable (write-back/write-through)
Area P0
H'7F000000
On-chip RAM space
H'80000000
0.5-Gbyte fixed physical space, cacheable (write-through: SH7708) (write-back/write-through: SH7708S, SH7708R)
Area P1
H'A0000000
0.5-Gbyte fixed physical space, non-cacheable
Area P2
H'C0000000
0.5-Gbyte virtual space, cacheable (write-back/write-through)
Area P3
H'E0000000
0.5-Gbyte control space, non-cacheable
Area P4
H'FFFFFFFF
Privileged mode
H'00000000
2-Gbyte virtual space, cacheable (write-back/write-through)
Area U0
H'7F000000
On-chip RAM space
H'80000000
Address error
H'FFFFFFFF
User mode

Figure 3.2 Virtual Address Space Mapping

Physical Address Space: The SH7708 Series supports a 32-bit physical address space, but the upper 3 bits are actually ignored and treated as a shadow. See section 10, Bus State Controller, for details.

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

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

When the MMU is disabled, the virtual address is used directly as the physical address. As the SH7708 Series 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. Either single or multiple virtual mode is selected according to the value set in the MMU control register. The items used in address comparison are the VPN and ASID. The VPN of the virtual address used to access external memory is compared with the VPN of the TLB entry selected by the index number.

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. The ASID is 8 bits in length and is held in PTEH within the MMU indicating the current process. With ASID, the TLB need not be purged when the process is switched.

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

3.1.4 Register Configuration

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

Table 3.1 Register Configuration

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

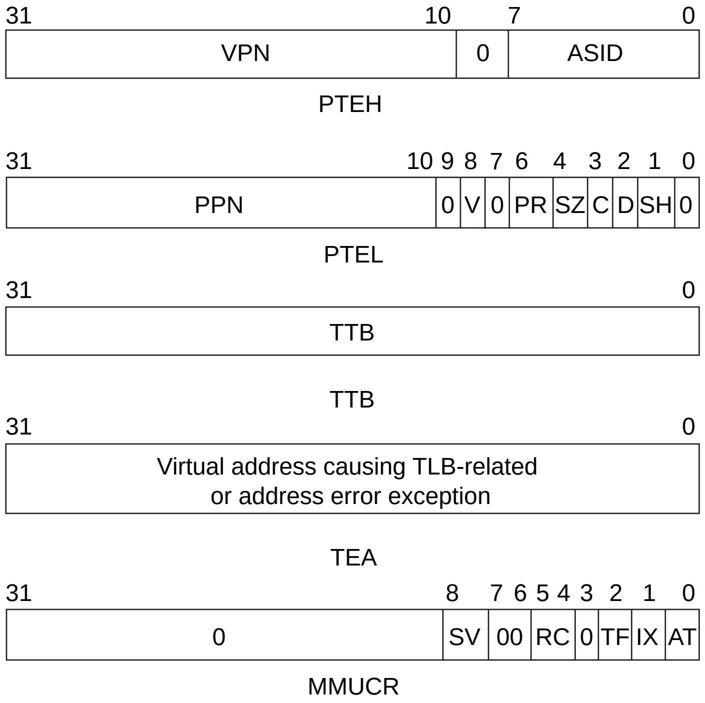
Notes: 1. Initialized by a power-on reset or manual reset.
2. SV bit: undefined
Other bits: 0

3.2 Register Description

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

1. The page table entry register high (PTEH) register residing at address H'FFFFFFF0, which consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the virtual address at which the exception is generated in the case of an MMU exception or address error exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the virtual address, but in this case the upper 22 bits of the virtual address are set. The VPN can also be modified by software. As the ASID, software sets the number of the currently executing process. The VPN and ASID are recorded in the TLB by the LDTLB instruction.
2. The page table entry register low (PTEL) register residing at address H'FFFFFFF4, and used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified in response to a software command.
3. The translation table base register (TTB) residing at address H'FFFFFFF8, which points to the base address of the current page table. The hardware does not set any value in TTB automatically. TTB is available to software for general purposes.
4. The TLB exception address register (TEA) register residing at address H'FFFFFFFC, which stores the virtual address corresponding to a TLB or address error exception. This value remains valid until the next exception or interrupt.
5. The MMU control register (MMUCR) residing at address H'FFFFFFF0, 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 (except MMUCR): Always read as 0. Writing is ignored.
(MMUCR) :Except bit 3 is read as 0. Bit 3 is don't care. Writing is should be 0.

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

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

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

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

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

Figure 3.3 MMU Register Contents

3.3 TLB Functions

3.3.1 Configuration of the TLB

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

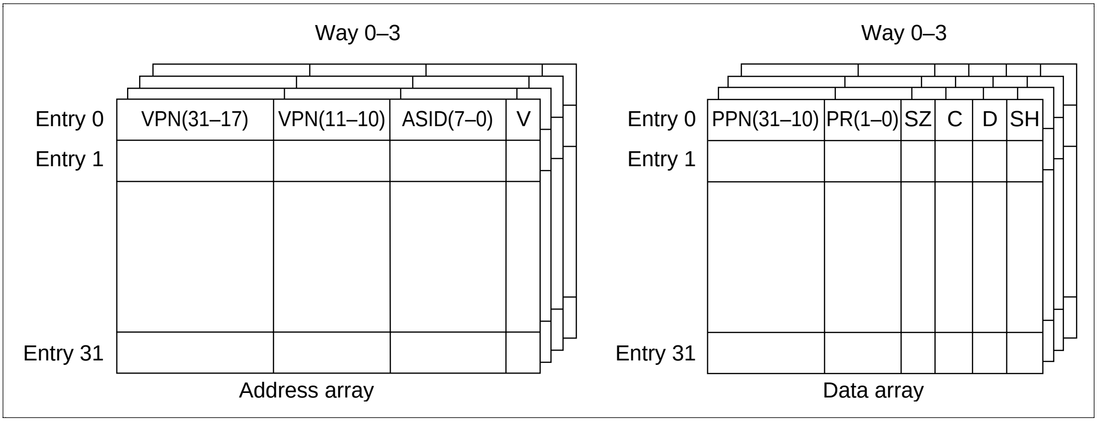


Figure 3.4 Overall Configuration of the TLB

31 10	9 0
VPN	Offset

Virtual address (1-kbyte page)

31 12	11 0
VPN	Offset

Virtual address (4-kbyte page)

(15)	(2)	(8)	(1)	(1)	(1)
VPN (31–17)	VPN (11–10)	ASID	SH	SZ	V

(22)	(2)	(1)	(1)
PPN	PR	C	D

TLB entry

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

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

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

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

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

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

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

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

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

Figure 3.5 Virtual Address and TLB Structure

3.3.2 TLB Indexing

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

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

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

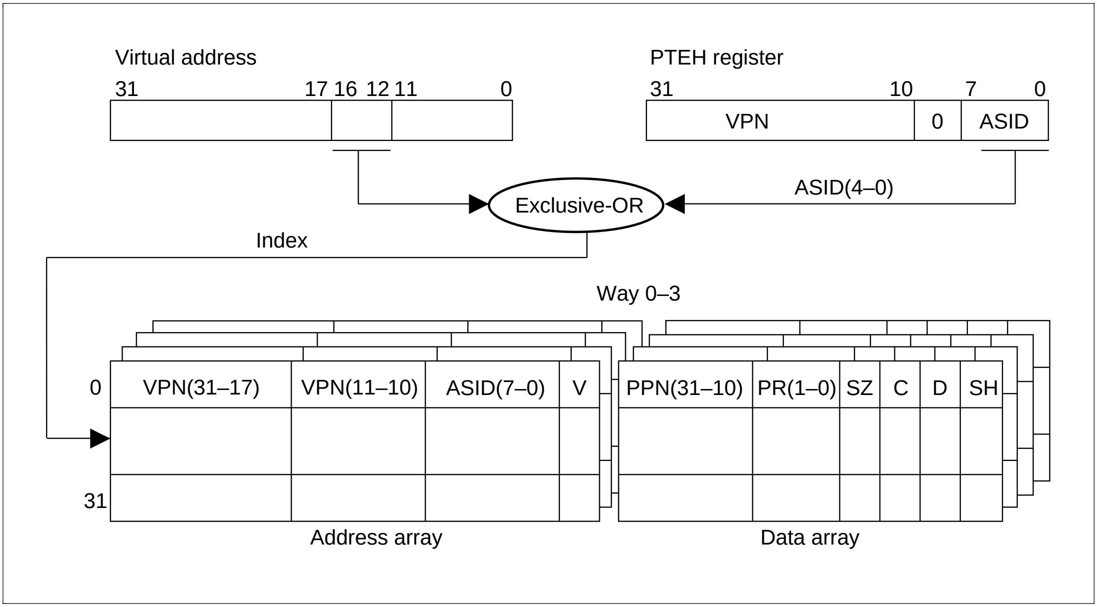


Figure 3.6 TLB Indexing (IX = 1)

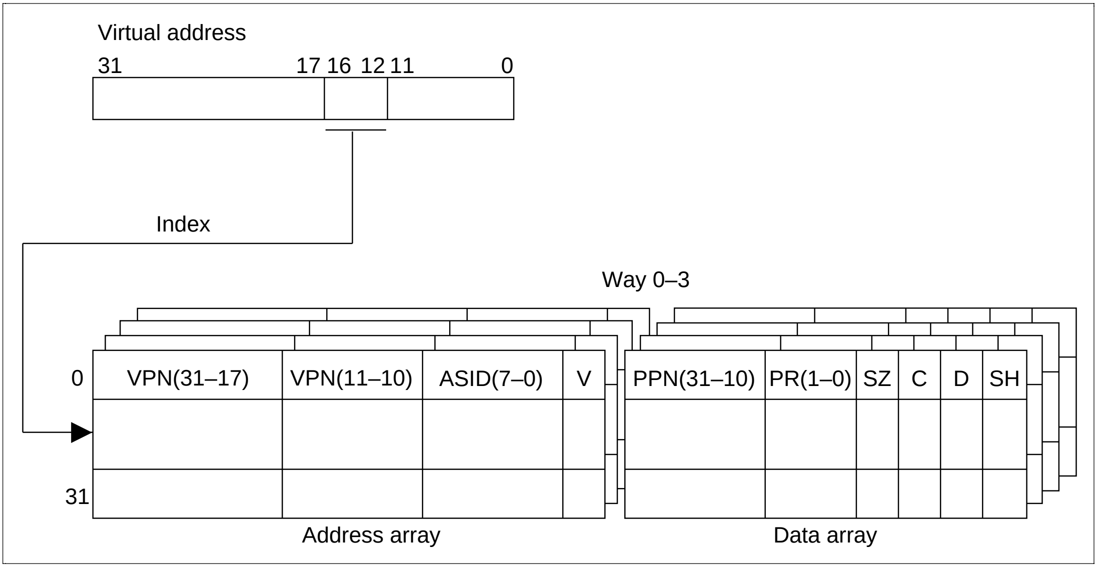


Figure 3.7 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

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

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

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

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

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

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

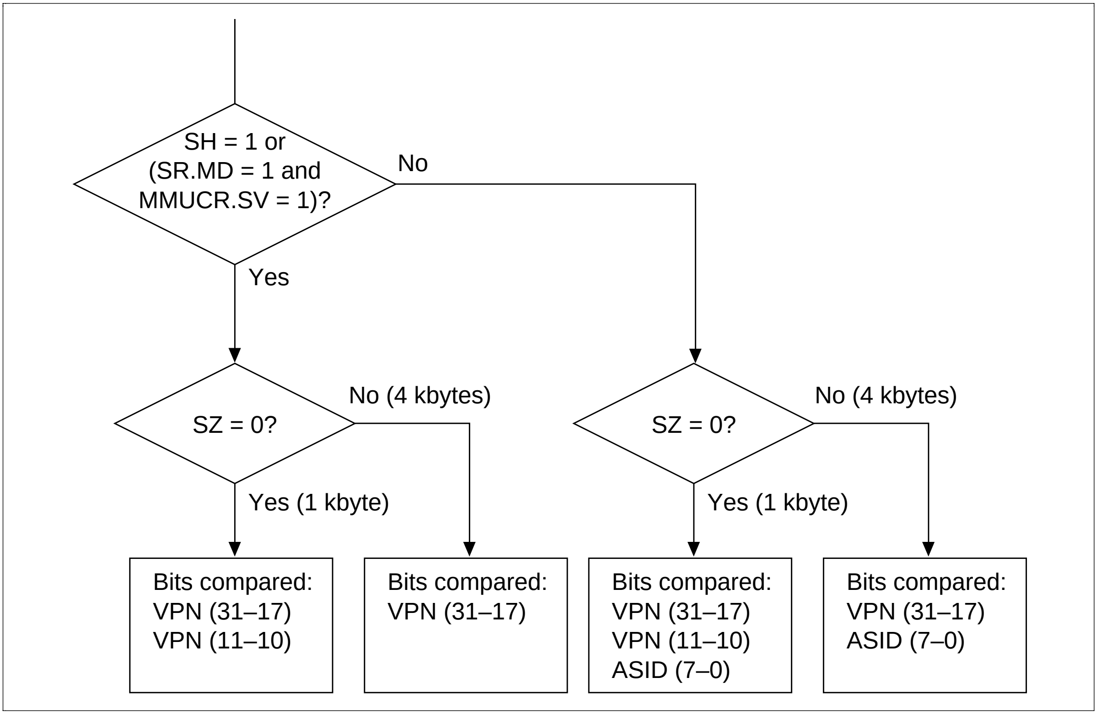


Figure 3.8 Objects of Address Comparison

3.3.4 Page Management Information

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

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

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

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

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

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

3.4 MMU Functions

3.4.1 MMU Hardware Management

MMU hardware management is of the following two kinds.

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

3.4.2 MMU Software Management

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

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

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

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

3.4.3 MMU Instruction (LDLTB)

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

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

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

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

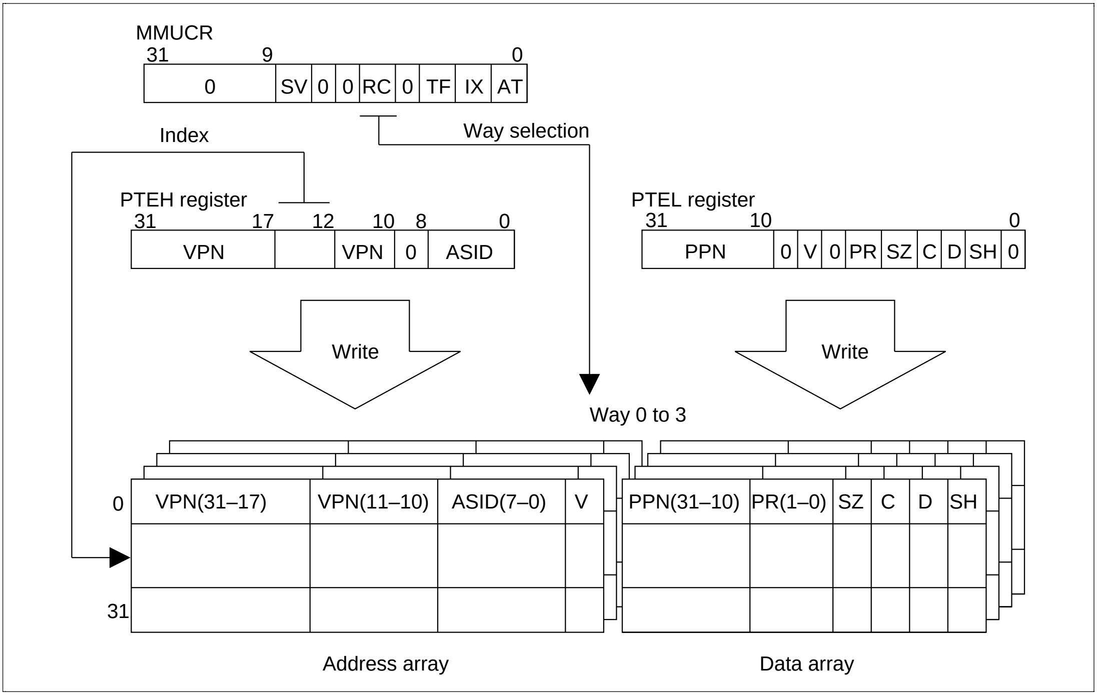


Figure 3.9 Operation of LDTLB Instruction

3.4.4 Avoiding Synonym Problems

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

To achieve high-speed operation of the SH7708 Series 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.

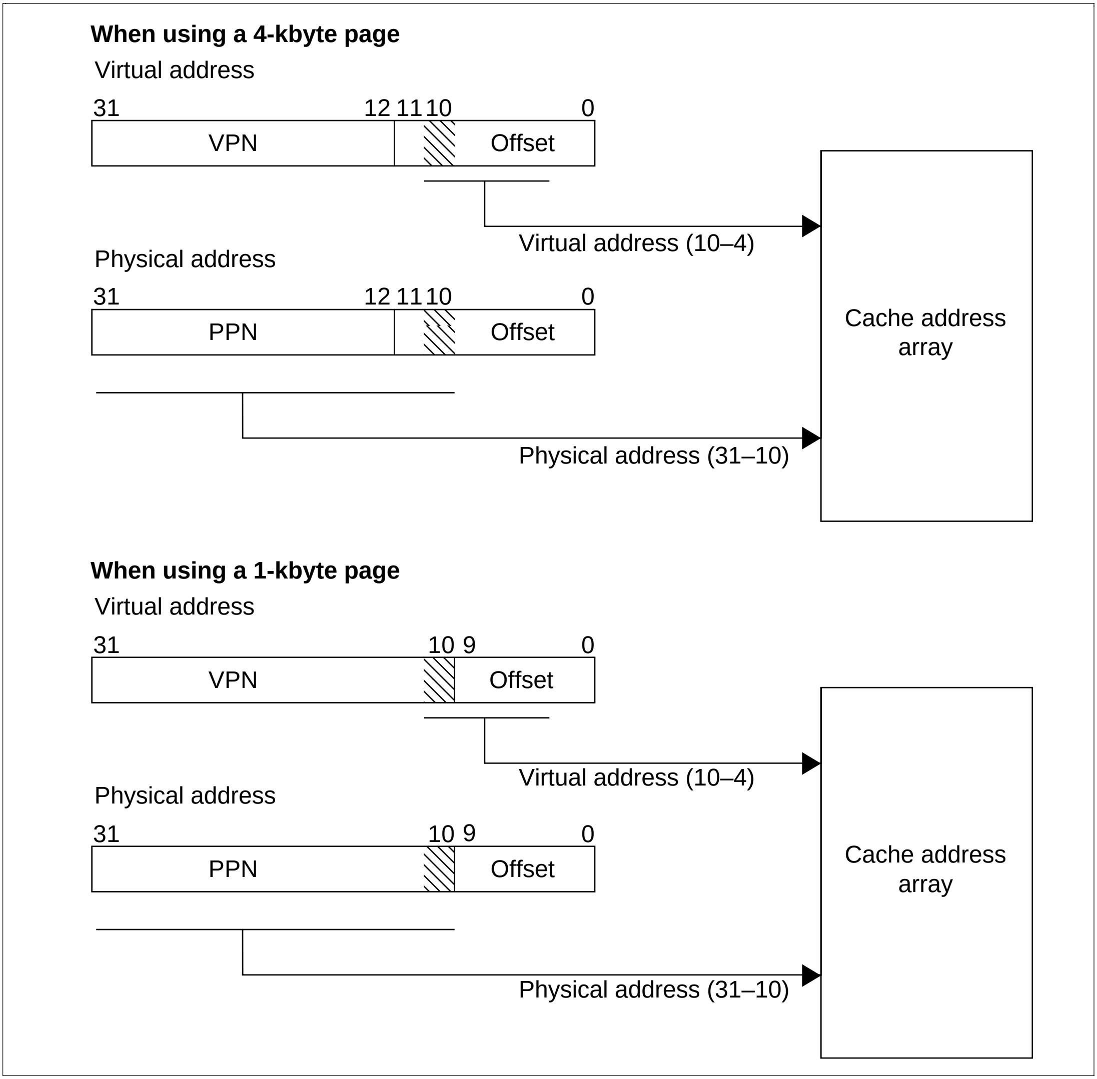


Figure 3.10 Synonym Problem

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 SH7708 Series 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 SH7708 Series in privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The RC field in the MMU control register (MMUCR) is incremented by 1 when all entries indexed are valid. When some entries indexed are invalid, the smallest way number of them is set in RC.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000400 to invoke the user-written TLB miss exception handler.

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

1. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the address translation table in external memory into the PTEL register in the SH7708 Series.

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. The RTE instruction should be issued at least two instructions after the LDTLB instruction.

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 SH7708 Series 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 SH7708 Series 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. The RTE instruction should be issued at least two instructions after the LDTLB instruction.

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 SH7708 Series hardware executes a set of prescribed operations, as follows:

1. The VPN number of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. The way number causing the exception is written to RC in MMUCR.
4. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
5. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the delayed branch instruction is written to the SPC.
6. The contents of SR at the time of the exception are written into SSR.
7. The mode (MD) bit in SR is set to 1 to place the SH7708 Series in privileged mode.
8. The block (BL) bit in SR is set to 1 to mask any further exception requests.
9. The register bank (RB) bit in SR is set to 1.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100, and the TLB protection violation exception handler starts.

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

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

3.5.4 Initial Page Write Exception

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

Hardware Operations: In an initial page write exception, the SH7708 Series hardware executes a set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH register.
2. The virtual address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7708 Series in privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100 to invoke the user-written initial page write exception handler.

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

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

Figure 3.11 shows the flowchart for MMU exceptions.

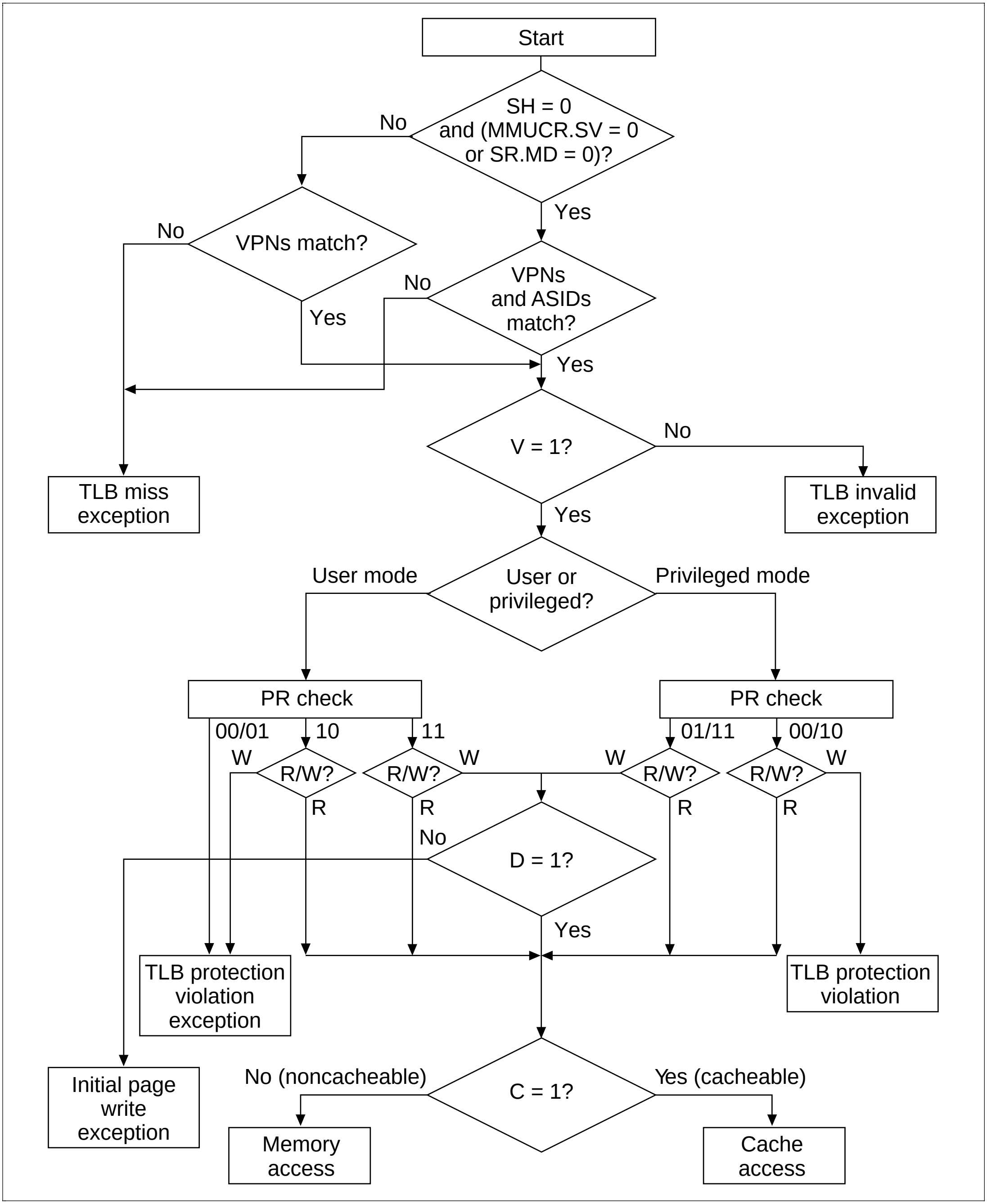


Figure 3.11 MMU Exception Generation Flowchart

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

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

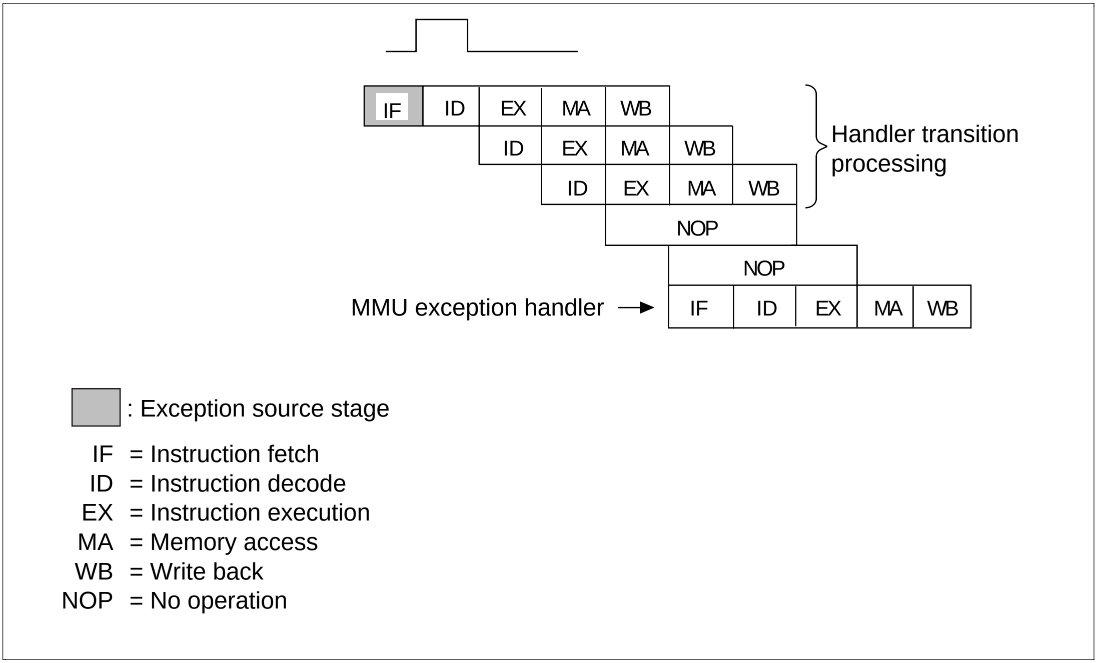


Figure 3.12 MMU Exception Signals in Instruction Fetch

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

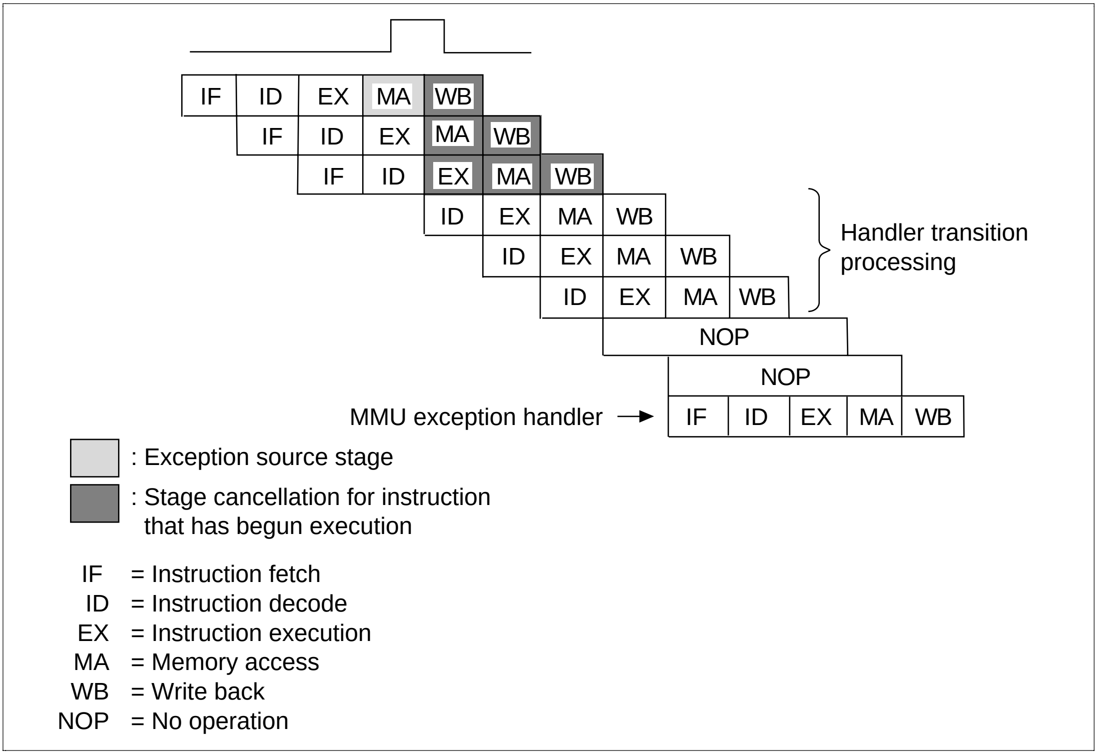


Figure 3.13 MMU Exception Signals in Data Access

3.6 Memory-Mapped TLB

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

3.6.1 Address Array

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

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

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

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

When reading, the VPN (31–17, 11–10), V bit, and ASID of the entry specified by the entry address and way number are read in the format of the data field in figure 3.14 without comparing addresses. Zero is read in the data field (16–12).

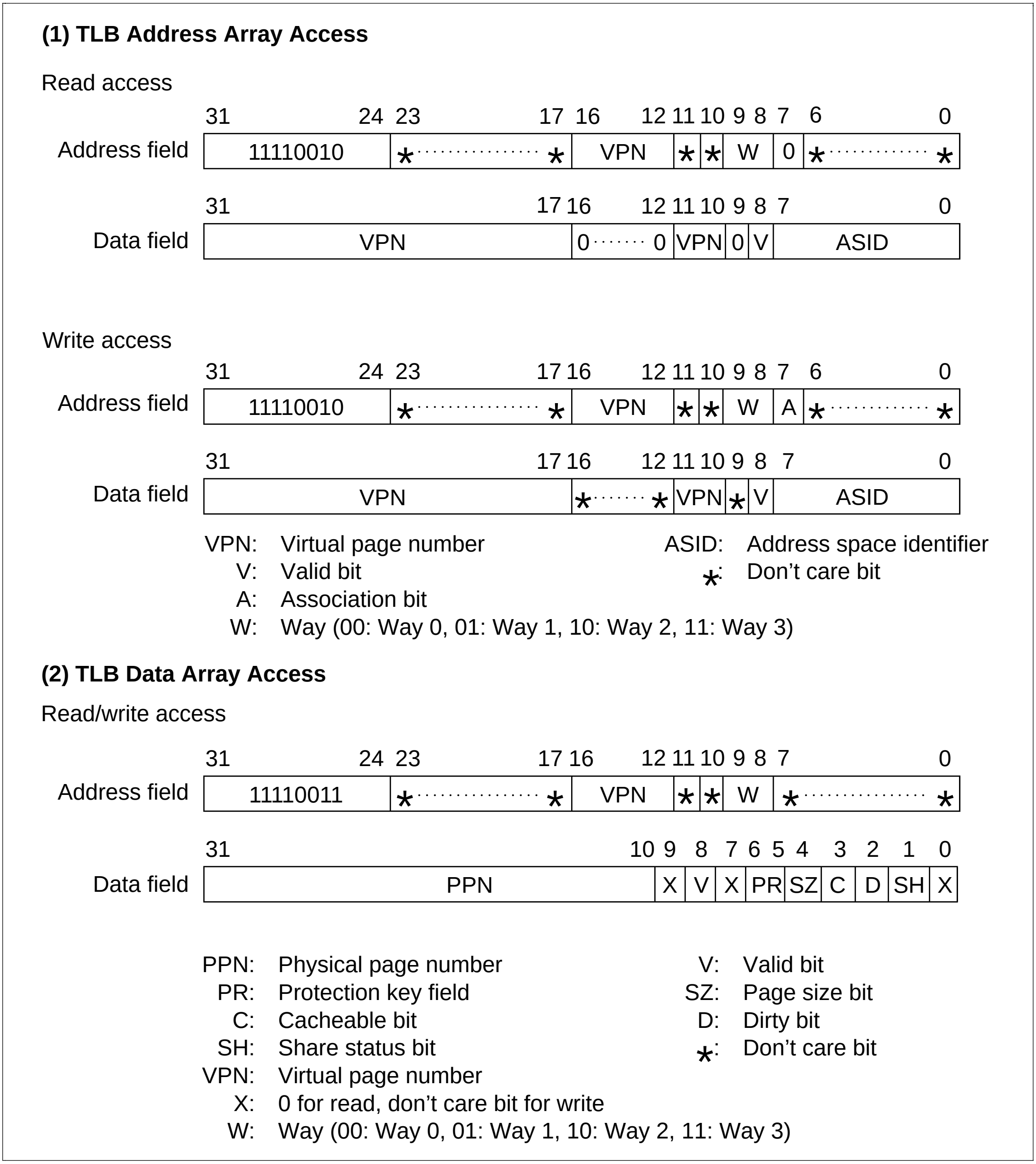
To invalidate a specific entry, specify the entry and write 0 to its V bit. When 1 is specified for the A bit, only the required VPN entry is invalidated.

3.6.2 Data Array

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

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

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



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

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 is compared to the VPN and ASID within the TLB entry selected by the entry address and data is written to the matching way. If no match is found, there is no operation. R0 specifies the write data and R1 specifies the address.

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

MOV.L  R0,@R1
```

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

```
; R0=H'F300 4300   VPN(16-12)=B'00100   Way 3

; MOV.L @R0,R1
```

3.7 Usage Note

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

Section 4 Exception Handling

4.1 Overview

4.1.1 Features

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

4.1.2 Register Configuration

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

Table 4.1 Register Configuration

Register	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

4.2 Exception Handling Function

4.2.1 Exception Handling Flow

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

A basic exception processing sequence consists of the following operations:

- The contents of the PC and SR are saved in the SPC and SSR, respectively.
- The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
- The mode (MD) bit in SR is set to 1 to place the SH7708 Series 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) register.
- Instruction execution jumps to the designated exception processing vector address to invoke the handler routine.

4.2.2 Exception Handling Vector Addresses

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

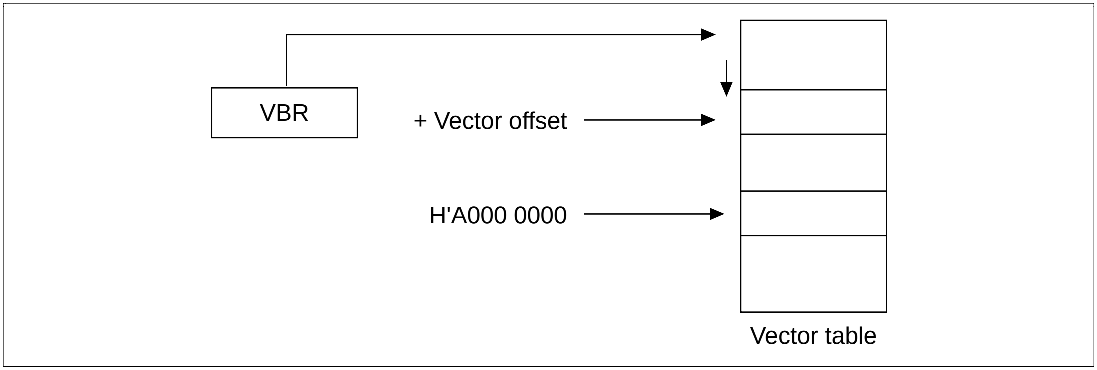


Figure 4.1 Vector Table

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

Table 4.2 Vectored Exception Events

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

Table 4.2 Vectored Exception Events (cont)

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

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

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

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

4.2.3 Acceptance of Exceptions

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

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

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

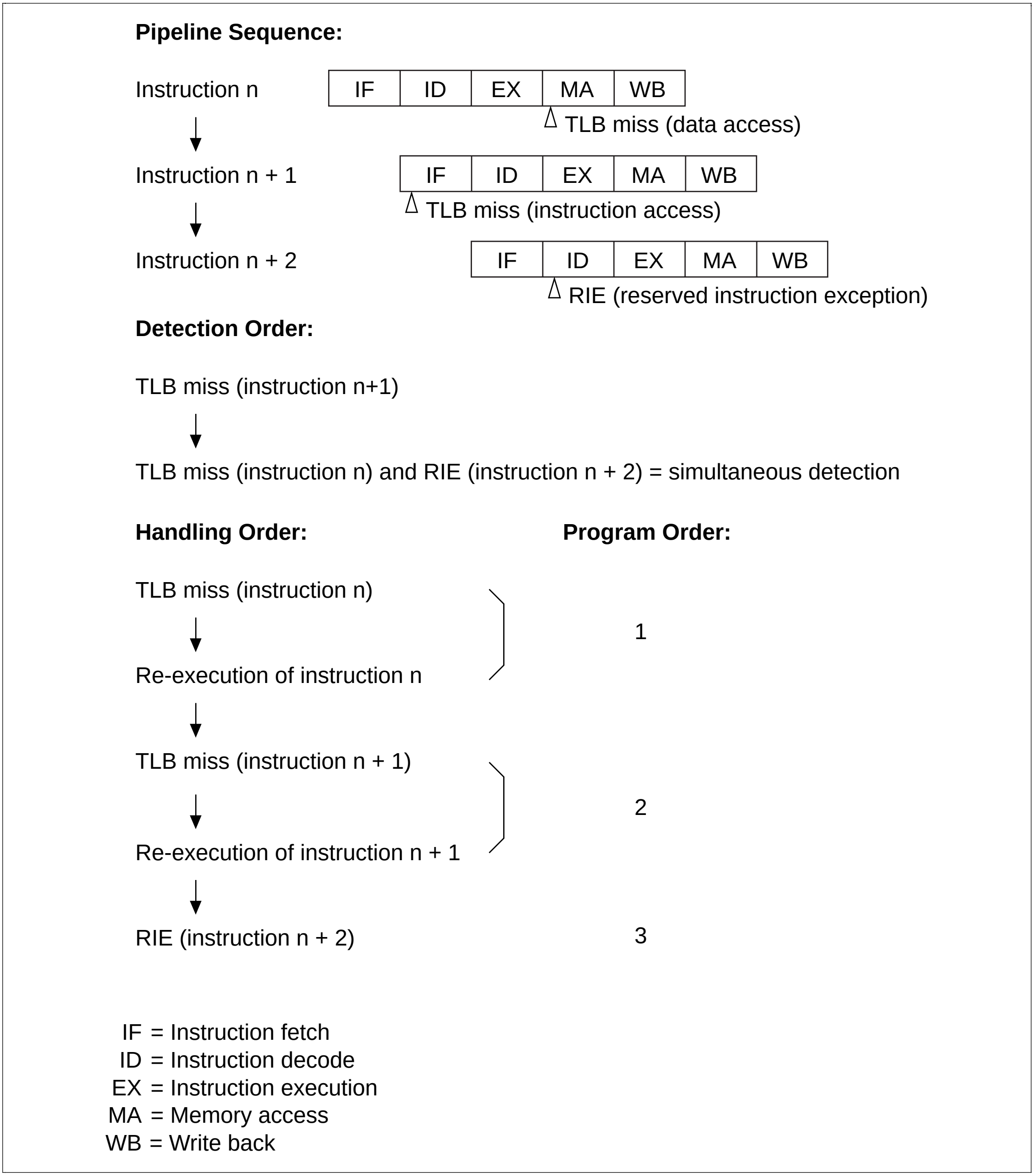


Figure 4.2 Example of Acceptance Order of General Exceptions

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

4.2.4 Exception Codes

Table 4.3 lists the exception codes written to bits 11–0 of the EXPEVT register (for reset or general exceptions) or the INTEVT 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 break point trap	H'1E0
General interrupt requests	Nonmaskable interrupt	H'1C0
	External hardware interrupts:	
	IRL3–IRL0 = 0000	H'200
	IRL3–IRL0 = 0001	H'220
	IRL3–IRL0 = 0010	H'240
	IRL3–IRL0 = 0011	H'260
	IRL3–IRL0 = 0100	H'280
	IRL3–IRL0 = 0101	H'2A0
	IRL3–IRL0 = 0110	H'2C0
	IRL3–IRL0 = 0111	H'2E0
	IRL3–IRL0 = 1000	H'300

Table 4.3 Exception Codes (cont)

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

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

4.2.5 Exception Request Masks

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

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

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

4.2.6 Returning from Exception Handling

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

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

4.3 Register Description

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

1. The exception event register (EXPEVT) resides at address H'FFFFFFD4, and contains a 12-bit exception code. The exception code set in EXPEVT is that for a reset or general exception event. The exception code is set automatically by hardware when an exception occurs. EXPEVT can also be modified by software.
2. The interrupt event register (INTEVT) resides at address H'FFFFFFD8, and contains a 12-bit exception code. The exception code set in EXPEVT is that for an interrupt request. The exception code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.
3. The TRAPA exception register (TRA) resides at address H'FFFFFFD0, and contains 8-bit immediate data (imm) for the TRAPA instruction. TRA is set automatically by hardware when a TRAPA instruction is executed. TRA can also be modified by software.

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

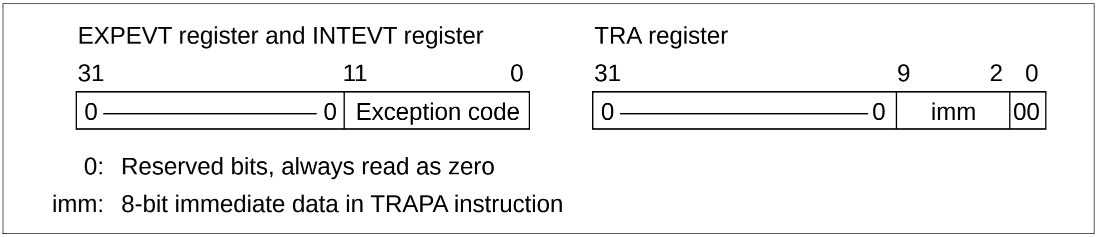

EXPEVT register and INTEVT register
31
11
0
0 — 0
Exception code
TRA register
31
9
2
0
0 — 0
imm
00
0: Reserved bits, always read as zero
imm: 8-bit immediate data in TRAPA instruction

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

4.4 Exception Handler Operation

4.4.1 Reset

The reset sequence is used to power up or restart the SH7708 Series 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 SH7708 Series in privileged mode.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The RB bit in SR is set to 1.
- An encoded value of H'000 in a power-on reset or H'020 in a manual reset is written to bits 11–0 of the EXPEVT register to identify the exception event.
- Instruction execution jumps to the user-written exception handler at address H'A0000000.

4.4.2 Interrupts

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

- The contents of the PC and SR are saved in SPC and SSR, respectively.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The MD bit in SR is set to 1 to place the SH7708 Series 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 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 SH7708 Series encounters any exception condition other than a reset or interrupt request, it executes the following operations:

- The contents of the PC and SR are saved in the SPC and SSR, respectively.
- The BL bit in SR is set to 1, masking any subsequent exceptions.
- The MD bit in SR is set to 1 to place the SH7708 Series 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 $\overline{\text{RESET}}$ low
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bits are set to 1 and the IMASK field is set to B'1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details. A power-on reset must always be performed when powering on.
- Manual Reset
  - Conditions: $\overline{\text{BREQ}}$ pin low and $\overline{\text{RESET}}$ low
  - Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB, and BL bits are set to 1 and the IMASK field is set to B'1111. The CPU and on-chip supporting modules are initialized. See the register 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.

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

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

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

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

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

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

- TLB protection exception

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

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

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

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

- Address error

— Conditions:

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

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

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

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

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

For details see section 3.5.5, Processing Flow in Event of MMU Exception.

- Unconditional trap

— Conditions: TRAPA instruction executed

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

- Reserved instruction exception
  - Conditions:
    a. When undefined code not in a delay slot is decoded

       Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
       Undefined instructions: H'Fxxx(SH7708, SH7708S), H'FxxF(SH7708R)

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

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

       Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S
       Undefined instructions: H'Fxxx(SH7708, SH7708S), H'FxxF(SH7708R)

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

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

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

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

4.5.3 Interrupts

- NMI
  - Conditions: NMI pin edge detection
  - Operations: The PC and SR after the instruction that receives the interrupt are saved to the SPC and SSR, respectively. H'01C0 is set in INTEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0600. This interrupt is not masked by SR.IMASK 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 IRL3–IRL0 level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the IRL3–IRL0 level is set in INTEVT. The corresponding code is given as H'200 + B'(IRL3–IRL0) × H'20. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set in SR.IMASK. See section 6, Interrupt Controller, for more information.
- On-chip module interrupts
  - Conditions: The value of the interrupt mask bits in SR is lower than the on-chip module (TMU, RTC, SCI, CPG, REF) interrupt level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the interrupt source is set in INTEVT. See table 6.4, Interrupt Exception Vectors and Rankings, for the corresponding codes. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. B'0000 to B'1111 are set in the interrupt priority level registers (IRPA, IRPB) within the interrupt controller. See section 6, Interrupt Controller, for more information.

4.6 Cautions

- Return from exception handling
  - Check the BL bit in SR with software. When the SPC and SSR have been saved to external memory, set the BL bit in SR to 1 before restoring them.
  - Issue an RTE instruction. Set the SPC in the PC and SSR in SR with the RTE instruction, branch to the SPC address, and return from exception handling.
- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is set to 0 by software. If there is a request and the reception conditions are satisfied, the interrupt is accepted after the execution of the instruction that sets the BL bit in SR to 0. During the sleep or standby mode, however, the interrupt will be accepted even when the BL bit in SR is 1.
  - Exception: No user break point trap will occur even when the break conditions are met. When one of the other exceptions occurs, a branch is made to the fixed address of the reset (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR registers are undefined.
- SPC when an Exception Occurs: The PC saved to the SPC when an exception occurs is as shown below:
  - Re-executing-type exceptions: The PC of the instruction that caused the exception is set in the SPC and re-executed after return from exception handling. If the exception occurred in a delay slot, however, the PC of the immediately prior delayed branch instruction is set in the SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.
  - Completed-type exceptions and interrupts: The PC of the instruction after the one that caused the exception is set in the SPC. If the exception was caused by a delayed conditional instruction, however, the branch destination PC is set in SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.
- Initial register values after reset
  - Undefined registers

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

    VBR = H'00000000

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

    PC = H'A0000000
- Ensure that an exception is not generated at an RTE instruction delay slot, as operation is not guaranteed in this case.
- When the BL bit in SR is 1, ensure that a TLB-related exception or address error is not caused by an LDC instruction that updates SR and the following instruction, as these may be identified as multiple exceptions and may start reset processing.

Section 5 Cache

5.1 Overview

5.1.1 Features

The cache specifications are listed in table 5.1.

Table 5.1 Cache Specifications

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

5.1.2 Cache Structure

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

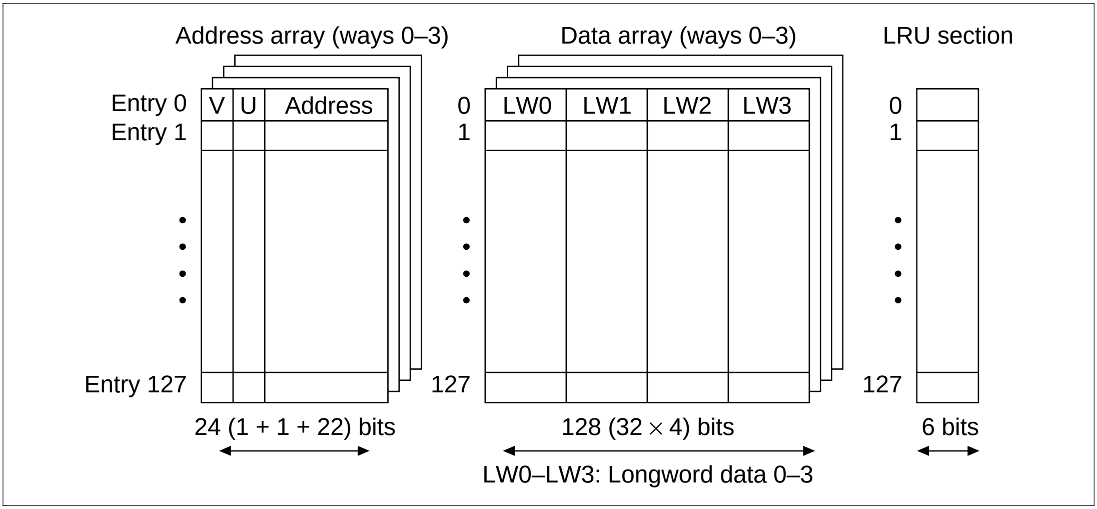


Figure 5.1 Cache Structure

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

In the SH7708 Series, the top three of the 32 physical address bits are used as shadow bits (see section 10), and therefore in a normal replace operation the top three bits of the vector address are cleared to 0.

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

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

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

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

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

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

Table 5.2 LRU and Way Replacement in Normal Mode

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

5.1.3 Register Configuration

Table 5.3 shows details of the cache control register.

Table 5.3 Register Configuration

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

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

5.2 Register Description

5.2.1 Cache Control Register (CCR)

The cache is enabled or disabled using the CE bit of the cache control register (CCR). CCR also has an RA bit (which switches the cache operation mode between RAM mode and normal mode), a CF bit (which invalidates all cache entries), a WT bit and a CB 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. CB bit is not supported in emulator.

Note: * SH7708S, SH7708R Only

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

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

Note:SH7708:Reserved bit
SH7708S,SH7708R:CB

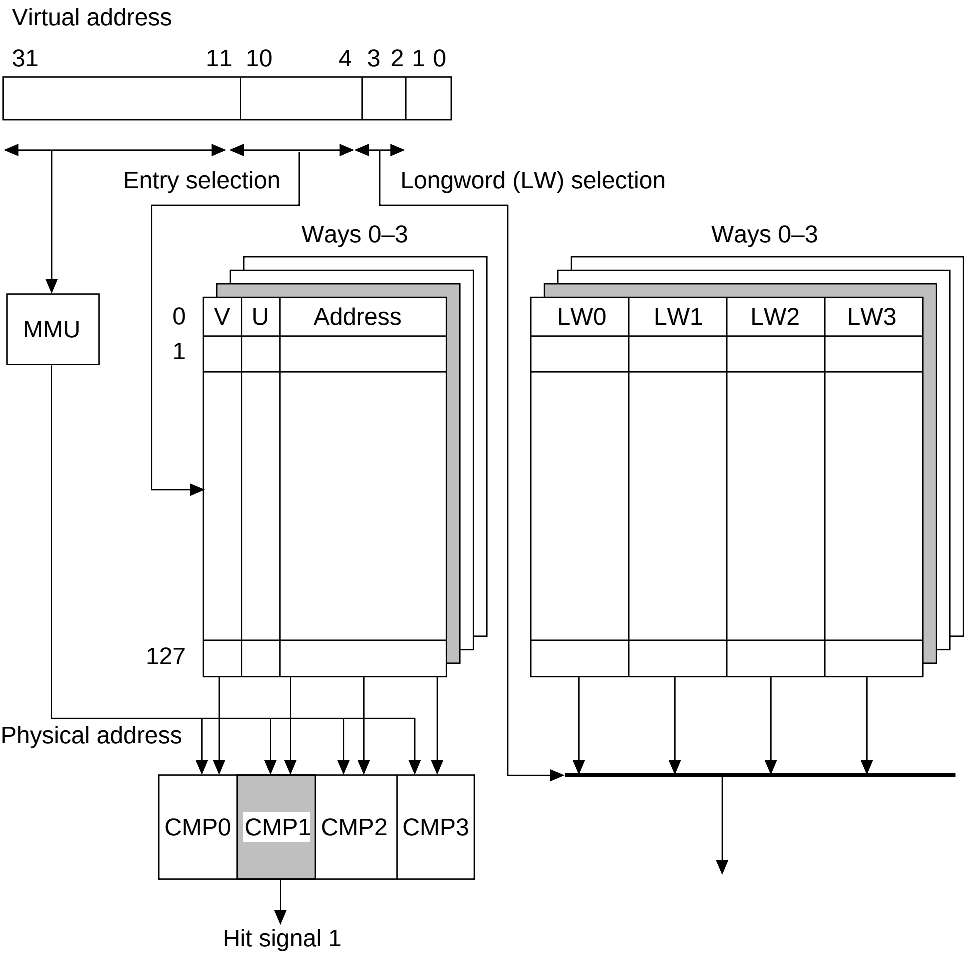
Figure 5.2 CCR Register Configuration

5.3 Cache Operation

5.3.1 Searching the Cache

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

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



CMP0: Comparison circuit 0
CMP1: Comparison circuit 1
CMP2: Comparison circuit 2
CMP3: Comparison circuit 3

Figure 5.3 Cache Search Scheme (Normal Mode)

5.3.2 Read Access

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

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

5.3.3 Write Access

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

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

5.3.4 Write-Back Buffer

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

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

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

Figure 5.4 Write-Back Buffer Configuration

5.3.5 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by the SH7708 Series 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 SH7708 Series, invalidate the entry that includes the updated data in the cache.

5.3.6 RAM Mode

In RAM mode, way 0 and way 1 function as a 4-kbyte two-way set associative cache, while way 2 and way 3 function as 4-kbyte internal RAM. The internal RAM is mapped onto H'7F000000 to H'7F000FFF with 4-kbyte shadow areas from H'7F001000 to H'7FFFFFFF. In RAM mode with the MMU enabled, a virtual address from H'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 operation mode, all entries in the cache should be invalidated.

5.4 Memory-Mapped Cache

To allow software management of the cache, it is mapped onto virtual address space P4. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF and the data array onto addresses H'F1000000 to H'F1FFFFFF. In privileged mode, the cache contents can be read or written using the MOV instruction. With the address array and data array, the access size is fixed at longword, and instruction fetches cannot be performed.

5.4.1 Address Array

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

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

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

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

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

In the SH7708 Series, the top 3 bits of the 32-bit physical address are treated as a shadow (see section 10, Bus State Controller (BSC)). Therefore, in the event of a cache miss, 0 is registered in the top 3 bits of the address array address tag.

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

5.4.2 Data Array

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

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

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

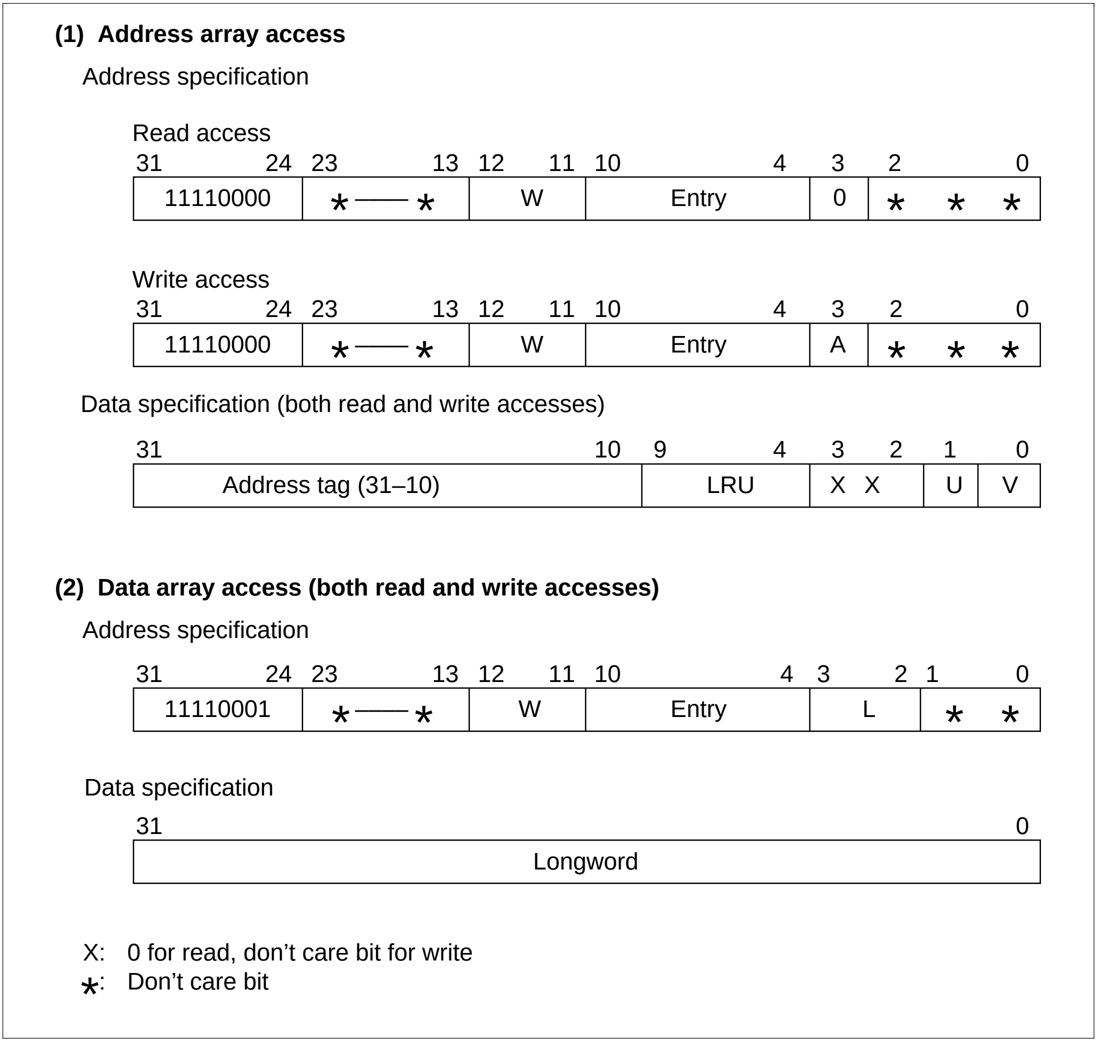

(1) Address array access
Address specification
Read access
31 24 23 13 12 11 10 4 3 2 0
11110000 | * —— * | W | Entry | 0 | * * *
Write access
31 24 23 13 12 11 10 4 3 2 0
11110000 | * —— * | W | Entry | A | * * *
Data specification (both read and write accesses)
31 10 9 4 3 2 1 0
Address tag (31–10) | LRU | X X | U | V
(2) Data array access (both read and write accesses)
Address specification
31 24 23 13 12 11 10 4 3 2 1 0
11110001 | * —— * | W | Entry | L | * *
Data specification
31 0
Longword
X: 0 for read, don't care bit for write
*: Don't care bit

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

5.5 Usage Examples

5.5.1 Invalidating Specific Entries

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

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

5.5.2 Reading the Data of a Specific Entry

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

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

Section 6 Interrupt Controller (INTC)

6.1 Overview

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

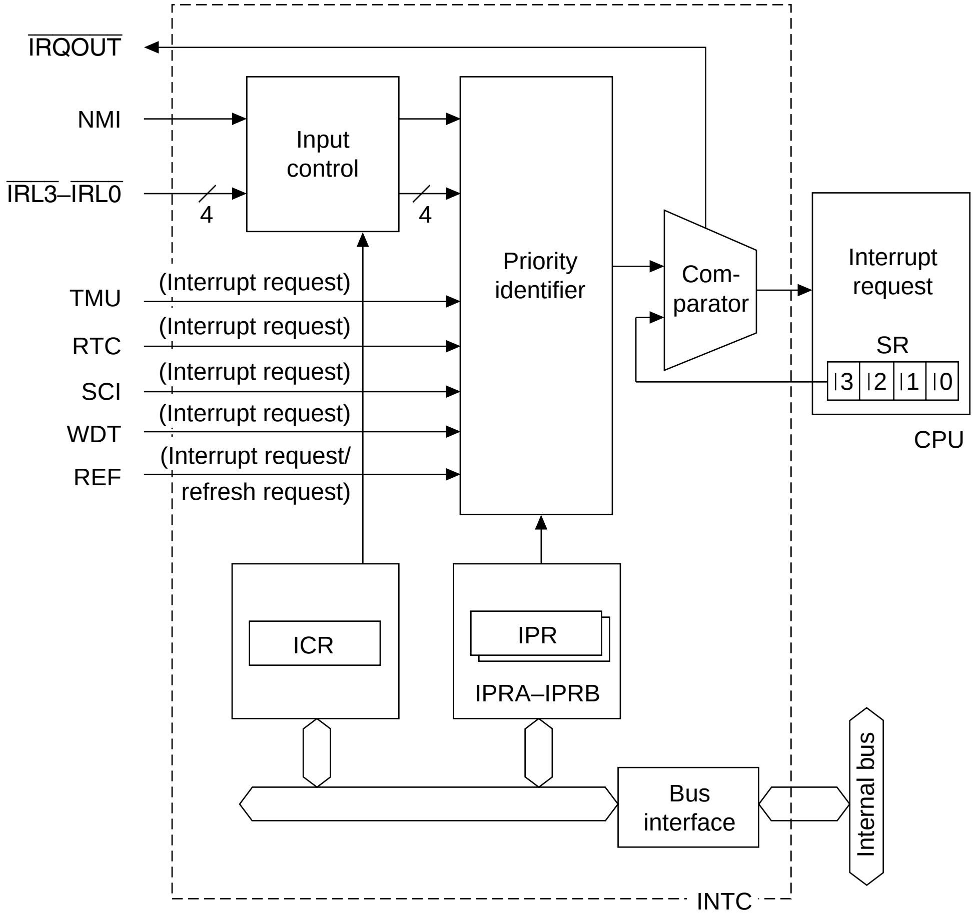
6.1.1 Features

INTC has the following features:

- 15 levels of interrupt priority can be set: By setting the two interrupt-priority registers, the priorities of on chip supporting module interrupts can be selected from 15 levels for different request sources.
- NMI noise canceller function: NMI input level bit indicates NMI pin status. By reading this bit in the interrupt exception handler, 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 ($\overline{\text{IRQOUT}}$ pin): For example, when the SH7708 Series has released the bus, 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 the SH7708 Series to request the bus.

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the INTC.



TMU: Timer unit
RTC: Realtime clock unit
SCI: Serial communication interface
WDT: Watch dog timer
REF: Memory refresh controller section of the bus state controller
ICR: Interrupt control register
IPRA–IPRB: Interrupt priority level setting registers A–B
SR: Status register

Figure 6.1 INTC Block Diagram

6.1.3 Pin Configuration

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	I	Input of nonmaskable interrupt request signal
Interrupt input pins	$\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$	I	Input of interrupt request signals (maskable by I3–I0 in SR)
Bus request output pin	$\overline{\text{IRQOUT}}$	O	Output of signal that notifies external devices that an interrupt or memory refresh request has occurred

6.1.4 Register Configuration

The INTC has the three registers listed in table 6.2.

Table 6.2 Register Configuration

			Initial		Access
Name	**Abbr.**	**R/W**	**Value*1**	**Address**	**Size**
Interrupt control register	ICR	R/W	*2	H'FFFFFEE0	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

Notes: 1. Initialized by a power-on reset or manual reset.
2. H'8000 when the NMI pin is high, H'0000 when the NMI pin is low.

6.2 Interrupt Sources

There are three types of interrupt sources: NMI, IRL, 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 Interrupts

The NMI interrupt has the highest priority level of 16. It is always accepted unless the BL bit in the status register in the CPU is set to 1, and is edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL setting. The NMI edge select bit (NMIE) in the interrupt control register (ICR) is used to select either the rising or falling edge. When the NMIE bit in the ICR register is changed, the NMI interrupt is not detected for 20 cycles after changing ICR. NMI interrupt exception handling does not affect the interrupt mask level bits (I3–I0) in the status register (SR).

6.2.2 IRL Interrupts

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

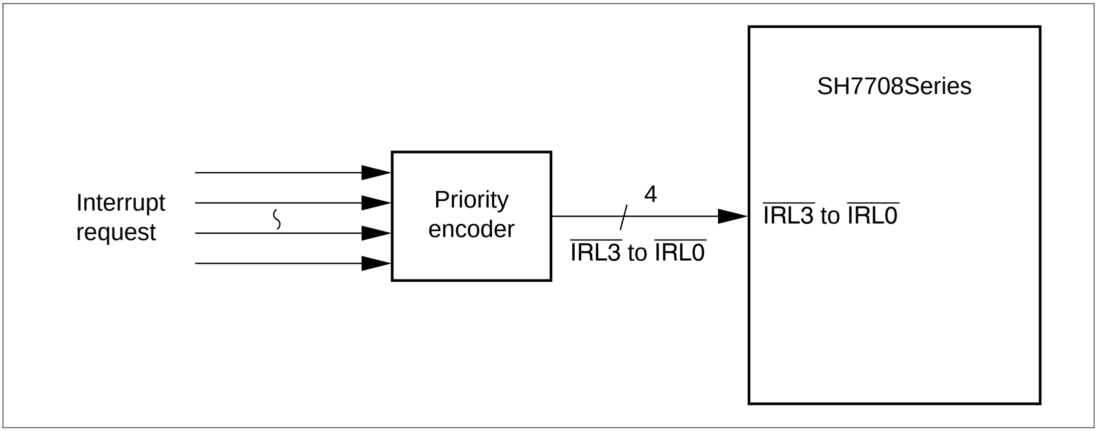


Figure 6.2 Example of IRL Interrupt Connection

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 $\overline{\text{IRL}}$ interrupt handling.

6.2.3 On-Chip Supporting Module Interrupts

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

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interface (SCI)
- Bus state controller (BSC)
- Watchdog timer (WDT)

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

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

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

On-chip supporting module interrupt source flag and interrupt enable flag updating should only be performed when the BL bit in the status register (SR) is set to 1. To prevent acknowledgment of an erroneous interrupt from an interrupt source that should have been updated, first read the on-chip peripheral register containing the relevant flag, then clear the BL bit to 0. This will secure the necessary timing internally. When updating a number of flags, there is no problem if only the register containing the last flag updated is read.

If flag updating is performed while the BL bit is cleared to 0, the program may jump to the interrupt service routine when the INTEVT register value is 0. In this case, interrupt handling is initiated due to the timing relationship between the flag update and interrupt request recognition within the chip. Processing can be continued without any problem by executing an RTE instruction.

6.2.4 Interrupt Exception Handling and Priority

Table 6.3 lists the codes for the interrupt event register (INTEVT), and the order of interrupt priority. Each interrupt source is assigned unique code. The start address of the interrupt handler is common to each interrupt source. This is why, for instance, the value of INTEVT is used as offset at the start of the interrupt handler and branched to identify the interrupt source.

The order of priority of the on-chip supporting module is set within the priority levels 0–15 at will by using the interrupt priority level set in registers A and B (IPRA–IPRB). The order of priority of the on-chip supporting module is set to zero by a reset.

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

Updating of interrupt priority level setting registers A and B should only be performed when the BL bit in the status register (SR) is set to 1. To prevent erroneous interrupt acknowledgment, first read one or other of the interrupt priority level setting registers, then clear the BL bit to 0. This will secure the necessary timing internally.

Table 6.4 Interrupt Exception Vectors and Rankings

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0	16	—	—	High
IRL	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0000	H'200	15	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0001	H'220	14	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0010	H'240	13	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0011	H'260	12	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0100	H'280	11	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0101	H'2A0	10	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0110	H'2C0	9	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 0111	H'2E0	8	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1000	H'300	7	—	—	↓
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1001	H'320	6	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1010	H'340	5	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1011	H'360	4	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1100	H'380	3	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1101	H'3A0	2	—	—	
	$\overline{\mathrm{IRL3}}$–$\overline{\mathrm{IRL0}}$ = 1110	H'3C0	1	—	—	
TMU0	TUNI0*1	H'400	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1*1	H'420	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2*1	H'440	0–15 (0)	IPRA (7–4)	High	
TMU2	TICPI2*2	H'460	0–15 (0)	IPRA (7–4)	Low	Low

Table 6.4 Interrupt Exception Vectors and Rankings (cont)

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
RTC	ATI*[3]	H'480	0–15 (0)	IPRA (3–0)	High	High
RTC	PRI*[4]	H'4A0	0–15 (0)	IPRA (3–0)	↓	
RTC	CUI*[5]	H'4C0	0–15 (0)	IPRA (3–0)	Low	
SCI	ERI*[6]	H'4E0	0–15 (0)	IPRB (7–4)	High	
SCI	RXI*[7]	H'500	0–15 (0)	IPRB (7–4)	↓	↓
SCI	TXI*[8]	H'520	0–15 (0)	IPRB (7–4)		
SCI	TEI*[9]	H'540	0–15 (0)	IPRB (7–4)	Low	
WDT	ITI	H'560	0–15 (0)	IPRB (15–12)	—	
REF	RCMI*[10]	H'580	0–15 (0)	IPRB (11–8)	High	
REF	ROVI*[11]	H'5A0	0–15 (0)	IPRB (11–8)	Low	Low

Notes: 1. TUNI0–TUNI2: Underflow interrupts, see section 11.
2. TICPI2: Input capture interrupt, see section 11.
3. ATI: Alarm interrupt, see section 12.
4. PRI: Periodic interrupt, see section 12.
5. CUI: Carry-up interrupt, see section 12.
6. ERI: Receive error interrupt, see section 13.
7. RXI: Receive-data-full interrupt, see section 13.
8. TXI: Transmit-data-empty interrupt, see section 13.
9. TEI: Transmit-data-end interrupt, see section 13.
10. ITI: Interval timer interrupt, see section 9.
11. RCMI: Compare-match interrupt, see section 10.
12. ROVI: Refresh counter overflow interrupt, see section 10.

6.3 INTC Registers

6.3.1 Interrupt Priority Registers A and B (IPRA, IPRB)

Interrupt priority registers A and B (IPRA and IPRB) are 16-bit read/write registers that set priority levels from 0 to 15 for on-chip supporting module interrupts. These registers are initialized to H'0000 by a reset. They 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 6.5 lists the relationship between the interrupt sources and the IPRA and IPRB bits.

Table 6.5 Interrupt Request Sources and IPRA, IPRB

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*1	SCI	Reserved *2

Notes: 1. REF is the memory refresh unit in the bus state controller. See section 10, Bus State Controller, for details.

2. Reserved bits: Always read 0. Only 0 should be written.

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

6.3.2 Interrupt Control Register (ICR)

ICR is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and indicates the input signal level to the NMI pin. This register is initialized by a power-on reset or manual reset. It is not initialized in standby mode.

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

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

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

Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. It 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 (Initial value)
1	Interrupt request is detected on rising edge of NMI input

Bits 14–9 and 7–0—Reserved: These bits always read 0. The write value should always be 0.

6.4 INTC Operation

6.4.1 Interrupt Sequence

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

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

Notes: 1. The interrupt mask bits (I3–I0) in the status register (SR) are not changed by acceptance of an interrupt in the SH7708 Series.

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

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

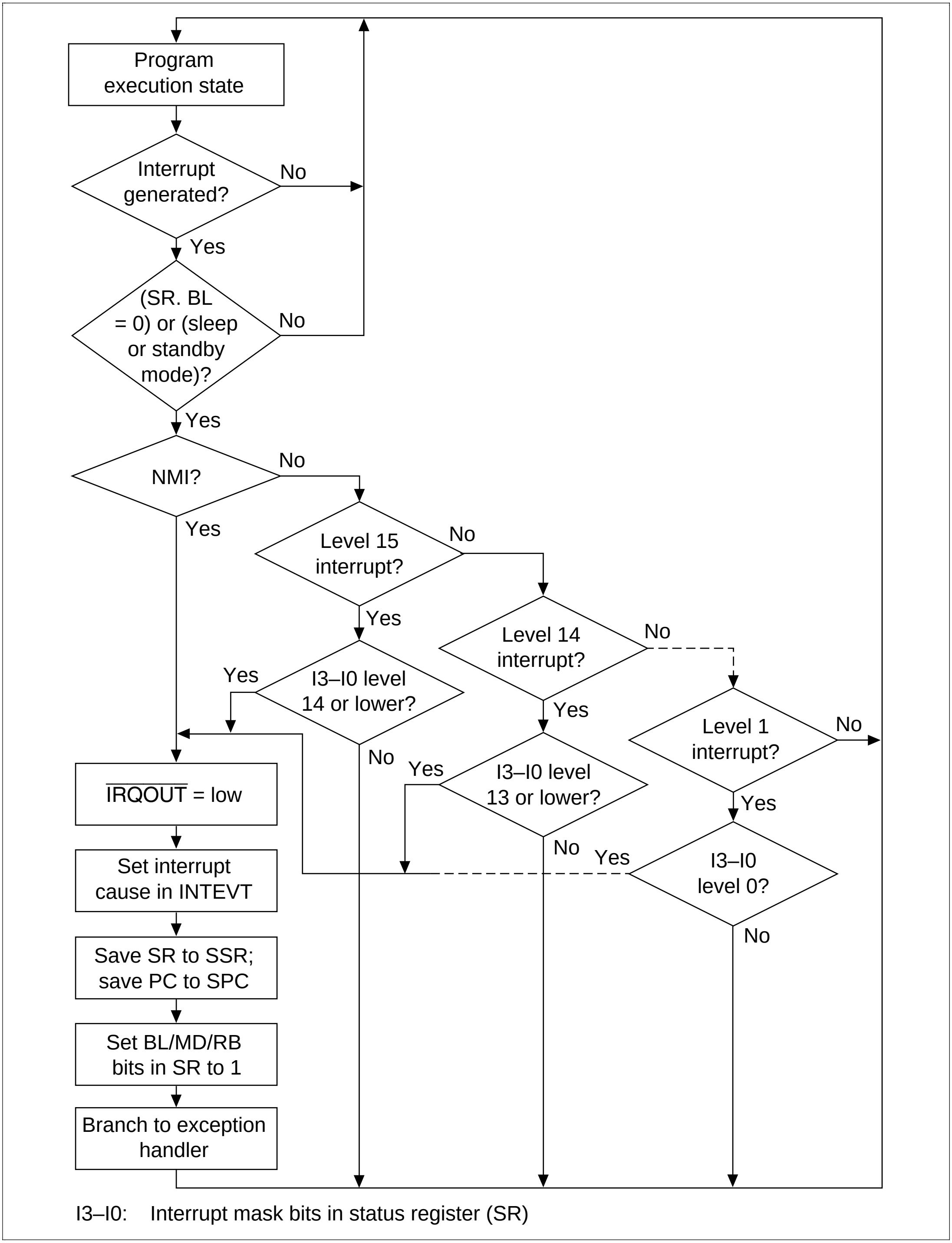

Program execution state
Interrupt generated?
No
Yes
(SR. BL = 0) or (sleep or standby mode)?
No
Yes
NMI?
No
Yes
Level 15 interrupt?
No
Yes
Level 14 interrupt?
No
Yes
I3–I0 level 14 or lower?
Yes
No
I3–I0 level 13 or lower?
Yes
No
Level 1 interrupt?
No
Yes
I3–I0 level 0?
Yes
No
$\overline{\text{IRQOUT}}$ = low
Set interrupt cause in INTEVT
Save SR to SSR; save PC to SPC
Set BL/MD/RB bits in SR to 1
Branch to exception handler
I3–I0: Interrupt mask bits in status register (SR)

Figure 6.3 Interrupt Operation Flowchart

6.4.2 Multiple Interrupts

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

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

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

6.5 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception handling is performed and fetching of the first instruction of the exception handler is started (the interrupt response time) is shown in table 6.6. 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 6.6 Interrupt Response Time

	Number of States			
Item	**NMI**	**IRL**	**Supporting Modules**	**Notes**
Time for priority decision and SR mask bit comparison	0.5 × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	0.5 × Icyc + 0.5 × Bcyc + 2 × Pcyc	0.5 × Icyc + 1.5 × Pcyc	
Wait time until end of sequence being executed by CPU	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	

Table 6.6 Interrupt Response Time (cont)

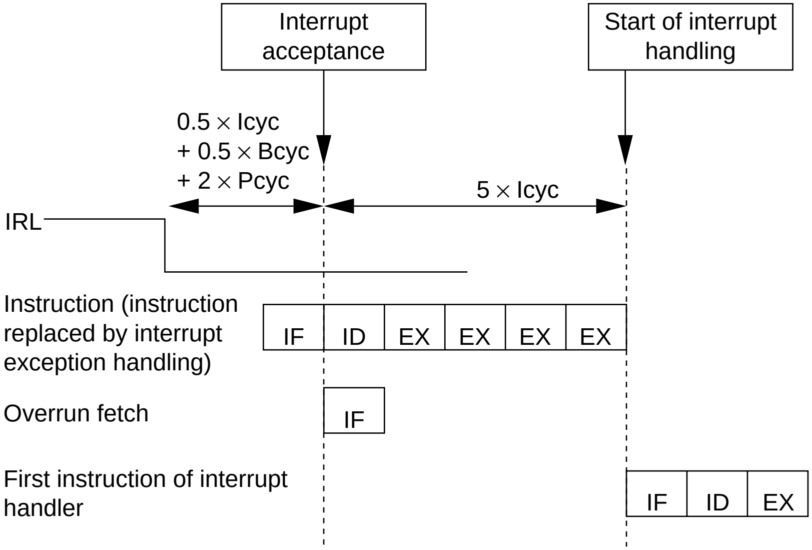
		Number of States			
Item		**NMI**	**RL**	**Peripheral Modules**	**Notes**
Response time	Total	(5.5 + X) × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	(5.5 + X) × Icyc + 0.5 × Bcyc + 2 × Pcyc	(5.5 + X) × Icyc + 1.5 × Pcyc	
	Minimum case*2	6.5	8	7	At 60 MHz operation: 0.10–0.14 µs
	Maximum case*3	7 + S	13 + S	10.5 + S	At 60 MHz operation: 0.23–0.34 µs (in case of operand cache-hit) At 60 MHz operation: 0.27–0.37 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 1 : 1 : 1.
3. The internal clock : CKIO : peripheral clock ratio is 1 : 1 : 1/4.


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

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

Figure 6.4 Example of Pipeline Operations when IRL Interrupt is Accepted

Section 7 User Break Controller (UBC)

7.1 Overview

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

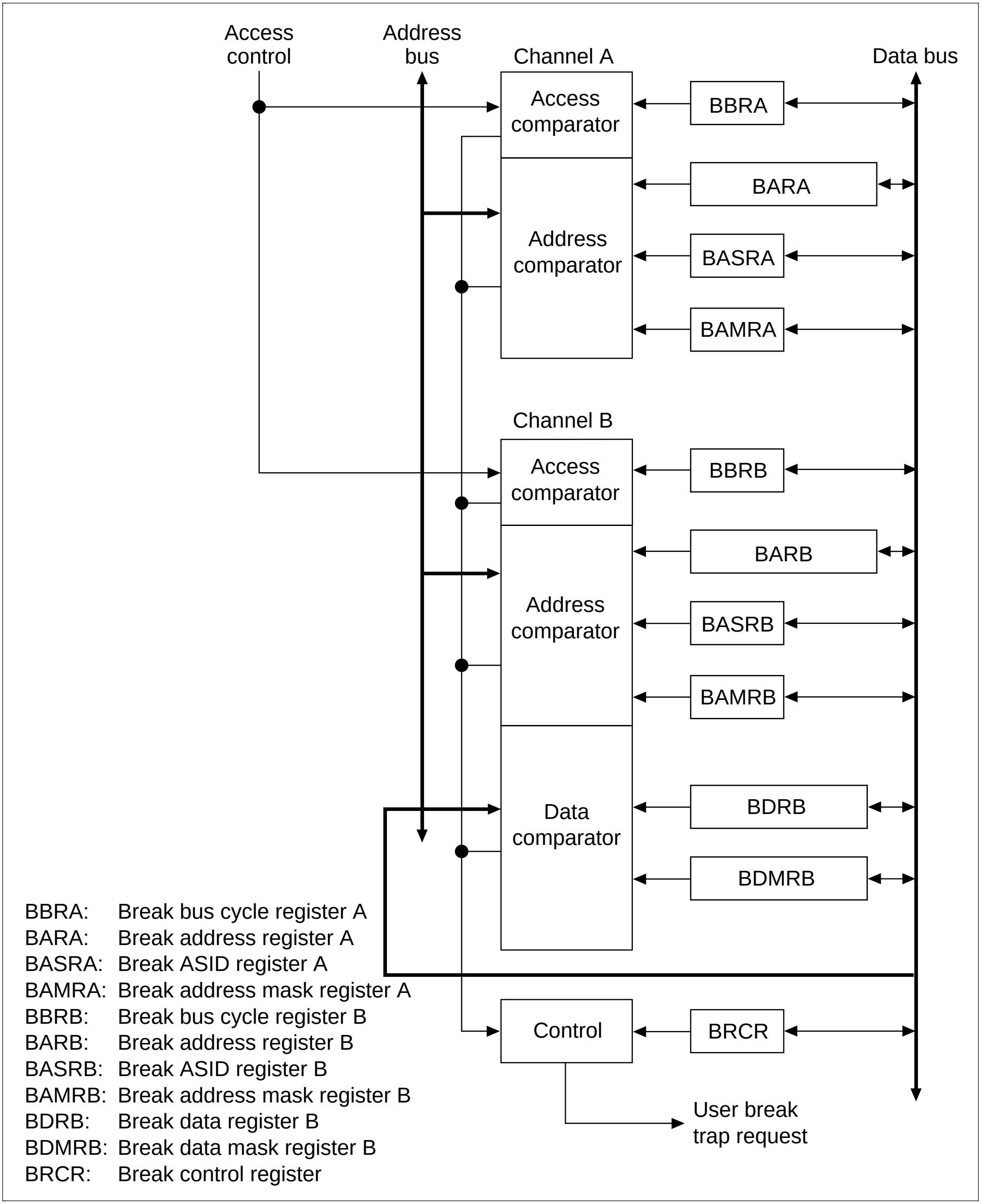
7.1.1 Features

The features of the user break controller are listed below.

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

7.1.2 Block Diagram

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



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

Figure 7.1 Logical Block Diagram of User Break Controller

7.1.3 Register Configuration

Table 7.1 shows the user break controller registers.

Table 7.1 UBC Registers

Channel	Register	Initial Value*[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 power-on reset 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, it is set in the BDMRB register. Data inclusion in or exclusion from break conditions is set in the DBEB bit of the BRCR register.

5. Sequential use of channels A and B is set in the SEQ bit of the BRCR. When sequential use is designated, a user break occurs when the channel B conditions are matched after matching of channel A conditions.
6. When a user break occurs, the CMFA and CMFB bits in the BRCR register are set to 1. If a break is to be generated again, the CMFA and CMFB bits should be cleared to 0.

7.2 UBC Register Functions

7.2.1 Break Address Register A (BARA)

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

Bit:	23	22	21	20	19	18	17	16
Bit name:	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
Bit name:	BAA15	BAA14	BAA13	BAA12	BAA11	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 read/write register that stores the virtual address of the channel A break condition. It is not initialized by a reset.

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

7.2.2 Break Address Register B (BARB)

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

7.2.3 Break ASID Register A (BASRA)

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

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

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

7.2.4 Break ASID Register B (BASRB)

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

7.2.5 Break Address Mask Register A (BAMRA)

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	BASMA	BAMA1	BAMA0
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/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 read/write register that specifies which bits in the break ASID specified in BASRA and which bits in the break address specified in BARA are masked. It is not initialized by a reset.

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

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

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

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

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

7.2.6 Break Address Mask Register B (BAMRB)

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

7.2.7 Break Bus Cycle Register A (BBRA)

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

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

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

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

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

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

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

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

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

7.2.8 Break Bus Cycle Register B (BBRB)

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

7.2.9 Break Data Register B (BDRB)

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

Bit:	23	22	21	20	19	18	17	16
Bit name:	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
Bit name:	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

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

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

7.2.10 Break Data Mask Register B (BDMRB)

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

Bit:	23	22	21	20	19	18	17	16
Bit name:	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
Bit name:	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

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

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

n = 31 to 0

Notes: 1. When the data bus value is contained in the break conditions, specify the operand size.
2. For byte data, set the same data in bits 0–7 and bits 8–15 of BDRB and BDMRB.
3. Bits 31–16 of BDRB and BDMRB are ignored for byte and word sizes.

7.2.11 Break Control Register (BRCR)

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

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

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

BRCR:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7.3 UBC Operation

7.3.1 User Break Operation Flow

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

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

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

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

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

7.3.2 Instruction Fetch Cycle Break

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

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

7.3.3 Data Access Cycle Break

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

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

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

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

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

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

7.3.4 Saved Program Counter (PC) Value

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

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

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

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

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

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

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

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

7.3.5 Examples of Use

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

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

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

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00000404:	Address H'00000404
	BAMRA = H'00:	Address mask H'00
	BBRA = H'0014:	Bus cycle, instruction fetch (post-execution), read (operand size not included in conditions)
Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'00008010:	Address H'00008010
	BAMRB = H'02:	Address mask H'02
	BBRB = H'0014:	Bus cycle, instruction fetch (pre-execution), read (operand size not included in conditions)
	BDRB = H'00000000:	Data H'00000000
	BDMRB = H'00000000:	Data mask H'00000000

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

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

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

Channel A:	BASRA = H'80:	ASID H'80
	BARA = H'00037226:	Address H'00037226
	BAMRA = H'00:	Address mask H'00
	BBRA = H'0016:	Bus cycle, instruction fetch (pre-execution), read, word
Channel B:	BASRB = H'70:	ASID H'70
	BARB = H'0003722E:	Address H'0003722E
	BAMRB = H'00:	Address mask H'00
	BBRB = H'0016:	Bus cycle, instruction fetch (pre-execution), read, word
	BDRB = H'00000000:	Data H'00000000
	BDMRB = H'00000000:	Data mask H'00000000

   A user break is generated before execution of the instruction at address H'0003722E with

ASID = H'70, after execution of the instruction at address H'00037226 with ASID = H'80.

3. Data access cycle break condition setting

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

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

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

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

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

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

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

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

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

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

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

7.3.6 Cautions

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

Section 8 Power-Down Modes

8.1 Overview

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

8.1.1 Power-Down Modes

The SH7708 Series have the following power-down modes:

1. Sleep mode
2. Standby mode
3. Module standby function (TMU, RTC, and SCI on-chip supporting modules)
4. Hardware standby mode (SH7708S, SH7708R only)

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

Table 8.1 Power-Down Modes

		State							
Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts (Register held)	Held	Held	Run	Held	Refresh	1. Interrupt 2. Reset
Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts (Register held)	Held	Held	Halts*[1]	Held	Self-refresh	1. Interrupt 2. Reset
Hardware standby mode	Drive CA pin low	Halts	Halts (Register held)	Held	Held	Halts*[3]	Held	Self-refresh	Power-on reset
Module standby	Set MSTP bit of STBCR to 1	Runs	Runs	Held	Held	Specified module halts	*[2]	Refresh	1. Clear MSTP bit to 0 2. Reset

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

2. Depends on the on-chip supporting module.

TMU external pin: Held

SCI external pin: Reset

3. The RTC still runs if the START bit in RCR2 is set to 1 (see section 12, Realtime Clock (RTC)). The TMU does not run.

8.1.2 Register Configuration

Table 8.2 shows the configuration of the control register for the power-down modes.

Table 8.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'00*	H'FFFFFF82	Byte

Note: * Initialized by a power-on reset. In a manual reset, the register contents are retained.

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	Processor Operating Status
High	High	O	Reset
	Low		Sleep mode
Low	High		Standby mode
	Low		Normal operation

Note: The “normal operation” status applies during refresh cycles even in sleep mode.

8.2 Register Description

8.2.1 Standby Control Register (STBCR)

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

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

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

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

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

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

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

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

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

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

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

8.3 Sleep Mode

8.3.1 Transition to Sleep Mode

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

8.3.2 Canceling Sleep Mode

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

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

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

8.4 Standby Mode

8.4.1 Transition to Standby Mode

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

Table 8.4 Register States in Standby Mode

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

The procedure for moving to standby mode is as follows:

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

8.4.2 Canceling Standby Mode

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

Canceling with an Interrupt: The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRL,*1 or on-chip supporting module (except the interval timer)*2 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 register. Interrupts are accepted during standby mode even when the BL bit in the SR register is 1. Immediately after an interrupt is detected, the phase of the clock output of the CKIO pin may be unstable, until the processor starts interrupt handling. (The canceling condition is that the IRL3–IRL0 level is higher than the mask level in the I3–I0 bits in the SR register.)

Notes: 1. When the RTC is being used, standby mode can be canceled using IRL3–IRL0.
2. Standby mode can be canceled with an RTC or TMU (only when running on the RTC clock) interrupt.

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

8.4.3 Clock Pause Function

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

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

8.5 Module Standby Function

8.5.1 Transition to Module Standby Function

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

Bit	Value	Description
MSTP2	0	TMU runs.
	1	Supply of clock to TMU is halted. Registers are initialized.*1
MSTP1	0	RTC runs.
	1	Supply of clock to RTC is halted. Register access is prohibited.*2
MSTP0	0	SCI operates.
	1	Supply of clock to SCI is halted.

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

8.5.2 Clearing the Module Standby Function

The module standby function can be cleared by clearing the MSTP2–MSTP0 bits to 0, or by a power-on reset or manual reset.

8.6 Timing of STATUS Pin Changes

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

The meaning of the STATUS descriptions is as follows:

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

The meaning of the clock units is as follows:

Bcyc:	Bus clock cycle
Pcyc:	Peripheral clock cycle

8.6.1 Timing for Resets

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

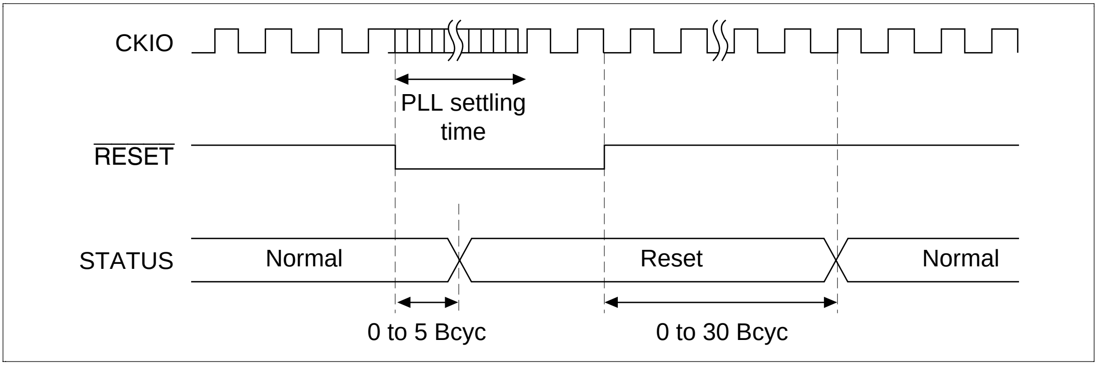


Figure 8.1 Power-On Reset (Clock Mode 0, 1, 2, and 7) STATUS Output

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

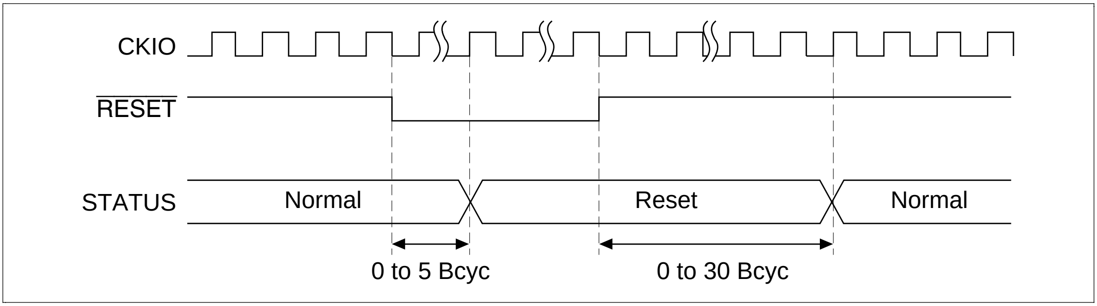
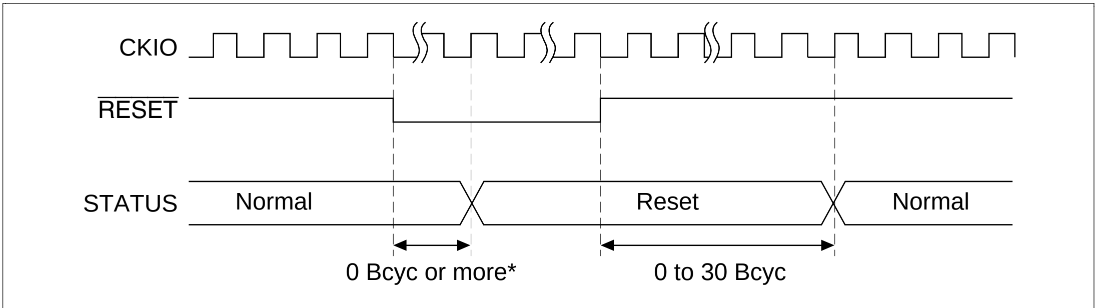


Figure 8.2 Power-On Reset (Clock Mode 3 and 4) STATUS Output

Manual Reset:



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

Figure 8.3 Manual Reset STATUS Output

8.6.2 Timing for Canceling Standbys

Standby to Interrupt:

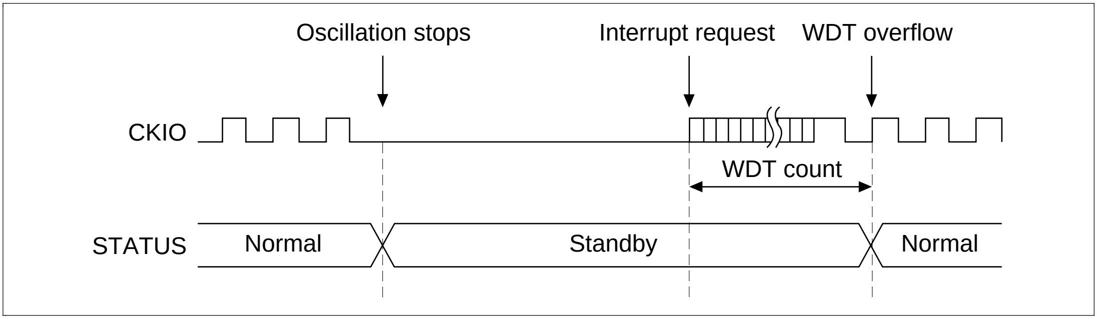
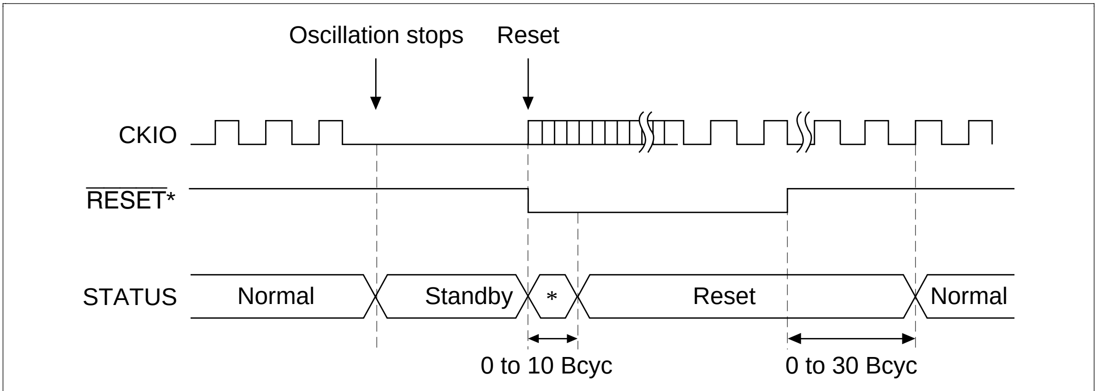


Figure 8.4 Standby to Interrupt STATUS Output

Standby to Power-On Reset:

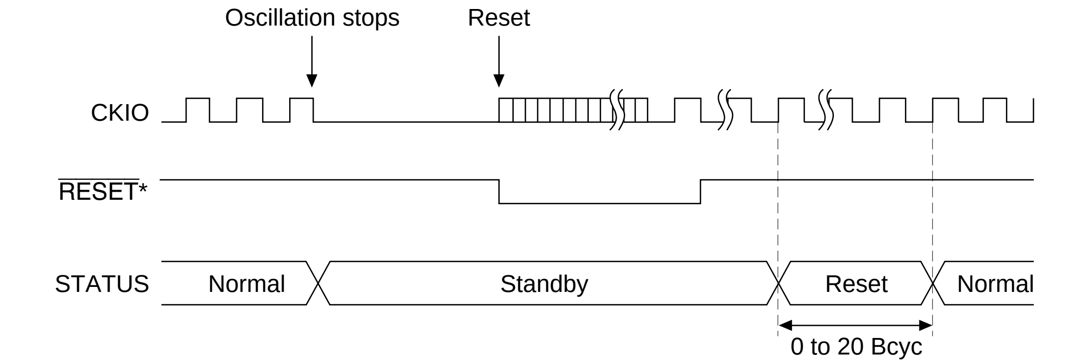


Note: When standby mode is cleared with a power-on reset, the WDT does not count. Keep RESET low during the PLL's oscillation settling time.

*: Undefined

Figure 8.5 Standby to Power-On Reset STATUS Output

Standby to Manual Reset:



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

Figure 8.6 Standby to Manual Reset STATUS Output

8.6.3 Timing for Canceling Sleep Mode

Sleep to Interrupt:

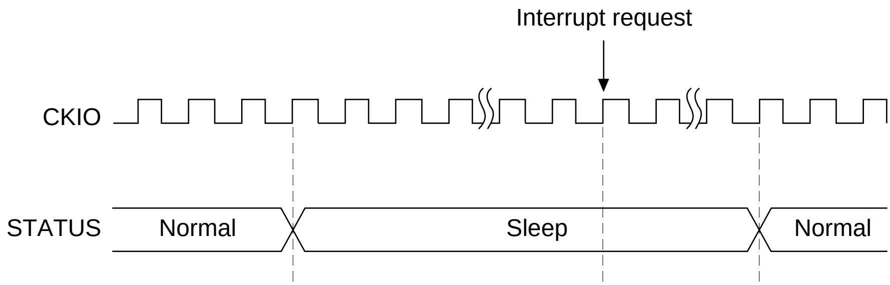
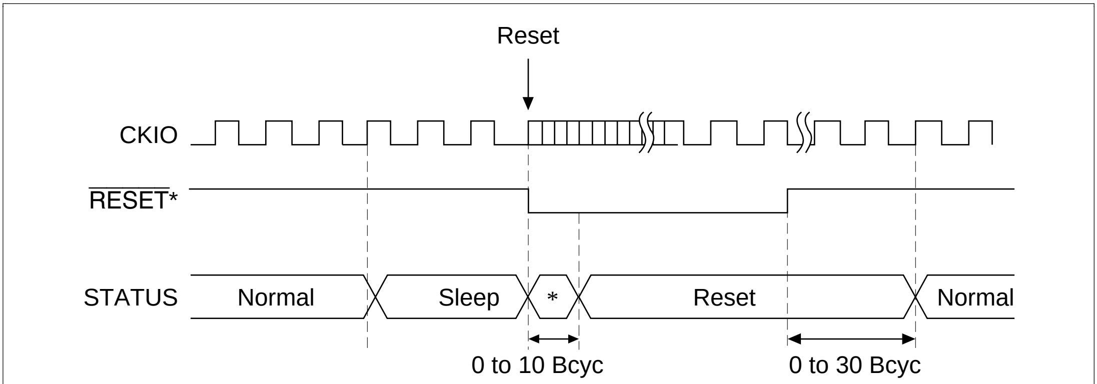


Figure 8.7 Sleep to Interrupt STATUS Output

Sleep to Power-On Reset:

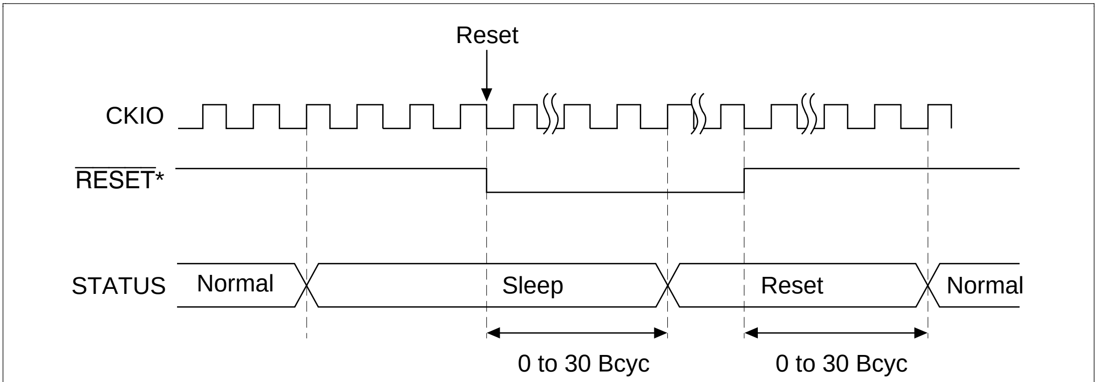


Note: When the PLL1's multiplication ratio is changed by a power-on reset, keep RESET low during the PLL's oscillation settling time.

*: Undefined

Figure 8.8 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset:



Note: Keep RESET low until the STATUS becomes reset.

Figure 8.9 Sleep to Manual Reset STATUS Output

8.7 Hardware Standby Mode

The hardware standby mode is provided only in the SH7708S and SH7708R. This mode is not supported in emulator.

8.7.1 Transition to Hardware Standby Mode

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

Hardware standby mode differs from standby mode as follows.

1. Interrupts and manual resets are not accepted.
2. The TCLK clock output is fixed low.
3. The TMU does not operate.
4. In the SH7708S and SH7708R, the RTC continues to operate even if power is not supplied to power supply pins other than those for RTC power. In this case, all output pins go to the non-drive state. With the SH7708, it is not possible to supply power only to the RTC power supply.

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

1. In standby mode

   The clock remains stopped and the chip enters the hardware standby state. Acceptance of interrupts and manual resets is disabled, TCLK output is fixed low, and the TMU halts.
2. During WDT operation when standby mode is canceled by an interrupt

   The chip enters hardware standby mode after standby mode is canceled and the CPU resumes operation.
3. In sleep mode

   The chip enters hardware standby mode after sleep mode is canceled and the CPU resumes operation.
4. During PLL standby (see section 9.6 for the PLL standby function)

   The chip enters hardware standby mode after forced implementation of the PLL OFF state.

Hold the CA pin low in hardware standby mode.

8.7.2 Canceling Hardware Standby Mode

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

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

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

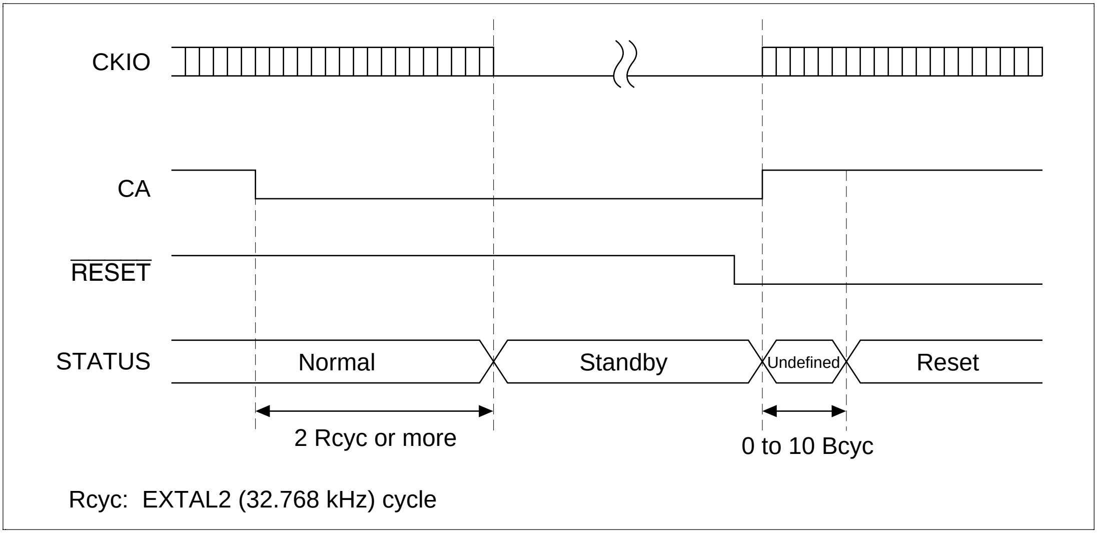
8.7.3 Hardware Standby Mode Timing

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

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

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

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



Rcyc: EXTAL2 (32.768 kHz) cycle

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

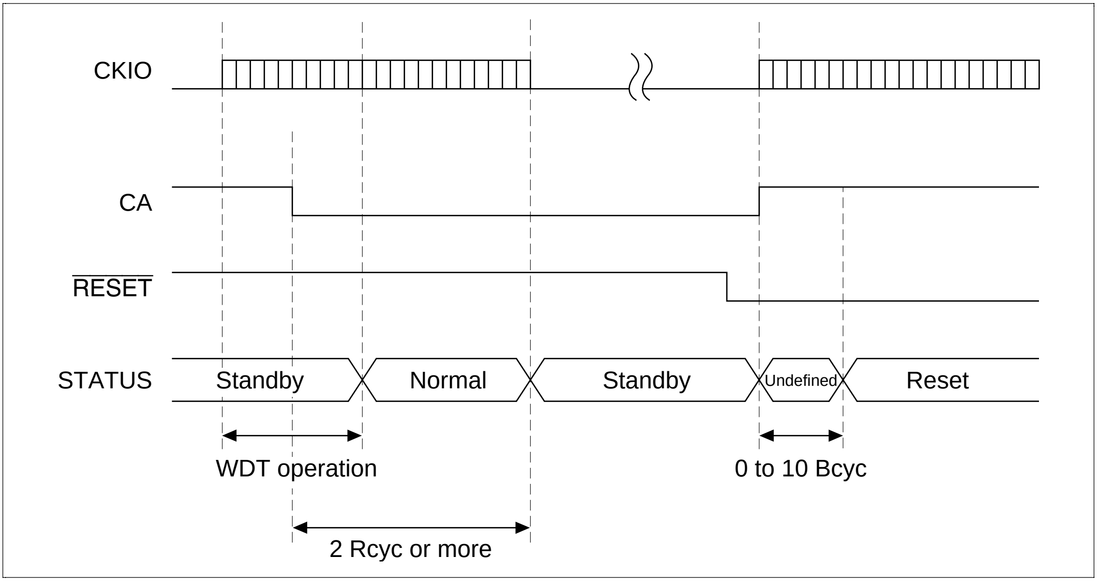


Figure 8.11 Hardware Standby Mode Timing (When CA Goes Low during WDT Operation on Standby Mode Cancellation)

Section 9 On-Chip Oscillation Circuits

9.1 Overview

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

9.1.1 Features

The CPG has the following features:

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

The WDT has the following features:

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

9.2 Overview of the CPG

9.2.1 CPG Block Diagram

A block diagram of the on-chip clock pulse generator is shown in figure 9.1(SH7708, SH7708S) and figure 9.2(SH7708R).

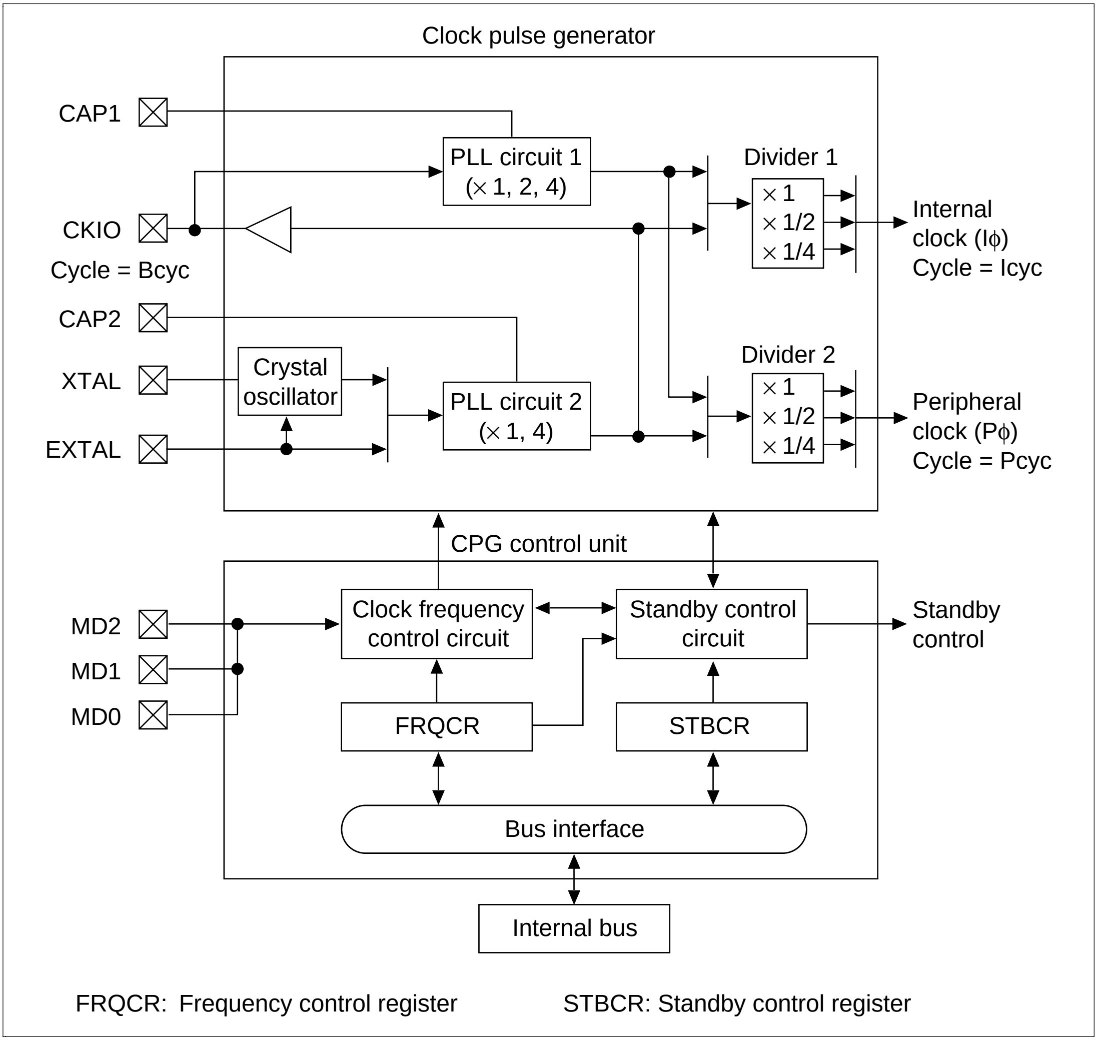


Figure 9.1 Block Diagram of Clock Pulse Generator(SH7708, SH7708S)

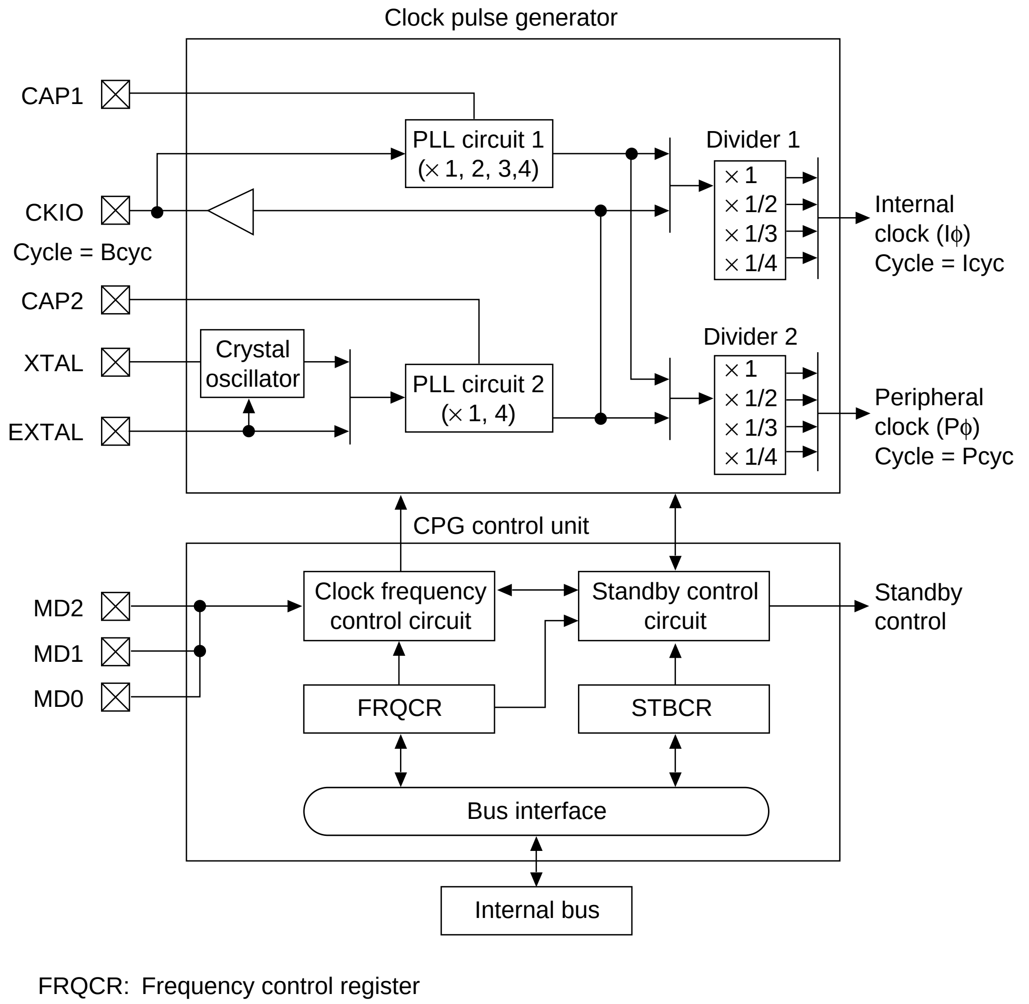


Figure 9.2 Block Diagram of Clock Pulse Generator(SH7708R)

The clock pulse generator blocks function as follows:

1. PLL Circuit 1: PLL circuit 1 doubles, triples*, quadruples, or leaves unchanged the input clock frequency from the CKIO terminal. The multiplication rate is set by the frequency control register. When this is done, the phase of the leading edge of the internal clock is controlled so that it will agree with the phase of the leading edge of the CKIO pin.
2. PLL Circuit 2: PLL circuit 2 leaves unchanged or quadruples the frequency of the crystal oscillator or the input clock frequency coming from the EXTAL pin. The multiplication ratio is fixed by the clock operation mode. The clock operation mode is set by pins MD0, MD1, and MD2. See table 9.3 for more information on clock operation modes.

3. Crystal Oscillator: This oscillator is used when a crystal oscillator element is connected to the XTAL and EXTAL pins. It operates according to the clock operating mode setting.
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the internal clock. The operating frequency can be 1, 1/2, 1/3*, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
5. Divider 2: Divider 2 generates a clock at the operating frequency used by the peripheral clock. The operating frequencies can be 1, 1/2, 1/3*, or 1/4 times the output frequency of PLL Circuit 1 or the clock frequency of the CKIO pin, as long as it stays at or below the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
6. Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using the MD pin and the frequency control register.
7. Standby Control Circuit: The standby control circuit controls the status of the clock pulse generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, on/off control of PLL circuit 1, PLL standby, the frequency multiplication ratio of PLL 1, and the frequency division ratio of the internal clock and the peripheral clock.
9. Standby Control Register: The standby control register has bits for controlling the power-down modes. See section 8, Power-Down Modes, for more information.

Note: * SH7708R only

9.2.2 CPG Pin Configuration

Table 9.1 lists the CPG pins and their functions.

Table 9.1 Clock Pulse Generator Pins and Functions

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

9.2.3 CPG Register Configuration

Table 9.2 shows the CPG register configuration.

Table 9.2 Register Configuration

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

Note: * Initialized by a power-on reset via the RESET pin. Register contents are retained in a power-on reset initiated by the WDT.

9.3 Clock Operating Modes

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

Table 9.3 Clock Operating Modes

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

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7708 Series. PLL circuit 1 is constantly on, and there are no frequency range restrictions compared to mode 3. An input clock frequency of 8 MHz to 60 MHz(SH7708, SH7708S) or 16 MHz to 60 MHz(SH7708R) can be used, and the CKIO frequency range is 8 MHz to 60 MHz(SH7708, SH7708S) or 16 MHz to 60 MHz(SH7708R).

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

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4 by PLL circuit 2 before being supplied inside the SH7708 Series, allowing a low-frequency external clock to be used. An input clock frequency of 5 MHz to 15 MHz can be used, and the CKIO frequency range is 20 MHz to 60 MHz.

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

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside the SH7708 Series, allowing a low crystal frequency to be used. A crystal oscillation frequency of 5 MHz to 15 MHz can be used, and the CKIO frequency range is 20 MHz to 60 MHz.

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

Mode 3: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside the SH7708 Series. 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 8 MHz to 15 MHz(SH7708, SH7708S) or 16 MHz to 25 MHz(SH7708R) can be used, and the CKIO frequency range is of 8 MHz to 15 MHz(SH7708, SH7708S) or 16 MHz to 25 MHz(SH7708R).

Mode 4: The on-chip crystal oscillator operates, with its output supplied inside the SH7708 Series 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 of 8 MHz to 15 MHz(SH7708, SH7708S) or 16 MHz to 25 MHz(SH7708R) can be used, and the CKIO frequency range is of 8 MHz to 15 MHz(SH7708, SH7708S) or 16 MHz to 25 MHz(SH7708R).

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and undergoes waveform shaping, and also frequency multiplication according to the setting, by PLL circuit 1 before being supplied to the SH7708 Series. In modes 0 to 6, the system clock is generated from the output of the SH7708 Series' CKIO pin. Consequently, if a large number of ICs are operating on the clock cycle, the CKIO pin load will be large. This mode, however, assumes a comparatively large-scale system. If a large number of ICs are operating on the clock cycle, a clock generator with a number of low-skew clock outputs can be provided, so that the ICs can operate synchronously by distributing the clocks to each one.

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

Table 9.4 Range of Usable Frequencies for Each Clock Operating Mode(SH7708, SH7708S)

Mode	FRQCR Register Value	PLL1	PLL2	Clock Ratio*1 (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
0	H'0102	ON (× 1)	ON (× 1)	1:1:1/4	8 MHz to 60 MHz	8 MHz to 60 MHz
	H'0101	ON (× 1)	ON (× 1)	1:1:1/2	8 MHz to 60 MHz	8 MHz to 60 MHz
	H'0100	ON (× 1)	ON (× 1)	1:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0111	ON (× 2)	ON (× 1)	2:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0115	ON (× 2)	ON (× 1)	1:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0122	ON (× 4)	ON (× 1)	4:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'0126	ON (× 4)	ON (× 1)	2:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'012a	ON (× 4)	ON (× 1)	1:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
1, 2	H'0102	ON (× 1)	ON (× 4)	4:4:1	5 MHz to 15 MHz	20 MHz to 60 MHz
	H'0101	ON (× 1)	ON (× 4)	4:4:2	5 MHz to 15 MHz	20 MHz to 60 MHz
	H'0100	ON (× 1)	ON (× 4)	4:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	H'0112	ON (× 2)	ON (× 4)	8:4:2	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	H'0111	ON (× 2)	ON (× 4)	8:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	H'0115	ON (× 2)	ON (× 4)	4:4:4	5 MHz to 7.5 MHz	20 MHz to 30 MHz
	H'0116	ON (× 2)	ON (× 4)	4:4:2	5 MHz to 7.5 MHz	20 MHz to 30 MHz
3, 4	H'0102	OFF	ON (× 1)	1:1:1/4	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'0101	OFF	ON (× 1)	1:1:1/2	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'0100	OFF	ON (× 1)	1:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01d1	ON (× 2)	ON (× 1)	2:1:1/2	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01d0	ON (× 2)	ON (× 1)	2:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01d4	ON (× 2)	ON (× 1)	1:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01d5	ON (× 2)	ON (× 1)	1:1:1/2	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01d6	ON (× 2)	ON (× 1)	1:1:1/4	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01e0	ON (× 4)	ON (× 1)	4:1:4	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01e4	ON (× 4)	ON (× 1)	2:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01e5	ON (× 4)	ON (× 1)	2:1:1/2	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01e8	ON (× 4)	ON (× 1)	1:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01e9	ON (× 4)	ON (× 1)	1:1:1/2	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'01ea	ON (× 4)	ON (× 1)	1:1:1/4	8 MHz to 15 MHz	8 MHz to 15 MHz

Table 9.4 Range of Usable Frequencies for Each Clock Operating Mode(SH7708, SH7708S) (cont)

Mode	FRQCR Register Value	PLL1	PLL2	Clock Ratio*[1] (I:B:P)	Input Clock/ Crystal Oscillator Frequency Range	CKIO Pin Frequency Range
5*[2]	H'0102	OFF	OFF	1/2:1/2:1/8	2 MHz to 30 MHz	1 MHz to 15 MHz
	H'0101	OFF	OFF	1/2:1/2:1/4	2 MHz to 30 MHz	1 MHz to 15 MHz
	H'0100	OFF	OFF	1/2:1/2:1/2	2 MHz to 30 MHz	1 MHz to 15 MHz
	H'01d1	ON (× 2)	OFF	1:1/2:1/4	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01d0	ON (× 2)	OFF	1:1/2:1/2	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01d4	ON (× 2)	OFF	1/2:1/2:1/2	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01d5	ON (× 2)	OFF	1/2:1/2:1/4	16 MHz to 30 Mhz	8 MHz to 15 Mhz
	H'01d6	ON (× 2)	OFF	1/2:1/2:1/8	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01e2	ON (× 4)	OFF	2:1/2:1/2	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01e6	ON (× 4)	OFF	1:1/2:1/2	16 MHz to 30 MHz	8 MHz to 15 MHz
	H'01ea	ON (× 4)	OFF	1/2:1/2:1/2	16 MHz to 30 MHz	8 MHz to 15 MHz
6*[2]	H'0102	OFF	OFF	1/2:1/2:1/8	2 MHz to 20 MHz	1 MHz to 10 MHz
	H'0101	OFF	OFF	1/2:1/2:1/4	2 MHz to 20 MHz	1 MHz to 10 MHz
	H'0100	OFF	OFF	1/2:1/2:1/2	2 MHz to 20 MHz	1 MHz to 10 MHz
	H'01d1	ON (× 2)	OFF	1:1/2:1/4	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01d0	ON (× 2)	OFF	1:1/2:1/2	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01d4	ON (× 2)	OFF	1/2:1/2:1/2	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01d5	ON (× 2)	OFF	1/2:1/2:1/4	16 MHz to 20 MHz	8 MHz to 10 Mhz
	H'01d6	ON (× 2)	OFF	1/2:1/2:1/8	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01e2	ON (× 4)	OFF	2:1/2:1/2	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01e6	ON (× 4)	OFF	1:1/2:1/2	16 MHz to 20 MHz	8 MHz to 10 MHz
	H'01ea	ON (× 4)	OFF	1/2:1/2:1/2	16 MHz to 20 MHz	8 MHz to 10 MHz
7	H'0102	ON (× 1)	OFF	1:1:1/4	8 MHz to 60 MHz	8 MHz to 60 MHz
	H'0101	ON (× 1)	OFF	1:1:1/2	8 MHz to 60 MHz	8 MHz to 60 MHz
	H'0100	ON (× 1)	OFF	1:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0112	ON (× 2)	OFF	2:1:1/2	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0111	ON (× 2)	OFF	2:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0115	ON (× 2)	OFF	1:1:1	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0116	ON (× 2)	OFF	1:1:1/2	8 MHz to 30 MHz	8 MHz to 30 MHz
	H'0122	ON (× 4)	OFF	4:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'0126	ON (× 4)	OFF	2:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz
	H'012a	ON (× 4)	OFF	1:1:1	8 MHz to 15 MHz	8 MHz to 15 MHz

Notes: 1. Taking input clock as 1.
Maximum frequencies: Iø = 60 MHz, Bø = 60 MHz, Pø = 30 MHz.

2. Modes 5 and 6 are available only in the SH7708.

Table 9.5 Range of Usable Frequencies for Each Clock Operating Mode(SH7708R)

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate*[1] (I:B:P)	Input Frequency Range	CKIO Frequency Range
0	H'0100	ON (× 1)	ON (× 1)	1:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0101	ON (× 1)	ON (× 1)	1:1:1/2	16 MHz to 60 MHz	16 MHz to 60 MHz
	H'0102	ON (× 1)	ON (× 1)	1:1:1/4	16 MHz to 60 Mhz	16 MHz to 60 Mhz
	H'0111	ON (× 2)	ON (× 1)	2:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2	16 MHz to 50 MHz	16 MHz to 50 MHz
	H'0115	ON (× 2)	ON (× 1)	1:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2	16 MHz to 50 MHz	16 MHz to 50 MHz
	H'0122	ON (× 4)	ON (× 1)	4:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'0126	ON (× 4)	ON (× 1)	2:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'012A	ON (× 4)	ON (× 1)	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’A100	ON (× 3)	ON (× 1)	3:1:1	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz
	H’E100	ON (× 3)	ON (× 1)	1:1:1	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz
	H’E101	ON (× 3)	ON (× 1)	1:1:1/2	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz
1, 2	H'0100	ON (× 1)	ON (× 4)	4:4:4	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
	H'0101	ON (× 1)	ON (× 4)	4:4:2	5 MHz to 15 MHz	20 MHz to 60 MHz
	H'0102	ON (× 1)	ON (× 4)	4:4:1	5 MHz to 15 MHz	20 MHz to 60 MHz
	H'0111	ON (× 2)	ON (× 4)	8:4:4	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
	H'0112	ON (× 2)	ON (× 4)	8:4:2	5 MHz to 12.5 MHz	20 MHz to 50 MHz
	H'0115	ON (× 2)	ON (× 4)	4:4:4	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
	H'0116	ON (× 2)	ON (× 4)	4:4:2	5 MHz to 12.5 MHz	20 MHz to 50 MHz
	H’A100	ON (× 3)	ON (× 4)	12:4:4	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
	H’E100	ON (× 3)	ON (× 4)	4:4:4	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
	H’E101	ON (× 3)	ON (× 4)	4:4:2	5 MHz to 8.3 MHz	20 MHz to 33.3 MHz
3	H'0100	OFF	ON (× 1)	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'0101	OFF	ON (× 1)	1:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'0102	OFF	ON (× 1)	1:1:1/4	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01D0	ON (× 2)	ON (× 1)	2:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01D1	ON (× 2)	ON (× 1)	2:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01D2	ON (× 2)	ON (× 1)	2:1:1/4	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01D4	ON (× 2)	ON (× 1)	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01D5	ON (× 2)	ON (× 1)	1:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’01D6	ON (× 2)	ON (× 1)	1:1:1/4	16 MHz to 25 MHz	16 MHz to 25 MHz

Table 9.5 Range of Usable Frequencies for Each Clock Operating Mode(SH7708R) (cont)

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate*[1] (I:B:P)	Input Frequency Range	CKIO Frequency Range
3	H’81C0	ON (× 3)	ON (× 1)	3:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’81C1	ON (× 3)	ON (× 1)	3:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’C1C0	ON (× 3)	ON (× 1)	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’C1C1	ON (× 3)	ON (× 1)	1:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E0	ON (× 4)	ON (× 1)	4:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’01E1	ON (× 4)	ON (× 1)	4:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E4	ON (× 4)	ON (× 1)	2:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E5	ON (× 4)	ON (× 1)	2:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E6	ON (× 4)	ON (× 1)	2:1:1/4	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E8	ON (× 4)	ON (× 1)	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01E9	ON (× 4)	ON (× 1)	1:1:1/2	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'01EA	ON (× 4)	ON (× 1)	1:1:1/4	16 MHz to 25 MHz	16 MHz to 25 MHz
4	H'0100	OFF	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'0101	OFF	ON (× 1)	1:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'0102	OFF	ON (× 1)	1:1:1/4	16 MHz to 20 MHz	16 MHz to 20MHz
	H'01D0	ON (× 2)	ON (× 1)	2:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01D1	ON (× 2)	ON (× 1)	2:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01D2	ON (× 2)	ON (× 1)	2:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01D4	ON (× 2)	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01D5	ON (× 2)	ON (× 1)	1:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’01D6	ON (× 2)	ON (× 1)	1:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’81C0	ON (× 3)	ON (× 1)	3:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’81C1	ON (× 3)	ON (× 1)	3:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’C1C0	ON (× 3)	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’C1C1	ON (× 3)	ON (× 1)	1:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E0	ON (× 4)	ON (× 1)	4:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H’01E1	ON (× 4)	ON (× 1)	4:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E4	ON (× 4)	ON (× 1)	2:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E5	ON (× 4)	ON (× 1)	2:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E6	ON (× 4)	ON (× 1)	2:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E8	ON (× 4)	ON (× 1)	1:1:1	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01E9	ON (× 4)	ON (× 1)	1:1:1/2	16 MHz to 20 MHz	16 MHz to 20 MHz
	H'01EA	ON (× 4)	ON (× 1)	1:1:1/4	16 MHz to 20 MHz	16 MHz to 20 MHz

Table 9.5 Range of Usable Frequencies for Each Clock Operating Mode(SH7708R) (cont)

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate*[1] (I:B:P)	Input Frequency Range	CKIO Frequency Range
7	H'0100	ON (× 1)	OFF	1:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0101	ON (× 1)	OFF	1:1:1/2	16 MHz to 60 MHz	16 MHz to 60 MHz
	H'0102	ON (× 1)	OFF	1:1:1/4	16 MHz to 60 MHz	16 MHz to 60 MHz
	H'0111	ON (× 2)	OFF	2:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0112	ON (× 2)	OFF	2:1:1/2	16 MHz to 50 MHz	16 MHz to 50 MHz
	H'0115	ON (× 2)	OFF	1:1:1	16 MHz to 33.3 MHz	16 MHz to 33.3 MHz
	H'0116	ON (× 2)	OFF	1:1:1/2	16 MHz to 50 MHz	16 MHz to 50 MHz
	H'0122	ON (× 4)	OFF	4:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'0126	ON (× 4)	OFF	2:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H'012A	ON (× 4)	OFF	1:1:1	16 MHz to 25 MHz	16 MHz to 25 MHz
	H’A100	ON (× 3))	OFF	3:1:1	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz
	H’E100	ON (× 3)	OFF	1:1:1	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz
	H’E101	ON (× 3)	OFF	1:1:1/2	25 MHz to 33.3 MHz	25 MHz to 33.3 MHz

Notes: 1. Input clock frequency is 1

2. Max frequency : Iϕ = 100 MHz, Bϕ = (CKIO) = 60 MHz, Pϕ = 30 MHz

Cautions:

1. When clock operating modes 3 and 4 are used:
   - The on/off state of PLL circuit 1 is set by the frequency control register.
   - PLL circuit 1 is initialized to the off state by a power-on reset.
   - Always turn PLL circuit 1 off before going into standby mode.
2. The input to divider 1 becomes the output of:
   - PLL circuit 1 when PLL circuit 1 is on.
   - PLL circuit 2 when PLL circuit 1 is off and PLL circuit 2 is on.
   - Divider 3 when PLL circuit 1 is off and PLL circuit 2 is off.
3. The input of divider 2 becomes the output of:
   - PLL circuit 1 when the clock operating mode is 0–2 or 7.
   - PLL circuit 2 when the clock operating mode is 3 and 4 and PLL circuit 2 is on.
   - Divider 3 when the clock operating mode is 3 and 4 and PLL circuit 2 is off.
4. The frequency of the internal clock (Iϕ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on.
   - Equal to the frequency of CKIO pin when PLL circuit 1 is off.

- Do not set the internal clock frequency lower than the CKIO pin frequency.

5. The frequency of the peripheral clock (Pϕ) becomes:
    - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2 when the clock operating mode is 0–2 or 7.
    - The product of the frequency of the CKIO pin and the division ratio of divider 2 when the clock operating mode is 3 and 4.
    - The peripheral clock frequency should not be set higher than the frequency of the CKIO pin, higher than 30 MHz(SH7708, SH7708S) / 33.3 MHz(SH7708R), or lower than 1/4 (SH7708, SH7708S) / 1/8(SH7708R) the internal clock (Iϕ).
6. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1. This frequency should be equal to or lower than 60 MHz(SH7708, SH7708S) / 100 MHz(SH7708R).
7. × 1, × 2, × 3* or × 4 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, × 1/3*, and × 1/4 can be selected as the division ratios of dividers 1 and 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 is determined by the mode. × 3 multiplication of PLL circuit 1 and × 1/3 of dividers 1 and 2 are not supported in emulator.

Note: SH7708R only

8. For more in formation about the range of usable freguencies for each clock operating mode,see table 9.4.

9.4 Register Descriptions

9.4.1 Frequency Control Register (FRQCR)

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

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

SH7708, SH7708S:

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	CKOEN
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	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 8—Clock Enable (CKOEN): Used to output a clock from the CKIO pin or to fix the level of the CKIO pin. Even when the level is fixed, the SH7708 Series will operate internally at the frequency before the level was fixed. In case of clock operating mode 7, the CKIO pin becomes an input pin irrespective of the value of this bit.

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

Bit 7—PLL Circuit Enable (PLLEN): Specifies the on/off state of PLL circuit 1. This bit is valid in clock operating modes 3–6. PLL circuit 1 goes on when the clock operating mode is 0–2 or 7 irrespective of the value of PLLEN.

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

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

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

Bits 5 and 4—Frequency Multiplication Ratio (STC1, STC0): These 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): These 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 mode, 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): These bits specify the division ratio of the peripheral clock frequency with respect to the frequency of the output frequency of PLL circuit 1 or the frequency of the CKIO pin.

Bit 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 frequency of the CKIO pin.

SH7708R:

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

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

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

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

Note: Do not set the output frequency of PLL circuit 1 higher than 100MHz.

Bits 14, 3 and 2—Internal Clock Frequency Division Ratio (IFC): These 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 mode, set × 1.

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

Note: Do not set the interunal clock frequency lower then the CKIO frequency.

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

Bit 13: PFC2	Bit 1: PFC1	Bit 0: PFC0	Description
0	0	0	× 1
0	0	1	× 1/2
1	0	0	× 1/3
1	1	0	× 1/4

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

Bit 8—Clock Enable (CKOEN): Used to output a clock in operating mode 3–6 from the CKIO pin or to fix the level of the CKIO pin in clock operation modes 3 and 4. Even when the level is fixed, the SH7708R will operate internally at the frequency before the level was fixed. In case of clock operating mode 0–2, set this bit to 1. In case of clock operating mode 7, the CKIO pin becomes an input pin irrespective of the value of this bit.

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

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

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

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

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

9.5 Changing the Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of dividers 1 and 2. All of these are controlled by software through the frequency control register. The methods are described below. In modes 3–6, the frequency can also be changed by turning PLL circuit 1 on and off, as described in section 9.6, PLL Standby Function.

9.5.1 Changing the Multiplication Rate

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

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

9.5.2 Changing the Division Ratio

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

1. In the initial state, 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 rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

9.6 PLL Standby Function

9.6.1 Overview of the PLL Standby Function

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

9.6.2 Usage

From Off to On:

1. Initially, PSTBY = 0, PLLEN = 0, and PLL circuit 1 is stopped. The output of PLL circuit 2 is used for divider 1 input.
2. When the multiplication rate 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 rate. The input to divider 1 is still the output of PLL circuit 2 at this point.
3. After PLL circuit 1 oscillation has stabilized, the input of divider 1 switches when PLLEN is set to 1 and the oscillation output of PLL circuit 1 is divided and becomes the internal clock. At this time, the division ratio can be changed by changing the settings of 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 output do not stop.

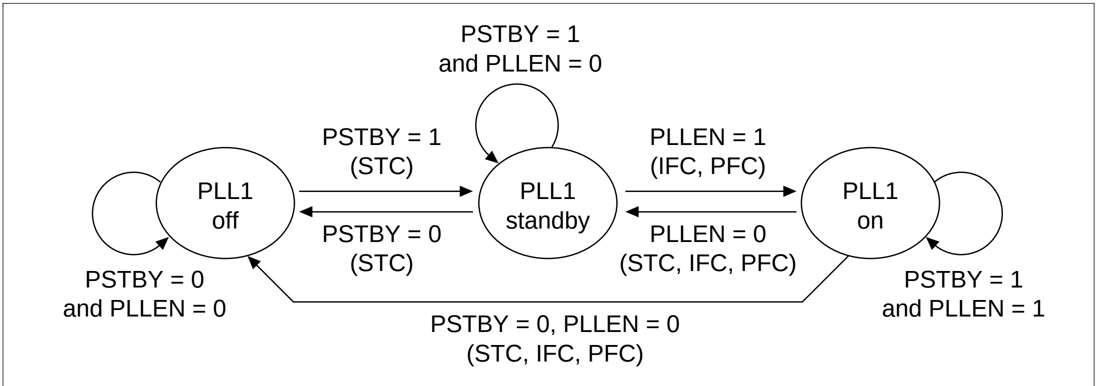
From On to Off:

1. When PLLEN is set to 0, the input of divider 1 switches to the output of PLL circuit 2. At this time, the division ratio can be changed by changing the settings of 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 in 1 above.

Notes: 1. There are some restrictions on the PLL standby state (PSTBY = 1, PLLEN = 0) as follows: The settings of the frequency control register’s CKOEN, STC2–STC0, IFC2–IFC0 and PFC2–PFC0 bits generally cannot be changed. In some cases, however, they can be changed if the PSTBY and PLLEN bit settings are also changed simultaneously (figure 9.3). The SLEEP instruction cannot be executed.

2. It is the responsibility of software to ensure the oscillation settling time. If PLLEN is set to 1 before the oscillation has settled, malfunctions may be caused by an unstable clock.

3. In clock modes 3 and 4, the SH7708 Series cannot go to standby mode while PLL circuit 1 is on. Always set PSTBY and PLLEN to 0 to stop PLL circuit 1 before going to standby mode.
4. When PSTBY and PLLEN are both changed from 0 to 1 together, the WDT will automatically start counting and the clock will switch when the WDT overflows. See section 9.5, Changing the Frequency, for setting the WDT.



Note: Bits in parentheses can be changed simultaneously.

Figure 9.3 State Transitions for the PLL Standby Function

9.7 Controlling Clock Output

The CKOEN bit in the FRQCR register can be used to switch between outputting a clock to the CKIO pin or having the level fixed.

9.7.1 Clock Modes 0–2

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

9.7.2 Clock Modes 3 and 4

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

9.8 Overview of the Watchdog Timer (WDT)

9.8.1 Block Diagram of the WDT

Figure 9.4 shows a block diagram of the WDT.

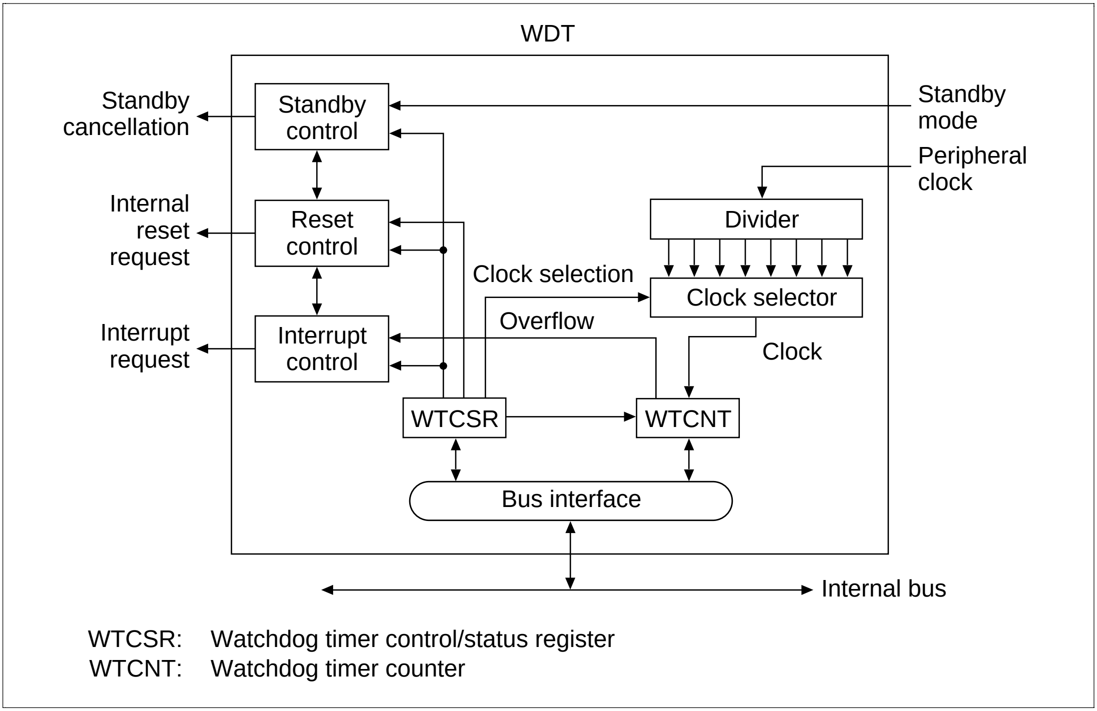


Figure 9.4 Block Diagram of the WDT

9.8.2 Register Configurations

The WDT has two registers that select the clock, switch the timer mode, and perform other functions. Table 9.5 shows the WDT register.

Table 9.5 Register Configuration

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

Note: * Write with a word access. Write H'5A and H'A5, respectively, in the upper bytes. Byte or longword writes are not possible. Read with a byte access.

9.9 WDT Registers

9.9.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit read/write counter that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. Its address is H'FFFFFF84. The WTCNT counter is initialized to H'00 only by a power-on reset through the RESET pin. Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to read WTCNT.

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

9.9.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags. Its address is H'FFFFFF86. The WTCSR register is initialized to H'00 only by a power-on reset through the RESET pin. When a WDT overflow causes an internal reset, the WTCSR retains its value. When used to count the clock settling time for canceling a standby, it retains its value after counter overflow. Use a word access to write to the WTCSR counter, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Bit:	7	6	5	4	3	2	1	0
	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Timer Enable (TME): Starts and stops timer operation. Clear this bit to 0 when using the WDT in standby mode or when changing the clock frequency.

Bit 7: TME	Description
0	Timer disabled: Count-up stops and WTCNT value is retained (Initial value)
1	Timer enabled

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

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

Note: If WT/$\overline{\text{IT}}$ is modified when the WDT is running, the up-count may not be performed correctly.

Bit 5—Reset Select (RSTS): Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored.

Bit 5: RSTS	Description
0	Power-on reset (Initial value)
1	Manual reset

Bit 4—Watchdog Timer Overflow (WOVF): Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode.

Bit 4: WOVF	Description
0	No overflow (Initial value)
1	WTCNT has overflowed in watchdog timer mode

Bit 3—Interval Timer Overflow (IOVF): Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode.

Bit 3: IOVF	Description
0	No overflow (Initial value)
1	WTCNT has overflowed in interval timer mode

Bits 2 to 0—Clock Select 2–0 (CKS2–CKS0): These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock. The overflow period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Division Ratio	Overflow Period (when Pϕ = 15 MHz)
0	0	0	1 (Initial value)	17 µs
		1	1/4	68 µs
	1	0	1/16	273 µs
		1	1/32	546 µs
1	0	0	1/64	1.09 ms
		1	1/256	4.36 ms
	1	0	1/1024	17.46 ms
		1	1/4096	69.84 ms

Note: If bits CKS2–CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

9.9.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 9.5. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

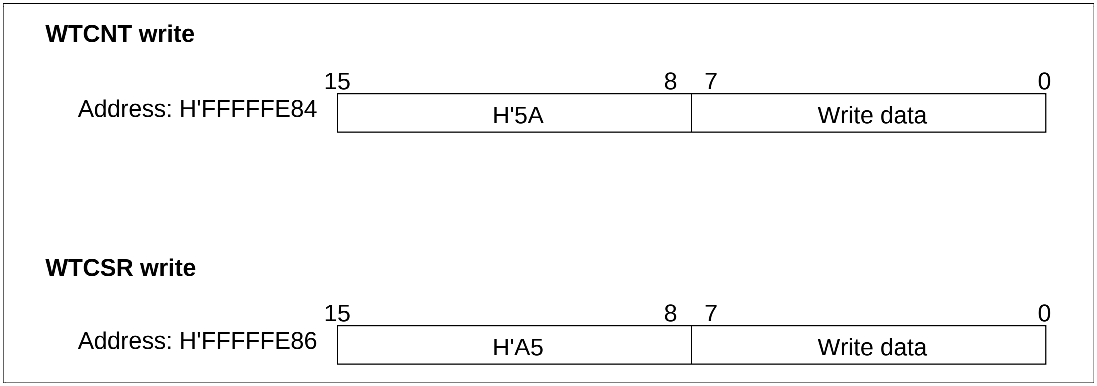


Figure 9.5 Writing to WTCNT and WTCSR

9.10 Using the WDT

9.10.1 Canceling Standbys

The WDT can be used to cancel standby mode with an NMI or other interrupts. The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the RESET pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal or detecting interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. The counter stops at the values H'00–H'01. The stop value depends on the clock ratio.

9.10.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2–CKS0 bits of WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the clock stops and the processor enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00–H'01. The stop value depends on the clock ratio.

9.10.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2–CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

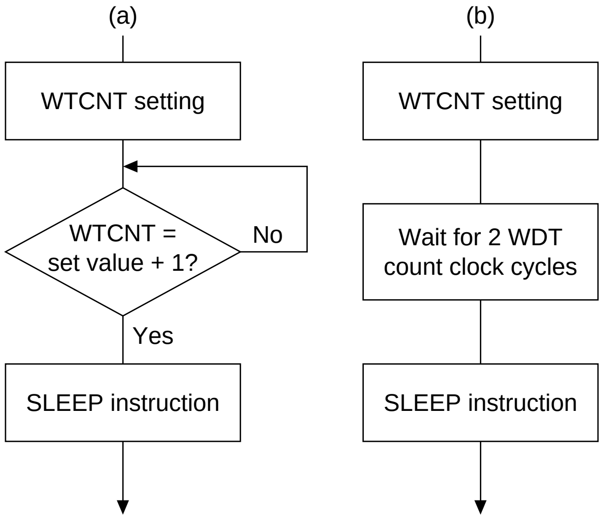
9.10.4 Using Interval Timer Mode

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

1. Clear the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 0, set the type of count clock in the CKS2–CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to INTC. The counter then resumes counting.

9.10.5 Usage Notes

WTCNT should be set as shown below when using the WDT of the SH7708 (not including the SH7708S and SH7708R) in watchdog timer mode or interval timer mode.

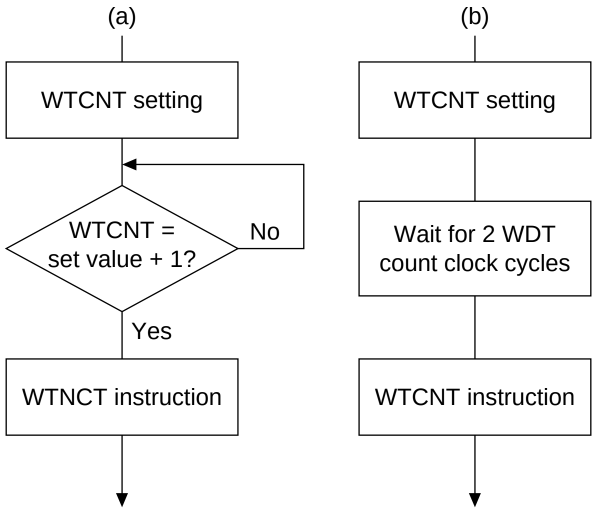


Perform either processing (a) or processing (b). No processing is required when the WDT is not used (WTCSR.TME = 0). If this processing is not performed, the WDT count will stop in sleep mode.

(a) Read WTCNT and confirm that it has been incremented, then execute a SLEEP instruction.

(b) Wait for two WDT count clock cycles after WTCNT is set, then execute a SLEEP instruction.

Figure 9.6 To Operate the WDT in Sleep Mode



Perform either processing (a) or processing (b). No processing is required when the WDT is not used (WTCSR.TME = 0). If this processing is not performed, the second WTCNT setting may not be carried out.

(a)

After the first WTCNT setting, read WTCNT and confirm that it has been incremented, then perform the next WTCNT setting.

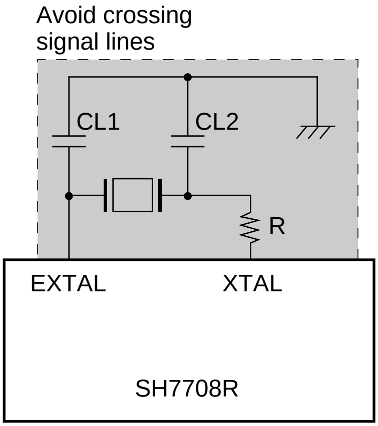
(b)

After the first WTCNT setting, wait for two WDT count clock

Figure 9.7 To Perform Repeated WTCNT Settings

9.11 Notes on Board Design

When Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



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

Figure 9.8 Points for Attention when Using Crystal Resonator

Decoupling Capacitors: As far as possible, insert a laminated ceramic capacitor of 0.01 to 0.1 µF as a passive capacitor for each V_{SS}/V_{CC} pair. Mount the passive capacitors as close as possible to the LSI power supply pins, and use components with a frequency characteristic suitable for the LSI operating frequency, as well as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 6-7, 17-18, 19-20, 30-31, 41-42, 49-50, 54-55, 59-60, 68-69, 82-81,83, 100-102, 115-116, 121-122, 127-128, 144-139

On-chip oscillator V_{SS}/V_{CC} pairs: 73-75, 76-78, 138-135

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{SS} (PLL1) and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and do not locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{CC} or V_{SS} and leave the XTAL pin open.

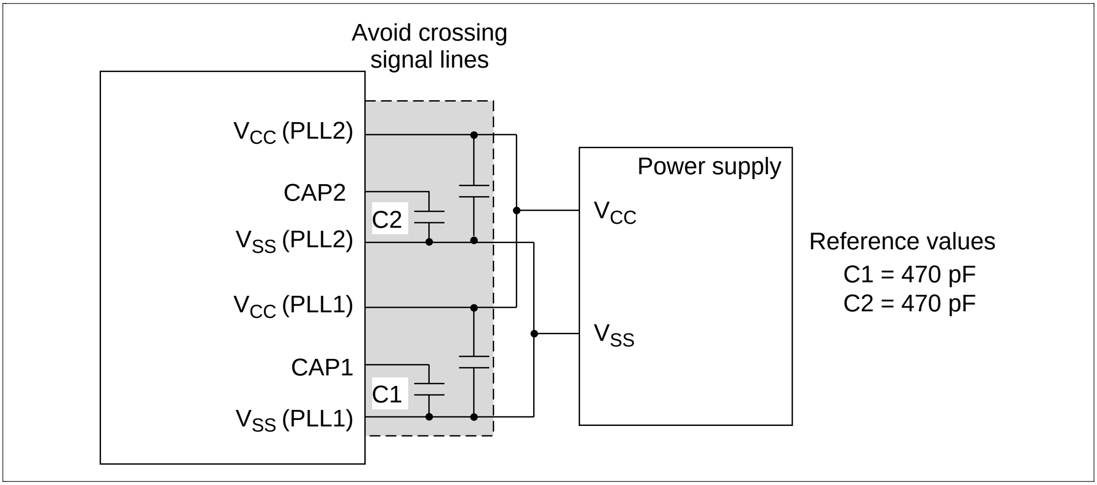


Figure 9.9 Points for Attention when Using PLL Oscillator Circuit

Power Supply Pin Wiring: To prevent mutual interference between power supply pins V_{CC} (PLL1), V_{CC} (PLL2) and power supply pins V_{SS} (PLL1), V_{SS} (PLL2), use independent wiring patterns for these two systems, and also other digital V_{CC} and V_{SS} pins, from the board power supply source.

External Device Connection: When connecting a device such as synchronous DRAM that requires timing adjustment, since the AC characteristics are based on the connection of a 30 pF capacitive load to the CKIO output, the total capacitive load of CKIO should be 30 pF or less. If the total capacitive load of the CKIO output exceeds 30 pF, a device requiring timing adjustment should be connected directly to CKIO, while in the case of a device without stringent timing adjustment requirements, the CKIO output should be received by a buffer to allow the total capacitive load of the CKIO output to be reduced.

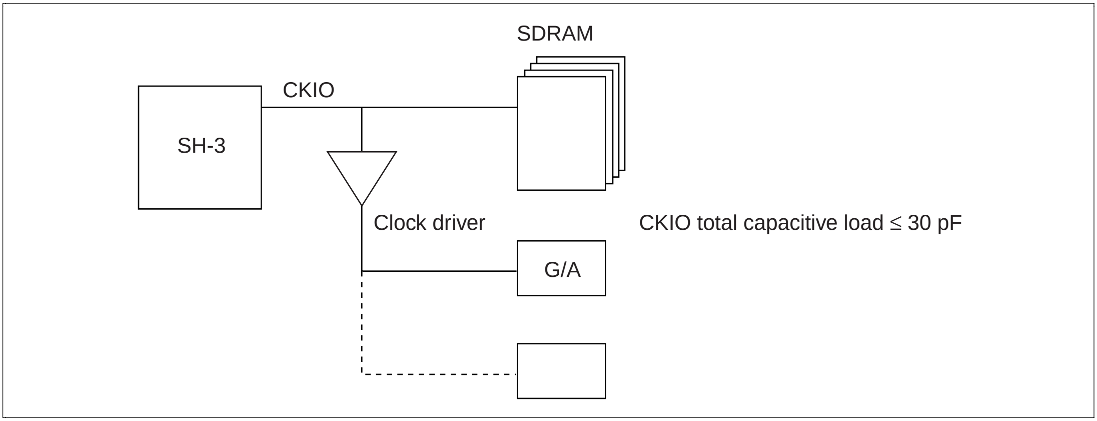


Figure 9.10 System Connection Example

Section 10 Bus State Controller (BSC)

10.1 Overview

The bus state controller (BSC) divides physical address space and outputs control signals for various types of memory and bus interface specifications. BSC functions enable the SH7708 Series to link directly with DRAM, synchronous DRAM, pseudo-SRAM, SRAM, ROM, and other memory storage devices without an external circuit. The BSC also allows direct connection to PCMCIA interfaces, simplifying system design and allowing high-speed data transfers in a compact system.

10.1.1 Features

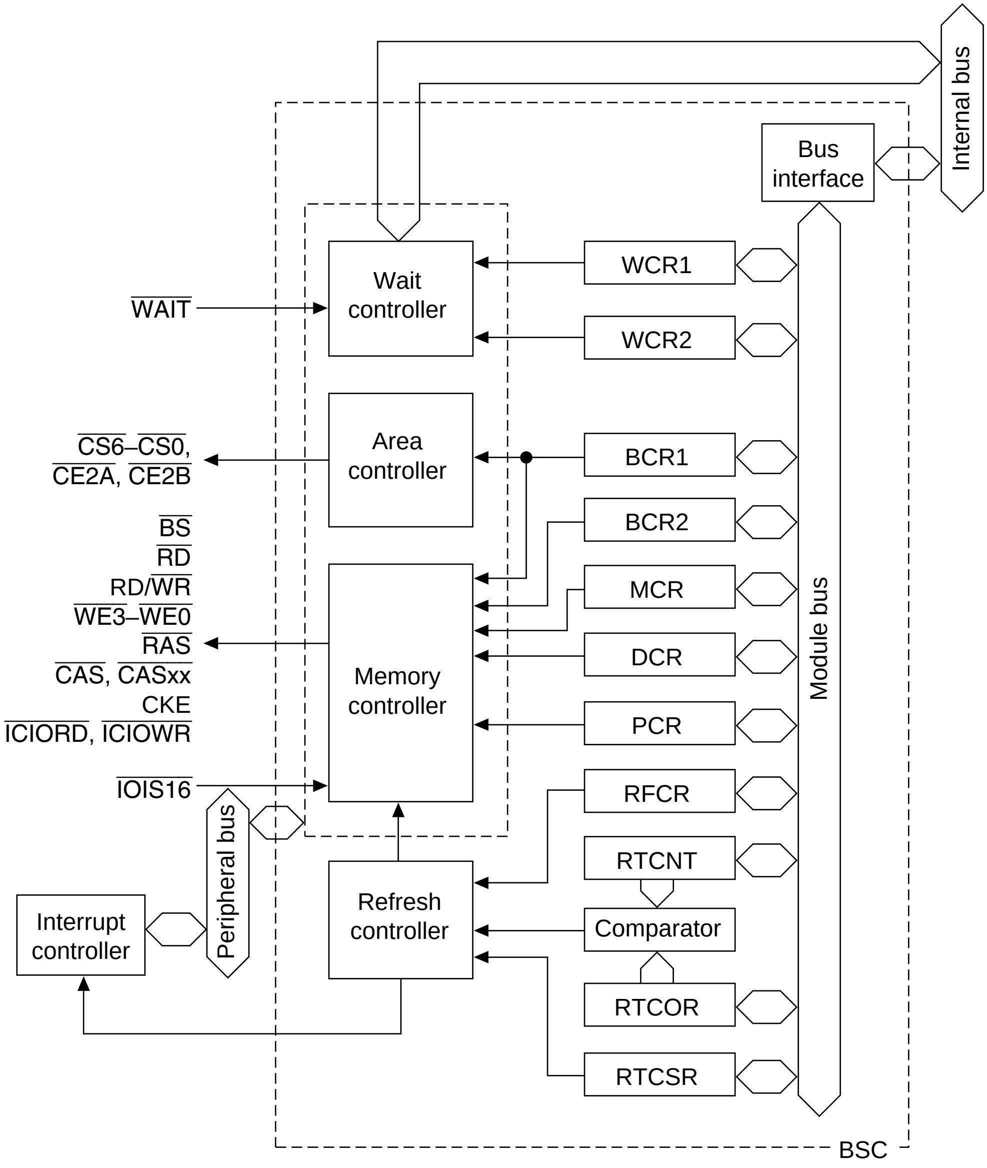
The BSC has the following features:

- Physical address space is divided into seven areas
  - A maximum 64 Mbytes for each of the seven areas, 0–6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the WAIT pin
  - Wait state insertion can be controlled through software. Register settings can be used to specify the insertion of 1–10 cycles independently for each area (areas 1 and 2 have a common setting)
  - The type of memory connected can be specified for each area, and control signals are output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous memory accesses to different areas or writes directly following reads of the same area
- Direct interface to DRAM
  - Multiplexes row/column addresses according to DRAM capacity
  - Supports burst operation (high-speed page mode, hyper page mode)
  - Supports CAS-before-RAS refresh and self-refresh
  - Performs low power 4-CAS-system byte control
  - Controls timing of DRAM direct-connection control signals according to register settings
- Direct interface to synchronous DRAM
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation
  - Has both auto-refresh and self-refresh functions
  - Controls timing of synchronous DRAM direct-connection control signals according to register setting

- Direct interface to pseudo-SRAM
  - Supports burst operation (static column mode)
  - Auto-refresh and self-refresh
- ROM burst interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface
  - Insertion of wait states controllable through software
  - Burst operation (page mode)
  - Bus sizing function for I/O bus width (little-endian mode only)
- Fine refreshing control
  - Supports refresh operation immediately after self-refresh operation in low-power DRAM by means of refresh counter overflow interrupt function
- Refresh counter can be used as an interval timer
  - Interrupt request generated at compare-match
  - Interrupt request generated at refresh counter overflow

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the bus state controller.



WCR:	Wait state control register	RFCR:	Refresh count register
BCR:	Bus control register	RTCNT:	Refresh timer count register
MCR:	Memory control register	RTCOR:	Refresh time constant register
DCR:	DRAM control register	RTCSR:	Refresh timer control/status register
PCR:	PCMCIA control register		

Figure 10.1 BSC Block Diagram

10.1.3 Pin Configuration

Table 10.1 lists the BSC pins.

Table 10.1 Pin Configuration

Pin Name	Signal	I/O	Description
Address bus	A25–A0	O	Address output
Data bus	D31–D24, D15–D0	I/O	Data I/O When port function is used, D31–D24 cannot be used. Leave these pins open.
Data bus/port	D23–D16/ PORT7–PORT0	I/O	When port function is not used, data I/O; when port function is used, port (I/O is set by register for each bit)
Bus cycle start	$\overline{\text{BS}}$	O	Shows start of bus cycle. During burst transfers, asserts every data cycle.
Chip select 6–0	$\overline{\text{CS6}}$–$\overline{\text{CS0}}$	O	Chip select signal to indicate area being accessed. $\overline{\text{CS5}}$ and $\overline{\text{CS6}}$ can also be used as $\overline{\text{CE1A}}$ and $\overline{\text{CE1B}}$ of PCMCIA.
Read/write	RD/$\overline{\text{WR}}$	O	Data bus direction indicator signal. DRAM/synchronous DRAM/PCMCIA write indicator signal.
Row address strobe	$\overline{\text{RAS}}$/$\overline{\text{CE}}$	O	When DRAM/synchronous DRAM is used, RAS signal. When pseudo-SRAM is used, $\overline{\text{CE}}$ signal.
Column address strobe	$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/ $\overline{\text{OE}}$	O	When synchronous DRAM is used, $\overline{\text{CAS}}$ signal. When DRAM is used, $\overline{\text{CAS}}$ signal for D7–D0. When pseudo-SRAM is used, $\overline{\text{OE}}$/$\overline{\text{RFSH}}$ signal.
Column address strobe LH	$\overline{\text{CASLH}}$	O	When DRAM is used, $\overline{\text{CAS}}$ signal for D15–D8
Column address strobe HL	$\overline{\text{CASHL}}$, $\overline{\text{CAS2L}}$	O	When DRAM is used, $\overline{\text{CAS}}$ signal for D23–D16. When the area 2 DRAM is being used, $\overline{\text{CAS}}$ signal for D7–D0.
Column address strobe HH	$\overline{\text{CASHH}}$, $\overline{\text{CAS2H}}$	O	When DRAM is used, $\overline{\text{CAS}}$ signal for D31–D24. When the area 2 DRAM is being used, $\overline{\text{CAS}}$ signal for D15–D8.

Table 10.1 Pin Configuration (Preliminary) (cont)

Pin Name	Signal	I/O	Description
Data enable 0	DQMLL/$\overline{\text{WE0}}$	O	When synchronous DRAM is used, selects D7–D0. For other memory, D7–D0 write strobe signal.
Data enable 1	DQMLU/$\overline{\text{WE1}}$	O	When synchronous DRAM is used, selects D15–D8. When PCMCIA is used, strobe signal that indicates the write cycle. For other memory, D15–D8 write strobe signal.
Data enable 2	DQMUL/$\overline{\text{WE2}}$/ $\overline{\text{ICIORD}}$	O	When synchronous DRAM is used, selects D23–D16. For other memory, D23–D16 write strobe signal. For PCMCIA, strobe signal indicating I/O read.
Data enable 3	DQMUU/$\overline{\text{WE3}}$/ $\overline{\text{ICIOWR}}$	O	When synchronous DRAM is used, selects D31–D24. For other memory, D31–D24 write strobe signal. For PCMCIA, strobe signal indicating I/O write.
Read	$\overline{\text{RD}}$	O	Strobe signal indicating read cycle
Wait	$\overline{\text{WAIT}}$	I	Wait state request signal (synchronous signal)
16-bit I/O	$\overline{\text{IOIS16}}$	I	Signal indicating PCMCIA 16-bit I/O. Valid only in little-endian mode. (Fix low in big-endian mode.)
Clock enable	CKE	O	Connected to clock enable control signal of synchronous DRAM
Bus release request	$\overline{\text{BREQ}}$	I	Bus release request signal
Bus release acknowledgment	$\overline{\text{BACK}}$	O	Bus release acknowledge signal
Area 0 bus width, PCMCIA card select	MD3/$\overline{\text{CE2A}}$*[1], MD4/$\overline{\text{CE2B}}$*[2]	I/O	Signal controlling bus width of physical space area 0. When PCMCIA is used, $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$.
Endian switching/ low address strobe	MD5/$\overline{\text{RAS2}}$*[3]	I/O	Signal setting endian for all spaces on reset. When area 2 DRAM is connected, area 2 DRAM $\overline{\text{RAS}}$ signal

Notes: 1. MD3/$\overline{\text{CE2A}}$ input/output switching is performed by BCR1.A5PCM. Output is selected when BCR1.A5PCM = 1.

2. MD4/$\overline{\text{CE2B}}$ input/output switching is performed by BCR1.A6PCM. Output is selected when BCR1.A6PCM = 1.

3. MD5/$\overline{\text{RAS2}}$ input/output switching is performed by BCR1.DRAMTP. Output is selected when BCR1.DRAMTP (2–0) = 101.

10.1.4 Register Configuration

The BSC has 11 registers (table 10.2). The synchronous DRAM also has a built-in synchronous DRAM mode register. 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*[2]	Address	Bus Width
Bus control register 1	BCR1	R/W	H'0000	H'FFFFFF60	16
Bus control register 2	BCR2	R/W	H'3FFC	H'FFFFFF62	16
Wait state control register 1	WCR1	R/W	H'3FFF	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
SDRAM mode register, area 2	SDMR	W	—	H'FFFFD000– H'FFFFDFFF	8
SDRAM mode register, area 3				H'FFFFE000– H'FFFFEFFF	

Notes: 1. For details see section 10.2.8, Synchronous DRAM Mode Register.

2. Initialized during a power-on reset.

10.1.5 Area Overview

Space Allocation: The SH7708 Series architecture provides for a 32-bit virtual address space. The virtual space is divided into five areas by the value of the upper bits of the address. The physical space is divided into eight areas with a 29-bit address space.

Virtual 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 shown in table 10.4, the SH7708 Series can be connected directly to seven areas of memory/PC card, and it outputs chip select signals ($\overline{\text{CS0}}$–$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$) for each of them. $\overline{\text{CS0}}$ is asserted during area 0 access; $\overline{\text{CS6}}$ is asserted during area 6 access. When DRAM, synchronous DRAM, or pseudo-SRAM is connected to area 2 or 3, signals such as $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, and DQM are also asserted. When PCMCIA interface is selected in area 5 or 6, in addition to $\overline{\text{CS5}}$/$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$/$\overline{\text{CE2B}}$ are asserted for the corresponding bytes accessed.

For virtual address spaces P0 and P3, when the memory management unit (MMU) is on, any physical address can be generated by the MMU for a virtual address. Consequently, figure 10.2 can be applied when the MMU is off, and when the MMU is on and the physical addresses corresponding to virtual addresses are identical except for the top 3 bits. When virtual addresses are translated to arbitrary physical addresses, refer to table 10.3, Physical Address Space Map.

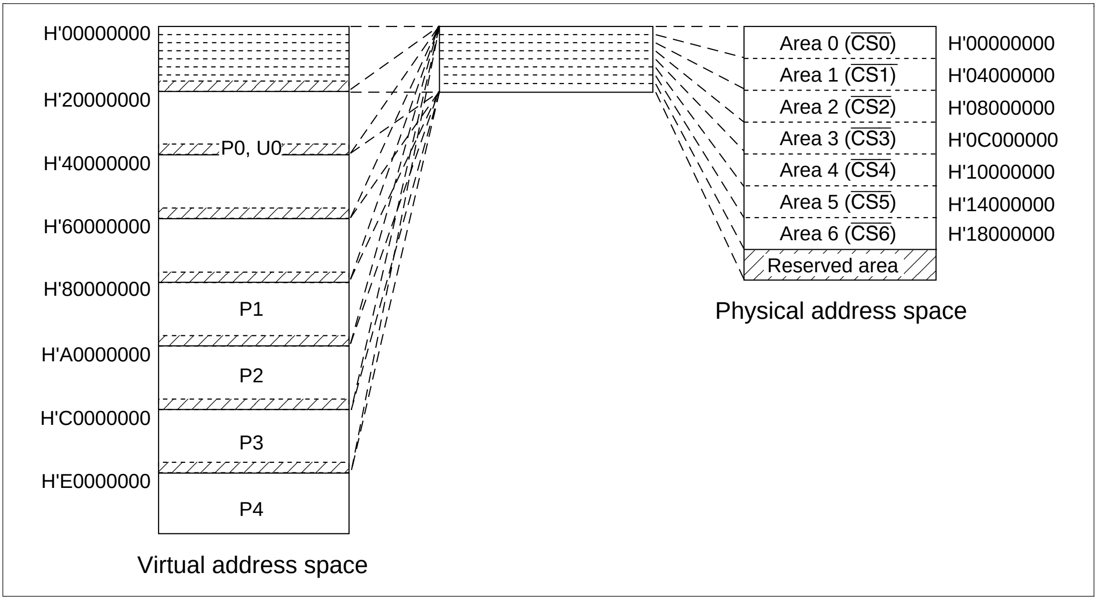

H'00000000
H'20000000
P0, U0
H'40000000
H'60000000
H'80000000
P1
H'A0000000
P2
H'C0000000
P3
H'E0000000
P4
Virtual address space
Area 0 ($\overline{\text{CS0}}$)
Area 1 ($\overline{\text{CS1}}$)
Area 2 ($\overline{\text{CS2}}$)
Area 3 ($\overline{\text{CS3}}$)
Area 4 ($\overline{\text{CS4}}$)
Area 5 ($\overline{\text{CS5}}$)
Area 6 ($\overline{\text{CS6}}$)
Reserved area
H'00000000
H'04000000
H'08000000
H'0C000000
H'10000000
H'14000000
H'18000000
Physical address space

Figure 10.2 Correspondence between Virtual 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	Normal memory*1, burst ROM	64 Mbytes	8, 16, 32*2
	H'00000000 + H'20000000 × n to H'03FFFFFF + H'20000000 × n		Shadow	n: 1–6
1	H'04000000 to H'07FFFFFF	Normal memory	64 Mbytes	8, 16, 32*3
	H'04000000 + H'20000000 × n to H'07FFFFFF + H'20000000 × n		Shadow	n: 1–6
2	H'08000000 to H'0BFFFFFF	Normal memory, synchronous DRAM, DRAM	64 Mbytes	8, 16, 32*3, *4
	H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n		Shadow	n: 1–6
3	H'0C000000 to H'0FFFFFFF	Normal memory, synchronous DRAM, DRAM, pseudo-SRAM	64 Mbytes	8, 16, 32*3, *5
	H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n		Shadow	n: 1–6
4	H'10000000 to H'13FFFFFF	Normal memory	64 Mbytes	8, 16, 32*3
	H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n		Shadow	n: 1–6
5*5	H'14000000 to H'15FFFFFF	Normal memory, PCMCIA, burst ROM	32 Mbytes	8, 16, 32 *3, *6
	H'16000000 to H'17FFFFFF		32 Mbytes	
	H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n		Shadow	n: 1–6
6	H'18000000 to H'19FFFFFF	Normal memory, PCMCIA, burst ROM	32 Mbytes	8, 16, 32 *3, *6
	H'1A000000 to H'1BFFFFFF			
	H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n		Shadow	n: 1–6
7*7	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n	Reserved area		n: 0–7

Notes: 1. Memory with an SRAM, ROM, or similar interface

2. Memory bus width specified by external pin

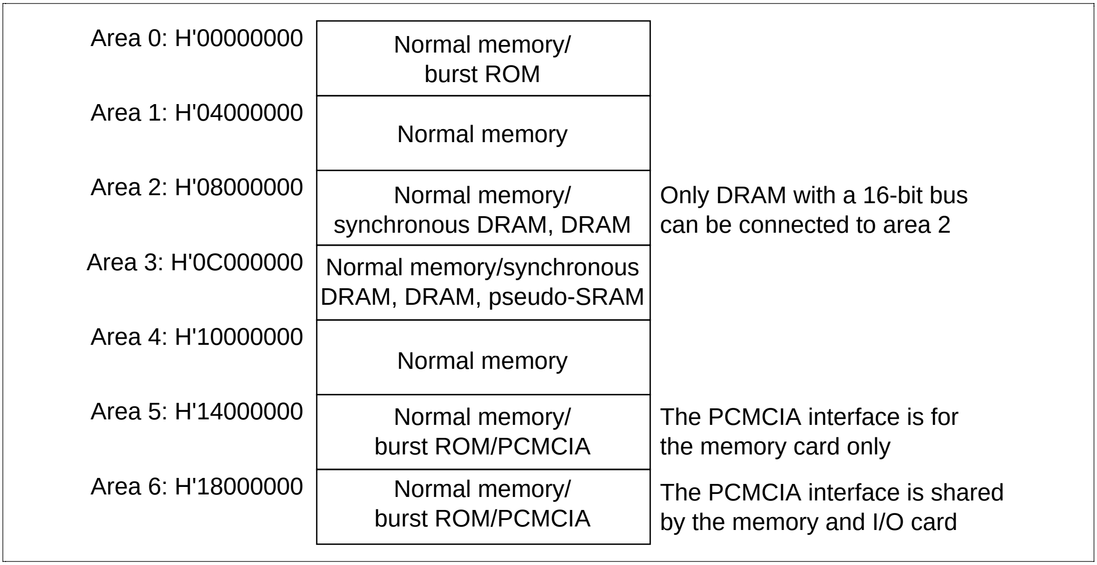
3. Memory bus width specified by register

4. With synchronous DRAM interface, bus width is 32 bits only.
   With DRAM interface, bus width is 16 bits only.

5. With synchronous DRAM interface, bus width is 32 bits only.
   With DRAM and pseudo-SRAM interface, bus width is either 16 or 32 bits only. When areas 2 and 3 are both DRAM interface areas, bus width is 16 bits only.

6. With PCMCIA interface, bus width is either 8 or 16 bits only.

7. Do not access a reserved area, as operation cannot be guaranteed in this case.



Area 0: H'00000000	Normal memory/ burst ROM	
Area 1: H'04000000	Normal memory	
Area 2: H'08000000	Normal memory/ synchronous DRAM, DRAM	Only DRAM with a 16-bit bus can be connected to area 2
Area 3: H'0C000000	Normal memory/synchronous DRAM, DRAM, pseudo-SRAM	
Area 4: H'10000000	Normal memory	
Area 5: H'14000000	Normal memory/ burst ROM/PCMCIA	The PCMCIA interface is for the memory card only
Area 6: H'18000000	Normal memory/ burst ROM/PCMCIA	The PCMCIA interface is shared by the memory and I/O card

Figure 10.3 Physical Space Allocation

Memory Size: The memory size in the SH7708Series 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). The relationship between the external pins (MD4 and MD3) and the bus width after a power-on reset is as follows.

MD4	MD3	Bus Width
0	0	Reserved (do not set)
	1	8 bits
1	0	16 bits
	1	32 bits

For areas 1–6, byte, word, and longword may be chosen for the bus width using bus control register 2 (BCR2) whenever normal memory, ROM, burst ROM, or the PCMCIA interface is used. When the DRAM or pseudo-SRAM interfaces are used, word or longword can be chosen as the bus width using the individual memory control register (MCR). Set the bus width to longword with MCR for synchronous DRAM interfaces.

When area 2 is used as a DRAM area, set the bus widths of areas 2 and 3 to word. When areas 5 and 6 are used as PCMCIA interfaces, set the bus width to byte or word. When using the port function, set each of the bus widths to byte or word for all areas. For more information, see section 10.2.2, Bus Control Register 2 (BCR2), and section 10.2.5, Individual Memory Control Register (MCR).

Shadow Space: Areas 0–6 are decoded by physical address bits A28–A26, which correspond to areas 000 to 110. Address bits 31–29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space obtained by adding to it H'20000000 × n (n = 1–6). The address range for area 7, which is on-chip I/O space, is H'FC000000 to H'FFFFFFFF. The address space H'1C000000 + H'20000000 × n–H'1FFFFFFF + H'20000000 × n (n = 0–6) corresponding to the area 7 shadow space is reserved, so should not be used.

10.1.6 PCMCIA Support

The SH7708 Series supports PCMCIA standard interface specifications in physical space areas 5 and 6.

The interface supported is basically the IC memory card interface and I/O card interface defined by PCMCIA Specifications Version 4.2. In addition, burst access is supported to enable high-speed access.

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	Characteristics
Access	Random access + burst access (ROM page mode correspondence added)
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Memory capacity	Maximum 32 Mbytes
I/O section capacity	Maximum 32 Mbytes
Other	Supports dynamic I/O bus sizing* and access to PCMCIA interface from both the address translation area and non-address translation area

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

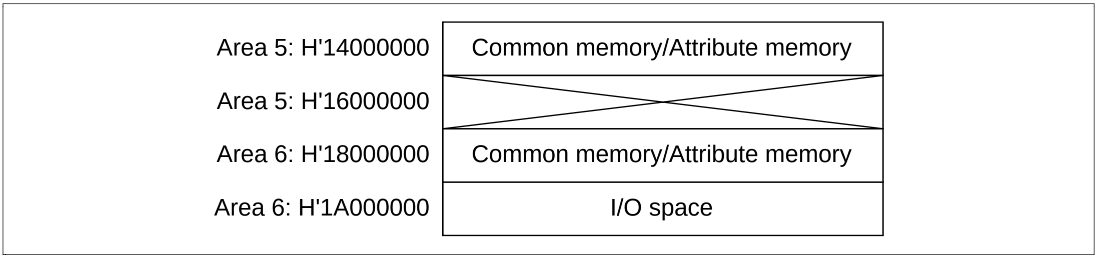


Figure 10.4 PCMCIA Space Allocation

Table 10.5 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			SH7708 Series
Pin	**Signal**	**I/O**	**Function**	**Signal**	**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	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CE1}}$	I	Card enable	$\overline{\text{CS5}}$ or $\overline{\text{CS6}}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{\text{OE}}$	I	Output enable	$\overline{\text{OE}}$	I	Output enable	$\overline{\text{RD}}$
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE}}$/$\overline{\text{PGM}}$	I	Write enable	$\overline{\text{WE1}}$
16	RDY/$\overline{\text{BSY}}$	O	Ready/Busy	$\overline{\text{IREQ}}$	O	Interrupt request	Sensed at port
17	V_{CC}		Operation power	V_{CC}		Operation power	—
18	VPP1		Program power	VPP1		Program/ peripheral power	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0

Table 10.5 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7708 Series
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**Pin**
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP*	O	Write protect	$\overline{\text{IOIS16}}$	O	16 bit I/O port	$\overline{\text{IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	Sensed at port
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2A}}$ or $\overline{\text{CE2B}}$
43	RFSH	I	Refresh request	RFSH	I	Refresh request	Output from port
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{ICIOWR}}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	V_{CC}		Power supply	V_{CC}		Power supply	—
52	VPP2		Program power	VPP2		Program/ peripheral power	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	Output from port
57	RFU		Reserved	RFU		Reserved	—
58	RESET	I	Reset	RESET	I	Reset	Output from port

Table 10.5 PCMCIA Support Interface (cont)

	IC Memory Card Interface			I/O Card Interface			SH7708 Series
Pin	**Signal**	**I/O**	**Function**	**Signal**	**I/O**	**Function**	**Pin**
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$
60	RFU		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	Output from port
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital voice signal	Sensed at port
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed at port
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	Sensed at port
68	GND		Ground	GND		Ground	—

Note: * The SH7708 Series does not provide WP support.

10.2 BSC Registers

10.2.1 Bus Control Register 1 (BCR1)

The bus control register 1 (BCR1) is a 16-bit read/write register that sets the functions and bus cycle status for each area. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or 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:	—	—	HIZMEM*[2]	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
Initial value:	0	0	0	0	0/1*[1]	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	A5BST0	A6BST1	A6BST0	DRAM TP2	DRAM TP1	DRAM TP0	A5PCM	A6PCM
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

Notes: 1. Samples the value of the external pin designating endian upon a power-on reset.
2. Reserved bit in the SH7708.

Bits 15 and 14—Reserved: These bits always read 0. The write value should always be 0.

Bits 13 (SH7708)—Reserved: These bits always read 0. The write value should always be 0. This bit is not supported in emulator.

Bit 13 (SH7708S, SH7708R) —High-Z Memory Control (HIZMEM): Specifies the state of A25 to A0, BS, CS, RD/WR, WE/DQM, RD, MD3/CE2A, and MD4/CE2B in standby mode.

Bit 13: HIZMEM	Description	
0	High-impedance (high-Z) in standby mode	(Initial value)
1	Drive state in standby mode	

Bit 12—High-Z Control (HIZCNT): Specifies the state of the RAS and CAS signals in the standby and bus-released states.

Bit 12: HIZCNT	Description
0	RAS and CAS signals become high-impedance (High-Z) in standby mode and in bus-released state. (Initial value)
1	RAS and CAS signals drive in standby mode and in bus-released state.

Bit 11—Endian Flag (ENDIAN): Samples the value of the external pin designating endian upon a power-on reset. Endian for all physical spaces is decided by this bit, which is read-only.

Bit 11: ENDIAN	Description
0	(On reset) Endian setting external pin (MD5) is low. Indicates the SH7708 Series is set as big-endian.
1	(On reset) Endian setting external pin (MD5) is high. Indicates the SH7708 Series 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	Access area 0 as normal memory. (Initial value)
	1	Access area 0 as burst ROM (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Access area 0 as burst ROM (8 consecutive accesses). Can be used only when bus width is 8 or 16.
	1	Access area 0 as burst ROM (16 consecutive accesses). Can be used only when bus width is 8.

Bits 8 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	Access area 5 as normal memory. (Initial value)
	1	Burst access of area 5 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 5 (8 consecutive accesses). Can be used only when bus width is 8 or 16.
	1	Burst access of area 5 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 6 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	Access area 6 as normal memory. (Initial value)
	1	Burst access of area 6 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 6 (8 consecutive accesses). Can be used only when bus width is 8 or 16.
	1	Burst access of area 6 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 4 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. Pseudo-SRAM, DRAM, and synchronous DRAM can also be directly connected.

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Areas 2 and 3 are normal memory (Initial value)
		1	Area 2: normal memory; area 3: PSRAM
	1	0	Area 2: normal memory; area 3: SDRAM
		1	Areas 2 and 3 are SDRAM
1	0	0	Area 2: normal memory; area 3: DRAM
		1	Areas 2 and 3 are DRAM *
	1	0	Reserved (cannot be set)
		1	Reserved (cannot be set)

Note: * When selecting these bits, set the area 2 and 3 bus widths as word. The MD5 pin output is the $\overline{\text{RAS2}}$ signal.

Bit 1—Area 5 Bus Type (A5PCM): Designates whether to access physical space area 5 as PCMCIA space.

Bit 1: A5PCM	Description
0	Access physical space area 5 as normal memory. (Initial value)
1	Access physical space area 5 as PCMCIA space.*

Note: * MD3 pin output is $\overline{\text{CE2A}}$.

Bit 0—Area 6 Bus Type (A6PCM): Designates whether to access physical space area 6 as PCMCIA space.

Bit 0: A6PCM	Description	
0	Access physical space area 6 as normal memory.	(Initial value)
1	Access physical space area 6 as PCMCIA space.*	

Note: MD4 pin output is $\overline{\text{CE2B}}$.

10.2.2 Bus Control Register 2 (BCR2)

Bus control register 2 (BCR2) is a 16-bit read/write register that selects the bus size of each area, and whether to use the 8-bit port. It is initialized to H'3FFC 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	A1SZ1	A1SZ0	—	PORTEN
Initial value:	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W

Bits 15, 14, and 1—Reserved: These bits always read 0. The write value should always be 0.

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

Bit 2n + 1: AnSZ1	Bit 2n: AnSZ0	Bit 0: PORTEN	Description
0	0	0	Reserved (not settable)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Longword (32-bit) size (Initial value)
0	0	1	Reserved (not settable)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Reserved (not settable)

Bit 0—Port Function Enable (PORTEN): Designates whether to use the D23–D16 pins as an 8-bit port. When using this function set the bus widths to word or byte in all areas.

Bit 0: PORTEN	Description
0	D23–D16 are not used as a port. (Initial value)
1	D23–D16 are used as a port.

10.2.3 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register 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 SH7708 Series automatically inserts idle states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FFF 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:	WAITSEL*	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0
Initial value:	0	0	1	1	1	1	1	1
R/W:	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:	A3IW1	A3IW0	A2IW1	A2IW0	A1IW1	A1IW0	A0IW1	A0IW0
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

Note: * SH7708R A-mask version only

Bits 15 and 14 —Reserved: These bits always read 0. The write value should always be 0.

Bit 15—WAIT Signal Sampling Timing Specification (WAITSEL) [A-mask version only]

Bit 15: WAITSEL	Description
0	WAIT signal is sampled at rise of CKIO In this case, the WAIT signal should be input synchronously (Initial value)
1	WAIT signal is sampled at fall of CKIO In this case, the WAIT signal can be input asynchronously

Bits 2n + 1, 2n—Area n (6–0) Intercycle Idle Specification (AnIW1, AnIW0): These bits specify the number of idles inserted between bus cycles when switching between physical space area n (6–0) 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	No idle cycles
	1	1 idle cycle inserted
1	0	2 idle cycles inserted
	1	3 idle cycles inserted (Initial value)

10.2.4 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit read/write register 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, but retains its contents.

Bit:	15	14	13	12	11	10	9	8
Bit name:	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1
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:	A4W0	A3W1	A3W0	A1-2W1	A1-2W0	A0W2	A0W1	A0W0
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 in physical space area 6. They also specify the burst pitch for burst transfer.

			Description			
			First Cycle		**Burst Cycle (Excluding First Cycle)**	
Bit 15: A6W2	**Bit 14: A6W1**	**Bit 13: A6W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	0	Ignored	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 (Initial value)	Enabled	10	Enabled

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

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 12: A5W2	Bit 11: A5W1	Bit 10: A5W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Ignored	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 (Initial value)	Enabled	10	Enabled

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

			Description	
Bit 9: A4W2	Bit 8: A4W1	Bit 7: A4W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Ignored
		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 in physical space area 3. External wait input is enabled only when normal memory is used, and is ignored when DRAM, synchronous DRAM, or pseudo-SRAM is used.

- For Normal Memory

		Description	
Bit 6: A3W0	**Bit 5: A3W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enabled
1	0	2	Enabled
	1	3	Enabled (Initial value)

- For DRAM, SDRAM, Pseudo-SRAM

		Description		
Bit 6: A3W1	**Bit 5: A3N0**	**DRAM: $\overline{\text{CAS}}$ Assert Period**	**SDRAM: CAS Latency**	**PSRAM: $\overline{\text{OE}}$, $\overline{\text{WE}}$ Assert Period**
0	0	1	1	1
	1	1	1	1
1	0	2	2	2
	1	3	3	3 (Initial value)

Bits 4 and 3—Areas 1 and 2 Wait Control (A1–2W1, A1–2W0): These bits specify the number of wait states inserted in physical space areas 1 and 2. External wait input is enabled only when normal memory is used, and is ignored when DRAM or synchronous DRAM is used.

- For Normal Memory

		Description	
Bit 4: A1-2W0	**Bit 3: A1-2W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enabled
1	0	2	Enabled
	1	3	Enabled (Initial value)

- For DRAM, Synchronous DRAM

		Description	
Bit 4: A1-2W0	**Bit 3: A1-2W0**	**DRAM: $\overline{\text{CAS}}$ Assert Period**	**SDRAM: CAS Latency**
0	0	1	1
	1	1	1
1	0	2	2
	1	3	3 (Initial value)

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

			Description			
			First Cycle		**Burst Cycle (Excluding First Cycle)**	
Bit 2: A0W2	**Bit 1: A0W1**	**Bit 0: A0W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	0	Ignored	2	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 (Initial value)	Enabled	10	Enabled

10.2.5 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit read/write register that specifies $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ timing and burst control for DRAM (area 3 only), synchronous DRAM (areas 2 and 3), and pseudo-SRAM, specifies address multiplexing, and controls refresh. This enables direct connection of DRAM, synchronous DRAM and pseudo-SRAM 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, TRWL1–TRWL0, TRAS1–TRAS0, BE, SZ, AMX1–AMX0, and EDOMODE are written to in the initialization after a power-on reset and are not then modified again. When RFSH and RMODE are written to, write the same values to the other bits. When using DRAM, pseudo-SRAM, and synchronous 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	TRWL1	TRWL0	TRAS1	TRAS0
	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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 $\overline{\text{RAS}}$ assertion after $\overline{\text{RAS}}$ negation. When synchronous DRAM interface is selected, they set the minimum number of cycles until output of the next bank-active command after precharge. When pseudo-SRAM interface is selected, they set the minimum number of cycles until the next $\overline{\text{CE}}$ assertion after $\overline{\text{CE}}$ negation.

		Description	
Bit 15: TPC1	**Bit 14: TPC0**	**Normally**	**Immediately after Self-Refresh**
0	0	1 cycle (Initial value)	2 cycles (Initial value)
	1	2 cycles	5 cycles
1	0	3 cycles	8 cycles
	1	4 cycles	11 cycles

Bits 11 and 10—Write-Precharge Delay (TRWL1, TRWL0): These bits set the synchronous DRAM write-precharge delay time. This designates the time between the end of a write cycle and the next bank-active command. This is valid only when synchronous DRAM is connected. After the write cycle, the next bank-active command is not issued for the period TPC + TRWL.

Bit 11: TRWL1	Bit 10: TRWL0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	Reserved (cannot be set)	

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 $\overline{\text{RAS}}$ assertion period for $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refreshes. When pseudo-SRAM interface is selected, they set the $\overline{\text{OE}}/\overline{\text{RFSH}}$ assertion period for auto-refreshes. When synchronous DRAM interface is selected, no bank-active command is issued during the period TPC + TRAS after an auto-refresh command.

In the SH7708, set the same values in the TRAS bits in MCR and DCR.

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: This bit always reads 0. The write value should always be 0.

Bit 6—Burst Enable (BE): Specifies whether to conduct a burst access of DRAM or pseudo-SRAM. When accessing synchronous DRAM, burst access is always carried out, regardless of this bit’s designation.

Bit 6: BE	Description	
0	Burst disabled	(Initial value)
1	With DRAM interface, high-speed page mode access With pseudo-SRAM interface, continuous data transfer in static column mode	

Bit 5—Memory Data Size (SZ): Specifies the memory data bus size for DRAM, synchronous DRAM, and pseudo-SRAM. Always set this bit to 1 when synchronous DRAM is used. Takes precedence over the BCR2 register designation.

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

Bits 4 and 3—Address Multiplex (AMX1, AMX0): These bits specify address multiplexing for DRAM and synchronous DRAM. The actual address shift value differs between DRAM interface and synchronous DRAM interface.

For DRAM Interface:

Bit 4: AMX1	Bit 3: AMX0	Description	
0	0	8-bit column address product	(Initial value)
	1	9-bit column address product	
1	0	10-bit column address product	
	1	11-bit column address product	

For Synchronous DRAM Interface:

Bit 4: AMX1	Bit 3: AMX0	Description	
0	0	16-Mbit product (1M × 16 bits)	(Initial value)
	1	16-Mbit product (2M × 8 bits)	
1	0	16-Mbit product (4M × 4 bits)	
	1	4-Mbit product (256k × 16 bits)	

Bit 2—Refresh Control (RFSH): Determines whether or not refreshing of DRAM, synchronous DRAM, and pseudo-SRAM is performed. The timer for generation of the refresh request frequency can also be used as an interval 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 or an auto-refresh is performed on DRAM, synchronous DRAM or pseudo-SRAM 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, synchronous DRAM or pseudo-SRAM will wait for the end of any executing external bus cycle before going into a self-refresh. All refresh requests to memory that is in the self-refresh state are ignored.

Bit 1: RMODE	Description	
0	CAS-before-RAS refresh (RFSH must be 1)	(Initial value)
1	Self-refresh (RFSH must be 1)	

Bit 0— Extended Data Out (EDOMODE): Specifies the timing of data sampling 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 valid only for DRAM connected to area 3. Do not set this bit to 1 when using synchronous DRAM or pseudo-SRAM.

Bit 0: EDOMODE	Description	
0	Set when using normal DRAM. Data is sampled during read cycle on the falling edge of CKIO.	(Initial value)
1	Set when using EDO mode DRAM. Data is sampled during read cycle on the rising edge of CKIO. Also, RAS signal negation is delayed 1/2 a CKIO machine cycle.	

10.2.6 DRAM Control Register (DCR)

The DRAM area control register (DCR) is a 16-bit read/write register that specifies $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ 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 resets or in standby mode. Do not access external memory outside area 2 until initialization of this register is complete.

Bit:	15	14	13	12	11	10	9	8
Bit name:	TPC1	TPC0	RCD1	RCD0	—	—	TRAS1	TRAS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	BE	—	AMX1	AMX0	RFSH	RMODE	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R

Bits 15 and 14—RAS Precharge Time (TPC1, TPC0): These bits set the RAS precharge time for the DRAM connected to area 2.

		Description	
Bit 15: TPC1	**Bit 14: TPC0**	**Normally**	**Immediately after Self-Refresh**
0	0	1 cycle (Initial value)	2 cycles (Initial value)
	1	2 cycles	5 cycles
1	0	3 cycles	8 cycles
	1	4 cycles	11 cycles

Bits 13 and 12—RAS–CAS Delay (RCD1, RCD0): These bits set the RAS–CAS delay time for the DRAM connected to area 2.

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): These bits set the RAS assert period for CAS-before-RAS refreshing of the DRAM connected to area 2.

In the SH7708, set the same values in the TRAS bits in MCR and DCR.

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 a burst access of the DRAM connected to area 2.

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

Bits 4 and 3—Address Multiplex (AMX1, AMX0): These bits specify address multiplexing for the DRAM connected to area 2.

Bit 4: AMX1	Bit 3: AMX0	Description	
0	0	8-bit column address product	(Initial value)
	1	9-bit column address product	
1	0	10-bit column address product	
	1	11-bit column address product	

Bit 2—Refresh Control (RFSH): Determines whether or not refreshing of the DRAM connected to area 2 is performed.

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

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

Bit 1: RMODE	Description	
0	CAS-before-RAS refresh (RFSH must be 1)	(Initial value)
1	Self-refresh (RFSH must be 1)	

10.2.7 PCMCIA Control Register (PCR)

The PCMCIA control register (PCR) is a 16-bit read/write register that specifies the $\overline{\text{OE}}$ and $\overline{\text{WE}}$ signal assert/negate timing for PCMCIA interfaces connected to areas 5 and 6. The $\overline{\text{OE}}$ and $\overline{\text{WE}}$ 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, and 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	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
Bit name:	A5TED1	A5TED0	A6TED1	A6TED0	A5TEH1	A5TEH0	A6TEH1	A6TEH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits always read 0. The write value should always be 0.

Bits 7 and 6—Area 5 Address $\overline{\text{OE}}/\overline{\text{WE}}$ Assert Delay (A5TED1, A5TED0): These bits specify the address to $\overline{\text{OE}}/\overline{\text{WE}}$ 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 $\overline{OE}/\overline{WE}$ Assert Delay (A6TED1, A6TED0): These bits specify the address to $\overline{OE}/\overline{WE}$ 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 $\overline{OE}/\overline{WE}$ Negate Address Delay (A5TEH1, A5TEH0): These bits specify the $\overline{OE}/\overline{WE}$ 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 $\overline{OE}/\overline{WE}$ Negate Address Delay (A6TEH1, A6TEH0): These bits specify the $\overline{OE}/\overline{WE}$ 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	

10.2.8 Synchronous DRAM Mode Register (SDMR)

The synchronous DRAM mode register (SDMR) is written to via the synchronous DRAM address bus and is a virtual 8-bit write-only register. It sets synchronous DRAM mode for areas 2 and 3. SDMR settings must be made before synchronous DRAM is accessed.

Writes to the synchronous DRAM mode register use the address bus rather than the data bus. If the value to be set is X and the SDMR address is Y, the value X is written in the synchronous DRAM mode register by writing in address X + Y. Since A0 of the synchronous DRAM is connected to A2 of the chip and A1 of the synchronous DRAM is connected to A3 of the chip, the value actually written to the synchronous DRAM is the X value shifted two bits right. For example, when H'0230 is written to the SDMR register of area 2, random data is written to the address

H'FFFD000 (address Y) + H'08C0 (value X), or H'FFFFD8C0. As a result, H'0230 is written to the SDMR register. When H'0230 is written to the SDMR register of area 3, random data is written to the address H'FFFE000 (address Y) + H'08C0 (value X) or H'FFFFE8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H'000 to H'0FFC.

Address bits

Bit:	31		12	11	10	9	8
Bit name:	SDMR address						
Initial value:	—	. . . .	—	—	—	—	—
R/W:	—	. . . .	—	W*	W*	W	W

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

Note: Depending on the type of synchronous DRAM.

10.2.9 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTSCR) is a 16-bit read/write register that specifies the refresh cycle, whether to generate an interrupt, and the cycle of that interrupt. RTSCR is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or in standby mode.

Note: The method for writing to RTCOR is different from that for general registers to prevent inadvertent overwriting. Using a word transfer instruction, place B'10100101 in the upper byte and the write data in the lower byte. For details, see section 10.2.13, Cautions on Accessing Refresh Control Related Registers.

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits always read 0.

Bit 7—Compare Match Flag (CMF): Status flag that 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*

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

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt request caused when the CMF bit in RTCSR is set to 1. Do not set this bit to 1 when using CAS-before-RAS refreshing or auto-refreshing.

Bit 6: CMIE	Description
0	Disables the interrupt request caused by CMF (Initial value)
1	Enables the interrupt request caused by CMF

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 scaled by the specified ratio.

Bit 5: CKS2	Bit 4: CKS1	Bit 3: CKS0	Description
0	0	0	Disables clock input (Initial value)
		1	Bus clock (CKIO)/4
	1	0	CKIO/16
		1	CKIO/64
1	0	0	CKIO/256
		1	CKIO/1024
	1	0	CKIO/2048
		1	CKIO/4096

Bit 2—Refresh Count Overflow Flag (OVF): Status flag that indicates when the number of refresh requests indicated in the refresh count register (RFCR) exceeds the limit set in the LMTS bit in RTCSR.

Bit 2: OVF	Description	
0	RFCR has not exceeded the count limit value set in LMTS Clearing Condition: When 0 is written to OVF	(Initial value)
1	RFCR has exceeded the count limit value set in LMTS Setting Condition: 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): Selects whether to suppress generation of interrupt requests by OVF when the OVF bit of RTCSR is set to 1.

Bit 1: OVIE	Description	
0	Disables interrupt requests caused by OVF	(Initial value)
1	Enables interrupt requests caused by OVF	

Bit 0—Refresh Count Overflow Limit Select (LMTS): Indicates the count limit value to be compared to the number of refreshes indicated in the refresh count register (RFCR). When the value RFCR 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 read/write register containing an 8-bit counter that counts up on an input clock. The clock select bits (CKS2–CKS0) in RTCSR select the input clock. When RTCNT matches RTCOR, the OVF bit in RTCSR is set and RTCNT is cleared. RTCNT is initialized to H'00 by a power-on reset; it continues incrementing after a manual reset; it is not initialized in standby mode, but retains its contents.

Note: The method for writing to RTCOR is different from that for general registers to prevent inadvertent overwriting. Using a word transfer instruction, place B'10100101 in the upper byte and the write data in the lower byte. For details, see section 10.2.13, Cautions on Accessing Refresh Control Related Registers.

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:								
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 read/write 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.

Note: The method for writing to RTCOR is different from that for general registers to prevent inadvertent overwriting. Using a word transfer instruction, place B'10100101 in the upper byte and the write data in the lower byte. For details, see section 10.2.13, Cautions on Accessing Refresh Control Related Registers.

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:								
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 read/write register containing a 10-bit counter that increments every time RTCOR and RTCNT match. When RFCR exceeds the count limit value set by 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.

Note: The method for writing to RFCR is different from that for general registers to prevent inadvertent overwriting. Using a word transfer instruction, place B'101001 in the top 6 bits of the upper byte, and the write data in the remaining bits. For details, see section 10.2.13, Cautions on Accessing Refresh Control Related Registers.

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/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 Cautions on Accessing Refresh Control Related Registers

RFCR, RTCSR, RTCNT, and RTCOR require that a specific code be appended to the data when it is written to prevent data from being mistakenly overwritten by program overruns or other write operations (figure 10.5). Perform reads and writes using the following methods:

1. When writing to RFCR, RTCSR, RTCNT, or 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, or RTCOR, use a 16-bit access. 0 is read from undefined bits.

	15							8	7 0
RTCSR, RTCNT, RTCOR	1	0	1	0	0	1	0	1	Write data

	15					10	9 0
RFCR	1	0	1	0	0	1	Write data

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

10.3 BSC Operation

10.3.1 Endian/Access Size and Data Alignment

The SH7708 Seires supports both big-endian mode, in which the 0 address is the most significant byte in the byte data, and little-endian mode, in which the 0 address is the least significant byte. Switching between the two is designated by an external pin (MD5 pin) at the time of a power-on reset. After a power-on reset, big-endian mode is set when MD5 is low, and little-endian mode is set when MD5 is high.

Three data bus widths are available for normal memory (byte, word, longword) and two data bus widths (word and longword) for DRAM and pseudo-SRAM. Only longword is available for synchronous DRAM. For the PCMCIA interface, choose from byte and word. This means data alignment is done by matching the device's data width and endian. The access unit must also be matched to the device's bus width. This also means that when longword data is read from a byte-width device, four read operations must be executed. In the SH7708 Series, 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

	Data Bus				Strobe Signal			
Operation	**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$, DQMUU**	**$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$, DQMUL**	**$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$, DQMLU**	**$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$, DQMLL**
Address 0 byte access	Data 7–0	—	—	—	Asserted	—	—	—
Address 1 byte access	—	Data 7–0	—	—	—	Asserted	—	—
Address 2 byte access	—	—	Data 7–0	—	—	—	Asserted	—
Address 3 byte access	—	—	—	Data 7–0	—	—	—	Asserted
Address 0 word access	Data 15–8	Data 7–0	—	—	Asserted	Asserted	—	—
Address 2 word access	—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 0 longword access	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted	Asserted

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

		Data Bus				Strobe Signal			
Operation		D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, CASHH, DQMUU	$\overline{\text{WE2}}$, CASHL, DQMUL	$\overline{\text{WE1}}$, CASLH, DQMLU	$\overline{\text{WE0}}$, CASLL, DQMLL
Address 0 byte access		—	—	Data 7–0	—	—	—	Asserted	—
Address 1 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 2 byte access		—	—	Data 7–0	—	—	—	Asserted	—
Address 3 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 0 word access		—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 2 word access		—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 0 longword access	1st time (address 0)	—	—	Data 31–24	Data 23–16	—	—	Asserted	Asserted
	2nd time (address 2)	—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted

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

		Data Bus				Strobe Signal			
Operation		**D31– D24**	**D23– D16**	**D15– D8**	**D7–D0**	**$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$, DQMUU**	**$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$, DQMUL**	**$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$, DQMLU**	**$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$, DQMLL**
Address 0 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 1 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 2 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 3 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 0 word access	1st time (address 0)	—	—	—	Data 15–8	—	—	—	Asserted
	2nd time (address 1)	—	—	—	Data 7–0	—	—	—	Asserted
Address 2 word access	1st time (address 2)	—	—	—	Data 15–8	—	—	—	Asserted
	2nd time (address 3)	—	—	—	Data 7–0	—	—	—	Asserted
Address 0 longword access	1st time (address 0)	—	—	—	Data 31–24	—	—	—	Asserted
	2nd time (address 1)	—	—	—	Data 23–16	—	—	—	Asserted
	3rd time (address 2)	—	—	—	Data 15–8	—	—	—	Asserted
	4th time (address 3)	—	—	—	Data 7–0	—	—	—	Asserted

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

	Data Bus				Strobe Signal			
Operation	**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**$\overline{\text{WE3}}$, CASHH, DQMUU**	**$\overline{\text{WE2}}$, CASHL, DQMUL**	**$\overline{\text{WE1}}$, CASLH, DQMLU**	**$\overline{\text{WE0}}$, CASLL, DQMLL**
Address 0 byte access	—	—	—	Data 7–0	—	—	—	Asserted
Address 1 byte access	—	—	Data 7–0	—	—	—	Asserted	—
Address 2 byte access	—	Data 7–0	—	—	—	Asserted	—	—
Address 3 byte access	Data 7–0	—	—	—	Asserted	—	—	—
Address 0 word access	—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 2 word access	Data 15–8	Data 7–0	—	—	Asserted	Asserted	—	—
Address 0 longword access	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted	Asserted

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

		Data Bus				Strobe Signal			
Operation		D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$, DQMUU	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$, DQMUL	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$, DQMLU	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$, DQMLL
Address 0 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 1 byte access		—	—	Data 7–0	—	—	—	Asserted	—
Address 2 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 3 byte access		—	—	Data 7–0	—	—	—	Asserted	—
Address 0 word access		—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 2 word access		—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
Address 0 longword access	1st time (address 0)	—	—	Data 15–8	Data 7–0	—	—	Asserted	Asserted
	2nd time (address 2)	—	—	Data 31–24	Data 23–16	—	—	Asserted	Asserted

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

		Data Bus				Strobe Signal			
Operation		D31–D24	D23–D16	D15–D8	D7–D0	$\overline{\text{WE3}}$, $\overline{\text{CASHH}}$, DQMUU	$\overline{\text{WE2}}$, $\overline{\text{CASHL}}$, DQMUL	$\overline{\text{WE1}}$, $\overline{\text{CASLH}}$, DQMLU	$\overline{\text{WE0}}$, $\overline{\text{CASLL}}$, DQMLL
Address 0 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 1 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 2 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 3 byte access		—	—	—	Data 7–0	—	—	—	Asserted
Address 0 word access	1st time (address 0)	—	—	—	Data 7–0	—	—	—	Asserted
	2nd time (address 1)	—	—	—	Data 15–8	—	—	—	Asserted
Address 2 word access	1st time (address 2)	—	—	—	Data 7–0	—	—	—	Asserted
	2nd time (address 3)	—	—	—	Data 15–8	—	—	—	Asserted
Address 0 longword access	1st time (address 0)	—	—	—	Data 7–0	—	—	—	Asserted
	2nd time (address 1)	—	—	—	Data 15–8	—	—	—	Asserted
	3rd time (address 2)	—	—	—	Data 23–16	—	—	—	Asserted
	4th time (address 3)	—	—	—	Data 31–24	—	—	—	Asserted

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).

Normal 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 and MD4 on a power-on reset. For details, see Memory Size in section 10.1.5. When the Area 0 space is accessed, the $\overline{\text{CS0}}$ signal is asserted. 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 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 ($\overline{\text{WAIT}}$). When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits.

Area 1: Area 1 physical 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).

Only normal memories like SRAM and ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using the A1SZ1–A1SZ0 bits in BCR2. When the Area 1 space is accessed, the $\overline{\text{CS1}}$ signal is asserted. 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 A12W1–A12W0 bits in WCR2. Also, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

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).

Normal memories like SRAM and ROM, as well as DRAM and synchronous 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 normal memory. For synchronous DRAM, set longword using the SZ bit in MCR. 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.

When the area 2 space is accessed, the $\overline{\text{CS2}}$ signal is asserted. When normal memories are 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 and the number of bus cycles is selected between 0 and 3 wait cycles using the A12W1 to A12W0 bits in WCR2. When normal memory is connected, only, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

When synchronous DRAM is connected, the $\overline{\text{RAS}}$ signal, $\overline{\text{CAS}}$ signal, RD/$\overline{\text{WR}}$ signal, and byte control signals DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed. Control of $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, data timing, and address multiplexing is set with MCR.

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).

Normal memories like SRAM and ROM, as well as DRAM, pseudo-SRAM, and synchronous 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 normal memory. For DRAM and pseudo-SRAM, word or longword can be selected using the SZ bit in MCR. When synchronous DRAM is connected, set to longword using the SZ bit in MCR.

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

When normal memories are 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 and the number of bus cycles is selected between 0 and 3 wait cycles using the A3W1–A3W0 bits in WCR2. When normal memory is connected, only, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

When synchronous DRAM is connected, the $\overline{\text{RAS}}$ signal, $\overline{\text{CAS}}$ signal, RD/$\overline{\text{WR}}$ signal, and byte control signals DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed. When DRAM is connected, the $\overline{\text{RAS}}$ signal, $\overline{\text{CASHH}}$ signal, $\overline{\text{CASHL}}$ signal, $\overline{\text{CASLH}}$ signal, $\overline{\text{CASLL}}$ signal, and RD/$\overline{\text{WR}}$ signal are all asserted and addresses multiplexed. When pseudo-SRAM is connected, the $\overline{\text{CE}}$ signal, $\overline{\text{OE}}$/$\overline{\text{RFSH}}$ signal, and $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, and $\overline{\text{WE3}}$ signals are asserted. For all of these, control of $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and data timing and of 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 normal 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 $\overline{\text{CS4}}$ signal is asserted. 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 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 ($\overline{\text{WAIT}}$).

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).

Normal memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. PCMCIA interfaces only use their IC memory card interface, so the address range becomes the 32 Mbytes at H'14000000 + H'20000000 × n – H'15FFFFFF + H'20000000 × n (n = 0–6, n = 1–6 is the shadow space).

For normal 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, and word can be selected as the bus width using the A5SZ1–A5SZ0 bits in BCR2.

When the area 5 space is accessed and normal memory is connected, the $\overline{\text{CS5}}$ signal is asserted. 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. When the PCMCIA interface is used, the $\overline{\text{CE1}}$ signal, $\overline{\text{CE2}}$ signal, $\overline{\text{OE}}$ signal, and $\overline{\text{WE}}$ 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 ($\overline{\text{WAIT}}$). When a burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits. The setup and hold times of address/$\overline{\text{CE1A}}$/$\overline{\text{CE2A}}$ for the read/write strobe signals can be set within a range of 0.5–3.5 cycles using the A5TED1–A5TED0 and A5TEH1–A5TEH0 bits 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).

Normal 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 normal 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, and word can be selected as the bus width using the A6SZ1–A6SZ0 bits in BCR2.

When the area 6 space is accessed and normal memory is connected, the $\overline{\text{CS6}}$ signal is asserted. 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. When the PCMCIA interface is used, the $\overline{\text{CE1B}}$ signal, $\overline{\text{CE2B}}$ signal, $\overline{\text{OE}}$ signal, and $\overline{\text{WE1}}$, $\overline{\text{ICORD}}$, and $\overline{\text{ICOWR}}$ 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 ($\overline{\mathrm{WAIT}}$). When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2–10 according to the number of waits. The setup and hold times of address/$\overline{\mathrm{CE1B}}$/$\overline{\mathrm{CE2B}}$ for the read/write strobe signals can be set within a range of 0.5–3.5 cycles using A6TED1–A6TED0 and A6TEH1–A6TEH0.

If PCMCIA is used in area 6, synchronous DRAM is used at the same time, and a synchronous DRAM auto-refresh (CAS-before-RAS refresh) request is issued simultaneously, area 6 card enable signals $\overline{\mathrm{CS6}}$ and $\overline{\mathrm{CE2B}}$ may be asserted earlier than usual, at the same time as the immediately preceding auto-refresh cycle. When both PCMCIA and synchronous DRAM are used, they should be used in area 5. When area 5 is used, the system design should provide for CS to be asserted early without causing any problems.

10.3.3 Basic Interface

Basic Timing: The basic interface of the SH7708 Series uses strobe signal output because 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 $\overline{\mathrm{BS}}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{\mathrm{CSn}}$ signal is negated on the T2 clock falling edge to secure the negation period. Therefore, 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 a 32-bit device, and 16 bits in a 16-bit device. When writing, only the $\overline{\mathrm{WE}}$ 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 continuously. The bus is not released during this transfer. For cache misses that occur during byte or word operand accesses or branching to odd word boundaries, the fill is always performed by longword accesses on the chip-external interface. Write-through area write access and noncacheable read/write access is based on the actual address size.

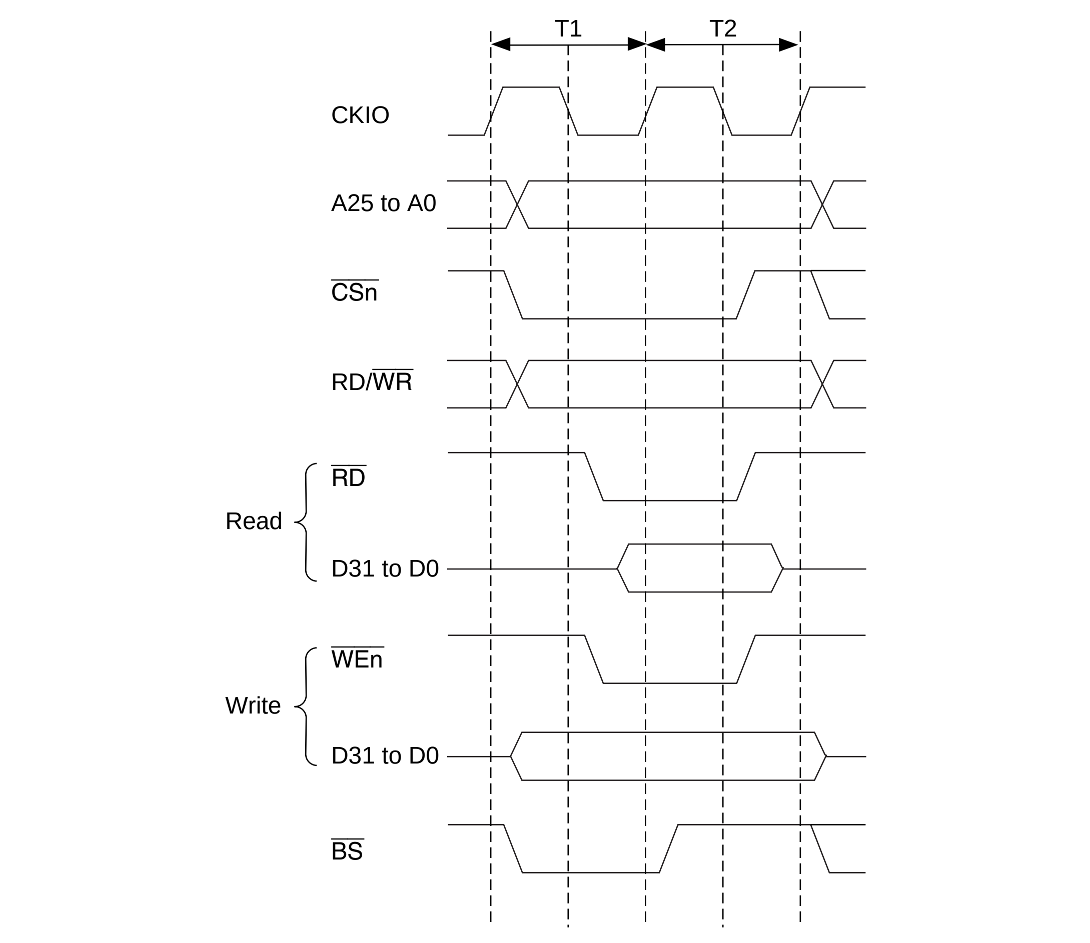


Figure 10.6 Basic Timing of Basic Interface

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

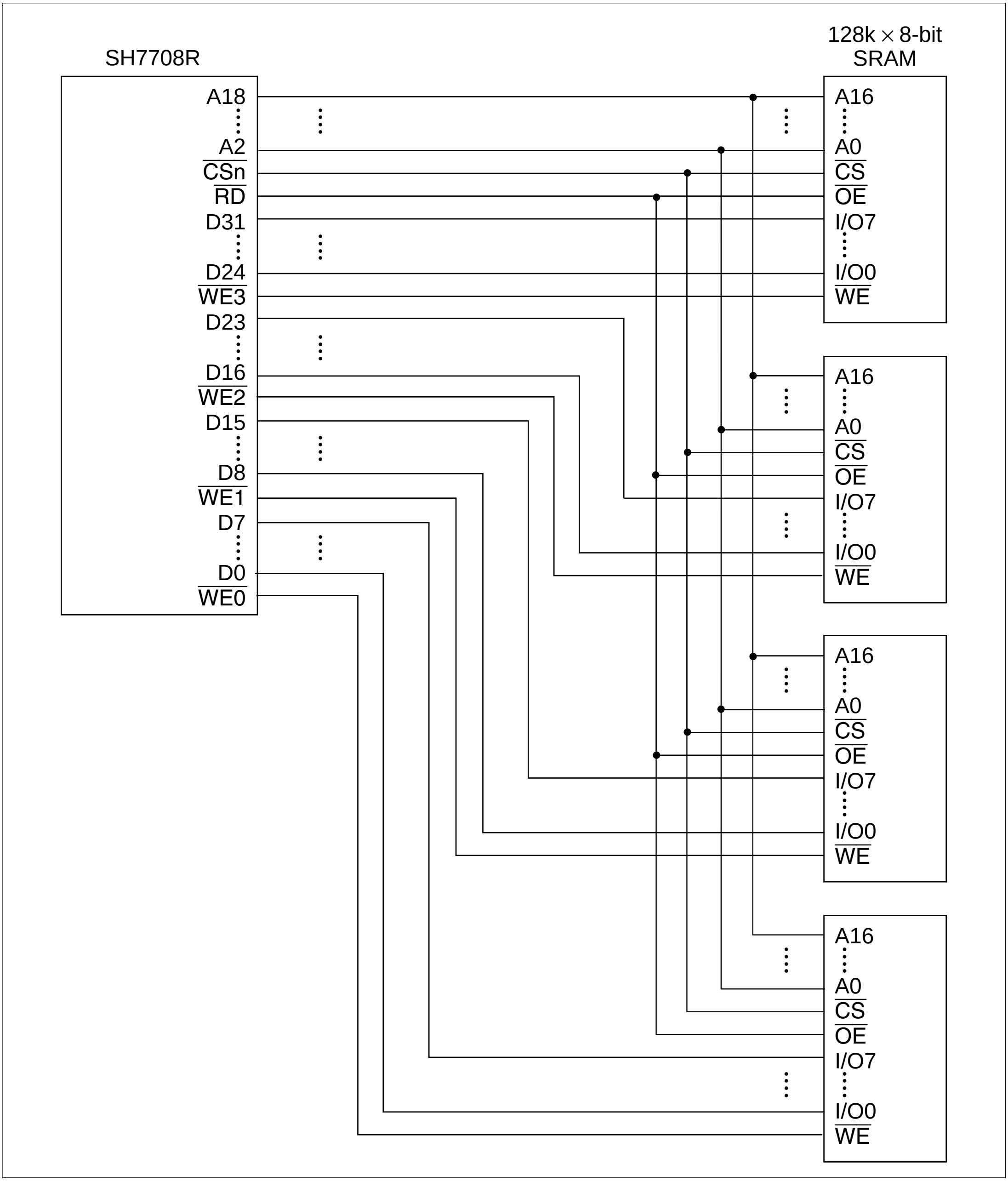


Figure 10.7 Example of 32-Bit Data Width SRAM Connection

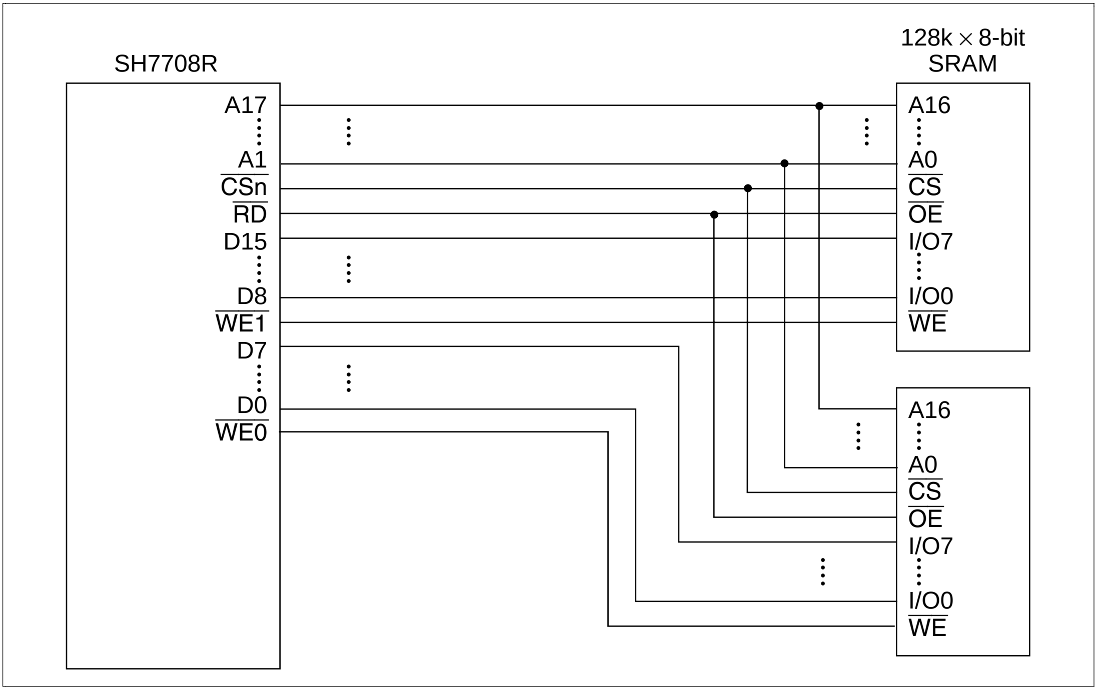


Figure 10.8 Example of 16-Bit Data Width SRAM Connection

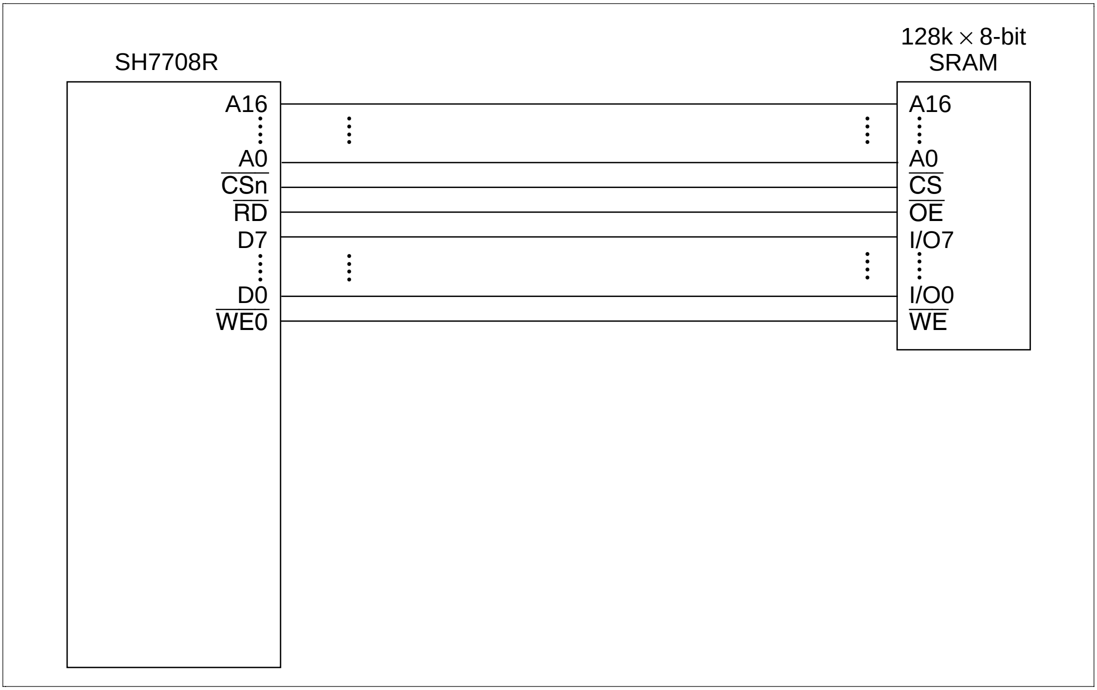


Figure 10.9 Example of 8-Bit Data Width SRAM Connection

Wait State Control: Wait state insertion on the basic interface can be controlled by the WCR2 settings. If the WCR2 wait specification bits corresponding to a particular area are not zero, a software wait is inserted in accordance with that specification. For details, see section 10.2.4, Wait Control Register 2 (WCR2).

The specified number of Tw cycles is inserted as wait cycles using the basic interface wait timing shown in figure 10.10.

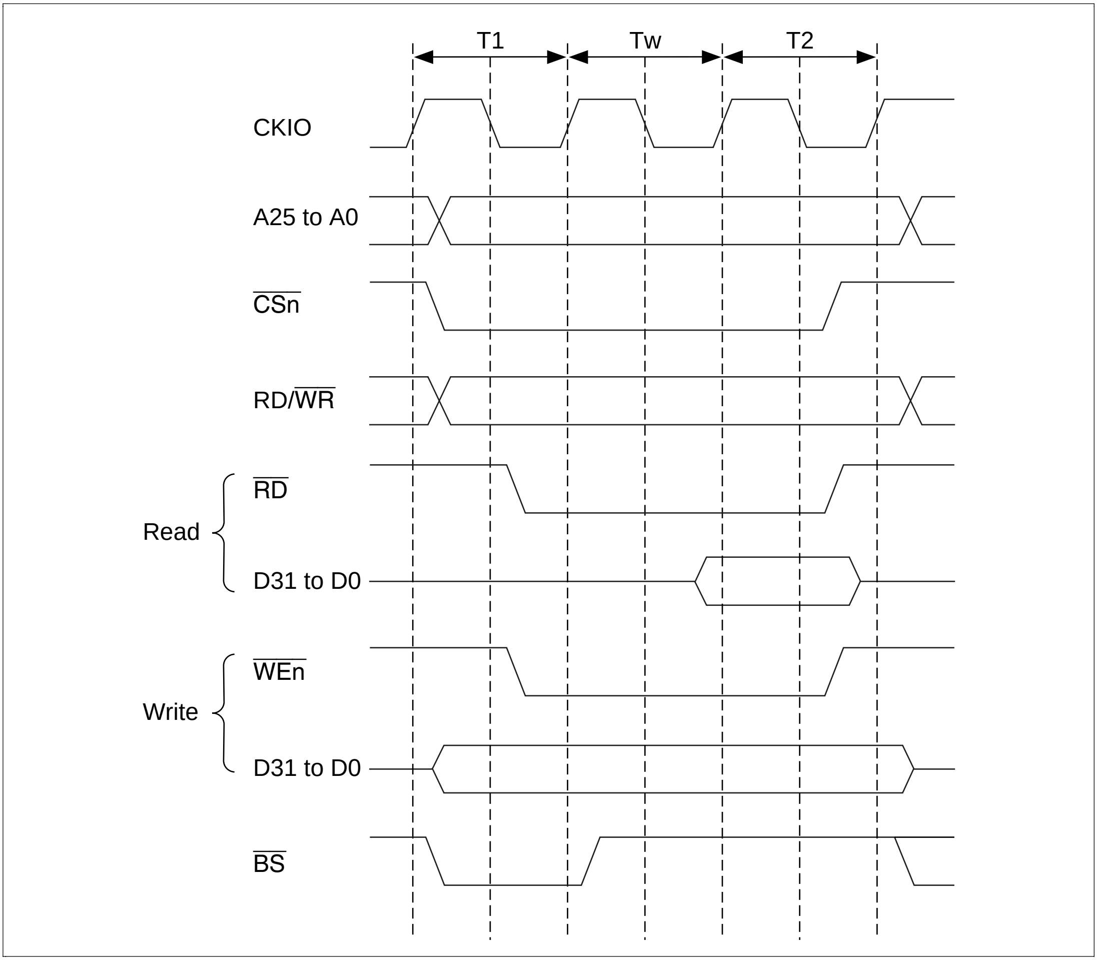


Figure 10.10 Basic Interface Wait Timing (Software Wait Only)

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, 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. $\overline{\text{WAIT}}$ is a synchronous signal.

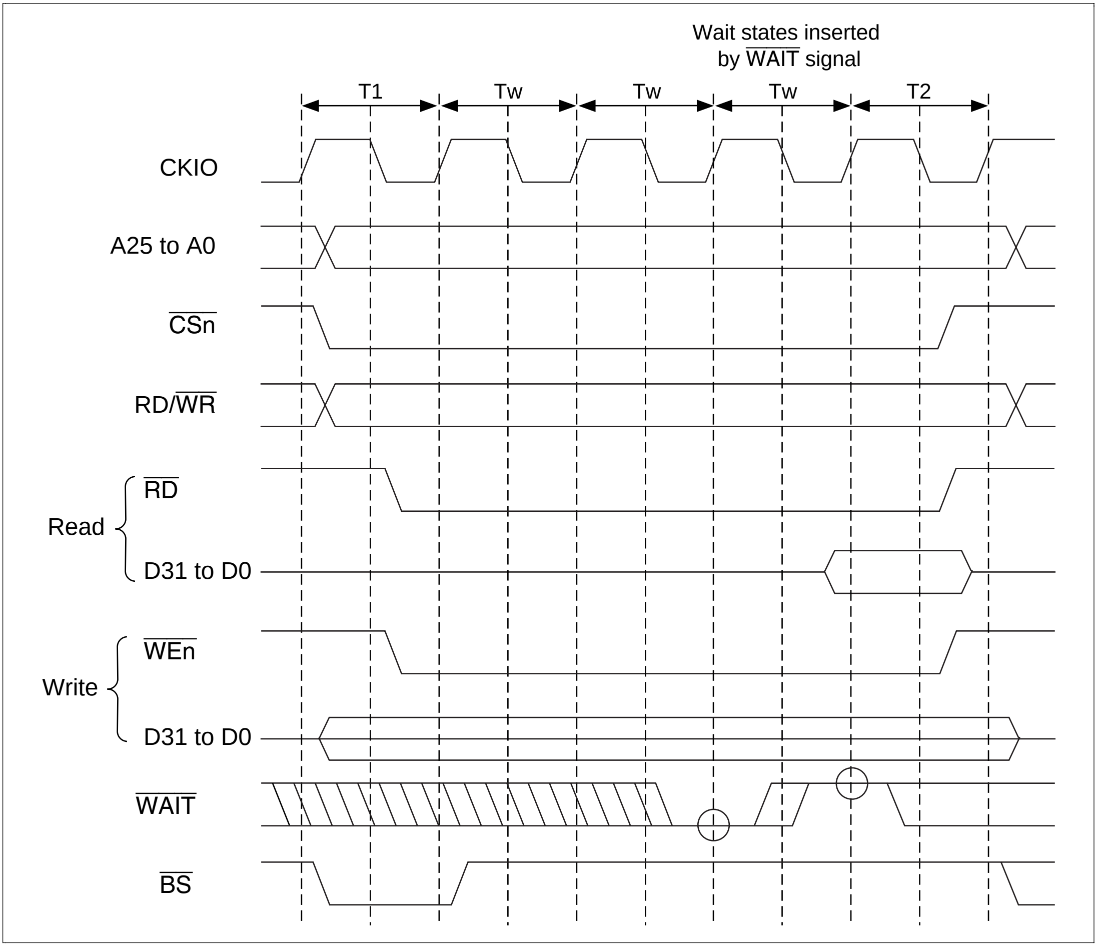


Figure 10.11 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal)

When the WAITSEL bit in the WCR1 register is set to 1, the $\overline{\text{WAIT}}$ signal is sampled at the fall of the clock.

If the setup time and hold times with respect to the falling edge of the clock are not satisfied, the value sampled at the next falling edge is used.

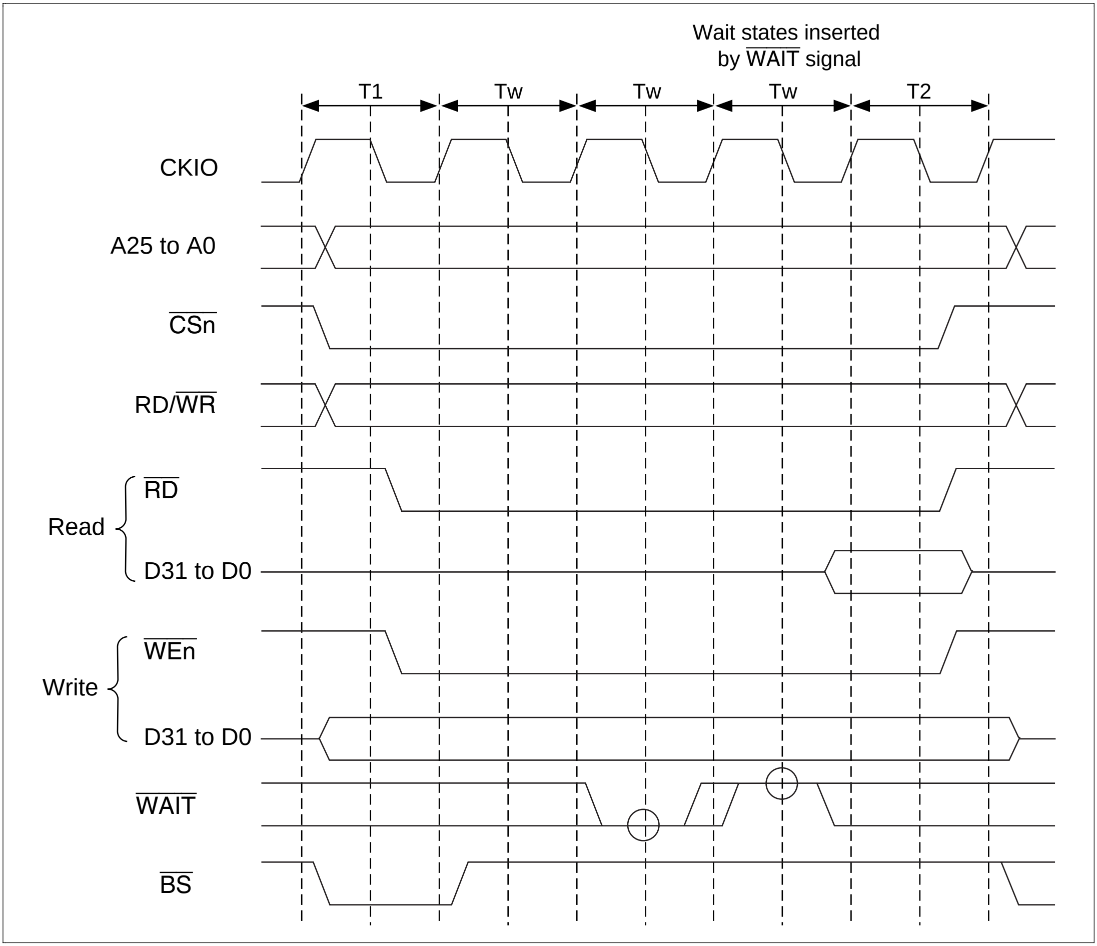


Figure 10.12 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal, WAITSEL = 1)

10.3.4 DRAM Interface

DRAM Connection Method: When the memory type bits (DRAMTP2–DRAMTP0) 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 SH7708 Series directly to DRAM.

16 or 32 bits can be selected as the interface data width for area 3 when bits DRAMTP2 to DRAMTP0 are set to 100, and 16 bits can be used for both area 2 and area when bits DRAMTP2 to DRAMTP0 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.

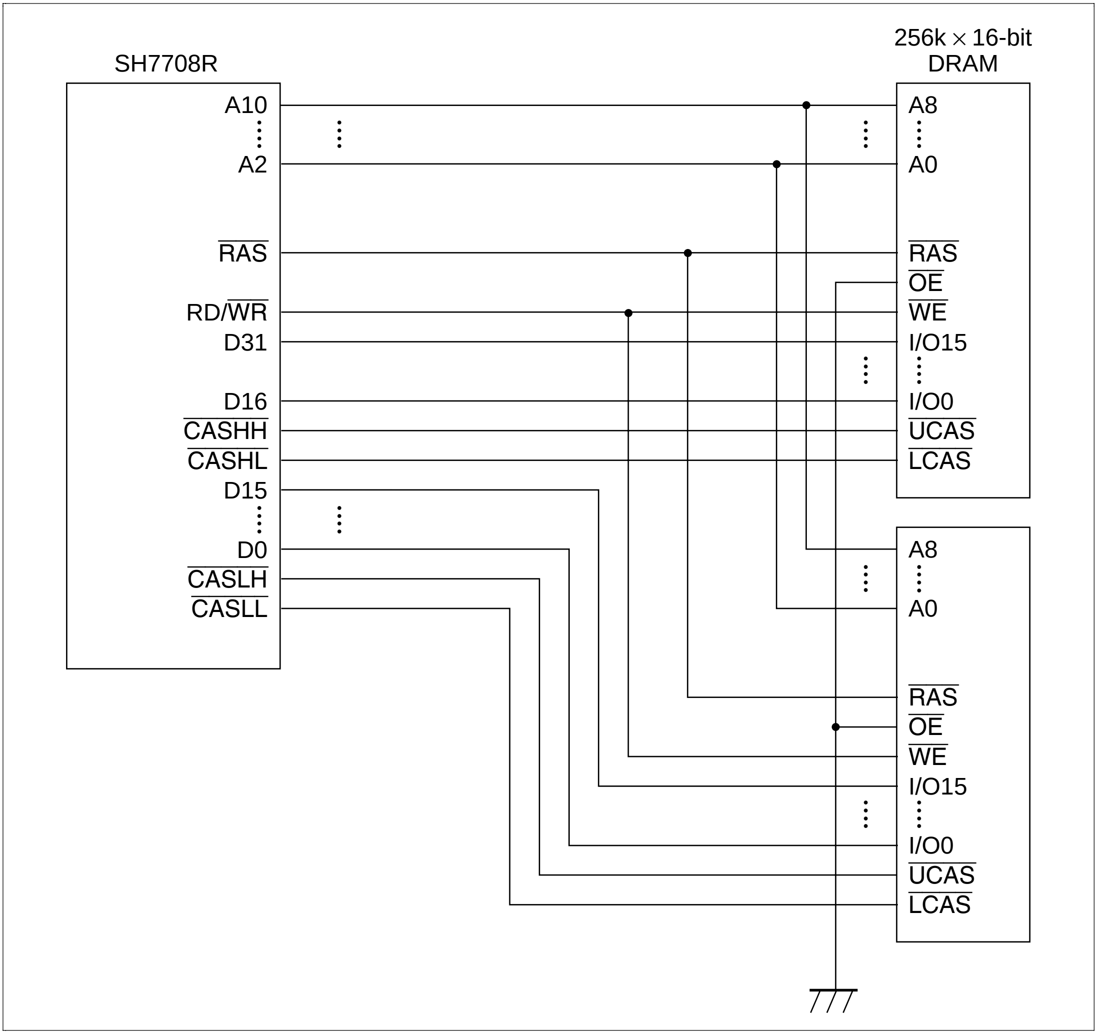


Figure 10.13 Example of DRAM Connection (32-Bit Data Width)

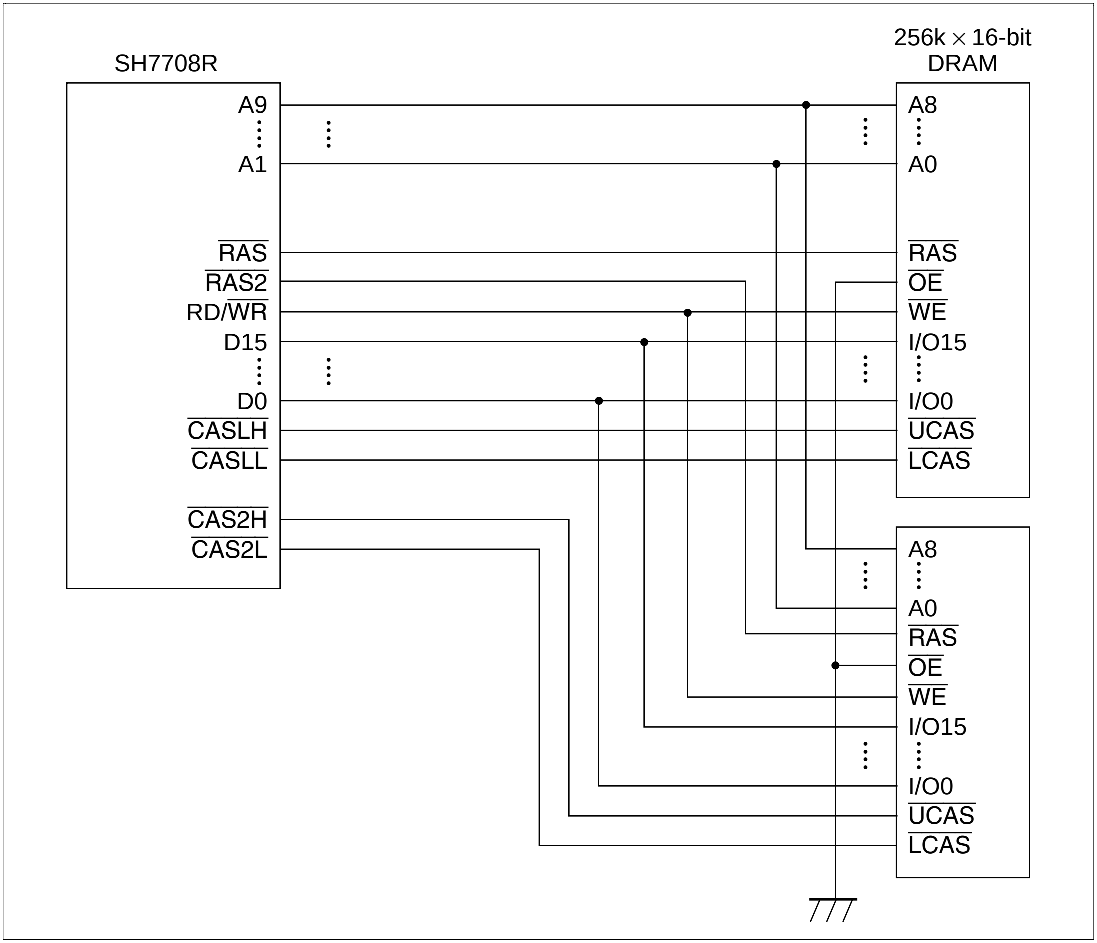


Figure 10.14 Example of DRAM Connection (16-Bit Data Width)

Address Multiplexing: When areas 2 and 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 SH7708 Series without using an external address multiplexer circuit. Any of the four multiplexing methods shown below can be selected by setting bits AMX1 and AMX0 in MCR for area 3 DRAM, or bits AMX1 and AMX0 in DCR for area 2 DRAM. The relationship between bits AMX1 and AMX0 and address multiplexing is shown in table 10.12. The address output pins subject to address multiplexing are A15 to A1. Pins A25 to A16 carry the original address.

Table 10.12 Relationship between AMX1-0 and Address Multiplexing

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

Basic Timing: Figure 10.15 shows the basic timing for DRAM access is 3 cycles. Tpc is the precharge cycle, Tr the $\overline{\text{RAS}}$ assert cycle, Tc1 the $\overline{\text{CAS}}$ assert cycle, and Tc2 the read data latch cycle.

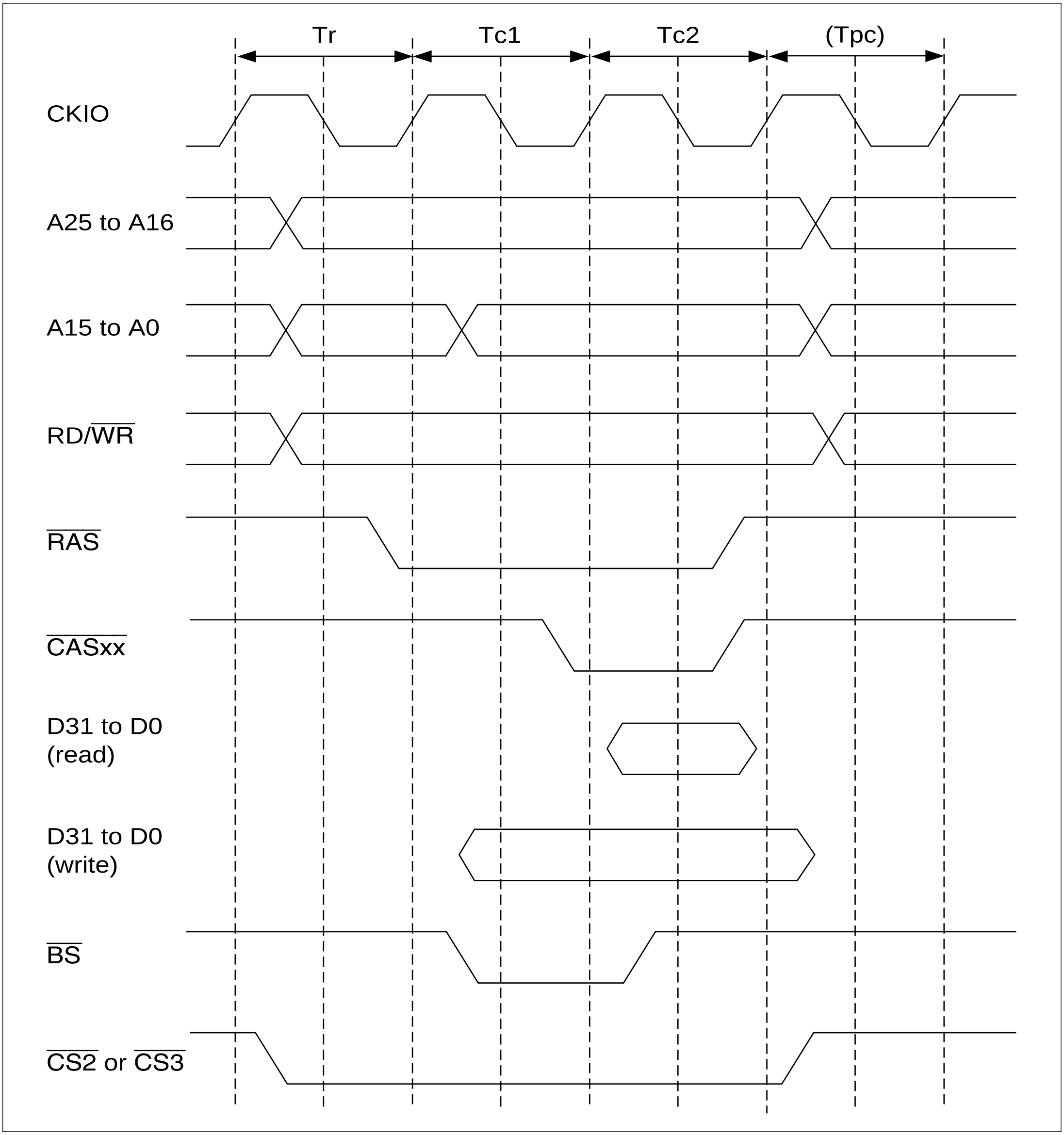


Figure 10.15 Basic Timing for DRAM Access

Wait State Control: As the clock frequency increases, it becomes impossible to complete all states in one cycle as in basic access. Therefore, provision is made for state extension by using the setting bits in WCR2, MCR, and DCR. The timing with state extension using these settings is shown in figure 10.16. Up to four additional Tpc cycles (cycles used to secure the RAS precharge time) can be inserted by means of the TPC bits in MCR and DCR. However, if there is a DRAM access request immediately after an auto-refresh (CAS-before-RAS refresh), the interval until the next assertion of RAS is 2 cycles, regardless of the values in MCR and DCR. The number of cycles from $\overline{\text{RAS}}$ assertion to $\overline{\text{CAS}}$ assertion can be set to between 1 and 4 by inserting Trw cycles by means of the RCD bits in MCR and DCR. The number of cycles from $\overline{\text{CAS}}$ assertion to the end of the access can be varied between 1 and 3 according to the setting of A1–2W (1,0) or A3W (1,0) in WCR2.

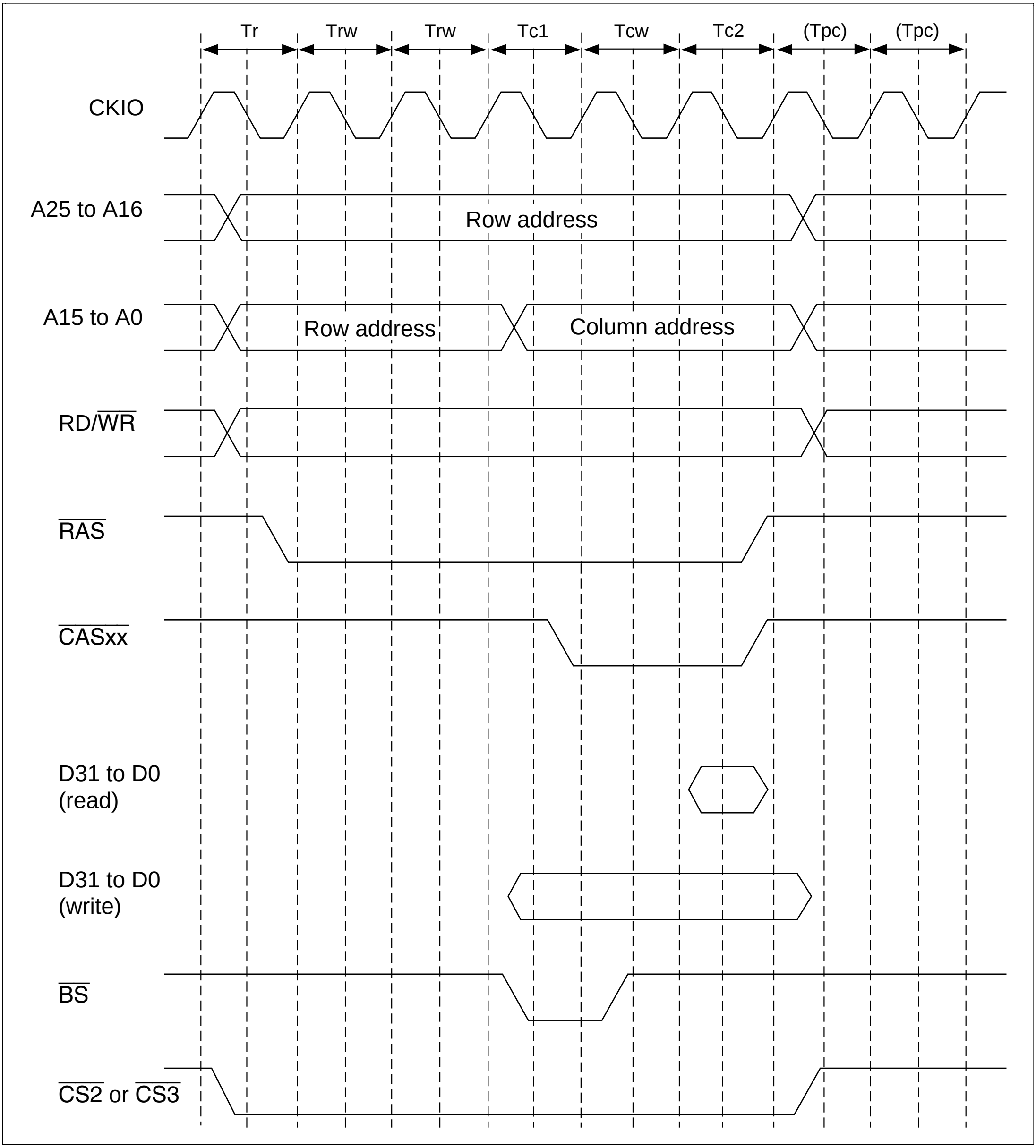


Figure 10.16 DRAM Wait State Timing

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

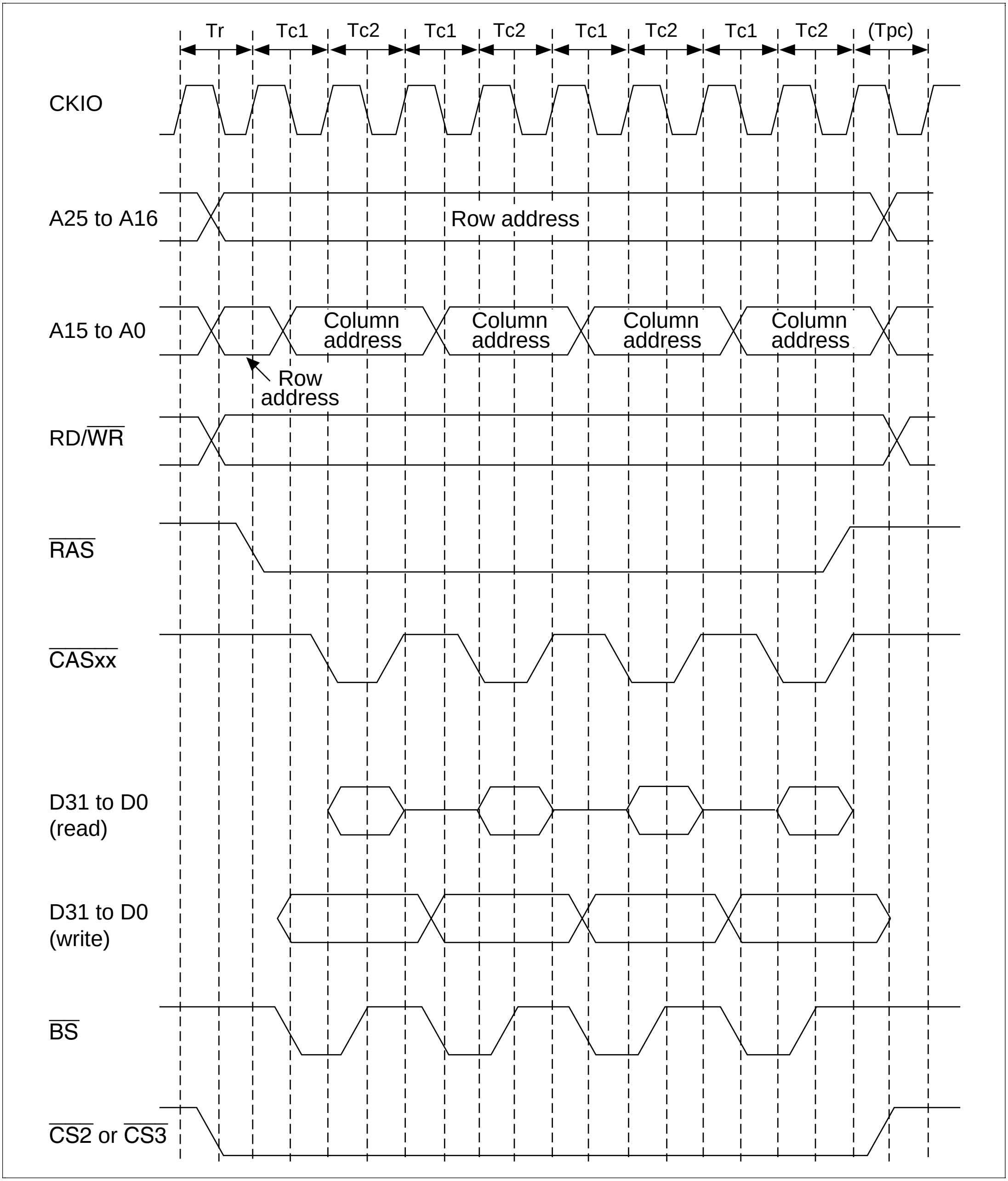


Figure 10.17 DRAM Burst Access Timing

EDO Mode: In DRAM, an extended data out (EDO) 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. (This is 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.) In the SH7708 Series, the EDO mode bit (EDOMODE) in MCR enables selection, for area 3 DRAM only, of either normal access/burst access using high-speed page mode or EDO mode normal access/burst access. EDO mode normal access is shown in figure 10.18, and burst access in figure 10.19.

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 $\overline{\text{CAS}}$ signal. This delays the data latch timing by 1/2 cycle to the rising edge of the CKIO clock, enabling the DRAM access time to be increased.

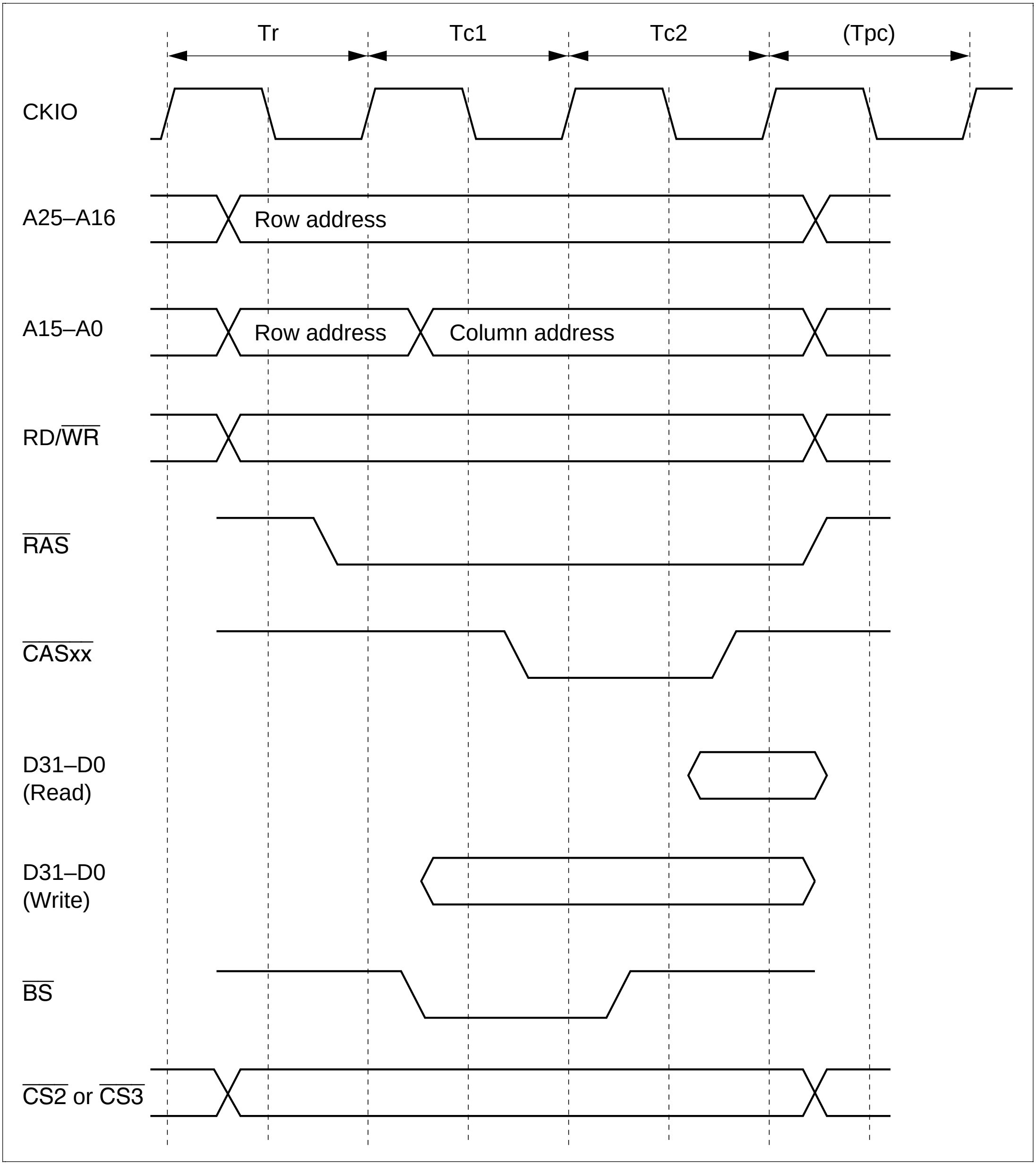


Figure 10.18 Normal Access Timing in DRAM EDO Mode

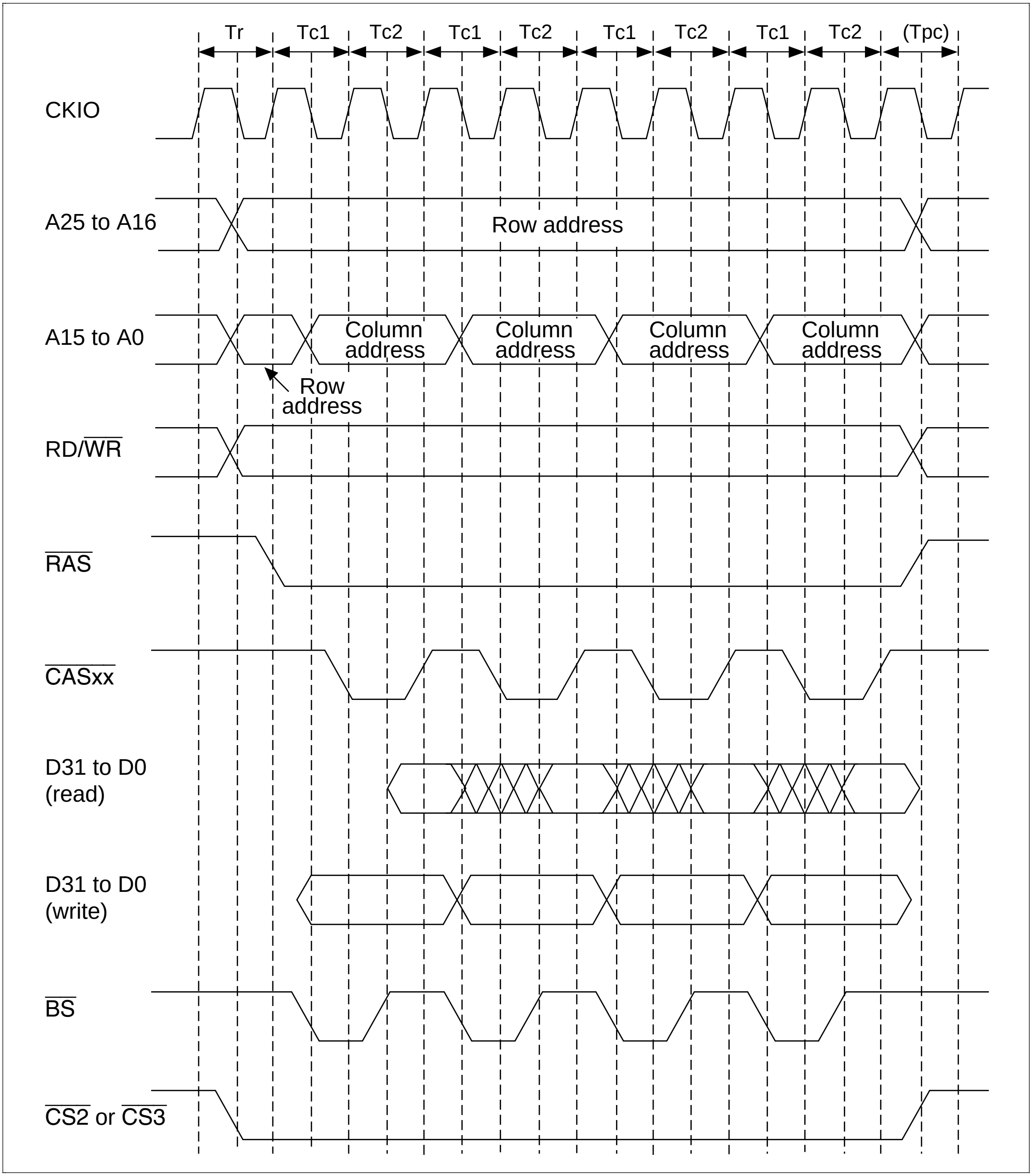


Figure 10.19 Burst Access Timing in DRAM EDO Mode

Refresh Timing: The bus state controller includes a function for controlling DRAM refreshing. 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. It also supports self-refresh mode.

When CAS-before-RAS refresh cycles are executed, refreshing is performed at intervals determined by the input clock selected by bits CKS2–CKS0 in RTCSR, and the value set in RTCOR. The value of bits CKS2–CKS0 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 to 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 SH7708 Series' 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 10.20 shows the operation of CAS-before-RAS refreshing.

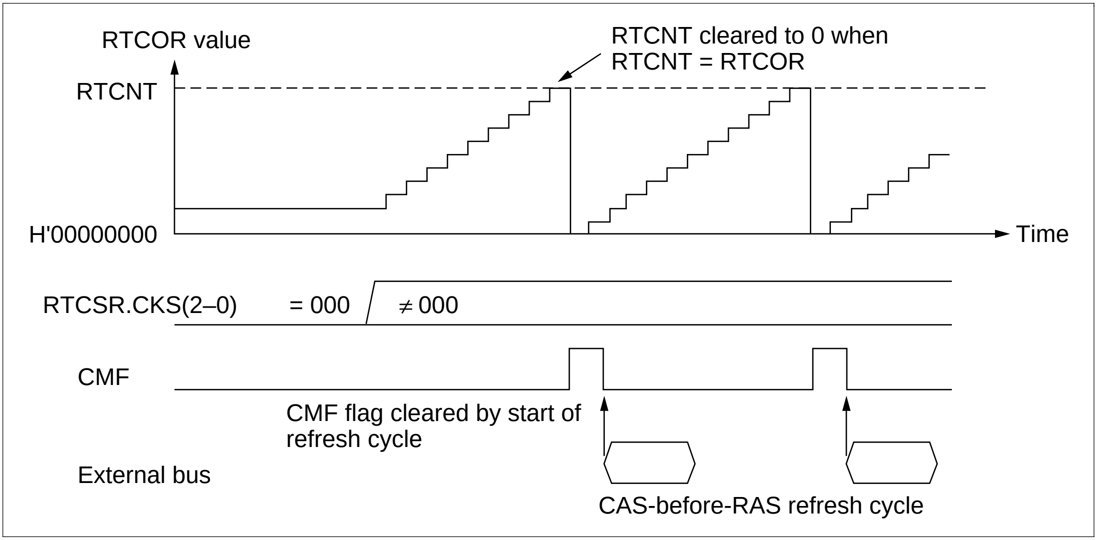


Figure 10.20 CAS-Before-RAS Refresh Operation

Figure 10.21 shows the timing of the CAS-before-RAS refresh cycle.

The number of $\overline{\text{RAS}}$ 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.

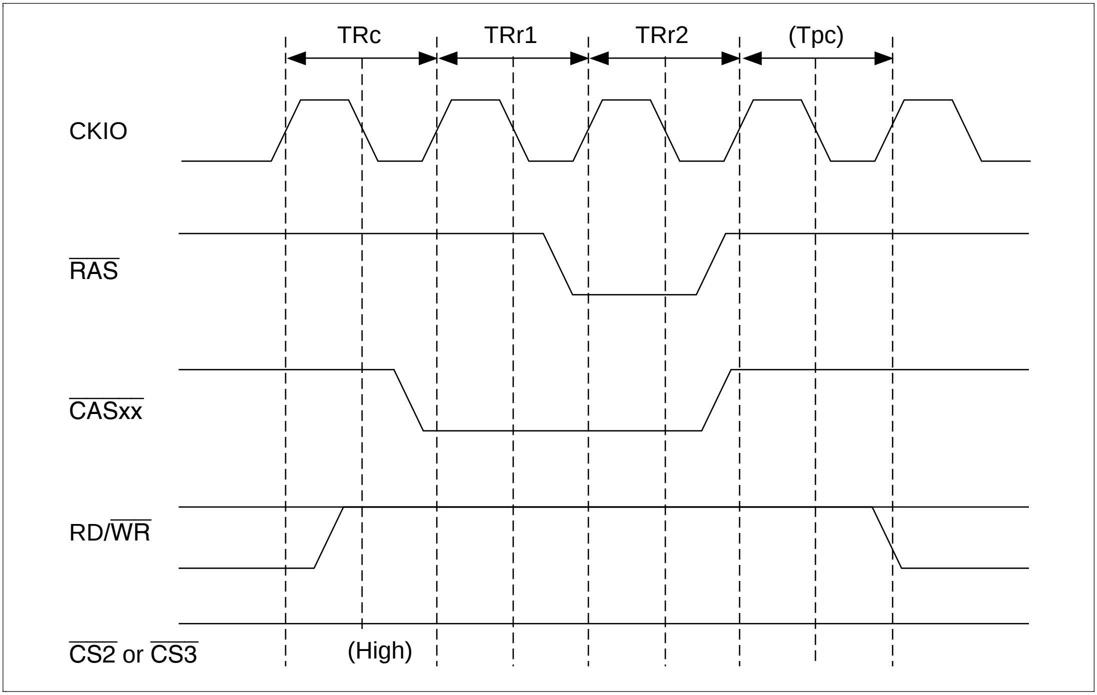

TRc
TRr1
TRr2
(Tpc)
CKIO
$\overline{\text{RAS}}$
$\overline{\text{CASxx}}$
RD/$\overline{\text{WR}}$
$\overline{\text{CS2}}$ or $\overline{\text{CS3}}$
(High)

Figure 10.21 DRAM CAS-Before-RAS Refresh Cycle Timing

The self-refreshing supported by the SH7708 Series is shown in figure 10.22.

After the self-refresh is cleared, the refresh controller immediately generates a refresh request. The RAS precharge time immediately after the end of the self-refreshing can be set by the TPC bits in MCR and DCR.

DRAMs include low-power products (L versions) with a long refresh cycle time (for example, the L version of the HM51W4160AL has a refresh cycle of 1024 cycles/128 ms compared with 1024 cycles/16 ms for the normal version). With these DRAMs, however, the same refresh cycle as the normal version is requested only when refreshing immediately after self-refreshing. Therefore, to ensure efficient DRAM refreshing, an overflow interrupt is generated and the refresh cycle is restored to its proper value. This occurs 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 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. Reset 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.

This procedure causes refreshing immediately following a self-refresh to occur in a short cycle. 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 a manual reset.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in 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. Controlling the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals to become high-impedance or continue to be output is performed with the HIZCNT bit in BCR1. This enables the DRAM to be kept in the self-refreshing state.

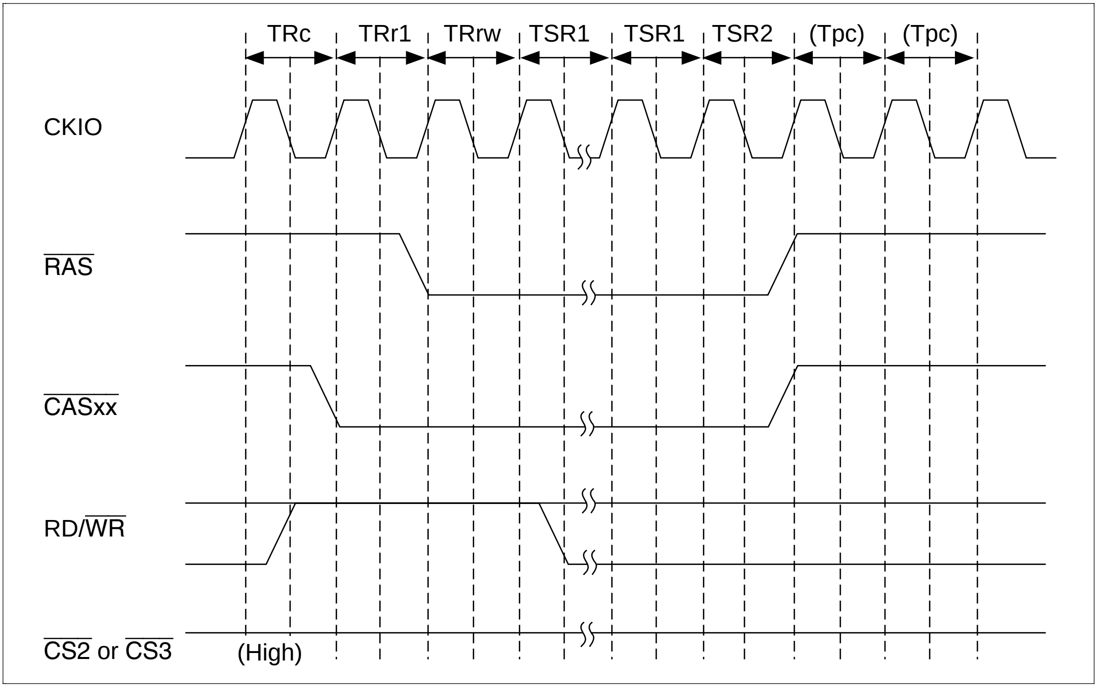


Figure 10.22 DRAM Self-Refresh Cycle Timing

Power-On Sequence: For DRAM after powering on, a minimum wait time of 100 µs or 200 µs, or more during which no access can be performed, should be provided, followed by the prescribed number (usually 8 or more) of dummy CAS-before-RAS refresh cycles. As the bus state controller does not perform any special operations for a power-on reset, the power-on sequence must be carried out by the initialization program executed after a power-on reset.

10.3.5 Synchronous DRAM Interface

Synchronous DRAM Direct Connection: Since synchronous DRAM can be selected by the $\overline{\text{CS}}$ signal, physical space areas 2 and 3 can be connected using $\overline{\text{RAS}}$ and other control signals in common. If the memory type bits (DRAMTP2–DRAMTP0) in BCR1 are set to 010, area 2 is normal memory space and area 3 is synchronous DRAM space; if set to 011, areas 2 and 3 are both synchronous DRAM space.

With the SH7708 Series, burst length 1 burst read/single write mode is supported as the synchronous DRAM operating mode. The data bus width is fixed at 32 bits, and the size bit (SZ) in MCR must be set to 1. The burst enable bit (BE) in MCR is ignored, a16-bit burst transfer is performed in a cache fill/copy-back cycle, and only one access is performed in a write-through area write or a noncacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$, DQMUU, DQMUL, DQMLU, DQMLL, and CKE. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid and fetched to the synchronous DRAM only when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. Synchronous DRAM can therefore be connected in parallel to a number of areas. CKE is negated (low) only when self-refreshing is performed, otherwise it is asserted (high).

Commands for synchronous DRAM are specified by $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, and special address signals. The commands are NOP, auto-refresh (REF), self-refresh (SELF), precharge all banks (PALL), precharge specified bank (P RE), row address strobe bank active (ACTV), read (READ), read with precharge (READA), write (WRIT), write with precharge (WRITA), and mode register write (MRS).

Byte specification is performed by DQMUU, DQMUL, DQMLU, and DQMLL. A read/write is performed for the byte for which the corresponding DQM is low. In big-endian mode, DQMUU specifies an access to address 4n, and DQMLL specifies an access to address 4n + 3. In little-endian mode, DQMUU specifies an access to address 4n + 3, and DQMLL specifies an access to address 4n.

Figure 10.23 shows an example of the connection of 256k × 16-bit synchronous DRAMs.

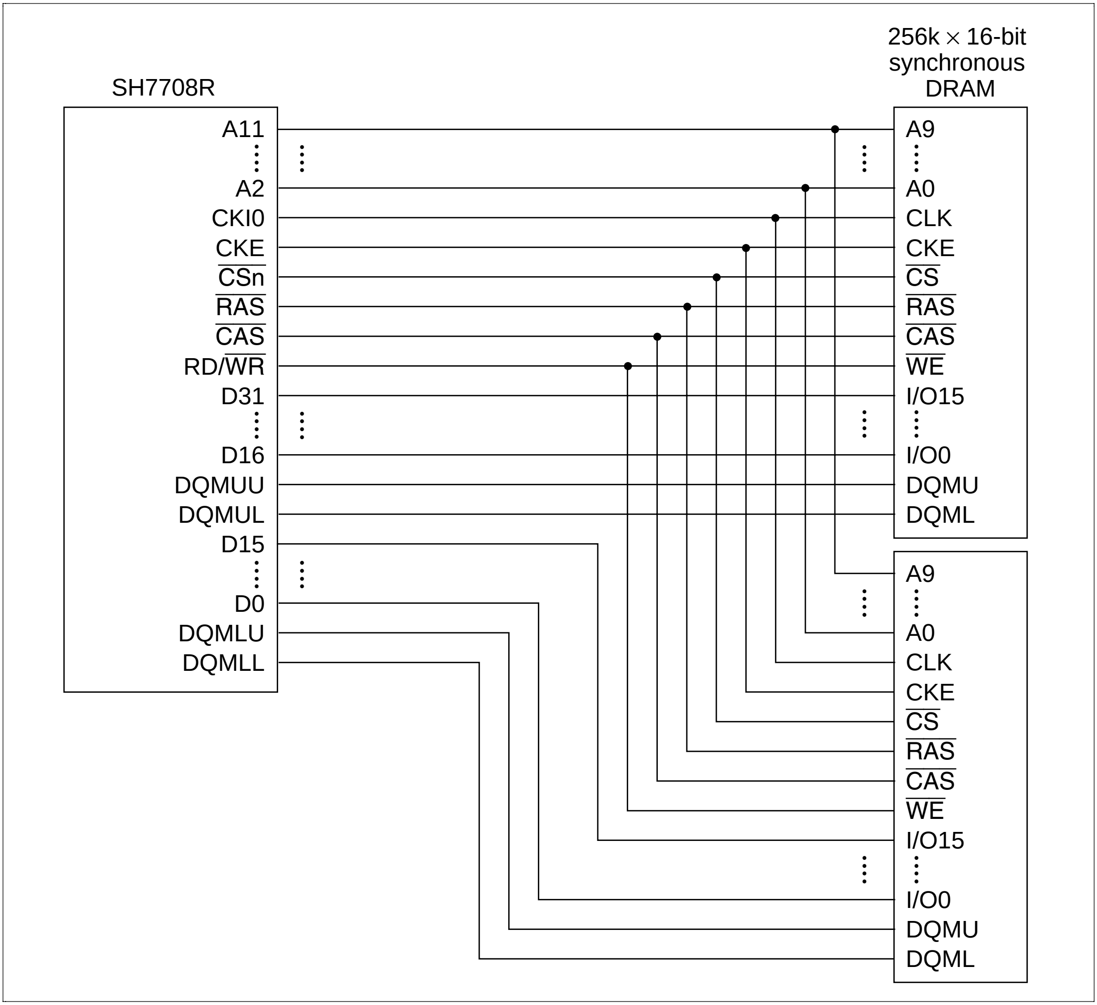


Figure 10.23 Example of Synchronous DRAM Connection

Address Multiplexing: Synchronous DRAM can be connected without external multiplexing circuitry in accordance with the address multiplex specification bits AMX1 and AMX0 in MCR. Table 10.13 shows the relationship between the address multiplex specification bits and the bits output at the address pins.

A25–A16 and A0 are not multiplexed; the original values are always output at these pins.

When A0, the LSB of the synchronous DRAM address, is connected to the SH7708 Series, it performs longword address specification. Connection should therefore be made in this order: connect pin A0 of the synchronous DRAM to pin A2 of the SH7708 Series, then connect pin A1 to pin A3. Table 10.14 shows the example of correspondence between SH7708 Series and synchronous DRAM address pins.

Table 10.13 Relationship between SZ, AMX, and Address Multiplex Output

Setting			External Address Pins							
AMX1	**AMX0**	**Output Timing**	**A1 to A8**	**A9**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**
0	0	Column address	A1 to A8	A9	A10	A11	L/H*[1]	A21*[2]	A14	A15
		Row address	A9 to A16	A17	A18	A19	A20	A21*[2]	A22	A23
0	1	Column address	A1 to A8	A9	A10	A11	L/H*[1]	A22*[2]	A14	A15
		Row address	A10 to A17	A18	A19	A20	A21	A22*[2]	A23	A24
1	0	Column address	A1 to A8	A9	A10	A11	L/H*[1]	A23*[2]	A14	A15
		Row address	A11 to A18	A19	A20	A21	A22	A23*[2]	A24	A25
1	1	Column address	A1 to A8	A9	L/H*[1]	A19*[2]	A12	A13	A14	A15
		Row address	A9 to A16	A17	A18	A19*[2]	A20	A21	A22	A23

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

2. Bank address specification.

Table 10.14 Example of Correspondence between SH7708 Series and Synchronous DRAM Address Pins (AMX (1-0) = 11)

SH7708 Series Address Pin			Synchronous	
	RAS Cycle	CAS Cycle	DRAM Address Pin	Function
A11	A19	A19	A9	BANK select bank address
A10	A18	L/H	A8	Address precharge setting
A9	A17	A9	A7	Address
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Not used	—
A0	A0	A0	Not used	

Burst Read: The timing chart for a burst read is shown in figure 10.24. In the following example it is assumed that four 2M × 8-bit synchronous DRAMs are connected and a 32-bit data width is used, and the burst length is 1. Following the Tr cycle in which ACTV command output is performed, a READ command is issued in the Tc1, Tc2, and Tc3 cycles, and a READA command in the Tc4 cycle. The read data is then accepted on the rising edge of the external command clock (CKIO) from cycle Td1 to cycle Td4. The Tpc cycle is used to wait for completion of auto-precharge based on the READA command inside the synchronous DRAM; no new access command can be issued to the same bank during this cycle, but access to synchronous DRAM for another area is possible. In the SH7708 Series, the number of Tpc cycles is determined by the TPC bit specification in MCR, and commands cannot be issued for the same synchronous DRAM during this interval.

The example in figure 10.24 shows the basic timing. To connect slower synchronous DRAM, the cycle can be extended by setting the WCR2 and MCR bits. The number of cycles from the ACTV command output cycle, Tr, to the READ command output cycle, Tc1, can be specified by the RCD bit in MCR, with a value of 0 to 3 specifying 1 to 4 cycles, respectively. For 2 or more cycles, a Trw cycle, in which an NOP command is issued for the synchronous DRAM, is inserted between the Tr cycle and the Tc cycle. The number of cycles from READ and READA command output cycles Tc1–Tc4 to the first read data latch cycle, Td1, can be specified as 1 to 3 cycles independently for areas 2 and 3 by means of A1–2W1 and A1–2W0 or A3W1 and A3W0 in WCR2. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

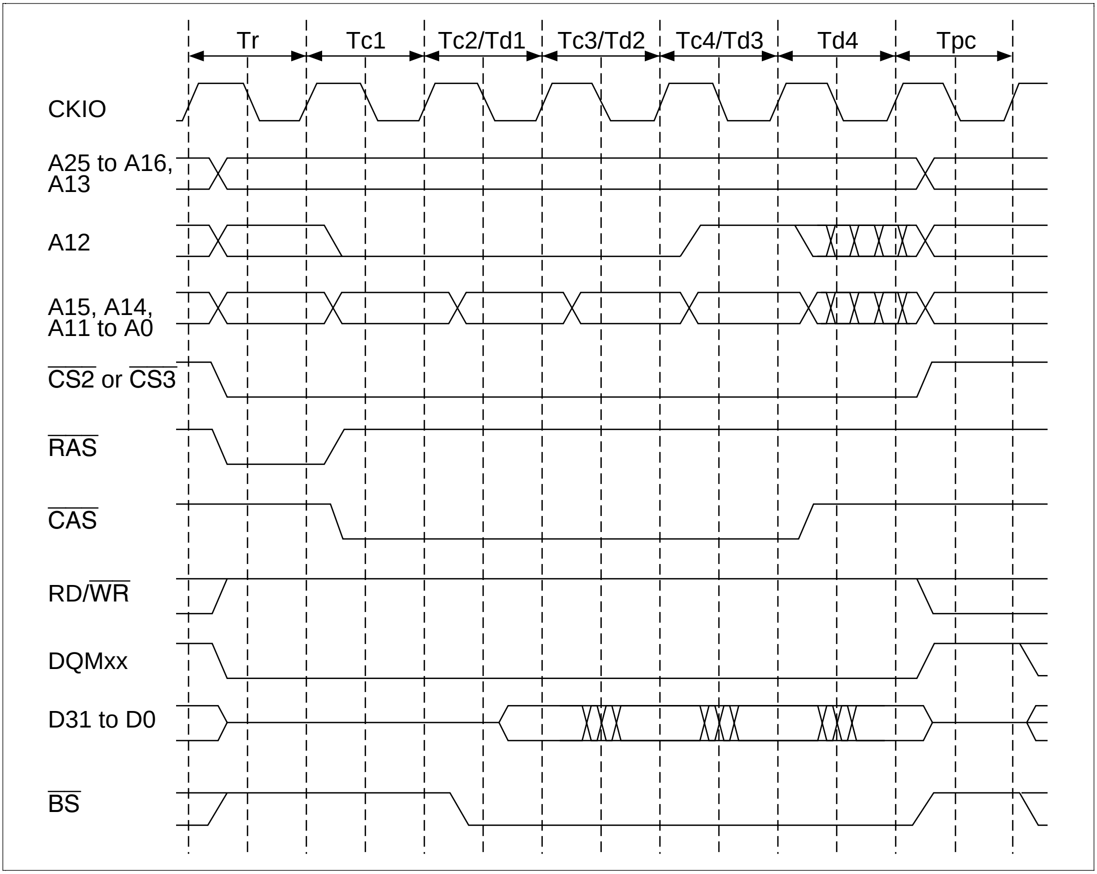


Figure 10.24 Basic Timing for Synchronous DRAM Burst Read

Figure 10.25 shows the burst read timing when RCD is set to 1, A3W1 and A3W0 are set to 10, and TPC is set to 1.

The $\overline{\text{BS}}$ cycle, which is asserted for one cycle at the start of a bus cycle for normal access space, is asserted in each of cycles Td1–Td4 in a synchronous DRAM cycle. When a burst read is performed, the address is updated each time $\overline{\text{CAS}}$ is asserted. As the unit of burst transfer is 16 bytes, address updating is performed for A3 and A2 only. In a fill operation in the event of a cache miss, the order of access is: the missed data is read first, then 16-byte boundary data including the missed data is read in wraparound mode.

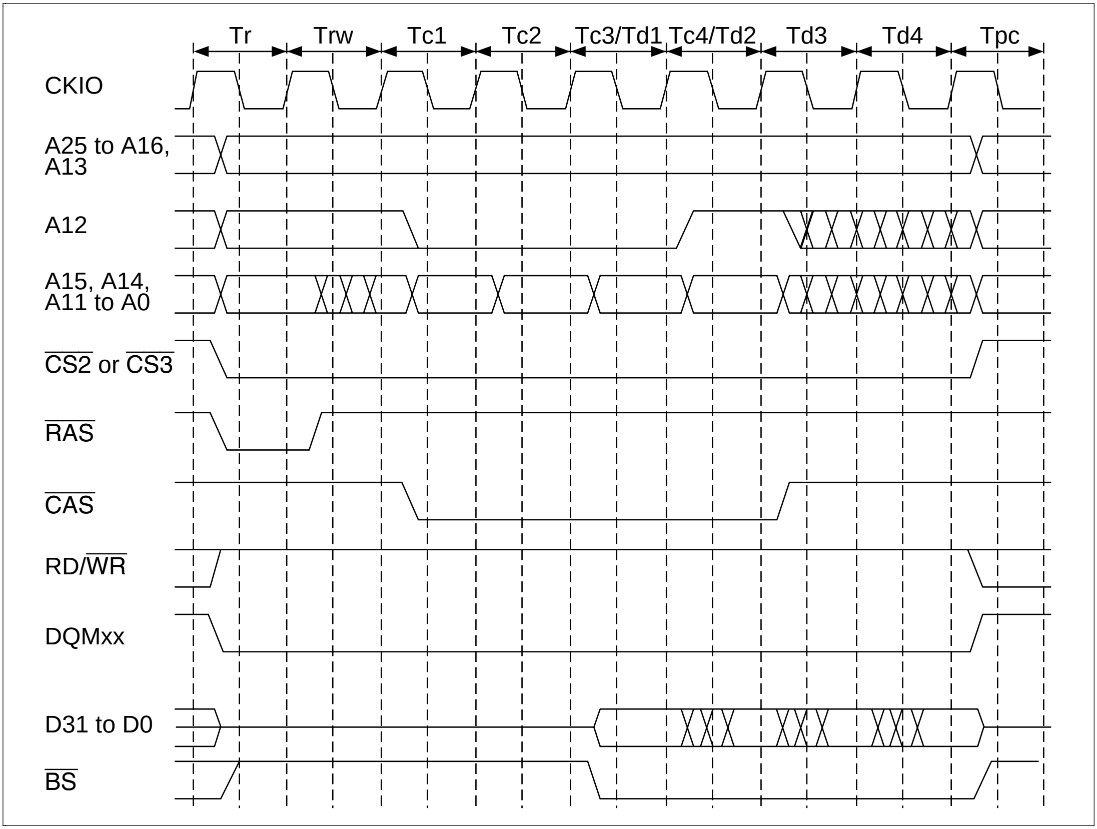


Figure 10.25 Synchronous DRAM Burst Read Wait Specification Timing

Single Read: Figure 10.26 shows the timing when a single address read is performed. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

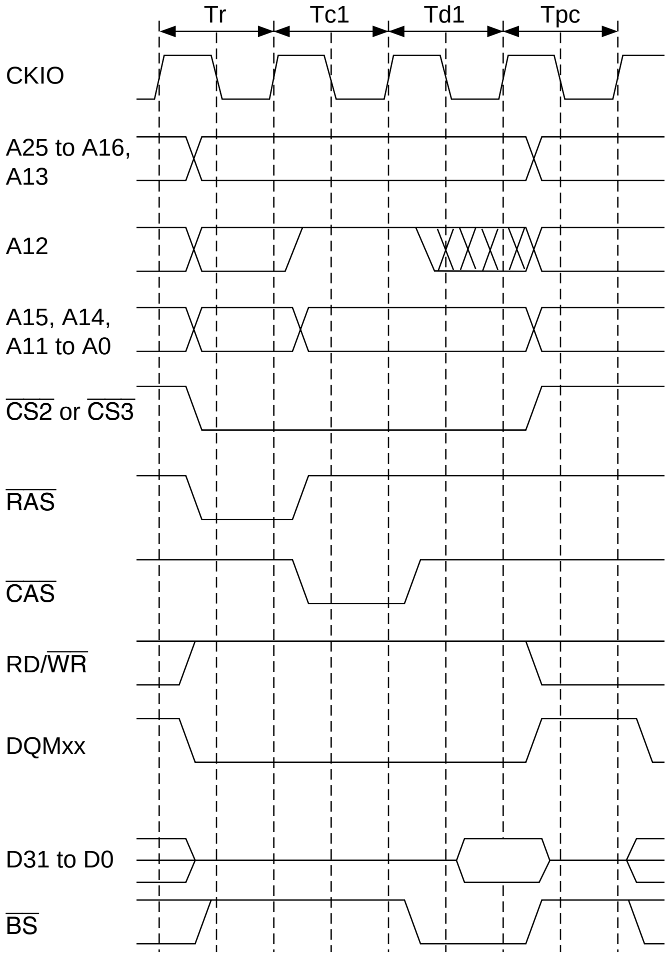


Figure 10.26 Basic Timing for Synchronous DRAM Single Read

Burst Write: The timing chart for a burst write is shown in figure 10.27. In the SH7708 Series, a burst write occurs only in the event of cache copy-back. In a burst write operation, following the Tr cycle in which ACTV command output is performed, a WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and a WRITA command that performs auto-precharge is issued in the Tc4 cycle. In the write cycle, the write data is output at the same time as the write command. For the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is added as a wait interval until precharging is started, following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

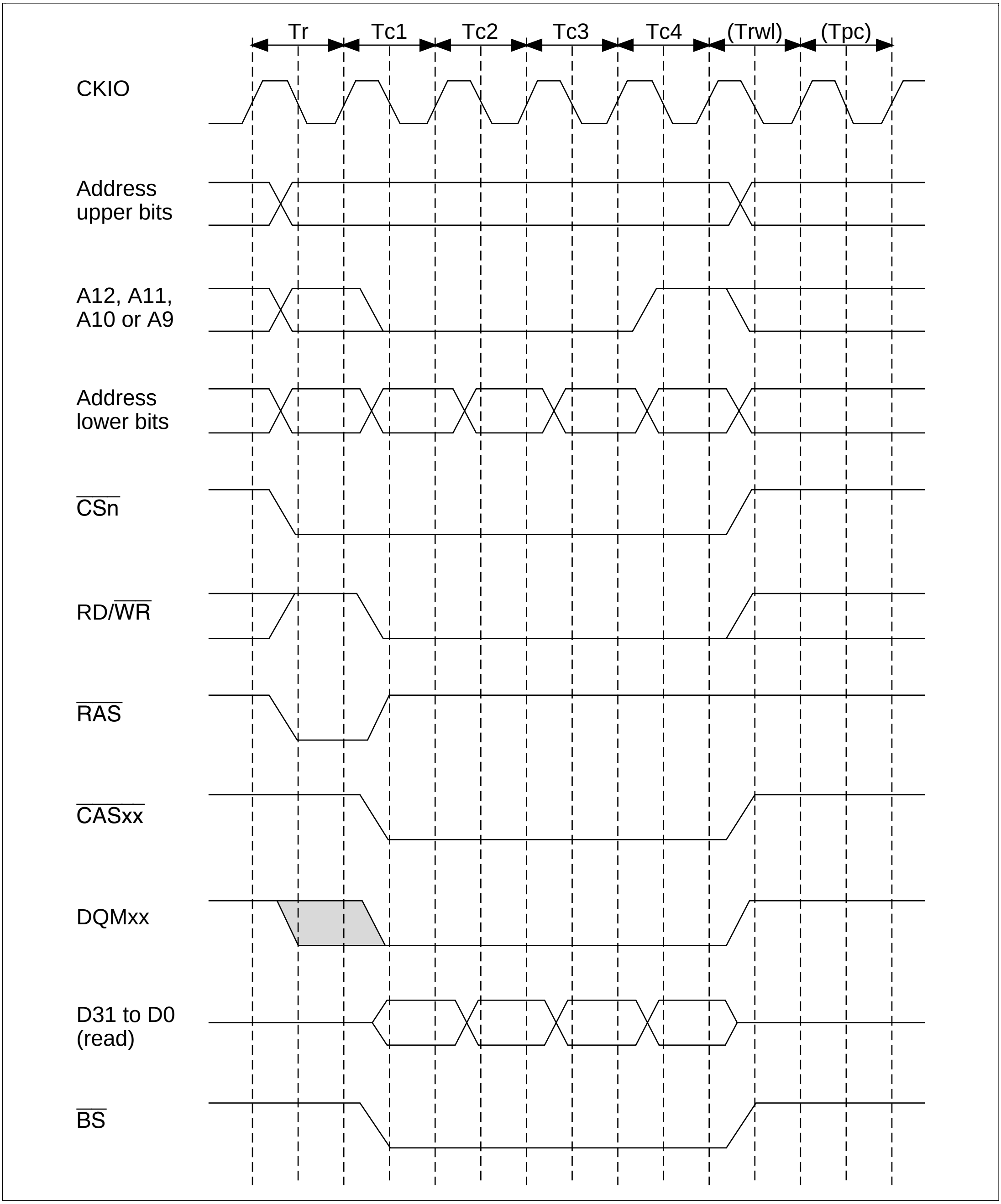


Figure 10.27 Basic Timing for Synchronous DRAM Burst Write

Single Write: The basic timing chart for write access is shown in figure 10.28. In a single write operation, following the Tr cycle in which ACTV command output is performed, a WRITA command that performs auto-precharge is issued in the Tc1 cycle. In the write cycle, the write data is output at the same time as the write command. For the write with auto-precharge command, precharging of the relevant bank is performed in synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

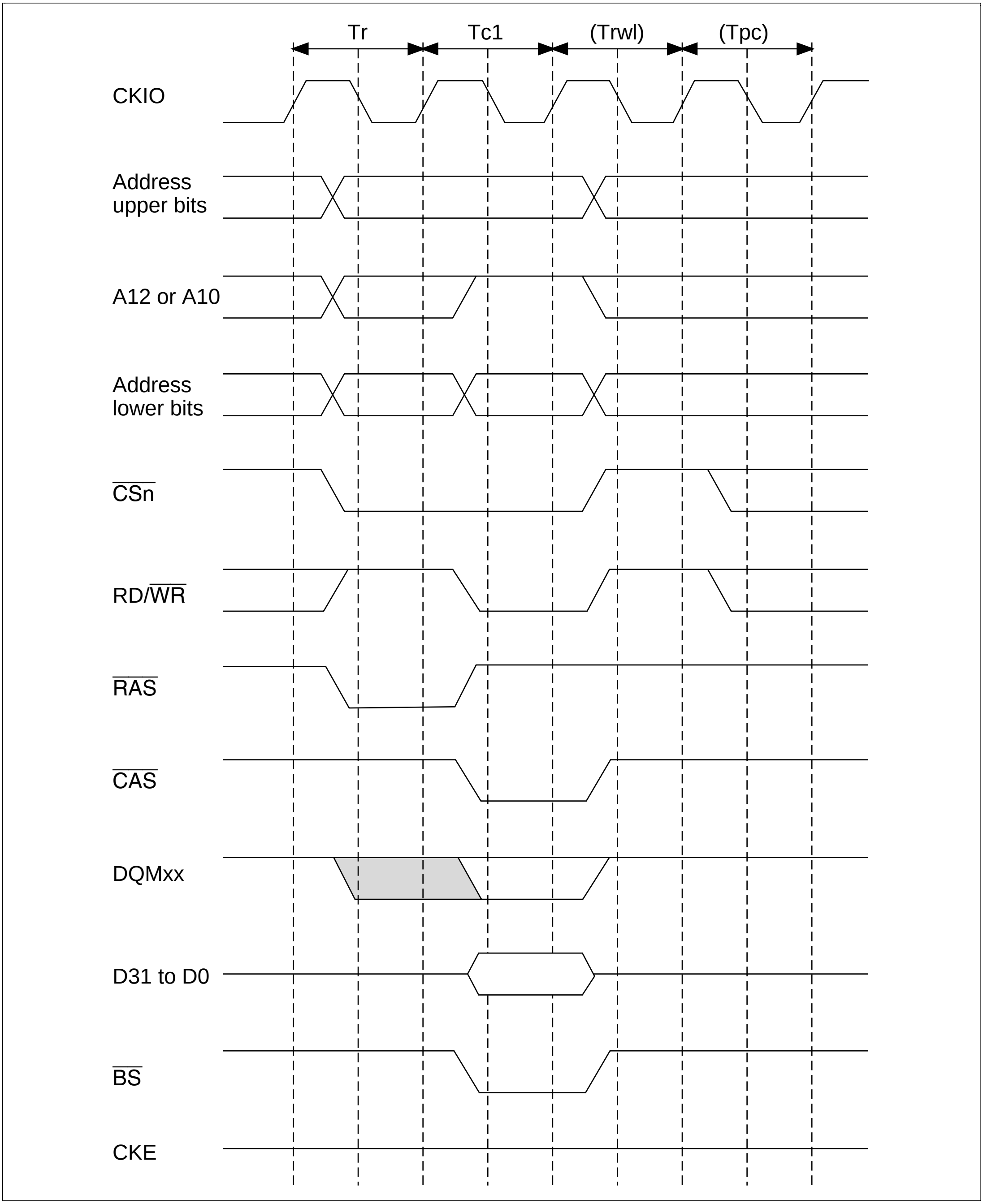


Figure 10.28 Basic Timing for Synchronous DRAM Single Write

Refreshing: The bus state controller is provided with a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-Refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2–CKS0 in RTCSR, and the value set in RTCOR. The value of bits CKS2–CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2 to CKS0 setting. When the clock is selected by CKS2–CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 10.29 shows the auto refresh operation. Figure 10.30 shows the auto-refresh cycle timing.

First, an REF command is issued in the TRr cycle. After the TRr cycle, new command output cannot be performed for the duration of the number of cycles specified by the TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. The TRAS and TPC bits must be set to satisfy the synchronous DRAM refresh cycle time stipulation (active/active command delay time).

Auto-refreshing is performed in normal operation, in sleep mode, and in a manual reset.

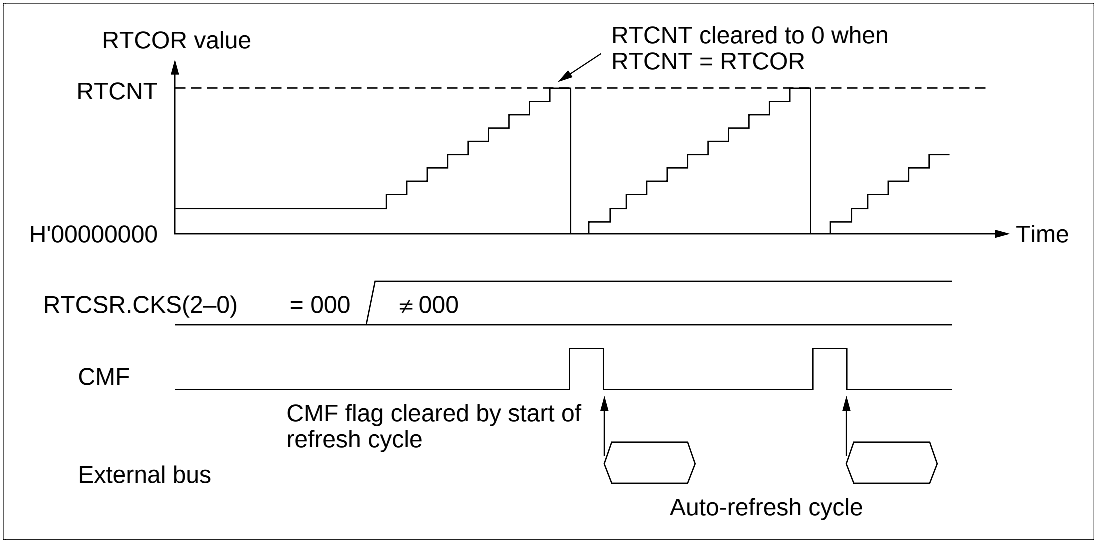


Figure 10.29 Auto-Refresh Operation

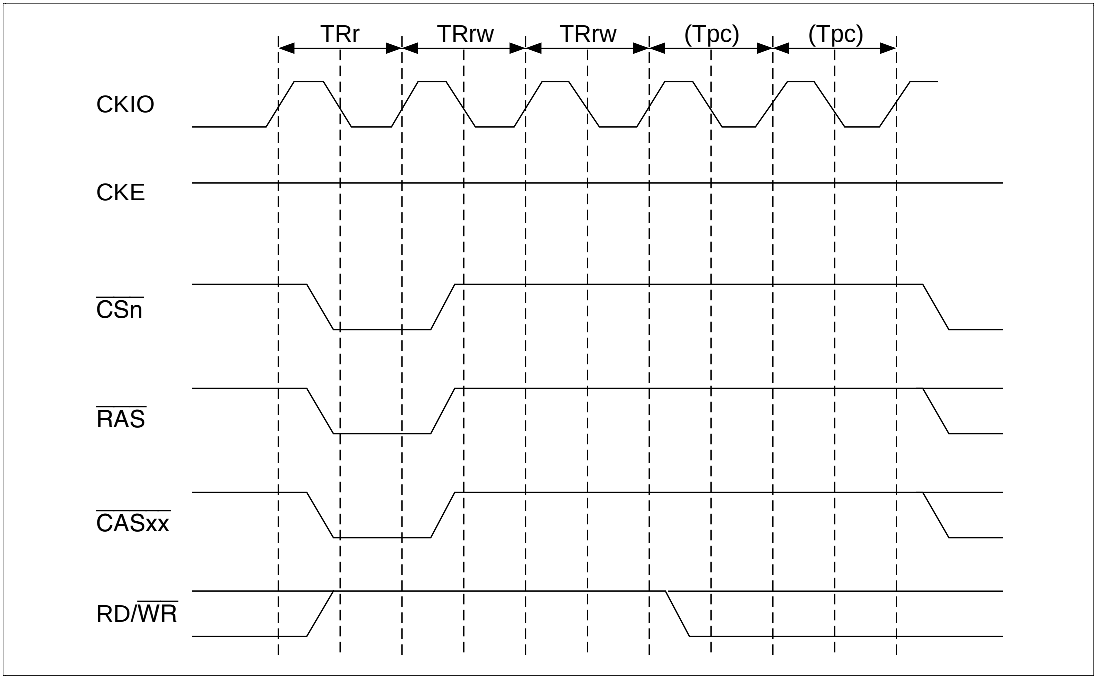


Figure 10.30 Synchronous DRAM Auto-Refresh Timing

2. Self-Refreshing

Self-refresh mode is a kind of standby mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the CKE signal is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TPC bits in MCR. Self-refresh timing is shown in figure 10.31. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the SH7708 Series’s standby function, and is maintained even after recovery from standby mode other than through a power-on reset. For a power-on reset, the bus state controller’s registers are initialized, thereby clearing the self-refresh state.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in a manual reset.

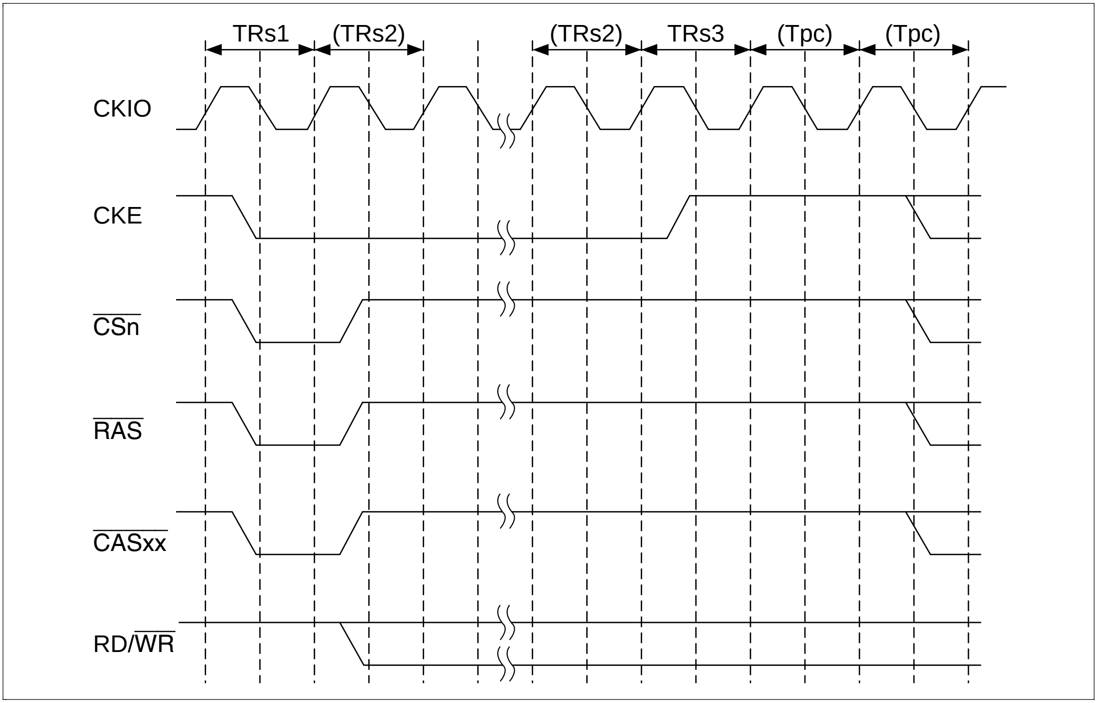


Figure 10.31 Synchronous DRAM Self-Refresh Timing

3. Relationship between refresh requests and bus cycle requests

If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. If a refresh request occurs when the bus has been released by the bus arbiter, refresh execution is deferred until the bus is acquired. If a match between RTCNT and RTCOR occurs while a refresh is waiting to be executed, thereby generating a new refresh request, the previous refresh request is eliminated. To perform normal refreshing, ensure that no bus cycle or bus mastership occurs that is longer than the refresh interval. When a refresh request is generated, the $\overline{\text{IRQOUT}}$ pin is asserted (driven low). Therefore, normal refreshing can be performed by having the $\overline{\text{IRQOUT}}$ pin monitored by a bus master other than the SH7708 Series requesting the bus, or the bus arbiter, and returning the bus to the SH7708 Series. When refreshing is started, and if no other interrupt request has been generated, the $\overline{\text{IRQOUT}}$ pin is negated (driven high).

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'FFFFD000 + X for area 2 synchronous DRAM, and to address H'FFFFE000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/write, CAS latency 1 to 3, wrap type = sequential, and burst length 1 supported by the SH7708 Series, arbitrary data is written in a byte-size access to the following addresses:

	Area 2	Area 3
CAS latency 1	FFFFD840	FFFFE840
CAS latency 2	FFFFD880	FFFFE880
CAS latency 3	FFFFD8C0	FFFFE8C0

Mode register setting timing is shown in figure 10.32.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge all banks (PALL) command is first issued in the TRp1 cycle, then a mode register write command is issued in the TMw1 cycle.

Before mode register setting, a 100 µs idle time (depending on the memory manufacturer) must be guaranteed after powering on requested by the synchronous DRAM. If the reset signal pulse width is greater than this idle time, there is no problem in performing mode register setting immediately. The number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is usually achieved automatically through various initialization methods after auto-refresh setting. However, a more dependable method is to set a short refresh request generation interval just as these dummy cycles are being executed. With simple read or write access, the address counter in the synchronous DRAM used for auto-refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.

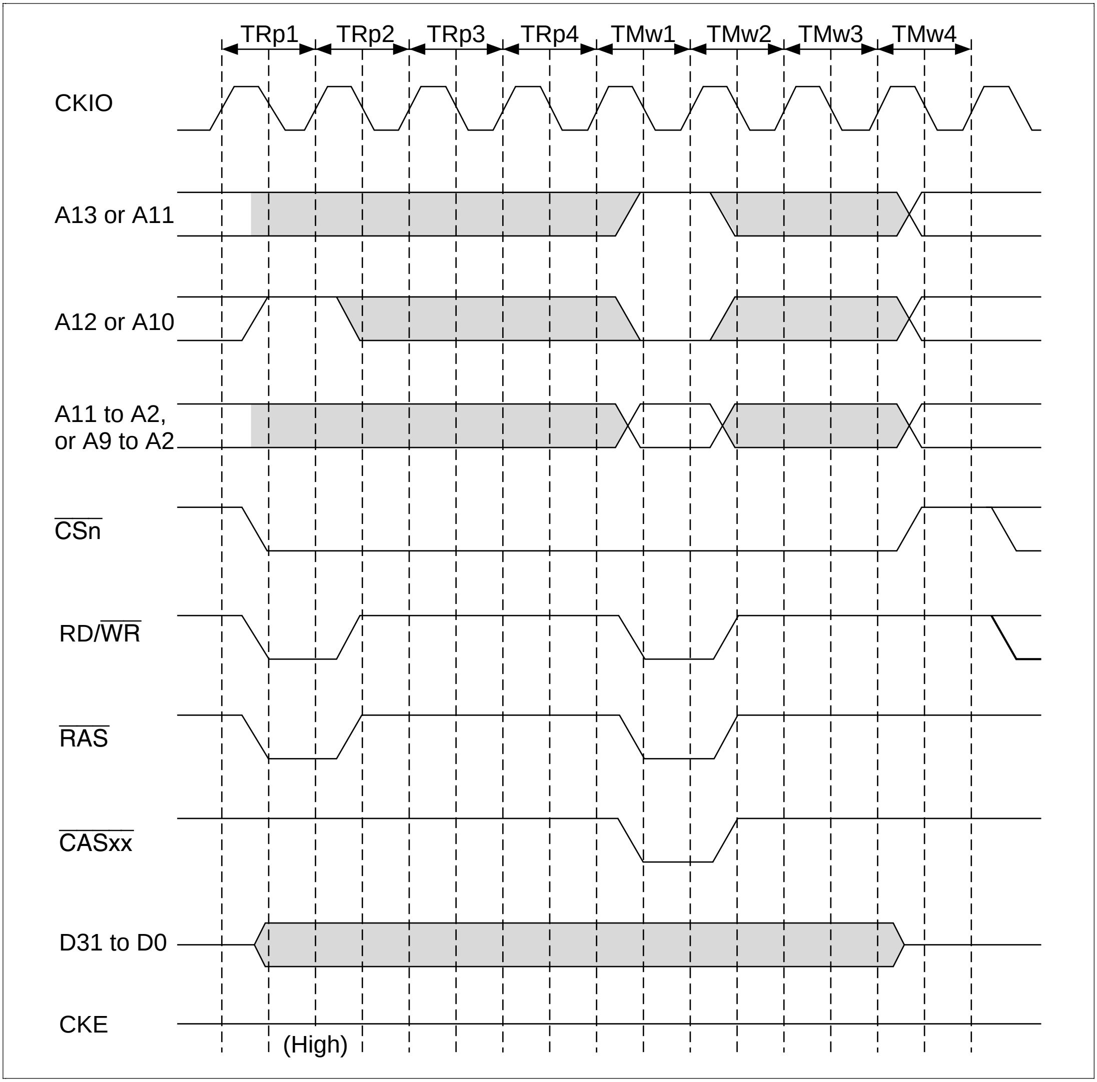


Figure 10.32 Synchronous DRAM Mode Write Timing

10.3.6 Pseudo-SRAM Direct Connection

When the memory type bits (DRAMTP2-0) in BCR1 are set to 001, physical space area 3 becomes pseudo-SRAM and the pseudo-SRAM interface function that allows pseudo-SRAM to be connected directly to the SH7708 Series can be used. An interface data width of 16 or 32 bits can be selected.

With directly connected pseudo-SRAM, the refresh signal and output enable signal are multiplexed. The signals used for connection are $\overline{\text{CE}}$, $\overline{\text{OE}}/\overline{\text{RFSH}}$, $\overline{\text{WE3}}$, $\overline{\text{WE2}}$, $\overline{\text{WE1}}$, and $\overline{\text{WE0}}$. $\overline{\text{WE3}}$ and $\overline{\text{WE2}}$ are not used with a 16-bit data width.

As access modes, burst access using the static column access function is supported in addition to ordinary read/write access.

Figure 10.33 shows an example of connection of 4M pseudo-SRAMs with multiplexed $\overline{\text{OE}}$ and $\overline{\text{RFSH}}$ signals, using a 32-bit data width.

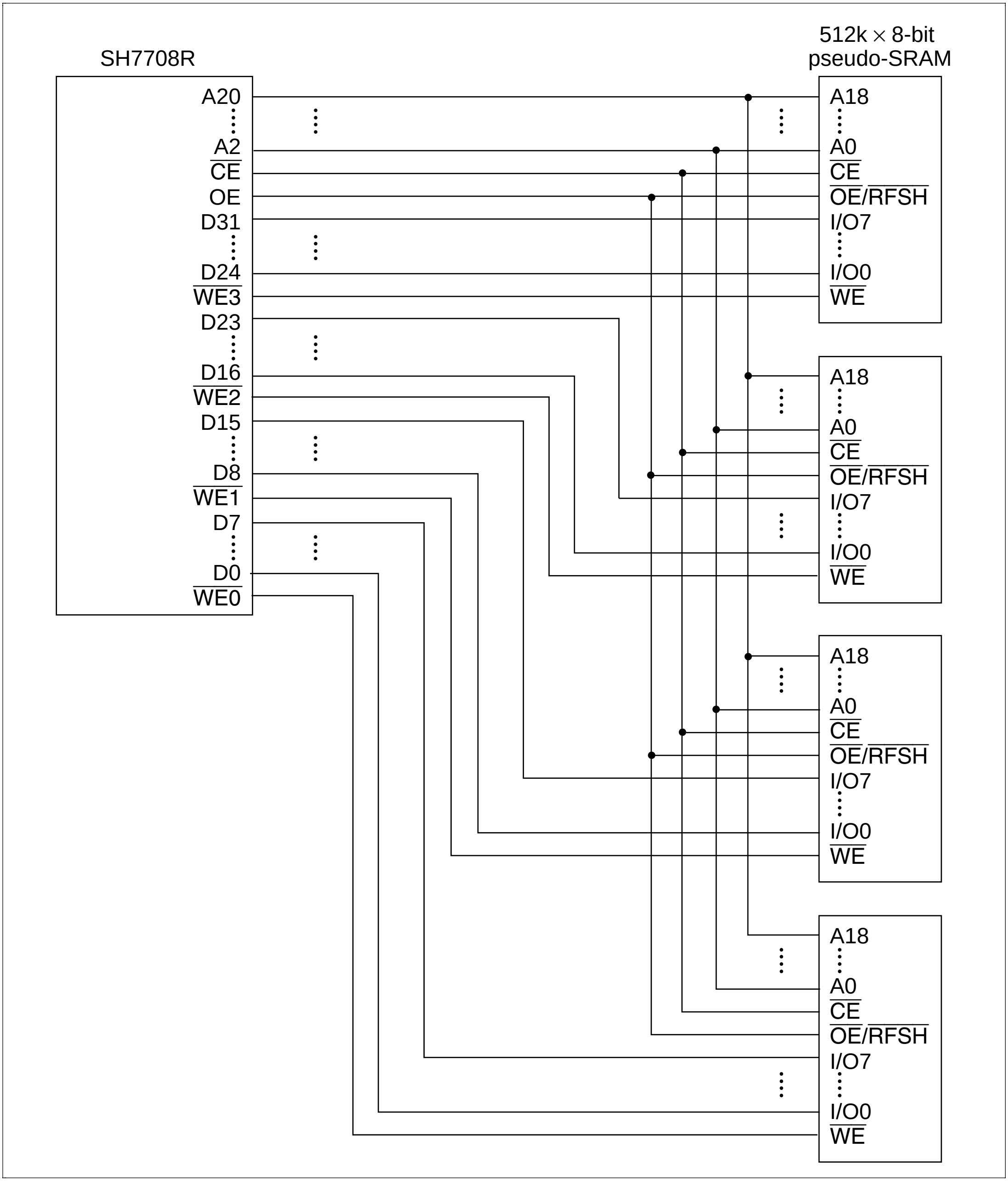


Figure 10.33 Example of Pseudo-SRAM Connection (4M-Bit Devices)

Basic Timing: Figure 10.34 shows the basic timing for pseudo-SRAM. Tpc is the precharge cycle, and Tr is the $\overline{\text{CE}}$ assert cycle. Tc1 is the write data cycle, $\overline{\text{BS}}$ the assert cycle, and Tc2 the read data latch cycle.

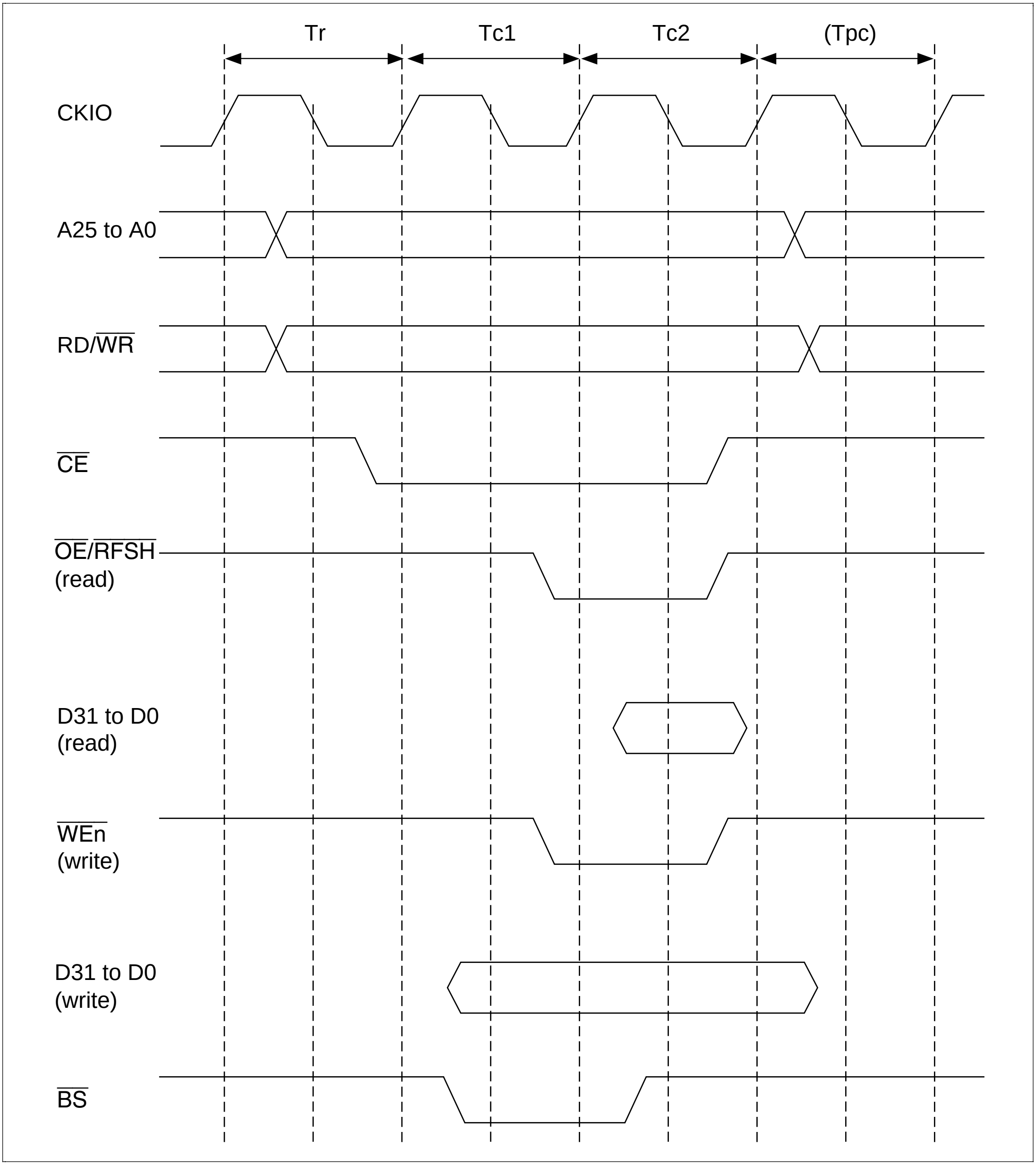


Figure 10.34 Basic Access Timing for Pseudo-SRAM

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 and MCR. The timing with state extension using these settings is shown in figure 10.35. Additional Tpc cycles (cycles used to secure the $\overline{\text{CE}}$ precharge time) can be inserted by means of the TPC bits in MCR. The number of $\overline{\text{OE}}$ and $\overline{\text{WEn}}$ assert cycles from RAS assertion to CAS assertion can be varied between 1 and 3 according to the setting of A3W1 and A3W0 in WCR2. Trw cycles can be inserted by means of the RCD bits in MCR, and the number of cycles from $\overline{\text{CE}}$ assertion to $\overline{\text{BS}}$ assertion and write data output can be varied between 1 and 4.

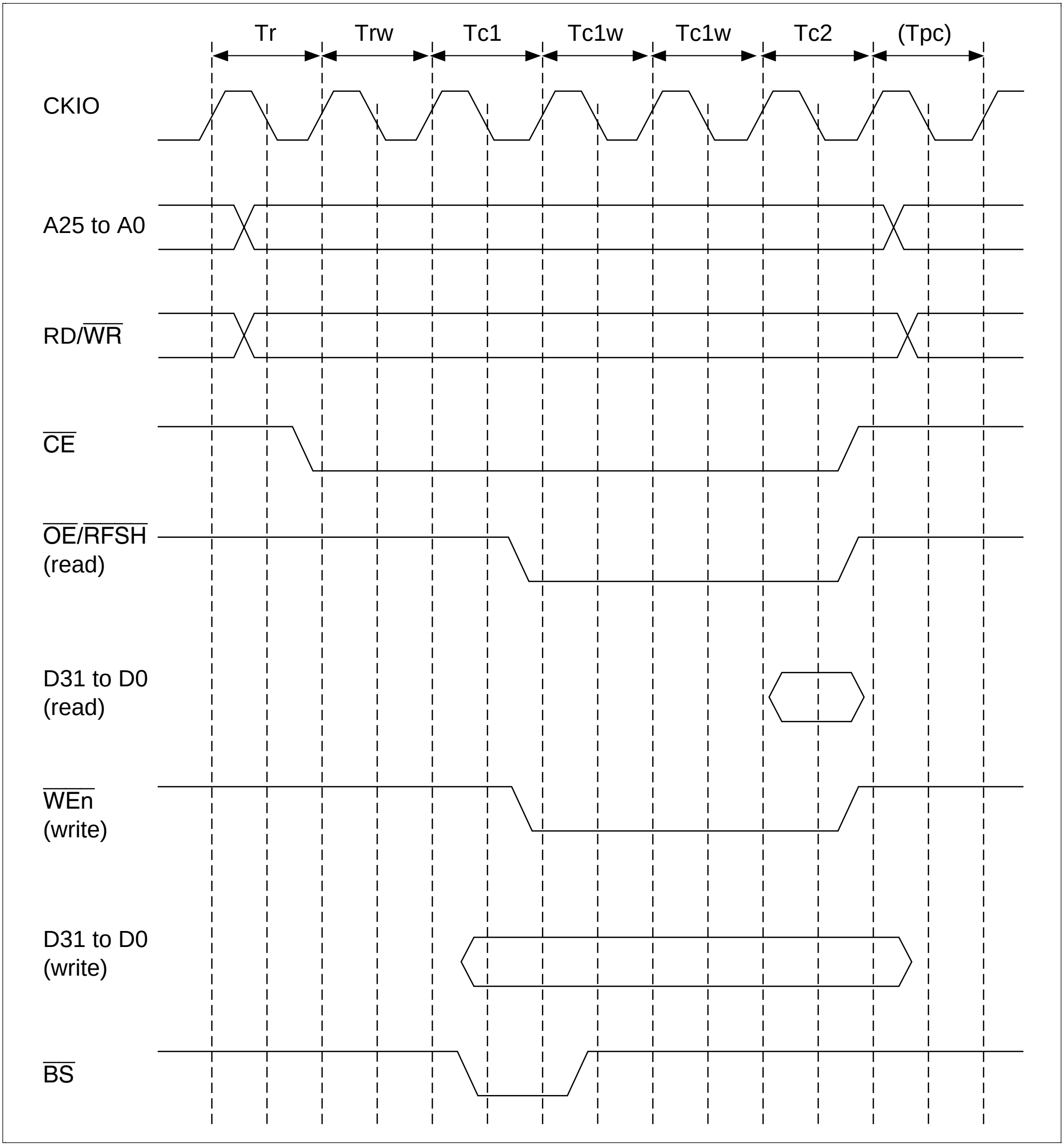


Figure 10.35 Pseudo-SRAM Wait State Timing

Burst Access: In addition to the normal access mode in which $\overline{\text{CE}}$ is asserted and negated in each access, some pseudo-SRAMs are provided with a static column mode for the case where consecutive accesses are made to the same row address. This mode allows fast access to data by keeping $\overline{\text{CE}}$ asserted and changing only the column address. Normal access or burst access using static column mode can be selected by means of the burst enable (BE) bit in MCR. The timing for burst access in static column mode is shown in figure 10.36. Cycles can also be inserted by the wait state control function when burst access is performed.

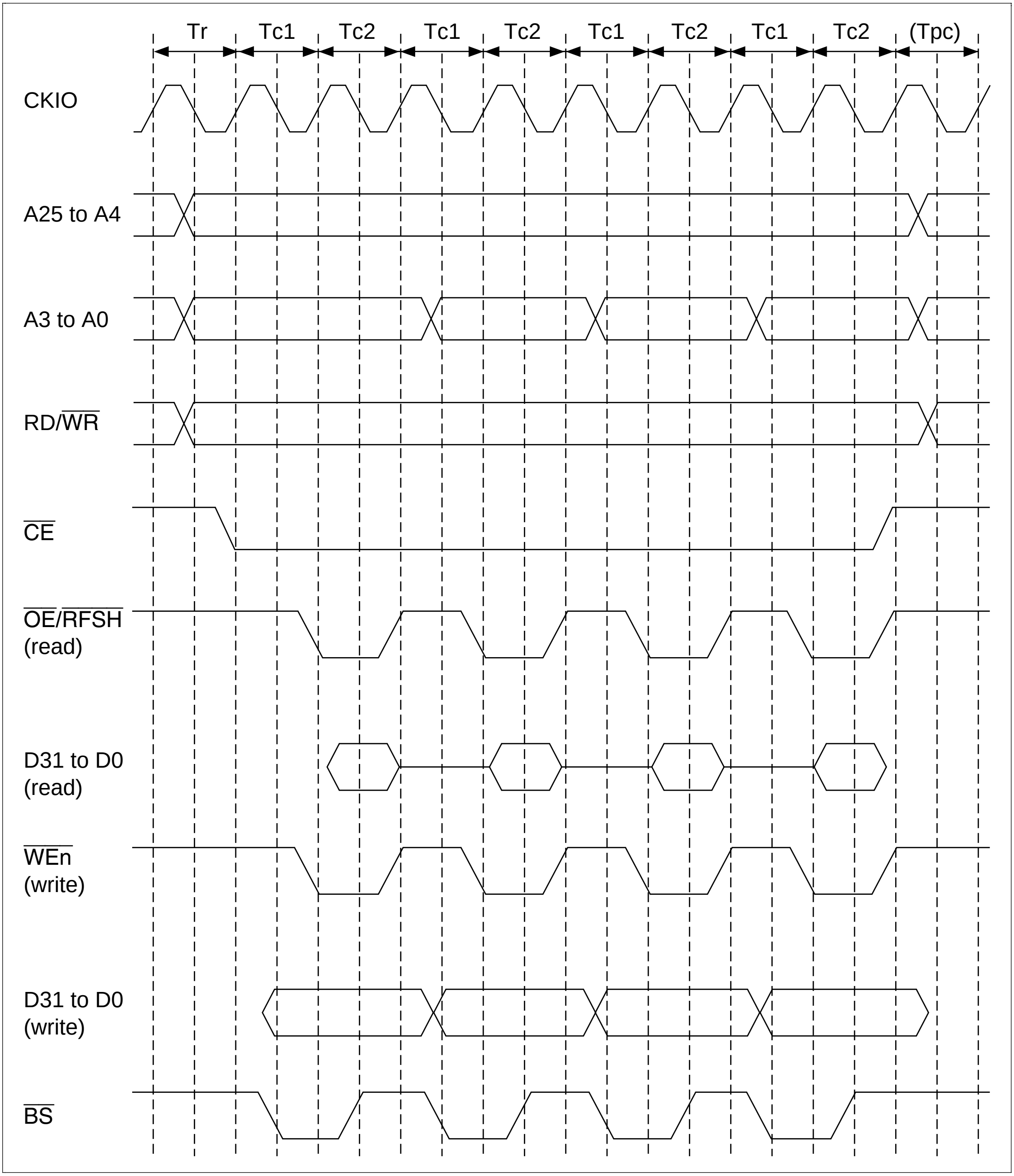


Figure 10.36 Pseudo-SRAM Static Column Mode

Refreshing: The bus state controller includes a function for controlling pseudo-SRAM refreshing. Distributed refreshing by means of auto-refresh cycles can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR.

Refreshing is performed at intervals determined by the input clock selected by bits CKS2–CKS0 in RTCSR, and the value set in RTCOR. The value of bits CKS2–CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the pseudo-SRAM used. First set the RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then set the CKS2–CKS0. When the clock is selected by CKS2–CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 10.37 shows the auto-refresh cycle timing.

The number of $\overline{\text{OE}}$ assert cycles for auto-refreshing is specified by the TRAS bits in MCR. The precharge time from $\overline{\text{OE}}$ negation until the next assertion of $\overline{\text{CE}}$ is determined by the setting of the TPC bits in MCR.

Auto-refreshing is performed in normal operation, in sleep mode, and in a manual reset.

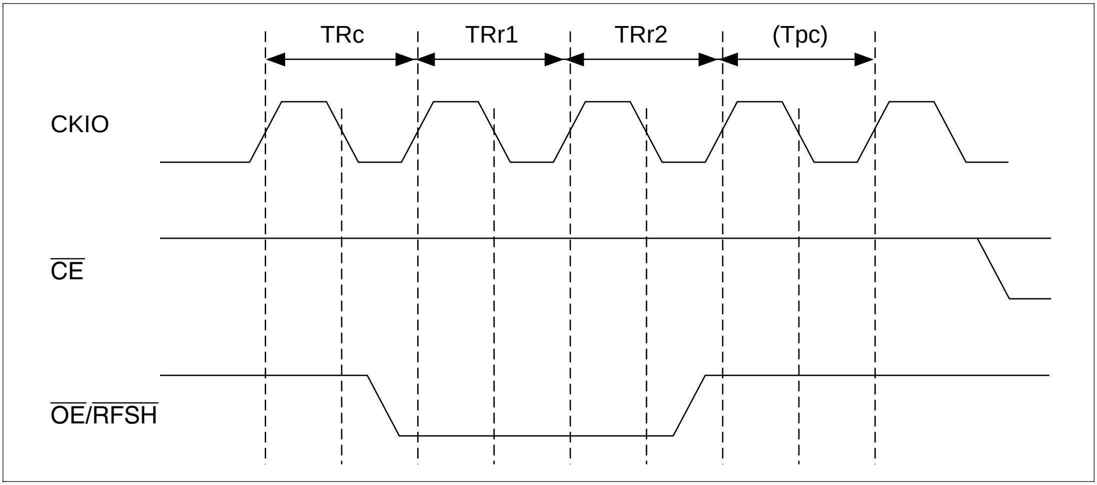


Figure 10.37 Pseudo-SRAM Auto-Refreshing

With pseudo-SRAM, self-refresh mode is entered by holding the $\overline{\text{RFSH}}$ signal low for at least the prescribed time. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the CKE signal is low. Pseudo-SRAM cannot be accessed while in self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, access to pseudo-SRAM is disabled for the number of cycles specified by the TPC bits in MCR, but if the refresh reset time needed to return from self-refreshing is longer than this interval, coding must be provided to ensure that no access—including auto-refresh—is made to pseudo-SRAM. Self-refresh timing is shown in figure 10.38. Settings must be made so that self-refresh clearing and data retention is performed correctly after self-refresh mode is cleared, and auto-refreshing is performed at the correct intervals. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in a manual reset.

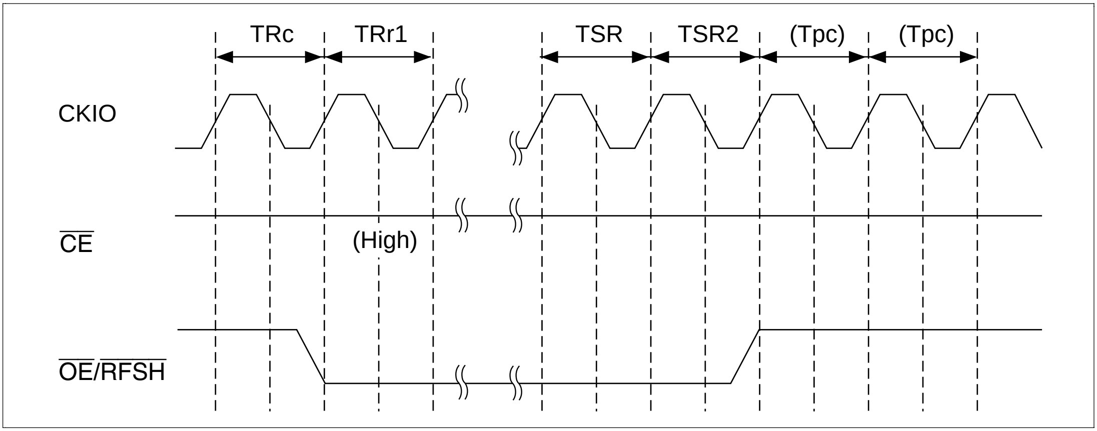


Figure 10.38   Pseudo-SRAM Self-Refreshing

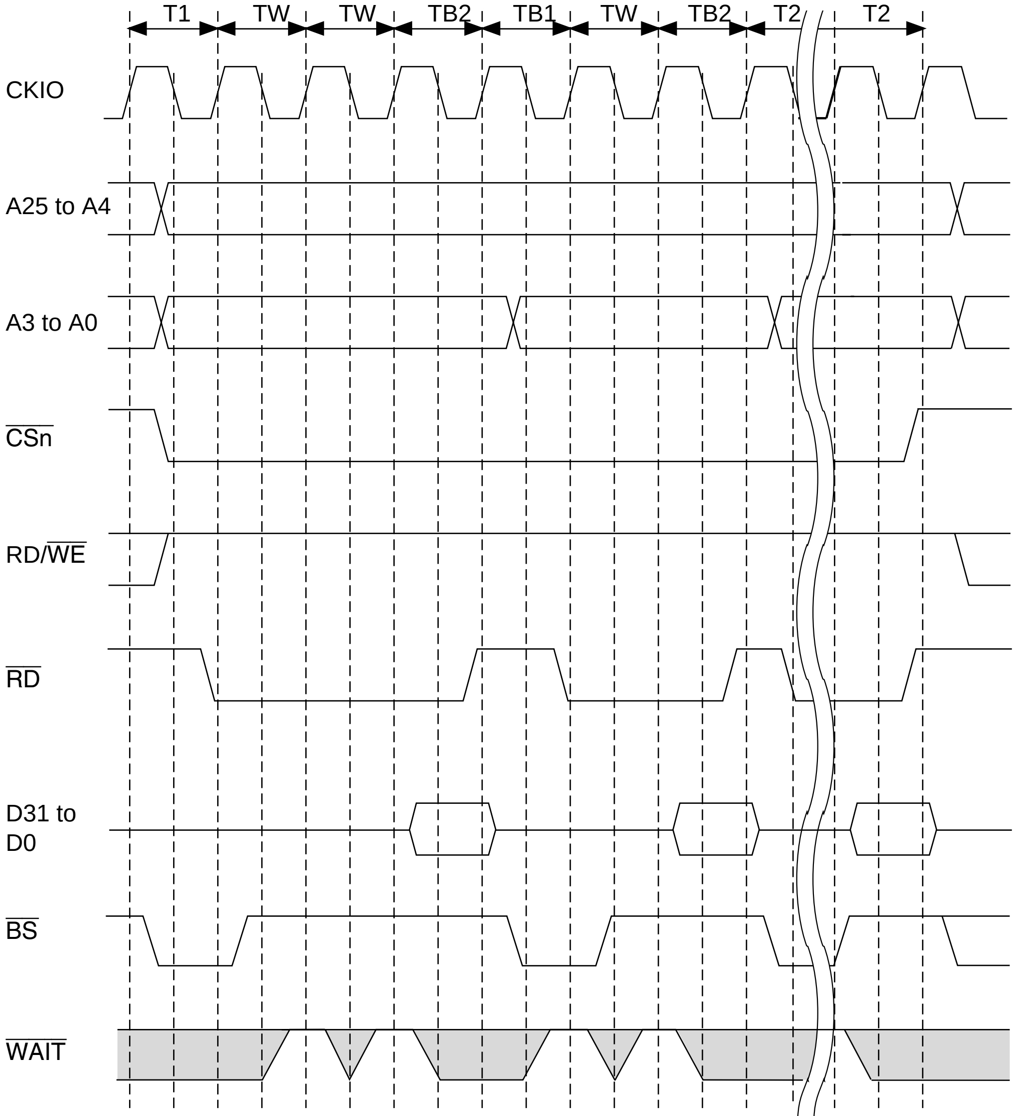
Power-On Sequence: After powering pseudo-SRAM on, a minimum wait time of 100 µs is requested during which no access can be performed, followed by the prescribed number (usually 8 or more) of dummy auto-refresh cycles. As the bus state controller does not perform any special operations for a power-on reset, the power-on sequence must be carried out by the initialization program executed after a power-on reset.

10.3.7 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 10.39. Two wait cycles are set. Basically, access is performed in the same way as for normal space, but when the first cycle ends, the $\overline{\text{CS0}}$ 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.

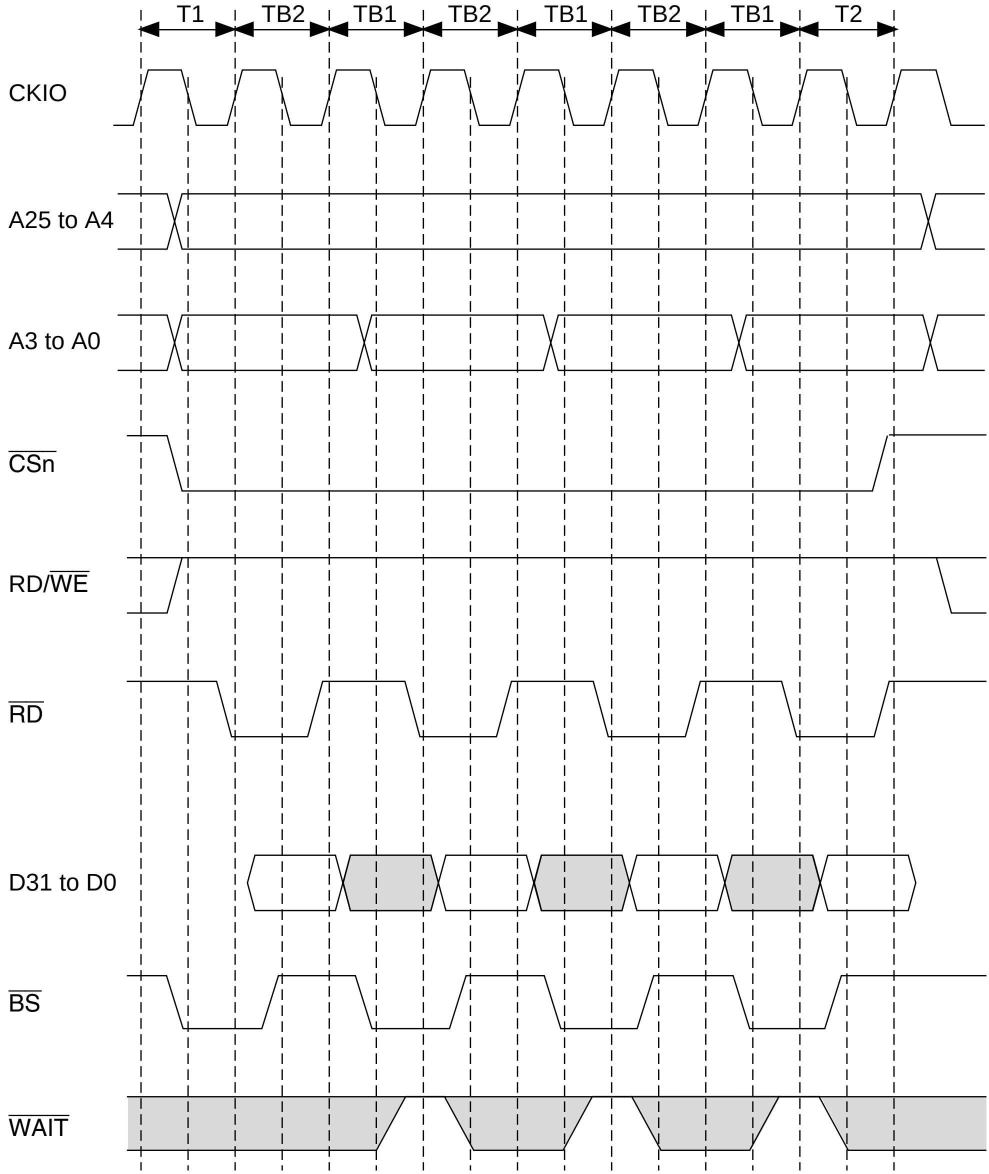
$\overline{\text{WAIT}}$ pin sampling is performed in the first access if one or more wait states are set, and is always performed in the second and subsequent accesses. $\overline{\text{WAIT}}$ is a synchronous signal.

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 10.40.



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

Figure 10.39 Burst ROM Wait Access Timing



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

Figure 10.40 Burst ROM Basic Access Timing

10.3.8 PCMCIA Interface

In the SH7708 Series, setting the A5PCM bit in BCR1 to 1 makes the bus interface for physical space area 5 an IC memory card interface as stipulated in JEIDA version 4.2 (PCMCIA2.1). Setting the A6PCM bit to 1 makes the bus interface for physical space area 6 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2. When the IC memory card interface is selected, a BCR1 register setting enables page mode burst access mode to be used. This burst access mode is not stipulated in JEIDA version 4.2, but allows high-speed data access using ROM provided with a burst mode, etc.

When the PCMCIA interface is used, a bus size of 8 or 16 bits can be set by bits A5SZ1 and A5SZ0, or A6SZ1 and A6SZ0, in BCR2.

Figure 10.41 shows an example of PCMCIA card connection to the SH7708 Series. To enable active insertion of the PCMCIA cards (i.e. insertion or removal while system power is being supplied), a 3-state buffer must be connected between the SH7708 Series’s bus interface and the PCMCIA cards.

As operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA specifications, the PCMCIA interface for the SH7708 Series in big-endian mode is stipulated independently.

If PCMCIA is used in area 6, synchronous DRAM is used at the same time, and a synchronous DRAM auto-refresh (CAS-before-RAS refresh) request is issued simultaneously, area 6 card enable signals $\overline{\text{CS6}}$ and $\overline{\text{CE2B}}$ may be asserted earlier than usual, at the same time as the immediately preceding auto-refresh cycle. When both PCMCIA and synchronous DRAM are used, they should be used in area 5. When area 5 is used, the system design should provide for CS to be asserted early without causing any problems.

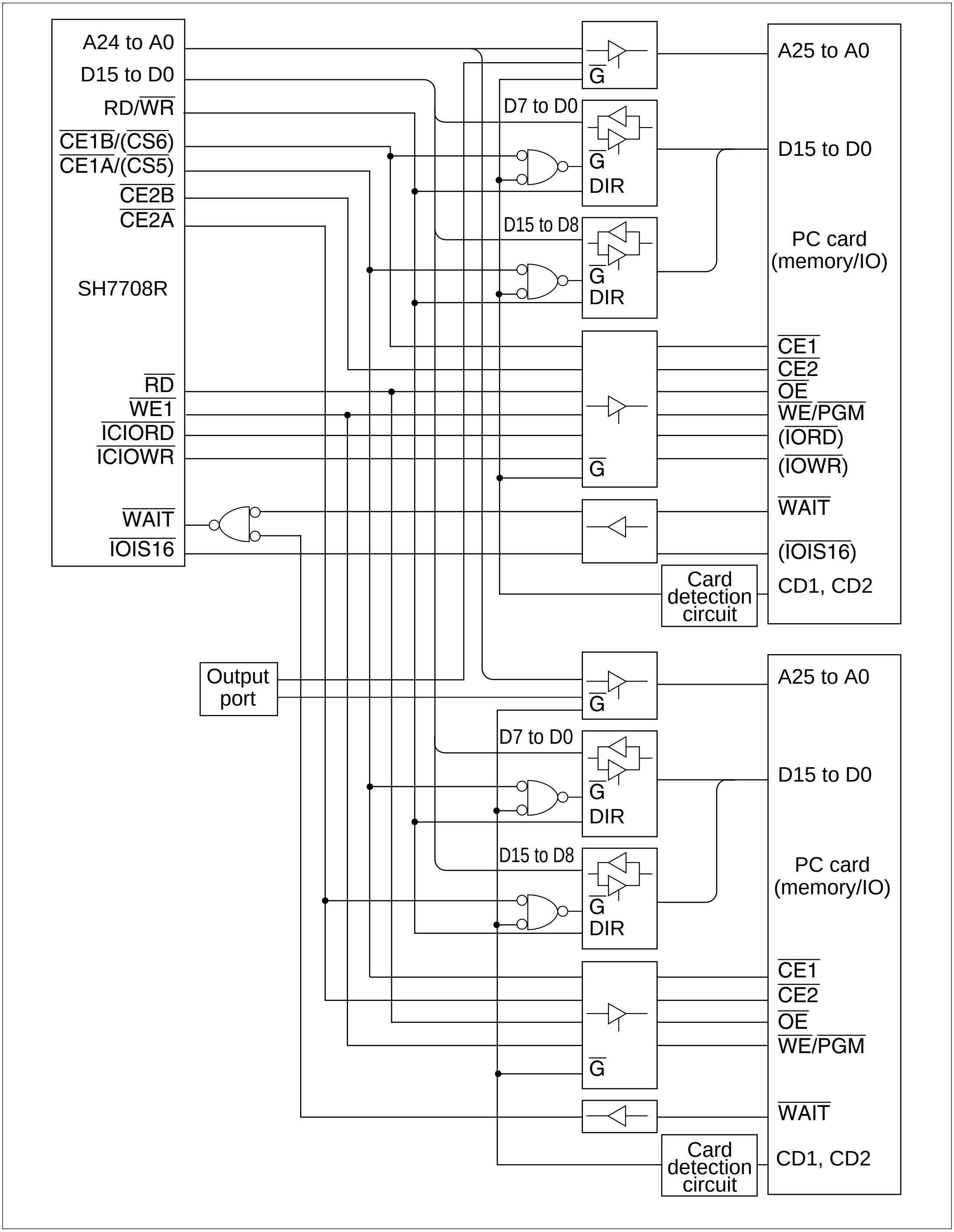


Figure 10.41 Example of PCMCIA Interface

Memory Card Interface Basic Timing: Figure 10.42 shows the basic timing for the PCMCIA IC memory card interface. When physical space areas 5 and 6 are designated as PCMCIA interface areas, bus accesses are automatically performed as IC memory card interface accesses when the lower address 32 Mbyte space of each area is accessed.

With a high external bus frequency (CKIO), the setup and hold times for the address (A24–A0), card enable ($\overline{\text{CS5}}$, $\overline{\text{CE2A}}$, $\overline{\text{CS6}}$, $\overline{\text{CE2B}}$), and write data (D15–D0) in a write cycle, become insufficient with respect to $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (the $\overline{\text{WE1}}$ pin in the SH7708 Series). The SH7708 Series provides for this by enabling setup and hold times to be set for physical space areas 5 and 6 in the PCR register. Also, software waits by means of a WCR2 register setting and hardware waits by means of the $\overline{\text{WAIT}}$ pin can be inserted in the same way as for the basic interface. $\overline{\text{WAIT}}$ is a synchronous signal. Figure 10.43 shows the PCMCIA memory bus wait timing.

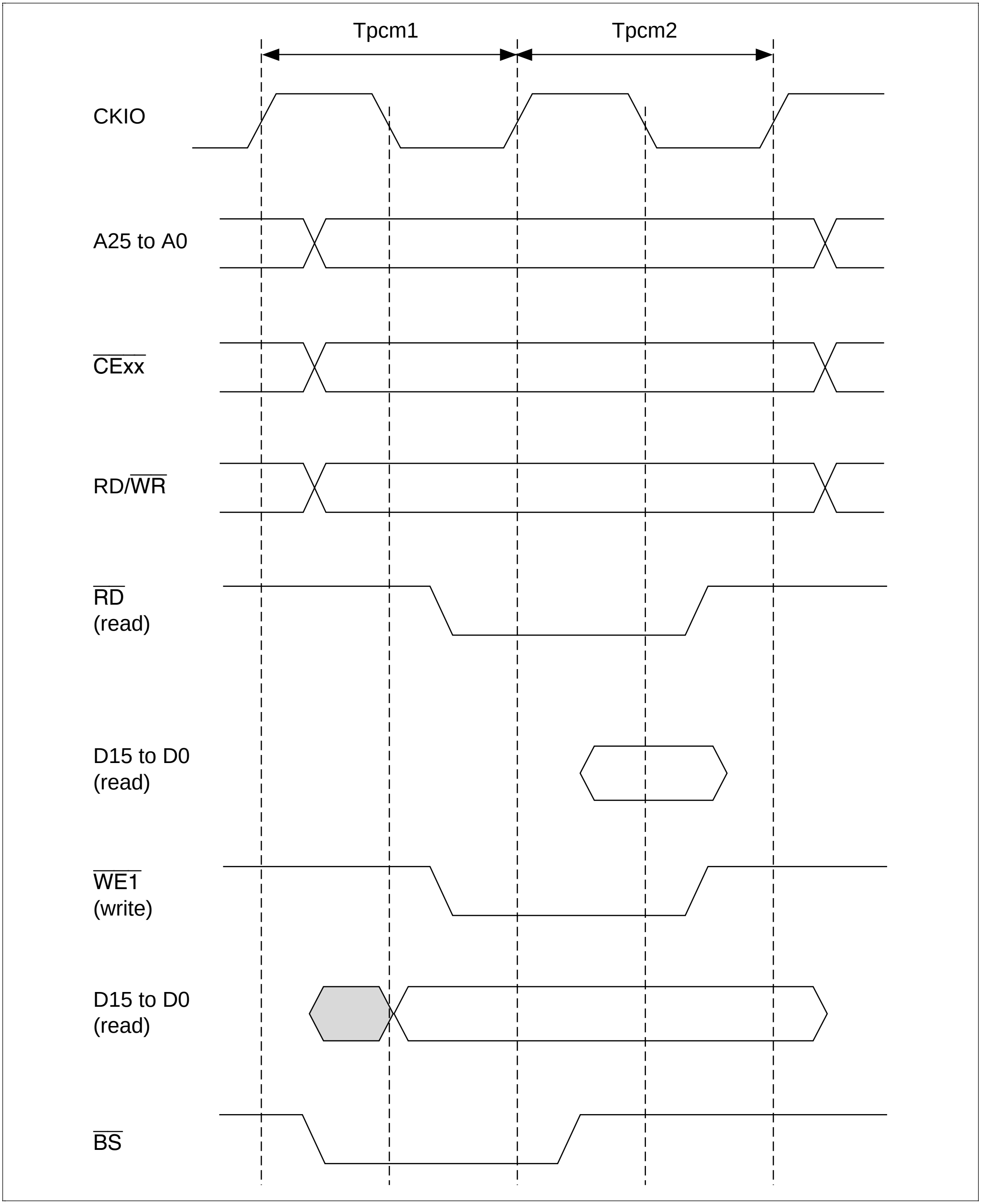


Figure 10.42 Basic Timing for PCMCIA Memory Card Interface

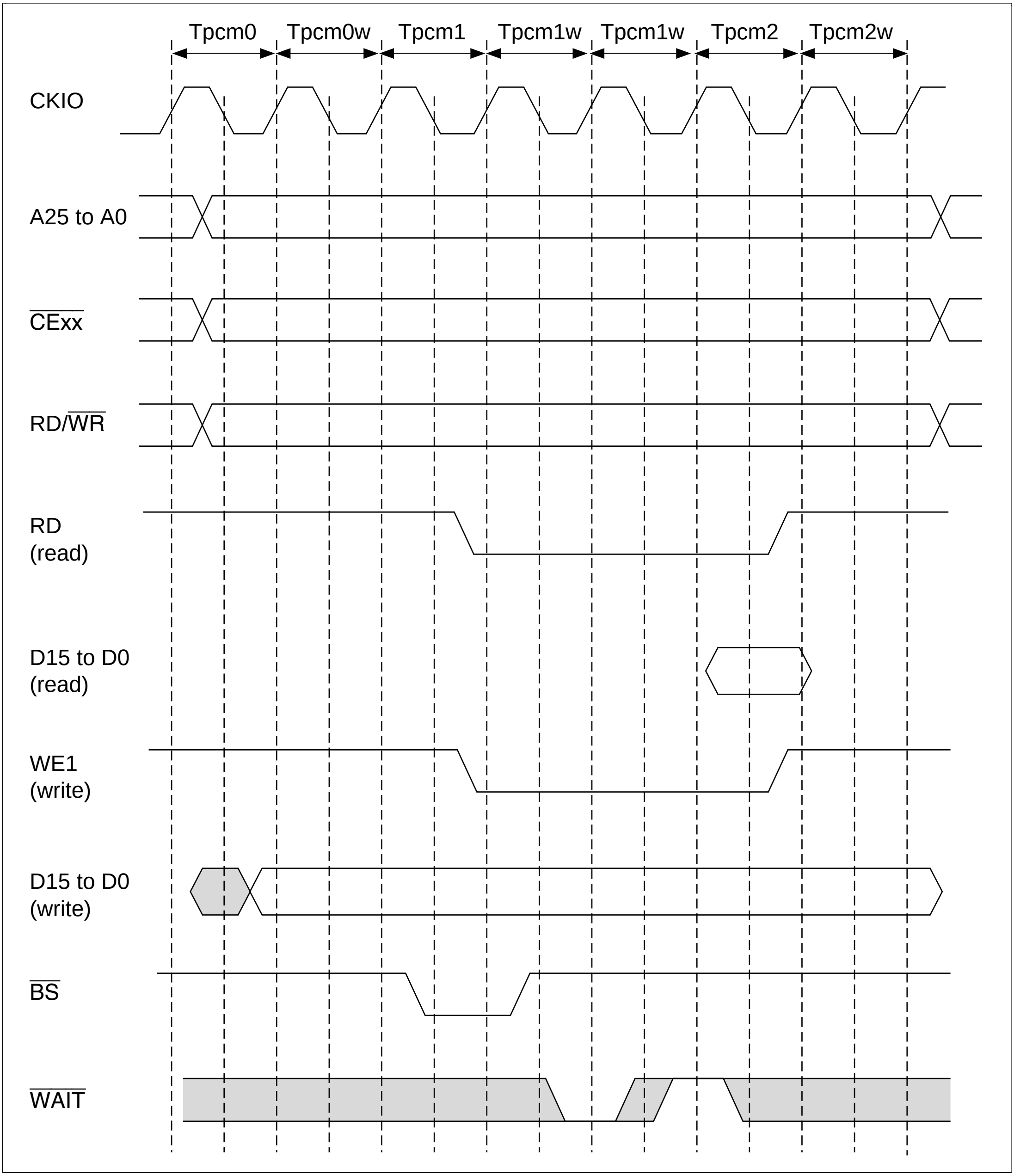


Figure 10.43 Wait Timing for PCMCIA Memory Card Interface

Memory Card Interface Burst Timing: In the SH7708 Series, when the IC memory card interface is selected, page mode burst access mode can be used, for read access only, by setting bits A5BST1 and A5BST0 in BCR for physical space area 5, or bits A6BST1 and A6BST0 for area 6. This burst access mode is not stipulated in JEIDA version 4.2 (PCMCIA2.1), but allows high-speed data access using ROM provided with a burst mode, etc.

Burst access mode timing is shown in figures 10.44 and 10.45.

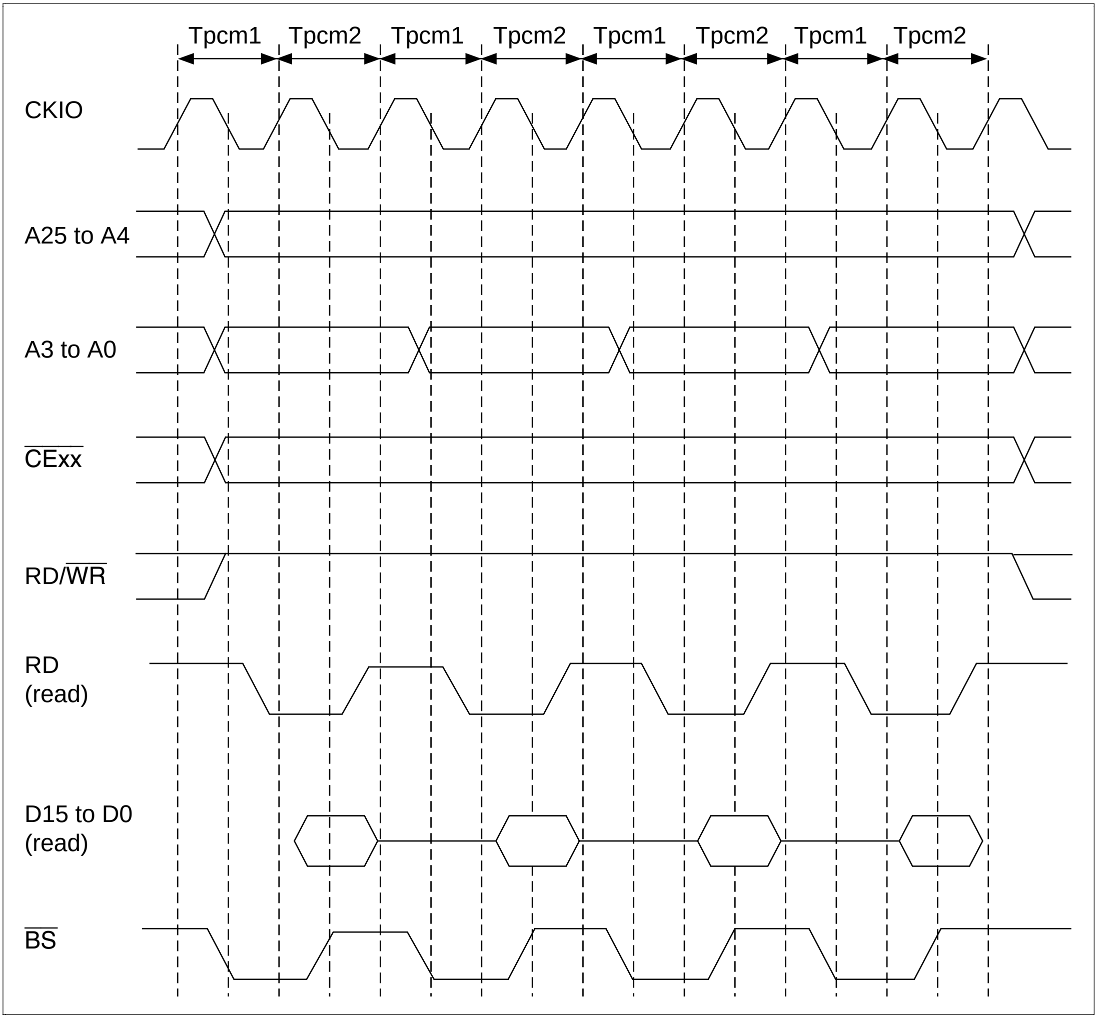


Figure 10.44 Basic Timing for PCMCIA Memory Card Interface Burst Access

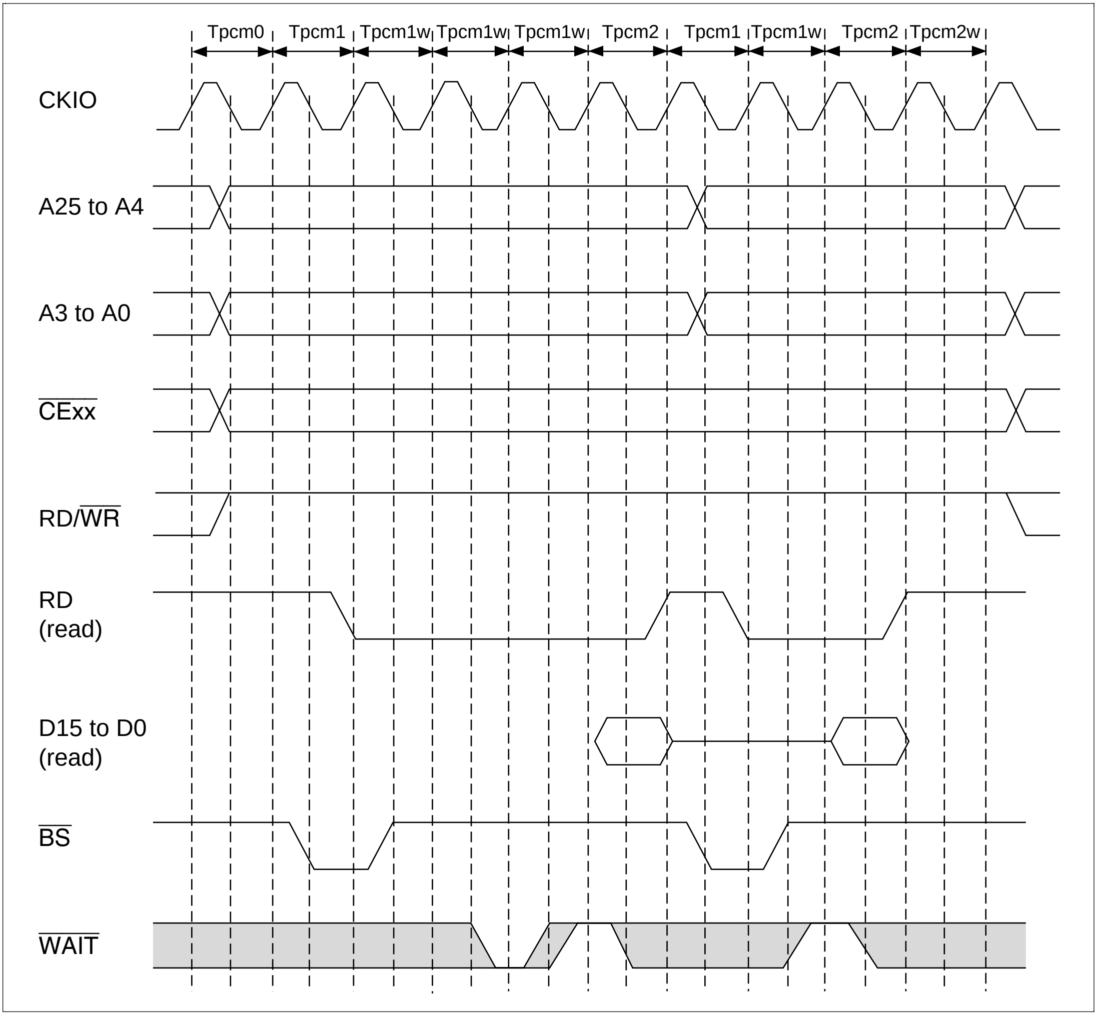


Figure 10.45 Wait Timing for PCMCIA Memory Card Interface Burst Access

When the entire 32-Mbyte memory space is used as IC memory card interface space, the common memory/attribute memory switching signal $\overline{\text{REG}}$ is generated using a port, etc. If 16-Mbytes or less of memory space is sufficient, using 16M bytes of memory space as common memory space and 16 Mbytes as attribute memory space enables the A24 pin to be used for the $\overline{\text{REG}}$ signal.

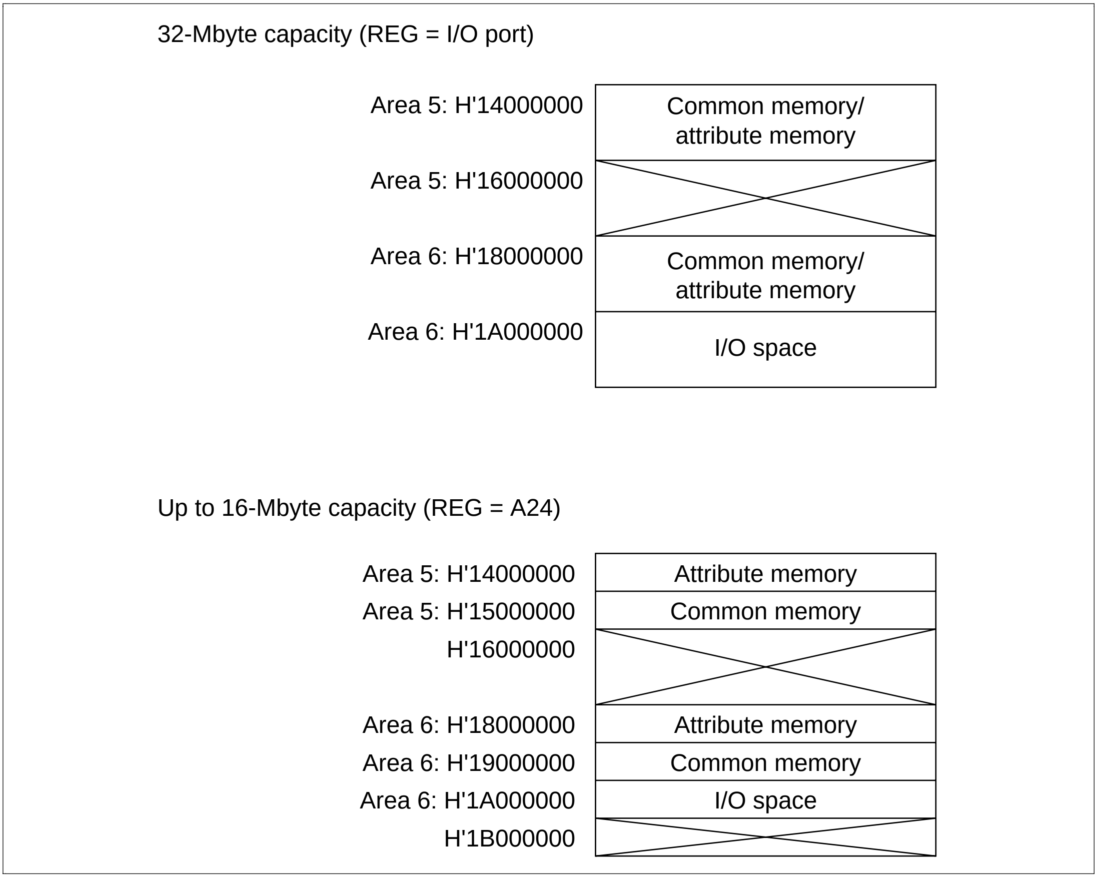


Figure 10.46 PCMCIA Space Allocation

I/O Card Interface Timing: Figures 10.47 and 10.48 show the timing for the PCMCIA I/O card interface.

The I/O card interface is supported only for physical space area 6. Switching between the I/O card interface and the IC memory card interface is performed according to the accessed address. When PCMCIA is designated for physical space area 6, the bus access is automatically performed as an I/O card interface access when a physical address from H'1A000000 to H'1BFFFFFF is accessed.

When accessing a PCMCIA I/O card, the access should be performed using a noncacheable area in virtual space (P2 or P3 space) or an area specified as noncacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, dynamic sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width is set for area 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port is recognized as being 8 bits in width. In this case, a data access for only 8 bits is performed in the I/O bus cycle being executed, followed automatically by a data access for the remaining 8 bits.

Figure 10.49 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed low.

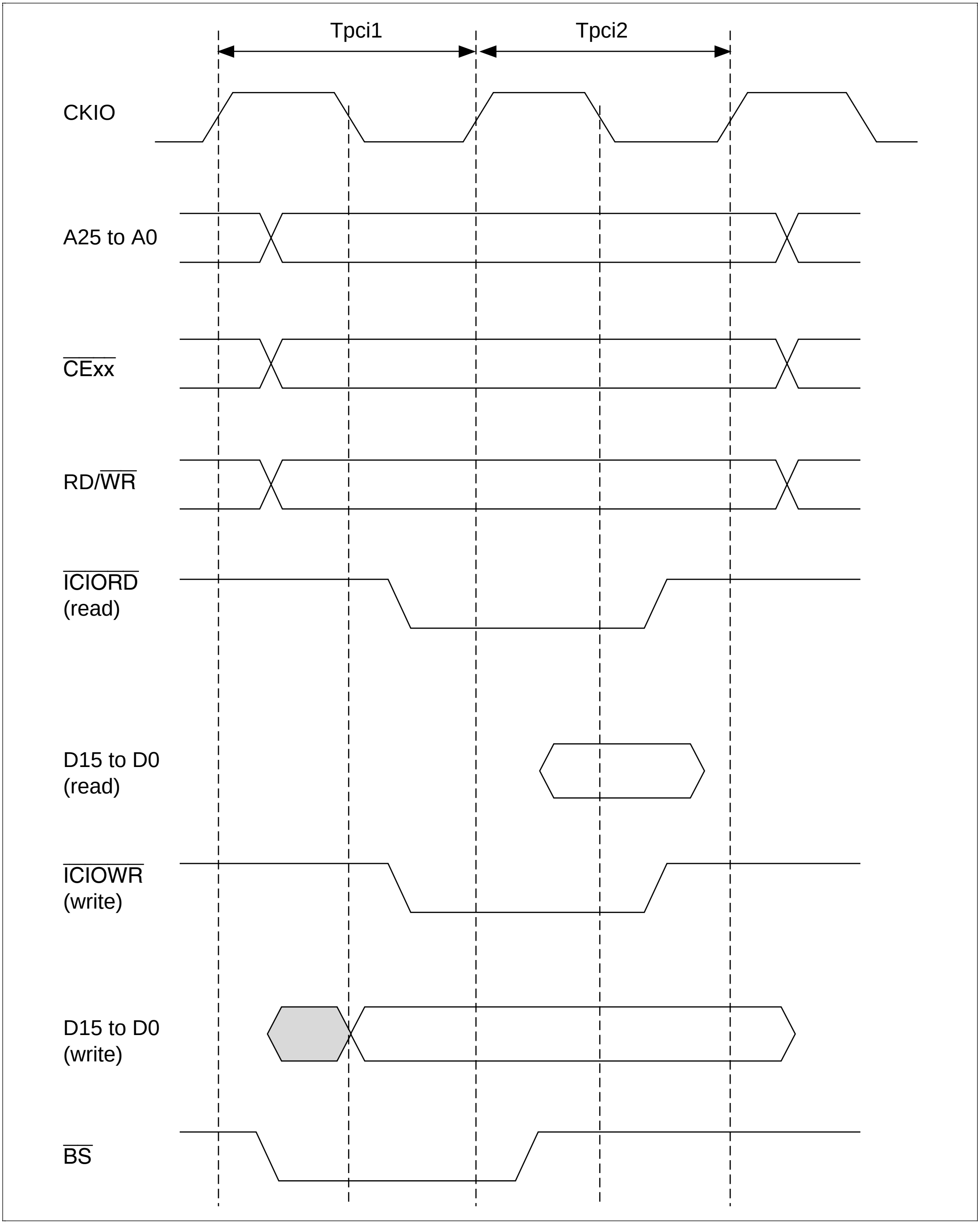


Figure 10.47 Basic Timing for PCMCIA I/O Card Interface

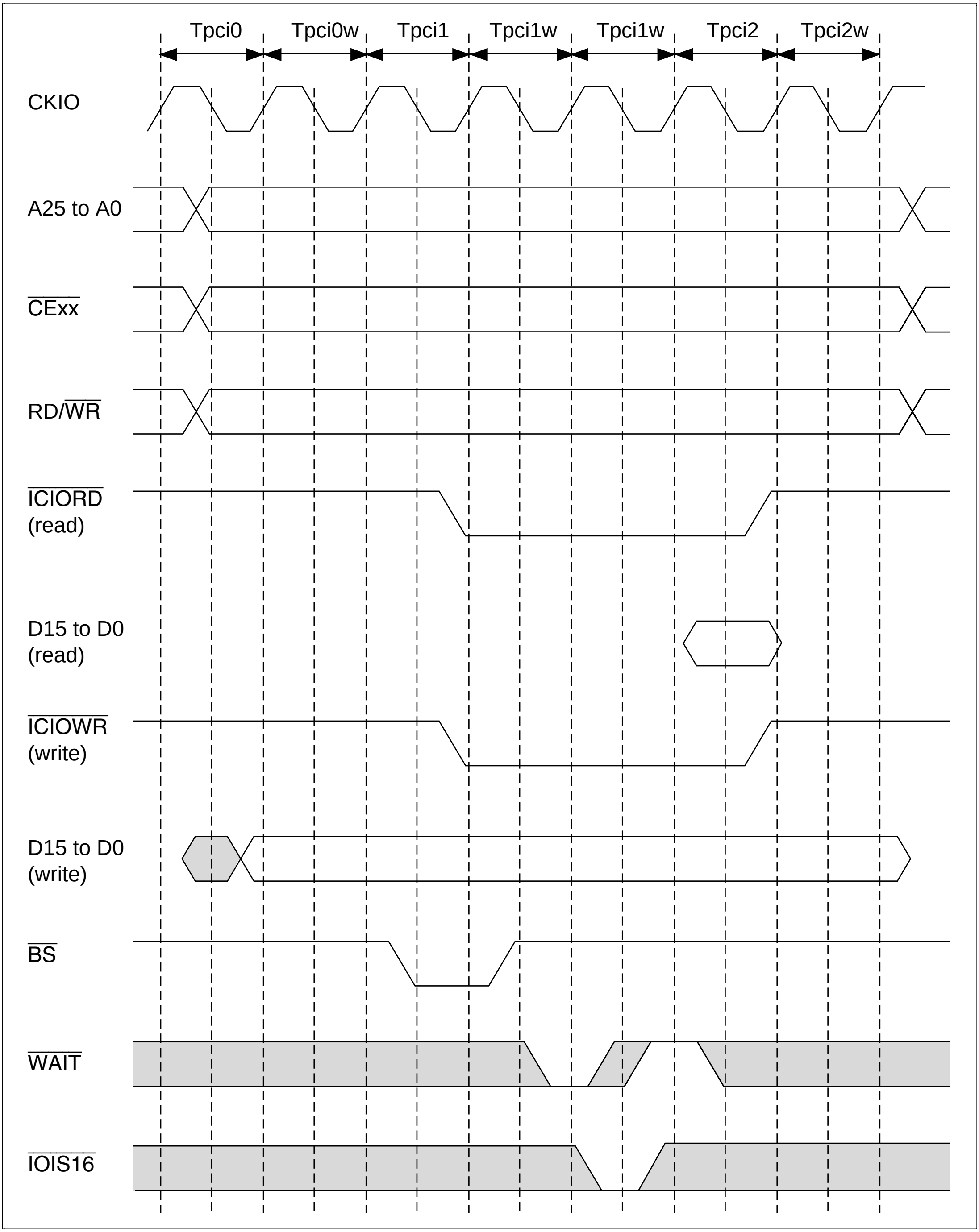


Figure 10.48 Wait Timing for PCMCIA I/O Card Interface

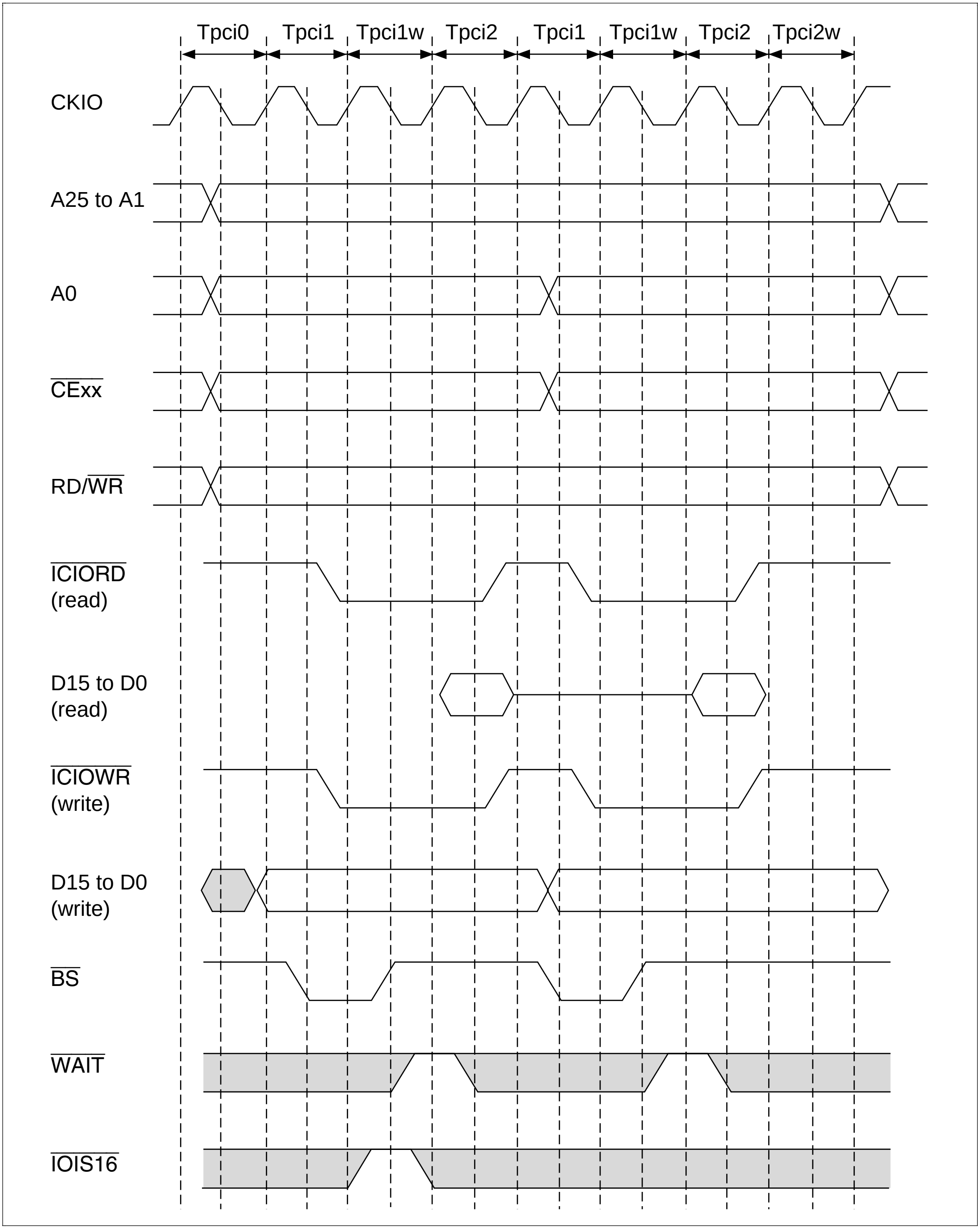


Figure 10.49 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

10.3.9 Waits between Access Cycles

A problem associated with higher external memory bus operating frequencies is that data buffer turn-off on completion of a read from a low-speed device may be too slow, causing a collision with data in the next access. This results in lower reliability or incorrect operation. To avoid this problem, a data collision prevention feature has been provided. This memorizes the preceding access area and the kind of read/write. If there is a possibility of a bus collision when the next access is started, a wait cycle is inserted before the access cycle thus preventing a data collision. There are two cases in which a wait cycle is inserted: when an access is followed by an access to a different area, and when a read access is followed by a write access from the SH7708 Series. When the SH7708 Series performs consecutive write cycles, the data transfer direction is fixed (from the SH7708 Series to other memory) and there is no problem. With read accesses to the same area, in principle, data is output from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and AnIW0 (n = 0–6) in WCR1 specify the number of idle cycles to be inserted between access cycles when a physical space area access is followed by an access to another area, or when the SH7708 Series performs a write access after a read access to physical space area n. If there is originally space between accesses, the number of idle cycles inserted is the specified number of idle cycles minus the number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empty cycles are inserted for arbitration purposes.

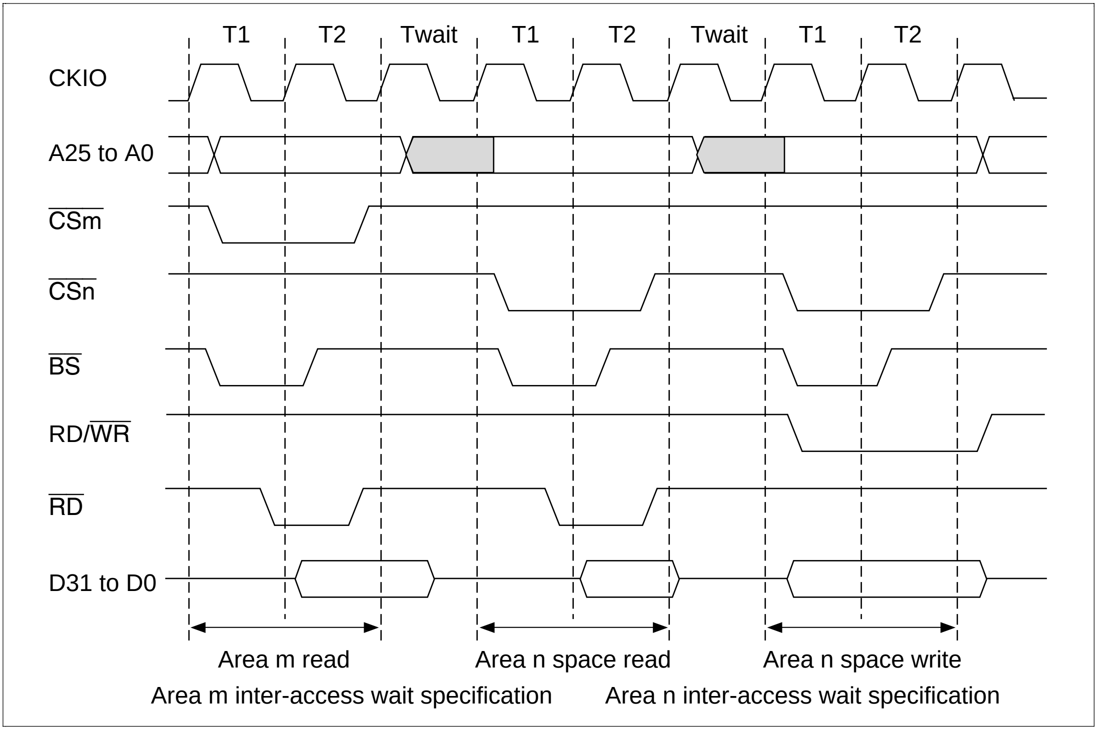


Figure 10.50 Waits between Access Cycles

10.3.10 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is received from an external device, buses are 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. In the case of multiple bus cycles generated because the data bus width is smaller than the access size—for example, in longword access to 8-bit-wide memory—bus arbitration is not performed between bus cycles. 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 status when the bus is released.

The SH7708 Series sometimes needs to retrieve a bus it has released. For example, when memory generates a refresh request or an interrupt request internally, the SH7708 Series must perform the appropriate processing. The SH7708 Series 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 SH7708 Series retrieves the bus and carries out the processing.

$\overline{\text{IRQOUT}}$ Pin Assertion Conditions:

- When a memory refresh request has been generated but the refresh cycle has not yet begun
- When an interrupt is generated with an interrupt request level higher than the setting of the interrupt mask bits (I3–I0) in the status register (SR). (This does not depend on the SR.BL bit.)

10.4 Usage Notes

10.4.1 Area 2 and 3 Bus Width Setting

Note the following concerning the SH7708 (not including the SH7708S and SH7708R).

When SDRAM, DRAM, or PSRAM is connected, cases such as the following can be envisaged in which the contents of register BCR2, which sets the bus width of each area, and the MCR.SZ bits, which set the bus width of SDRAM, DRAM, and PSRAM, are different.

- A setting other than 32 bits in the BCR2 register and MCR register when SDRAM is connected
- A setting of 16 bits in the BCR2 register and 32 bits in the MCR register when DRAM is connected in area 3 only
- A setting of 32 bits in the BCR2 register and 16 bits in the MCR register when DRAM is connected in area 3 only
- A setting other than 16 bits in the BCR2 register and MCR register when DRAM is connected in areas 2 and 3
- Different bus width settings in the BCR register and MCR register when PSRAM is connected

In these cases, one of the following settings should be made.

- When SDRAM is connected, use a 32-bit bus width setting in the BCR2 register and MCR register.
- When DRAM is connected in area 3 only, use the same bus width setting (16 or 32 bits) in the BCR2 register and MCR register.
- When DRAM is connected in areas 2 and 3, use a 16-bit bus width setting in the BCR2 register and MCR register.
- When PSRAM is connected, use the same bus width setting (16 or 32 bits) in the BCR2 register and MCR register.

10.4.2 When Area 6 is Designated for PCMCIA, with a 16-Bit Bus Width

Note the following concerning the SH7708 Series (SH7708, SH7708S, SH7708R).

In the case of consecutive word and byte accesses, or consecutive longword and byte accesses, in area 6, CE1B and CE2B are asserted simultaneously in the byte access. As a result, a byte write becomes a word-size write. In a byte read, the byte data at the specified address is read correctly.

Use one of the following methods to avoid this problem.

1. Do not use area 6 PCMCIA.
2. Use an 8-bit bus width for area 6 PCMCIA.
3. If a 16-bit bus width is used for area 6 PCMCIA, do not perform byte-size write accesses.
4. In consecutive accesses a. and b. below

   a. Word access → byte access (write)

   b. Longword access → byte access (write)

   in which this problem arises, the problem can be avoided by performing a dummy read in a non-cacheable area as shown in (a') and (b') below.

   a'. Word access → dummy read → byte access (write)

   b'. Longword access → dummy read → byte access (write)

10.4.3 Self-Refreshing

1. Note the following concerning the SH7708 (not including the SH7708S and SH7708R).

   When DRAM is connected in both area 2 and area 3, if

- self-refreshing is executed for the memory in only one of the areas, or
- a value corresponding to the above condition is written to the MCR or DCR register as a program operation

   after setting execution of CBR refreshing for both areas, use the procedure shown in (a), (b), and (c) below.

   a. Make a setting so that refreshing is not performed for either area 2 or area 3 (clear the RFSH bit to 0 in both the MCR and DCR registers).
   b. Execute self-refreshing for one of the areas.
   c. Make a new setting so that auto-refreshing is executed for the area in which self-refreshing is not used.

2. Note the following concerning the SH7708 Series (SH7708, SH7708S, SH7708R).

   If the refresh mode bit (the RMODE bit in registers MCR and DCR) is to be changed while the refresh control bit (the RFSH bit in registers MCR and DCR) is set to 1, use the procedure shown in a. and b. below.

   a. Clear the refresh control bit to 0.
   b. When refreshing is not being performed, set the refresh control bit to 1 again and then change the refresh mode.

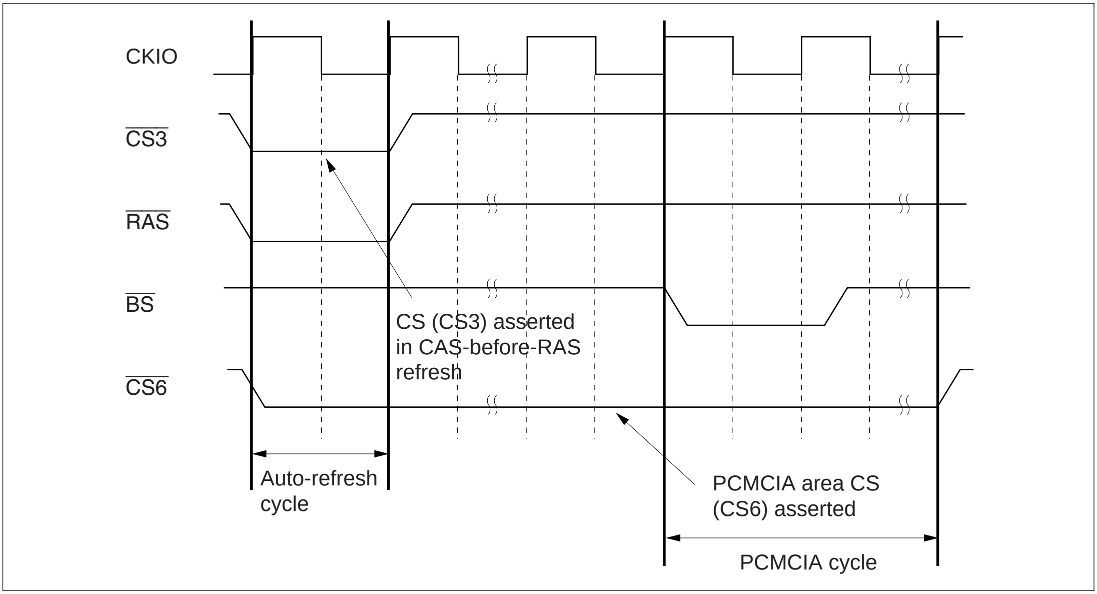
10.4.4 PCMCIA Area Access

In the SH7708 Series (SH7708, SH7708S, SH7708R), if an SDRAM auto-refresh (CAS-before-RAS) request occurs during an area 6 PCMCIA area access, CS6 is simultaneously asserted, in addition to the CS signal for the area to be accessed, when the auto-refresh is performed.

If the following specifications are made, CS3 and CS6 will be asserted simultaneously:

Area 3: SDRAM
Area 6: PCMCIA

CS6 remains asserted until the end of the following PCMCIA bus cycle.



Restrictions: When using PCMCIA in area 6 and also using SDRAM, the area 6 CS signal (CS6) may be asserted earlier than usual, at the same time as the preceding auto-refresh cycle.

Remedy: This anomaly occurs only in area 6. There is no problem with area 5.

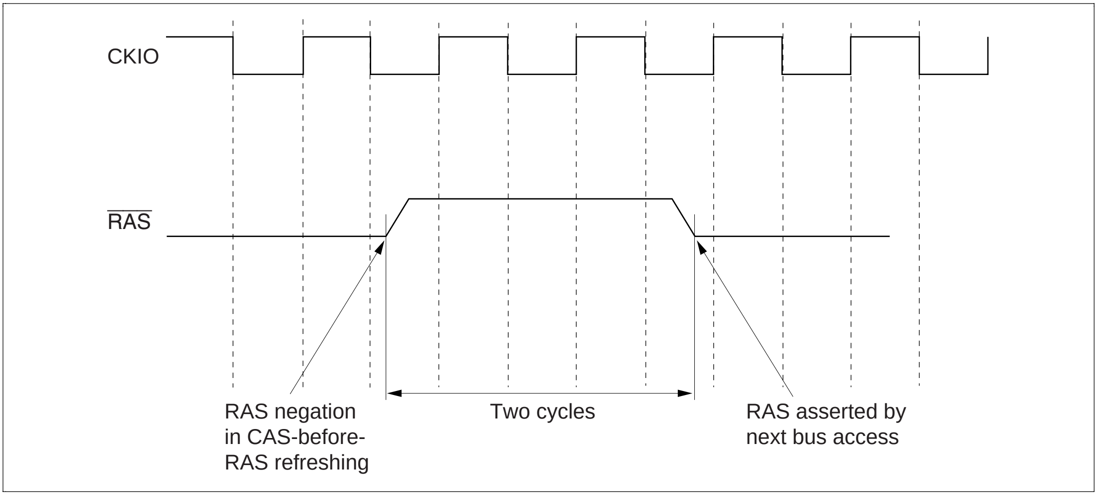
Therefore, if use of both SDRAM and PCMCIA in the system is being considered, area 5 should be used.

If area 6 is used, the system design should allow for CS to be asserted early without causing any problems.

10.4.5 Note on DRAM Connection for Areas 2 and 3

When area 2 or area 3 is connected to DRAM in the SH7708 or SH7708S, if a DRAM access request occurs immediately after a CAS-before-RAS refresh, the RAS precharge cycle value (tPC) always assumes its initial value regardless of the value set in MCR. Thus, the interval from RAS negation to the next RAS assertion in CAS-before-RAS refreshing is a two-cycle output interval (see figure below).

If a request does not immediately follow a CAS-before-RAS refresh, the number of RAS precharge cycles set by the MCR value are secured.


CKIO
$\overline{\text{RAS}}$
RAS negation in CAS-before-RAS refreshing
Two cycles
RAS asserted by next bus access

Restriction: If a DRAM access request occurs immediately after a CAS-before-RAS refresh, the interval from RAS negation to the next RAS assertion is two cycles.

Section 11 Timer (TMU)

11.1 Overview

The SH7708 Series uses a three-channel 32-bit timer unit (TMU).

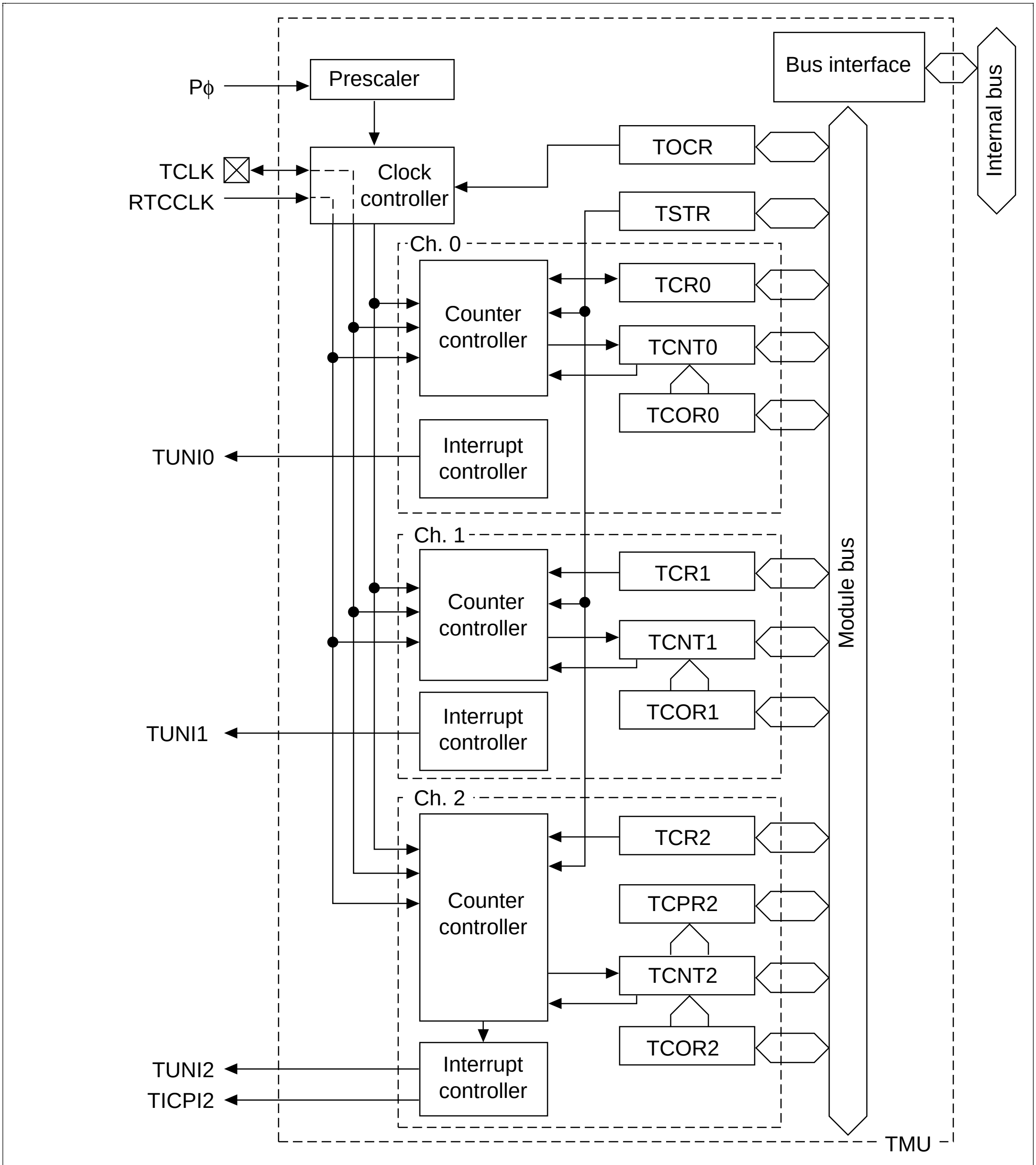
11.1.1 Features

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter
- Channel 2 is provided with an input capture function
- All channels are provided with 32-bit constant registers and 32-bit down counters that can be read or written to at any time
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection between 6 counter input clocks: External clock (TCLK), on-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256. (Pϕ is the internal clock for peripheral modules and can be selected as 1/4, 1/2, or the same frequency as that of the CPU operating clock ϕ.) See section 9, On-Chip Oscillation Circuits, for more information on the clock pulse generator.
- All channels can operate when the SH7708 Series is in standby mode: When the RTC output clock is being used as the counter input clock, the SH7708 Series is still able to count in standby mode.
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the peripheral module used has an internal bus width of 16 bits, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the counter read value caused by this time lag, a synchronization circuit is built into the TCNT so that the entire 32-bit data in the TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels: Operate the SH7708 Series 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.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the TMU.



TOCR: Timer output control register
TSTR: Timer start register
TCR: Timer control register

TCNT: 32-bit timer counter
TCOR: 32-bit timer constant register
TCPR2: 32-bit input capture register

Figure 11.1 TMU Block Diagram

11.1.3 Pin Configuration

Table 11.1 shows the pin configuration of the TMU.

Table 11.1 Pin Configuration

Channel	Pin	I/O	Description
Clock input/clock output	TCLK	I/O	External clock input pin/input capture control input pin/realtime clock (RTC) output pin

11.1.4 Register Configuration

Table 11.2 shows the TMU register configuration.

Table 11.2 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/W	Undefined	H'FFFFFEB8	32

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

11.2 TMU Registers

11.2.1 Timer Output Control Register (TOCR)

TOCR is an 8-bit read/write register that selects whether to use the external TCLK pin as an external clock or an input capture control usage input pin, or an output pin for the on-chip RTC output clock. TOCR is initialized to H'00 by a power-on reset or manual reset, but is not initialized in standby mode, and its contents are retained.

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

Bits 7 to 1—Reserved: These bits always read 0. The write value should always be 0.

Bit 0—Timer Clock Pin Control (TCOE): Selects use of the timer clock pin (TCLK) as an external clock input pin or input pin for input capture control for the on-chip timer, or as an output pin for the on-chip RTC output clock.

Bit 0: TCOE	Description
0	Timer clock pin (TCLK) used as external clock input or input capture control input pin for the on-chip timer (Initial value)
1	Timer clock pin (TCLK) used as output pin for on-chip RTC output clock

11.2.2 Timer Start Register (TSTR)

TSTR is an 8-bit read/write register that selects whether to run or halt the timer counters (TCNT) for channels 0–2. TSTR is initialized to H'00 by a power-on reset or manual reset. In standby mode, when the PLL1 multiplication factor is changed in clock mode 0, 1, 2, or 7, or when the MSTP2 bit is set to 1 in STBCR, TSTR is initialized only when the input clock selected for the channel is an external clock (TCLK) or the peripheral clock (Pø).

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

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

Bit 2—Counter Start 2 (STR2): Selects whether to run or halt timer counter 2 (TCNT2).

Bit 2: STR2	Description	
0	Halt TCNT2 count	(Initial value)
1	Start TCNT2 counting	

Bit 1—Counter Start 1 (STR1): Selects whether to run or halt timer counter 1 (TCNT1).

Bit 1: STR1	Description	
0	Halt TCNT1 count	(Initial value)
1	Start TCNT1 counting	

Bit 0—Counter Start 0 (STR0): Selects whether to run or halt timer counter 0 (TCNT0).

Bit 0: STR0	Description	
0	Halt TCNT0 count	(Initial value)
1	Start TCNT0 counting	

11.2.3 Timer Control Register (TCR)

The timer control registers (TCR) control the timer counters (TCNT) and interrupts. The TMU has three TCR, registers one for each channel.

The TCR registers are 16-bit read/write registers that control the issuance of interrupts when the flag indicating timer counter (TCNT) underflow has been set to 1, and also carry out counter clock selection. When the external clock has been selected, they also select its edge. Additionally, TCR2 controls the channel 2 input capture function and the issuance of interrupts during input capture. The TCRs are initialized to H'0000 by a power-on reset and manual reset. In standby mode, when the PLL1 multiplication factor is changed in clock mode 0, 1, 2, or 7, or when the MSTP2 bit is set to 1 in STBCR, the TCRs retain their contents when the input clock selected for the channel is an external clock (TCLK) or the peripheral clock (Pø), and continue operating when the selected clock is the on-chip RTC output clock (RTCCLK).

Channel 0 and 1 TCR Bit Configuration:

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Channel 2 TCR Bit Configuration:

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 10, 9 (except TCR2), 7, and 6 (except TCR2)—Reserved: These bits always read 0. The write value should always be 0.

Bit 9—Input Capture Interrupt Flag (ICPF): A function of channel 2 only: the flag is set when input capture is requested via the TCLK pin.

Bit 9: ICPF	Description
0	No input capture request has been issued. Clearing condition: When 0 is written to ICPF (Initial value)
1	Input capture has been requested via the TCLK pin. Setting condition: When an input capture is requested via the TCLK pin*

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

Bit 8—Underflow Flag (UNF): Status flag that indicates occurrence of a TCNT underflow.

Bit 8: UNF	Description
0	TCNT has not underflowed. Clearing condition: When 0 is written to UNF (Initial value)
1	TCNT has underflowed (H'00000000 → H'FFFFFFFF). Setting condition: When TCNT underflows*

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

Bits 7 and 6—Input Capture Control (ICPE1, ICPE0): A function of channel 2 only: determines whether the input capture function can be used, and when used, whether or not to enable interrupts.

When using this input capture function it is necessary to set the TCLK pin to input mode with the TCOE bit in the TOCR register. Additionally, use the CKEG bit to designate use of either the rising or falling edge of the TCLK pin to set the value in TNCT2 in the input capture register (TCPR2).

Bit 7: ICPE1	Bit 6: ICPE0	Description
0	0	Input capture function is not used. (Initial value)
	1	Reserved (cannot be set)
1	0	Input capture function is used. Interrupts due to ICPF (TICPI2) are not enabled.
	1	Input capture function is used. Interrupts due to ICPF (TICPI2) 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 (TUNI) are not enabled. (Initial value)
1	Interrupts due to UNF (TUNI) 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 edge

Bits 2 to 0—Timer Prescalers 2–0 (TPSC2–TPSC0): These bits select the TCNT count clock.

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	Internal clock: count on Pϕ/4 (Initial value)
		1	Internal clock: count on Pϕ/16
	1	0	Internal clock: count on Pϕ/64
		1	Internal clock: count on Pϕ/256
1	0	0	Internal clock: count on clock output of on-chip RTC (RTCCLK)
		1	External clock: count on TCLK pin input
	1	0	Reserved (Do not set)
		1	Reserved (Do not set)

11.2.4 Timer Constant Register (TCOR)

The timer constant registers are 32-bit registers. The TMU has three TCOR registers, one for each of the three channels.

TCOR is a 32-bit read/write register. When a TCNT count-down results in an underflow, the TCOR value is set in TCNT and the count-down continues from that value. TCOR is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initialized in standby mode, and retains its contents.

TCOR:

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	19	18	17	16
Bit name:								
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

11.2.5 Timer Counters (TCNT)

The timer counters are 32-bit read/write registers. The TMU has three timer counters, one for each channel.

TCNT counts down upon input of a clock. The clock input is selected using the TPSC2–TPSC0 bits in the timer control register (TCR).

When a TCNT count-down results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the count-down continues from that value.

Because the internal bus for the SH7708 Series on-chip supporting modules is 16 bits wide, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. Since TCNT counts sequentially, this time lag can create discrepancies between the data in the upper and lower halves. To correct the discrepancy, a buffer register is connected to TCNT so that upper and lower halves are not read separately. The entire 32-bit data in TCNT can thus be read at once.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset. In standby mode, when the PLL1 multiplication factor is changed in clock mode 0, 1, 2, or 7, or when the MSTP2 bit is set to 1 in STBCR, TCNT retains its contents when the input clock selected for the channel is an external clock (TCLK) or the peripheral clock (Pø), and continues operating when the selected clock is the on-chip RTC output clock (RTCCLK).

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

11.2.6 Input Capture Register (TCPR2)

The input capture register (TCPR2) is a read-only 32-bit register built only into timer 2. Control of TCPR2 setting conditions due to the TCLK pin is affected by the input capture function bits (ICPE1/ICPE2 and CKEG1/CKEG0)) in TCR2. When a TCPR2 setting indication due to the TCLK pin occurs, the value of TCNT2 is copied into TCPR2.

TCNT2 is not initialized by a power-on reset or manual reset, or in standby mode.

TCPR2:

Bit:	31	30	29	28	27	26	25	24
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
Bit name:								
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

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

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

11.3 TMU Operation

11.3.1 Overview

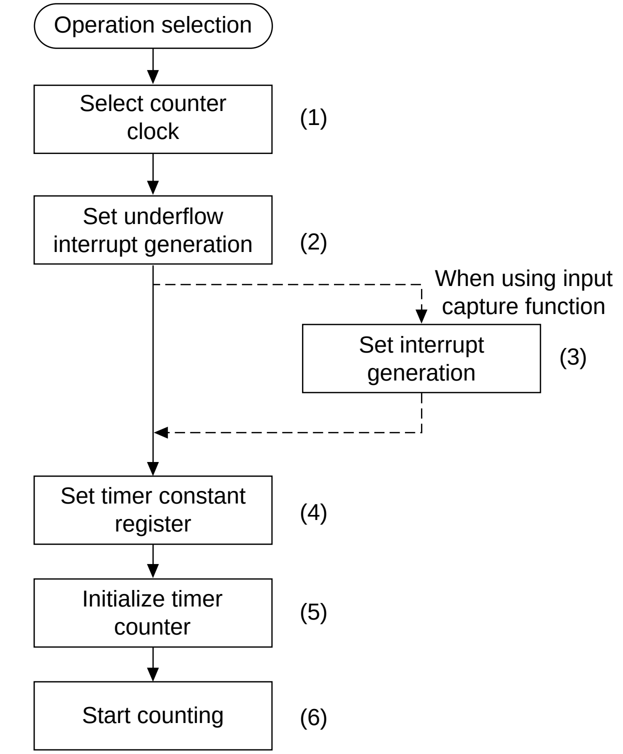
Each of the three channels has a 32-bit timer counter (TCNT) and a 32-bit timer constant register. The TCNT counts down. The auto-reload function enables synchronized counting and counting by external events. Channel 2 has an input capture function.

11.3.2 Basic Functions

Counter Operation: When the STR0–STR2 bits in the timer start register (TSTR) are set, the corresponding timer counter (TCNT) starts counting. When a TCNT underflows (H'00000000 → H'FFFFFFFF), the UNF flag of the corresponding timer control register (TCR) is set. At this time, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the down-count operation is continued.

The count operation is set as follows (figure 11.2):

1. Select the counter clock with the TPSC2–TPSC0 bits in the timer control register (TCR). If the external clock is selected, set the TCLK pin to input mode with the TOCE bit in TOCR, and select its edge with the CKEG1 and CKEG0 bits in TCR.
2. Use the UNIE bit in TCR to set whether to generate an interrupt when TCNT underflows.
3. When using the input capture function, set the ICPE bits in TCR, including the choice of whether or not to use the interrupt function (channel 2 only).
4. Set a value in the timer constant register (TCOR) (the cycle is the set value plus 1).
5. Set the initial value in the timer counter (TCNT).
6. Set the STR bit in the timer start register (TSTR) to 1 to start operation.



Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 11.2 Setting the Count Operation

Auto-Reload Count Operation: Figure 11.3 shows the TCNT auto-reload operation.

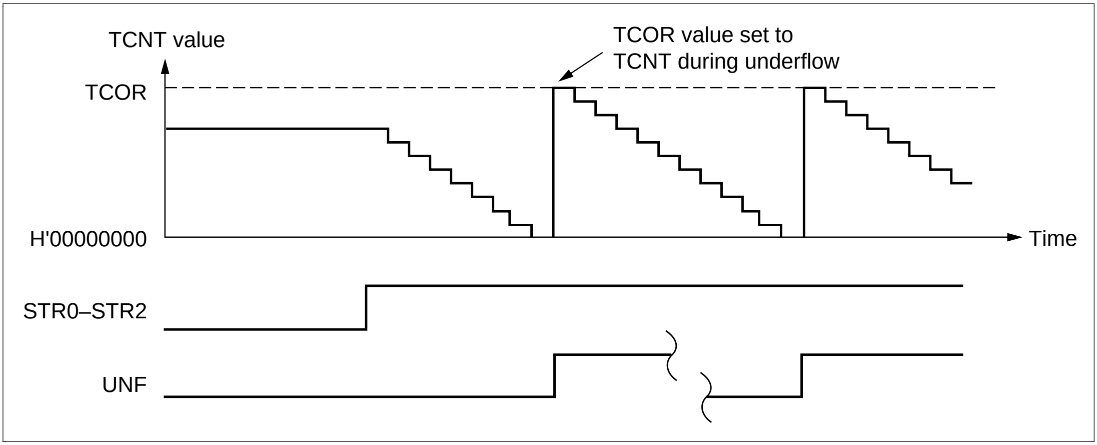


Figure 11.3 Auto-Reload Count Operation

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 11.4 shows the timing.

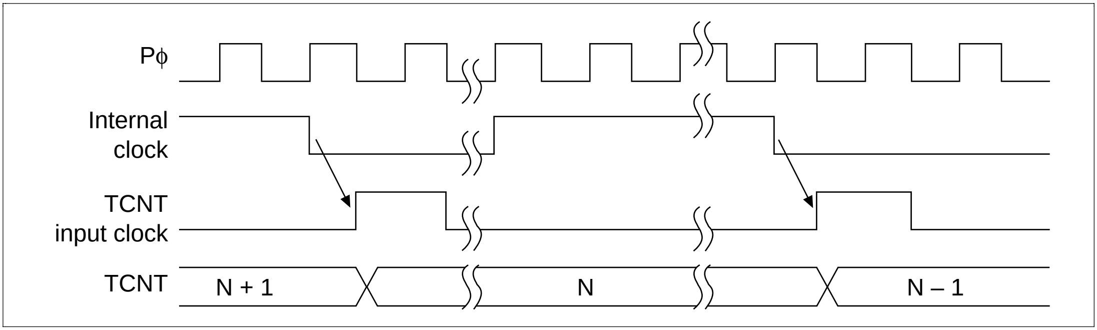


Figure 11.4 Count Timing when Internal Clock Is Operating

- External Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the external clock (TCLK) as the timer clock. Use the CKEG1 and CKEG0 bits in TCR to select the detection edge. Rise, fall or both may be selected. The pulse width of the external clock must be at least 1.5 peripheral module clock cycles for single edges or 2.5 peripheral module clock cycles for both edges. A shorter pulse width will result in inaccurate operation. Figure 11.5 shows the timing for both-edge detection.

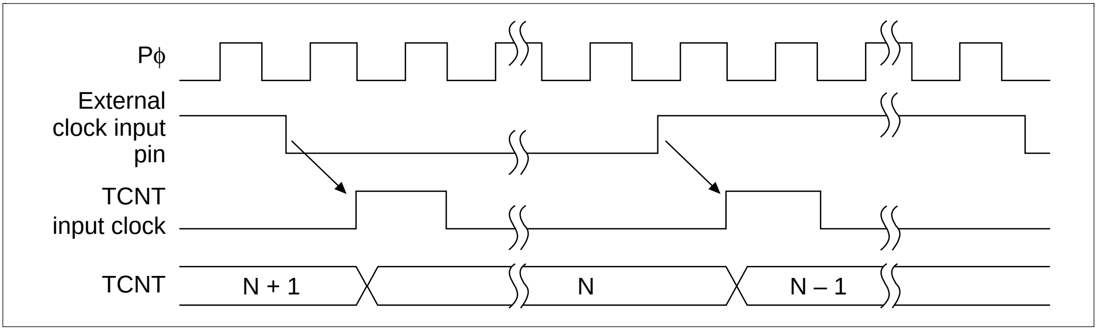


Figure 11.5 Count Timing when External Clock Is Operating (Both Edges Detected)

- On-Chip RTC Clock Operation: Set the TPSC2–TPSC0 bits in TCR to select the on-chip RTC clock as the timer clock. Figure 11.6 shows the timing.

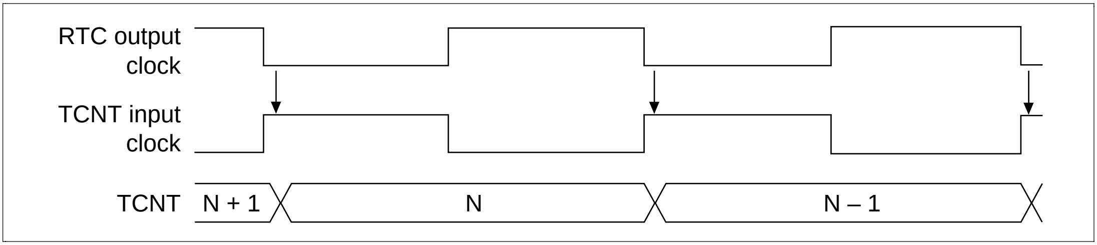


Figure 11.6 Count Timing when On-Chip RTC Clock Is Operating

Input Capture Function: Channel 2 has an input capture function (figure 11.7). 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.

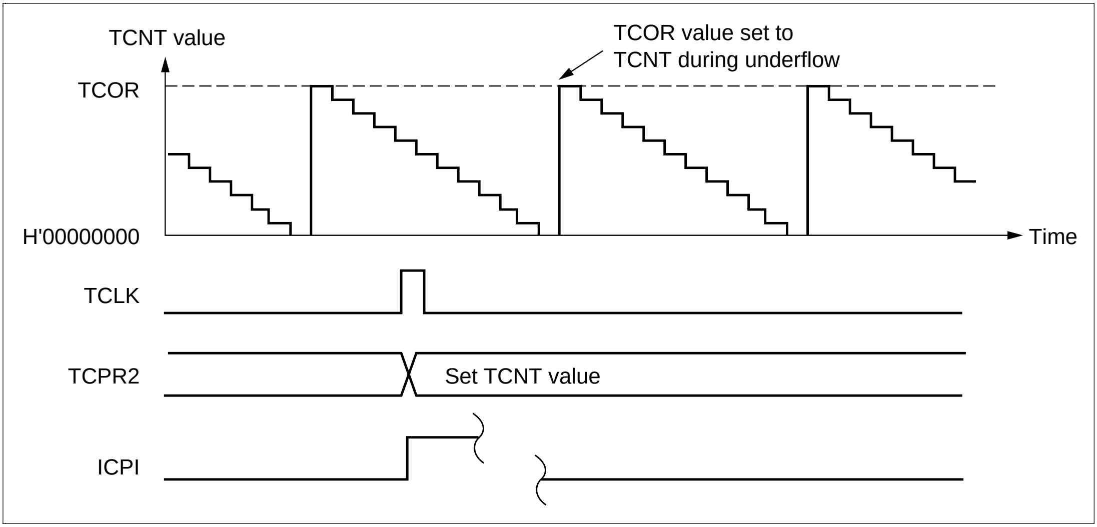


Figure 11.7 Operation Timing when Using the Input Capture Function (Using TCLK Rising Edge)

11.4 Interrupts

There are two sources of TMU interrupts: underflow interrupts (TUNI) and interrupts when using the input capture function (TICPI2).

11.4.1 Status Flag Set Timing

UNF is set to 1 when the TCNT underflows (H'00000000 → H'FFFFFFFF). Figure 11.8 shows the timing.

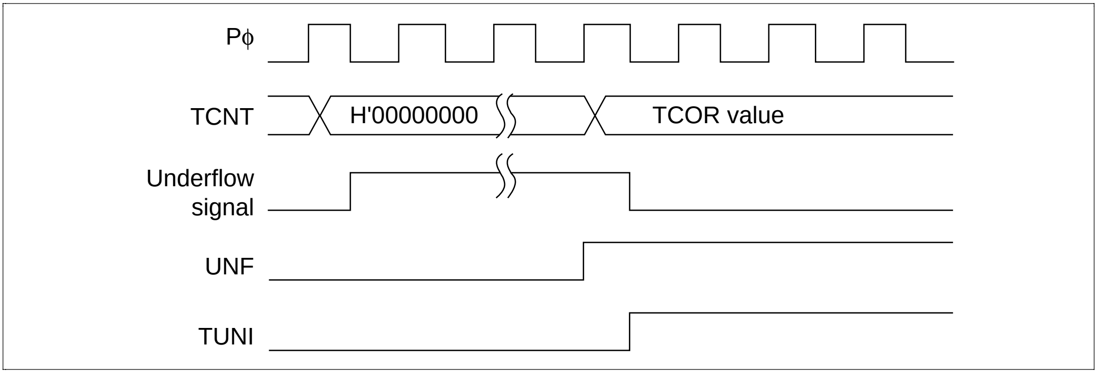


Figure 11.8 UNF Set Timing

11.4.2 Status Flag Clear Timing

The status flag can be cleared by writing a 0 from the CPU. Figure 11.9 shows the timing.

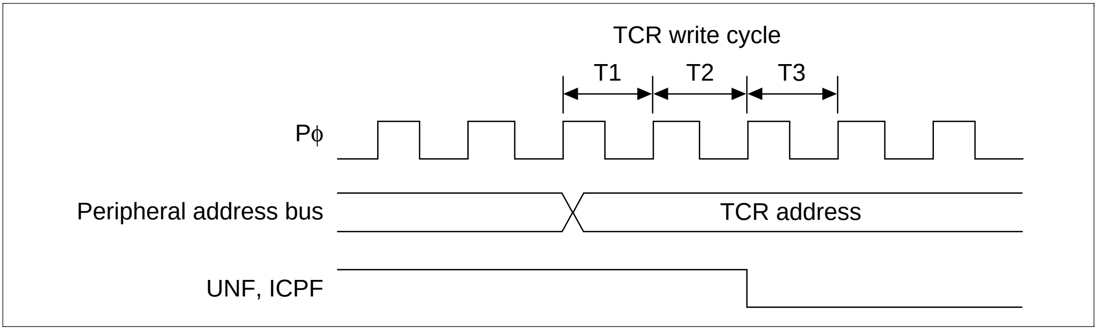


Figure 11.9 Status Flag Clear Timing

11.4.3 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) for these interrupts and interrupt handling occurs according to the codes.

The relative priorities of channels can be changed using the interrupt controller (see section 6, Interrupt Controller) and section 4, Exception Handling. Table 11.3 lists TMU interrupt sources.

Table 11.3 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

11.5 Usage Notes

11.5.1 Writing to Registers

Synchronization processing is not performed for timer counting during register writes. When writing to registers, always clear the appropriate start bits for the channel (STR2–STR0) in the timer start register (TSTR) to halt timer counting.

11.5.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. When timer counting and register read processing are performed simultaneously, the register value before TCNT counting down (with synchronization processing) is read.

11.5.3 Clearing UNF in the TCR Register

When the RTC (realtime clock) is specified for use as the TMU (timer) clock source in the SH7708 and SH7708S the procedure shown in the flowchart below should be used to clear the underflow flag (UNF) in the timer control register (TCR).

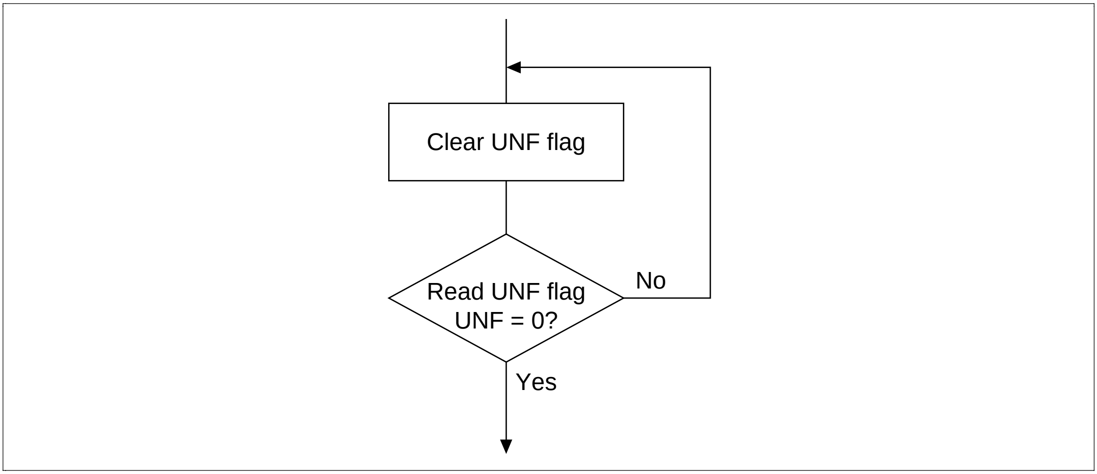


Figure 11.10 UNF Flag Clearing

Section 12 Realtime Clock (RTC)

12.1 Overview

The SH7708 Series has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

12.1.1 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

12.1.2 Block Diagram

The following abbreviations are used in the block diagram of the RTC (figure 12.1):

R64CNT:	64-Hz counter	RSECAR:	Second alarm register
RSECCNT:	Second counter	RMINAR:	Minute alarm register
RMINCNT:	Minute counter	RHRAR:	Hour alarm register
RHRCNT:	Hour counter	RWKAR:	Day of the week alarm register
RWKCNT:	Day of the week counter	RDAYAR:	Date alarm register
RDAYCNT:	Date counter	RMONAR:	Month alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2

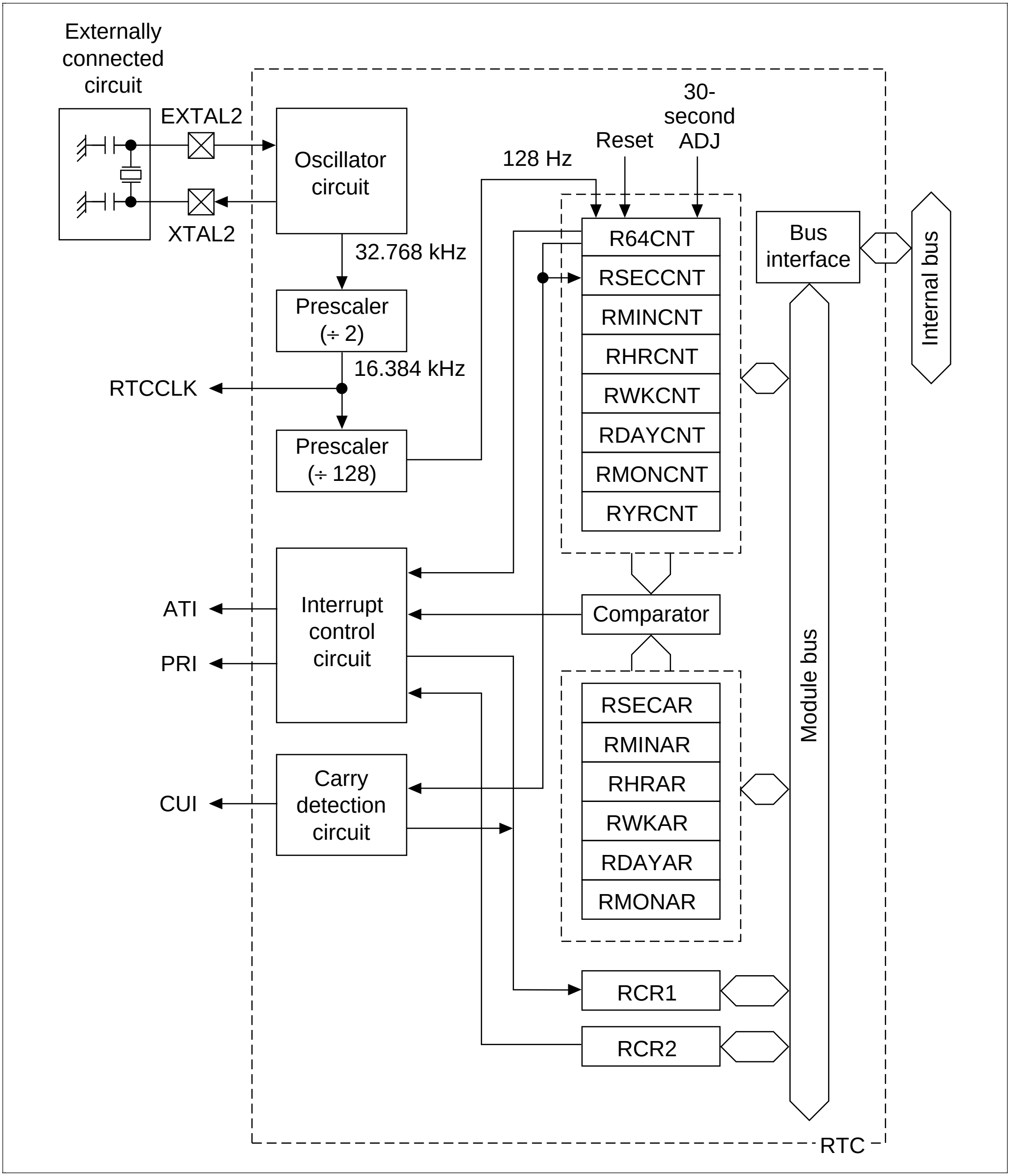


Figure 12.1 RTC Block Diagram

12.1.3 Pin Configuration

Table 12.1 shows the RTC pin configuration.

Table 12.1 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/realtime clock (RTC) output pin (shared by TMU)
Dedicated power-supply pin for RTC	Vcc (RTC)	—	Dedicated power-supply pin for RTC*
Dedicated GND pin for RTC	Vss (RTC)	—	Dedicated GND pin for RTC*

Note: * With the SH7708S and SH7708R, except in hardware standby mode, all V_{CC} and V_{SS} pins should be connected to the system power supply (power should be supplied constantly). In hardware standby mode, power should be supplied at least to V_{CC} (RTC) and V_{SS} (RTC). If power is not supplied to V_{CC} and V_{SS} pins other than V_{CC} (RTC) and V_{SS} (RTC), hold the CA pin low. With the SH7708, it is not possible to supply power only to the RTC power supply.

12.1.4 RTC Register Configuration

Table 12.2 shows the RTC register configuration.

Table 12.2 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 bits of each register are initialized.

12.2 RTC Registers

12.2.1 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

12.2.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded second section of the RTC. The count operation is performed by a carry for each second of the 64 Hz counter.

The range that can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded minute section of the RTC. The count operation is performed by a carry for each minute of the second counter.

The range that can be set is 00–59 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded hour section of the RTC. The count operation is performed by a carry for each 1 hour of the minute counter.

The range that can be set is 00–23 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.5 Day of the Week Counter (RWKCNT)

The day of the week counter (RWKCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded day of week section of the RTC. The count operation is performed by a carry for each day of the date counter.

The range that can be set is 0–6 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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 12.3.

Table 12.3 Day-of-Week Codes (RWKCNT)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

12.2.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded date section of the RTC. The count operation is performed by a carry for each day of the hour counter.

The range that can be set is 01–31 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded month section of the RTC. The count operation is performed by a carry for each month of the date counter.

The range that can be set is 00–12 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded year section of the RTC. The least significant 2 digits of the western calendar year are displayed. The count operation is performed by a carry for each year of the month counter.

The range that can be set is 00–99 (decimal). Errant operation will result if any other value is set. Carry out write processing after halting the count operation with the START bit in RCR2, or perform a write using the carry flag as shown in figure 12.2.

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

12.2.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded second section counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSECAR fields are not initialized by a power-on reset or manual reset, or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	ENB	10 seconds			1 second			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded minute section counter RMINCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–59 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR fields are not initialized by a power-on reset or manual reset, or in standby mode. Contents are retained in a manual reset and 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

12.2.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded hour section counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 00–23 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fields are not initialized by a power-on reset or manual reset, or in standby mode. Contents are retained in a manual reset and 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

12.2.12 Day of the Week Alarm Register (RWKAR)

The day of the week alarm register (RWKAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded day of week section counter RWKCNT of the RTC. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR registers, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 0–6 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR fields are not initialized by a power-on reset or manual reset, or in standby mode. Contents are retained in a manual reset and 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 12.4.

Table 12.4 Day-of-Week Codes (RWKAR)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

12.2.13 Date Alarm Register (RDAYAR)

The date alarm register (RDAYAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded date section counter RDAYCNT of the RTC. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 01–31 (decimal) + ENB bit. Errant operation will result if any other value is set. The RDAYCNT range that can be set changes with some months and in leap years. Please confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR fields are not initialized by a power-on reset or manual reset, or in standby mode. Contents are retained in a manual reset and 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

12.2.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded month section counter RMONCNT of the RTC. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among the registers RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincide, an RTC alarm interrupt is generated.

The range that can be set is 01–12 (decimal) + ENB bit. Errant operation will result if any other value is set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONAR fields are not initialized by a power-on reset or manual reset, or in standby mode. Contents are retained in a manual reset and 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

12.2.15 RTC Control Register 1 (RCR1)

The RTC control register 1 (RCR1) is an 8-bit read/write register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag. Because flags are sometimes set after an operand read, do not use this register in read-modify-write processing.

RCR1 is initialized to H'00 by a power-on reset. In a manual reset, all bits are initialized to 0 except for the CF flag, which is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

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

Bit 7—Carry Flag (CF): Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to R64CNT or RSECCNT occurs. A count register value read at this time cannot be guaranteed; another read is required.

Bit 7: CF	Description
0	No count up of R64CNT or RSECCNT. Clearing condition: When 0 is written to CF (Initial value)
1	Count up of R64CNT or RSECCNT. Setting condition: When 1 is written to CF

Bits 6, 5, 2, and 1—Reserved: These bits always read 0. The write value should always be 0.

Bit 4—Carry Interrupt Enable Flag (CIE): When the carry flag (CF) is set to 1, the CIE bit enables interrupts.

Bit 4: CIE	Description
0	A carry interrupt is not generated when the CF flag is set to 1 (Initial value)
1	A carry interrupt is generated when the CF flag is set to 1

Bit 3—Alarm Interrupt Enable Flag (AIE): When the alarm flag (AF) is set to 1, the AIE bit allows interrupts.

Bit 3: AIE	Description
0	An alarm interrupt is not generated when the AF flag is set to 1 (Initial value)
1	An alarm interrupt is generated when the AF flag is set to 1

Bit 0—Alarm Flag (AF): The AF flag is set to 1 when the alarm time set in an alarm register (only registers with ENB bit set to 1) matches the clock and calendar time. This flag is cleared to 0 by writing 0, but retains its previous value if 1 is written.

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.

12.2.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit read/write register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count start/stop control. It is initialized to H'09 by a power-on reset. It is 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 6–4—Periodic Interrupt Flags (PES2–PES0): These bits specify the periodic interrupt.

Bit 6: PES2	Bit 5: PES1	Bit 4: PES0	Description
0	0	0	No periodic interrupts generated (Initial value)
		1	Periodic interrupt generated every 1/256 second
	1	0	Periodic interrupt generated every 1/64 second
		1	Periodic interrupt generated every 1/16 second
1	0	0	Periodic interrupt generated every 1/4 second
		1	Periodic interrupt generated every 1/2 second
	1	0	Periodic interrupt generated every 1 second
		1	Periodic interrupt generated every 2 seconds

Bit 3—RTCEN: Controls the operation of the crystal oscillator for the RTC.

Bit 3: RTCEN	Description
0	Halts the crystal oscillator for the RTC.
1	Runs the crystal oscillator for the RTC. (Initial value)

Bit 2—30-Second Adjustment (ADJ): When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit will be simultaneously reset. This bit always reads 0.

Bit 2: ADJ	Description	
0	Runs normally.	(Initial value)
1 (write)	30-second adjustment.	

Bit 1—Reset (RESET): When 1 is written, initializes the divider circuit. This bit always reads 0.

Bit 1: RESET	Description	
0	Runs normally.	(Initial value)
1	Divider circuit is reset.	

Bit 0—Start Bit (START): Halts and restarts the counter (clock).

Bit 0: START	Description	
0	Second, minute, hour, day, week, month, year counter halts.	
1	Second, minute, hour, day, week, month, year counter runs normally.	(Initial value)

Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

12.3 RTC Operation

12.3.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

12.3.2 Setting the Time

Part (a) in figure 12.2 shows how to set the time when the clock is stopped. This works when the entire calendar or clock is to be set.

Part (b) in figure 12.2 describes how to set the clock when the clock is running. This works when only part of the calendar or clock needs to be reset (e.g., changing only the seconds or only the hour). The write status is checked using the carry flags. When there is a carry during the writing of new data, the new data is automatically updated. Since this causes errors in the data, the data must be rewritten if the carry flag is set to 1.

The interrupt function can be used to determine the status of the carry flag.

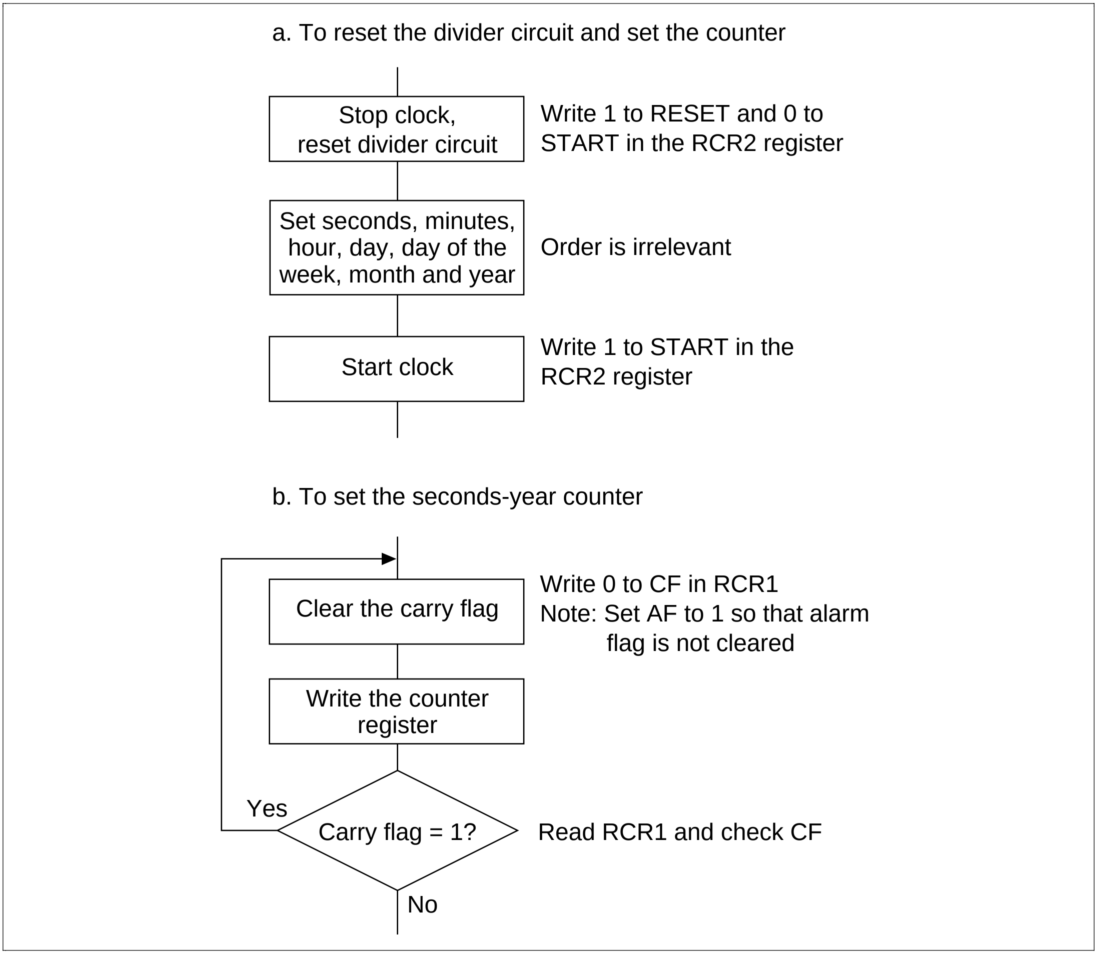

a. To reset the divider circuit and set the counter
Stop clock, reset divider circuit
Write 1 to RESET and 0 to START in the RCR2 register
Set seconds, minutes, hour, day, day of the week, month and year
Order is irrelevant
Start clock
Write 1 to START in the RCR2 register
b. To set the seconds-year counter
Clear the carry flag
Write 0 to CF in RCR1
Note: Set AF to 1 so that alarm flag is not cleared
Write the counter register
Yes
Carry flag = 1?
Read RCR1 and check CF
No

Figure 12.2 Setting the Time

12.3.3 Reading the Time

Figure 12.3 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 12.3 shows the method of reading the time without using interrupts; part (b) in figure 12.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

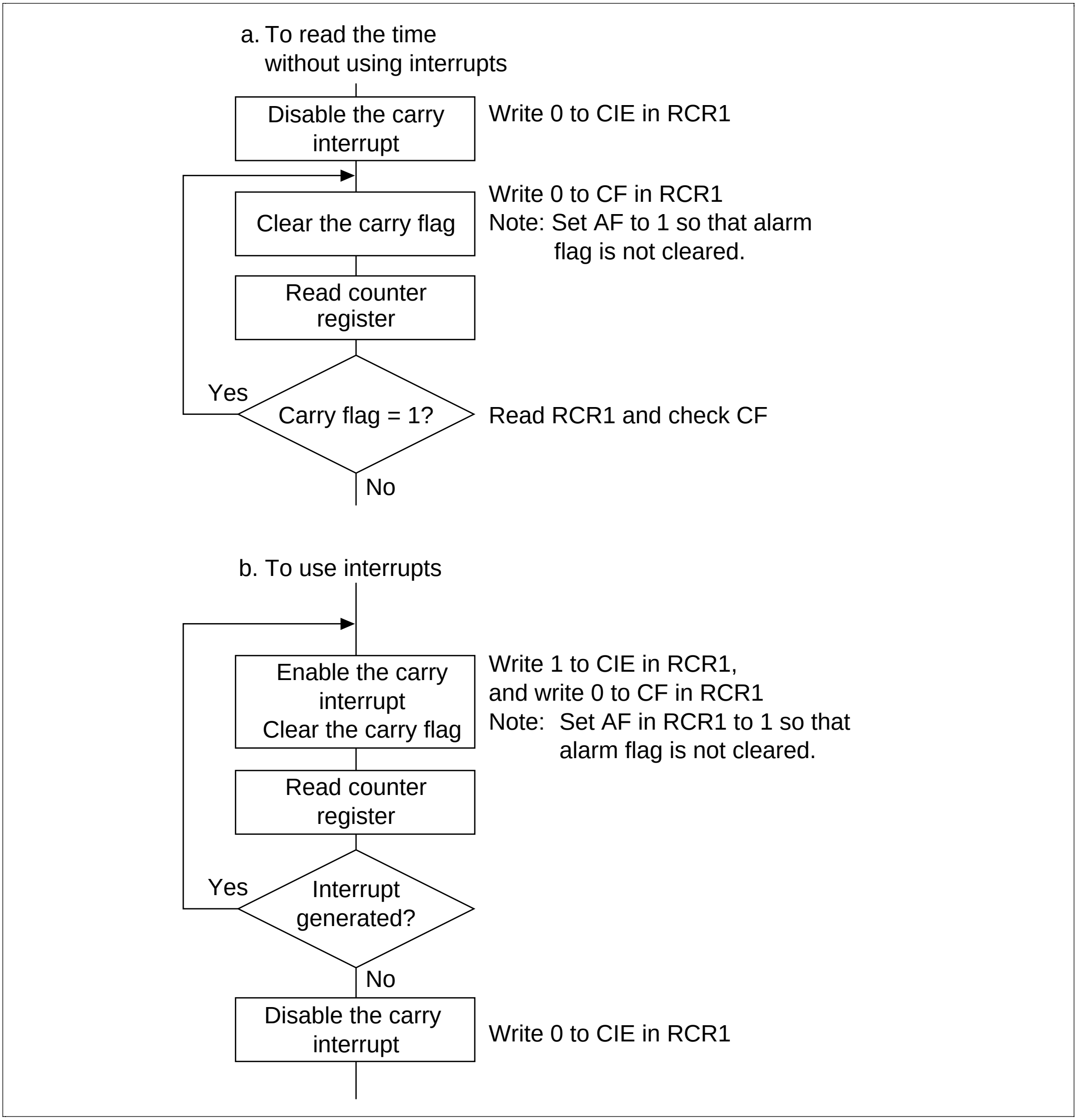

a. To read the time without using interrupts
Disable the carry interrupt
Write 0 to CIE in RCR1
Clear the carry flag
Write 0 to CF in RCR1
Note: Set AF to 1 so that alarm flag is not cleared.
Read counter register
Yes
Carry flag = 1?
Read RCR1 and check CF
No
b. To use interrupts
Enable the carry interrupt
Clear the carry flag
Write 1 to CIE in RCR1, and write 0 to CF in RCR1
Note: Set AF in RCR1 to 1 so that alarm flag is not cleared.
Read counter register
Yes
Interrupt generated?
No
Disable the carry interrupt
Write 0 to CIE in RCR1

Figure 12.3 Reading the Time

12.3.4 Alarm Function

Figure 12.4 shows how to use the alarm function.

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, or any combination of these. Set the ENB bit (bit 7) in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is placed to 0.

When the clock and alarm times match, a 1 is set in the AF bit (bit 0) in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the AIE bit (bit 3) in RCR1, an interrupt is generated when an alarm occurs.

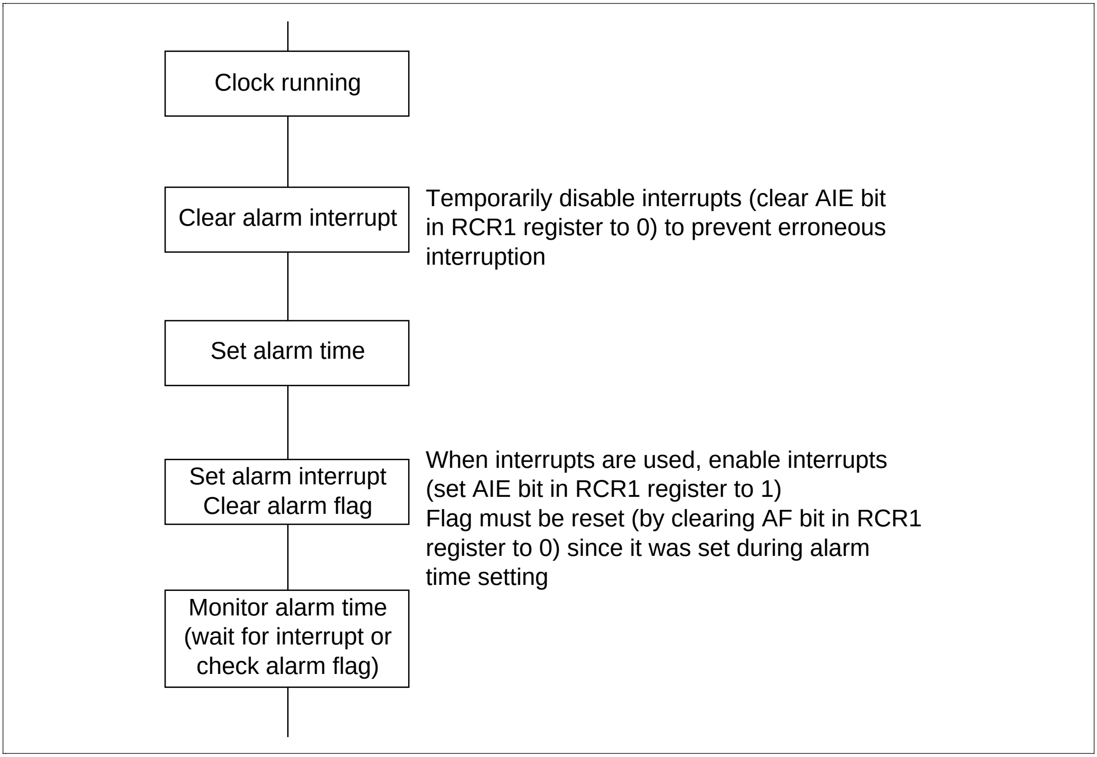

Clock running
Clear alarm interrupt
Temporarily disable interrupts (clear AIE bit in RCR1 register to 0) to prevent erroneous interruption
Set alarm time
Set alarm interrupt
Clear alarm flag
When interrupts are used, enable interrupts (set AIE bit in RCR1 register to 1)
Flag must be reset (by clearing AF bit in RCR1 register to 0) since it was set during alarm time setting
Monitor alarm time (wait for interrupt or check alarm flag)

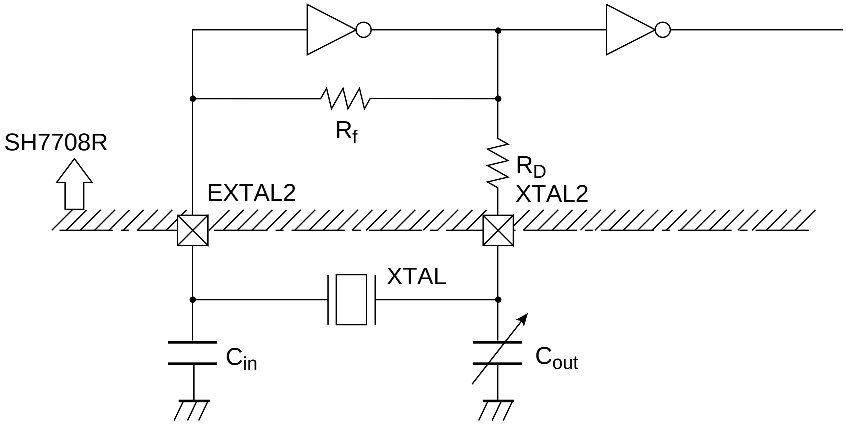
Figure 12.4 Using the Alarm Function

12.3.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 12.5, and the RTC crystal oscillator circuit in figure 12.5.

Table 12.5 Recommended Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes: 1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.

2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ

3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.

4. The crystal oscillation settling time depends on the mounted circuit constants, floating capacitance, etc., and should be decided after consultation with the crystal resonator manufacturer.

5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip.
(Correct oscillation may not be possible if there is externally induced noise in the EXTAL2 and XTAL2 pins.)

6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 12.5 Example of Crystal Oscillator Circuit Connection

12.4 Usage Notes

12.4.1 Flag Clearing

With the SH7708 (not including the SH7708S and SH7708R), the procedure shown in the flowchart below should be used to clear the carry flag (CF) and alarm flag (AF) in realtime clock (RTC) register RCR1 (RTC control register 1) and the peripheral interrupt flag (PEF) in register RCR2 (RTC control register 2).

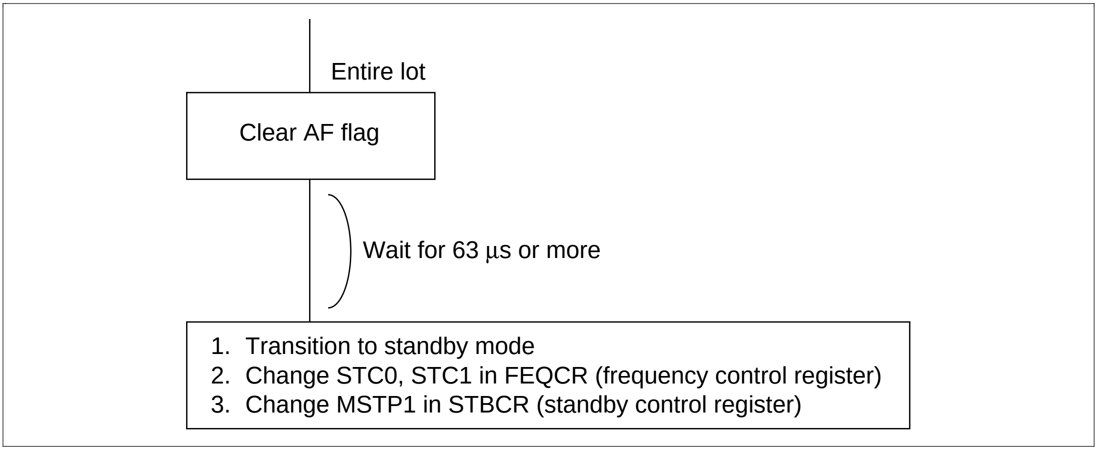

Entire lot
Clear AF flag
Wait for 63 µs or more
1. Transition to standby mode
2. Change STC0, STC1 in FEQCR (frequency control register)
3. Change MSTP1 in STBCR (standby control register)

Figure 12.6 AF Flag Clearing

If an interval of at least 63 µs is not allowed, the AF flag may not be set even if the setting conditions are satisfied. Use the same procedure for clearing the CF and PEF flags.

12.4.2 Register Writes during RTC Count

In the SH7708 Series (SH7708, SH7708S, and SH7708R), the following RTC registers cannot be written to while the RTC is counting (while bit 0 of RCR2 = 1).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be stopped before writing to any of these registers.

12.4.3 Usage Note about Periodic Interrupt on RTC

Figure 12.7 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.

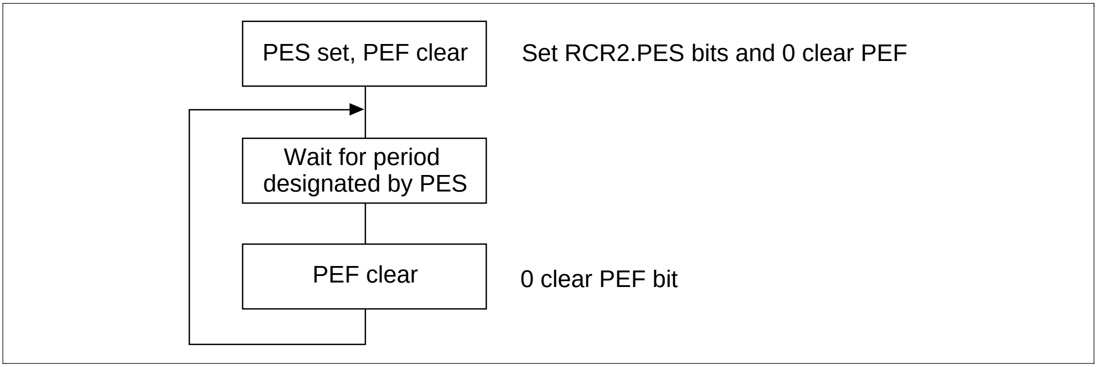

PES set, PEF clear
Set RCR2.PES bits and 0 clear PEF
Wait for period designated by PES
PEF clear
0 clear PEF bit

Figure 12.7 Using the Periodic Interrupt Function

Section 13 Serial Communication Interface (SCI)

13.1 Overview

The SH7708 Series has an on-chip serial communication interface (SCI) that supports both asynchronous and synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors. The SCI supports a smart card interface, which is a serial communications feature for IC card interfaces that conforms to the ISO/IEC standard 7816-3 for identification cards. See section 14, Smart Card Interface, for more information.

13.1.1 Features

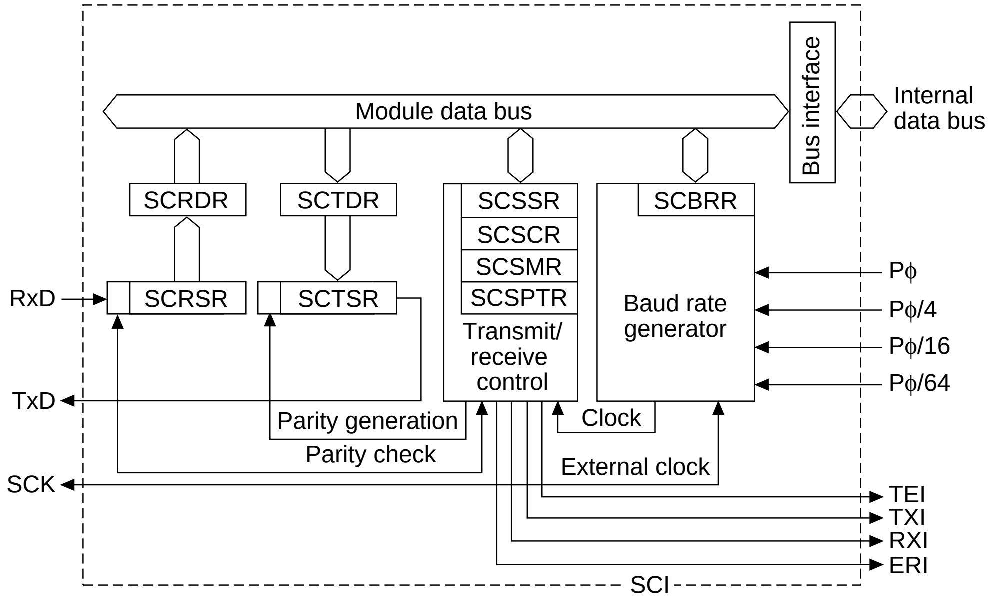
SCI features are listed below.

- Asynchronous or synchronous can be selected as the serial communication mode.
- Asynchronous mode:
  - Serial data communication is synchronized in start-stop mode 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. The maximum bit rate is 937.5 kbps. 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 serial port register (SCSPTR) 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. The maximum bit rate is 5 Mbps. There is one serial data communication format.
  - Data length: Eight bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.
- When the SCI is not in use, it can be stopped by halting the clock supplied to it, saving power.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the SCI.



SCRSR:	Receive shift register	SCSMR:	Serial mode register
SCRDR:	Receive data register	SCSCR:	Serial control register
SCTSR:	Transmit shift register	SCSSR:	Serial status register
SCTDR:	Transmit data register	SCBRR:	Bit rate register
		SCSPTR:	Serial port register

Figure 13.1 SCI Block Diagram

13.1.3 Pin Configuration

The SCI has the serial pins summarized in table 13.1.

Table 13.1 SCI Pins

Pin Name	Abbreviation	Input/Output	Function
Serial clock pin	SCK	Input/output	Clock input/output
Receive data pin	RxD	Input	Receive data input
Transmit data pin	TxD	Output	Transmit data output

Note: These pins function as mode input pins MD0–MD02 after a power-on reset. They are made to function as serial pins by performing SCI operation settings with the TE, RE, CKEI, and CKE0 bits in SCSCR and the $C/\overline{A}$ bit in SCSMR. Break status transmission and detection can be performed by means of the SCI's SCSPTR register.

13.1.4 Register Configuration

Table 13.2 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 13.2 Registers

Name	Abbreviation	R/W	Initial Value*[2]	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
Serial port register	SCSPTR	R/W	Undefined (Initialized)*[3]	H'FFFFFF7C	8

Notes: 1 Only 0 can be written, to clear the flags.

2 Initialized by power-on reset or manual reset.

3. All bits except 2 and 0 are initialized to 0. The value of bits 2 and 0 is undefined.

13.2 Register Descriptions

13.2.1 Receive Shift Register (SCRSR)

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 SCRSR directly.

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

13.2.2 Receive Data Register (SCRDR)

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 or module standby mode.

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

13.2.3 Transmit Shift Register (SCTSR)

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:	—	—	—	—	—	—	—	—

13.2.4 Transmit Data Register (SCTDR)

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 or module standby mode.

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 Serial Mode Register (SCSMR)

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 or module standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	C/$\overline{\text{A}}$	CHR	PE	O/$\overline{\text{E}}$	STOP	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Communication Mode (C/$\overline{\text{A}}$): Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7: C/$\overline{\text{A}}$	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 8 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/$\overline{\text{E}}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{\text{E}}$) mode setting.

Bit 4—Parity Mode (O/$\overline{\text{E}}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{\text{E}}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{\text{E}}$ setting is ignored in synchronous mode, and in asynchronous mode when parity addition and checking is disabled.

Bit 4: O/$\overline{\text{E}}$	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 In transmitting, a single 1-bit is added at the end of each transmitted character.	(Initial value)
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 13.3.3, Multiprocessor Communication.

Bit 2: MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available: Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 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

13.2.6 Serial Control Register (SCSCR)

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 or module standby mode.

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 locked at 1.
1	Transmitter enabled Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SCSSR) is cleared to 0 after writing of transmit data into 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 during reception. The MPIE setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation). (Initial value) MPIE is cleared to 0 by writing 0 to it, or when 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 SCSCR are set to 1), and allows the FER and ORER bits to be set.

Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if SCTDR does not contain new transmit data when the MSB is transmitted.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) requests are disabled.* (Initial value)
1	Transmit-end interrupt (TEI) requests are enabled.*

Note: * The TEI request can be cleared by reading the TDRE bit in the serial status register (SCSSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0, or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only in 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). Before selecting the SCI operating mode in the serial mode register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clock source, see table 13.9 in section 13.3, 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 serial clock output (Initial value)
	1	Asynchronous mode	Internal clock, SCK pin used for clock output*[1]
		Synchronous mode	Internal clock, SCK pin used for serial clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*[2]
		Synchronous mode	External clock, SCK pin used for serial clock input
	1	Asynchronous mode	External clock, SCK pin used for clock input*[2]
		Synchronous mode	External clock, SCK pin used for serial 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.

13.2.7 Serial Status Register (SCSSR)

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 that cannot be written. SCSSR is initialized to H'84 by a reset and in standby or module standby mode.

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 data is written in SCTDR.
1	SCTDR does not contain valid transmit data. (Initial value) TDRE is set to 1 when the chip is reset or enters standby mode, the TE bit in the serial control register (SCSCR) is cleared to 0, or SCTDR contents are loaded into SCTSR, so new data can be written in SCTDR.

Bit 6—Receive Data Register Full (RDRF): Indicates that SCRDR contains received data.

Bit 6: RDRF	Description
0	SCRDR does not contain valid received data. (Initial value) RDRF is cleared to 0 when the chip is reset or enters standby mode, software reads RDRF after it has been set to 1, then writes 0 in RDRF, or data is read from SCRDR.
1	SCRDR contains valid received data. RDRF is set to 1 when serial data is received normally and transferred from SCRSR to SCRDR.

Note: 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 receive data is lost.

Bit 5—Overrun Error (ORER): Indicates that data reception aborted due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally. (Initial value) Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value. ORER is cleared to 0 when the chip is reset or enters standby mode or software reads ORER after it has been set to 1, then writes 0 in ORER.
1	A receive overrun error occurred. 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. ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1.

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 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 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 and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE, or data is written in SCTDR.
1	End of transmission. (Initial value) TEND is set to 1 when the chip is reset or enters standby mode, TE is cleared to 0 in the serial control register (SCSCR), or TDRE is 1 when the last bit of a one-byte serial character is transmitted.

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in asynchronous mode. MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0. (Initial value) If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value.
1	Multiprocessor bit value in receive data is 1.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in 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.

13.2.8 Serial Port Register (SCSPTR)

The serial port register (SCSPTR) is an 8-bit register that the CPU can always read and write. It controls I/O and data of the port multiplexed with the serial communications interface (SCI) pins. Input data can be read from the RxD pin and output data can be transmitted to the TxD pin; this controls breaks for serial transmission and reception.

SCSPTR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT
Initial value:	0	0	0	0	0	—	0	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits always read 0. The write value should always be 0.

Bit 3—Serial Port Clock Port I/O (SPB1IO): Specifies serial port SCK pin input/output. When the SCK pin is actually set as a port output pin and outputs the value set by the SPB1DT bit, the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR should be cleared to 0.

Bit 3: SPB1IO	Description	
0	The SPB1DT value is not output to the SCK pin.	(Initial value)
1	The SPB1DT bit value is output to the SCK pin.	

Bit 2—Serial Port Clock Port Data (SPB1DT): Specifies the serial port SCK pin input/output data. Input or output is specified by the SPB1IO bit (see the description of SPB1IO for details). When output is specified, the value of the SPB1DT bit is output to the SCK pin. The SCK pin value is read from the SPB1DT bit regardless of the value of the SPB1IO bit. The initial value of this bit after a power-on reset is undefined.

Bit 2: SPB1DT	Description	
0	I/O data level is low.	(Initial value)
1	I/O data level is high.	

Bit 1—Serial Port Break I/O (SPB0IO): Specifies the serial port TxD pin output condition. When the TxD pin is actually set as a port output pin and outputs the value set by the SPB0DT bit, the TE bit in SCSCR should be cleared to 0.

Bit 1: SPB0IO	Description	
0	The SPB0DT bit value is not output to the TxD pin.	(Initial value)
1	The SPB0DT bit value is output to the TxD pin.	

Bit 0—Serial Port Break Data (SPB0DT): Specifies the serial port I/O data. Use the SPB0IO bit to specify input or output of TxD pin. See the description of SPB0IO for details. SPB0DT bit is output to the TxD pin when specified as output. The RxD pin value is read from the SPB0IO bit regardless of the SPB0IO bit value. The initial value is undefined.

Bit 0 : SPB0DT	Description	
0	I/O data level is low.	(Initial value)
1	I/O data level is high.	

Block diagrams of the SCI I/O port pins are shown in figures 15.2 to 15.4 in section 15, I/O Ports.

13.2.9 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 or standby mode. Each channel has independent baud rate generator control, so different values can be set in the two channels.

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

The SCBRR setting is calculated as follows:

Asynchronous mode: $N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

Synchronous mode: $N = [P\phi/(8 \times 2^{2n-1} \times B)] \times 10^6 - 1$

B: Bit rate (bit/s)

N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)

Pφ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 13.3.)

Table 13.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error for asynchronous mode is given by the following formula:

$$\text{Error (\%)} = \{P(\phi \times 10^{6})/[(N + 1) \times B \times 64 \times 2^{2n-1}] - 1 \} \times 100$$

Table 13.4 lists examples of SCBRR settings in asynchronous mode; table 13.5 lists examples of SCBRR settings in synchronous mode.

Table 13.4 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pϕ (MHz)								
	2			2.097152			2.4576		
Bit Rate (Bit/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	1	–14.67	0	1	0.00

Table 13.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	Pφ (MHz)								
	3			**3.6864**			**4**		
Bit Rate (Bit/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

	Pφ (MHz)								
	4.9152			**5**			**6**		
Bit Rate (Bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	86	0.31	2	88	–0.25	2	106	–0.44
150	1	255	0.00	2	64	0.16	2	77	0.16
300	1	127	0.00	1	129	0.16	1	155	0.16
600	0	255	0.00	1	64	0.16	1	77	0.16
1200	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	7	0.00	0	7	1.73	0	9	–2.34
31250	0	4	–1.70	0	4	0.00	0	5	0.00
38400	0	3	0.00	0	3	1.73	0	4	–2.34

Table 13.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	Pφ (MHz)								
	6.144			7.3728			8		
Bit Rate (Bit/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

	Pφ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (Bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 13.4 Bit Rates and SCBRR Settings in Asynchronous Mode (cont)

	Pφ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (Bit/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	3	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	2	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

	Pφ (MHz)											
	24			24.576			28.7			30		
Bit Rate (Bit/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 13.5 Bit Rates and SCBRR Settings in Synchronous Mode

	Pφ (MHz)									
Bit Rate	**4**		**8**		**16**		**28.7**		**30**	
(Bit/s)	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**
110	—	—	—	—	—	—	—	—	—	—
250	2	249	3	124	3	249	—	—	—	—
500	2	124	2	249	3	124	3	223	3	233
1k	1	249	2	124	2	249	3	111	3	116
2.5k	1	99	1	199	2	99	2	178	2	187
5k	0	199	1	99	1	199	2	89	2	93
10k	0	99	0	199	1	99	1	178	1	187
25k	0	39	0	79	0	159	1	71	1	74
50k	0	19	0	39	0	79	0	143	0	149
100k	0	9	0	19	0	39	0	71	0	74
250k	0	3	0	7	0	15	—	—	0	29
500k	0	1	0	3	0	7	—	—	0	14
1M	0	0*	0	1	0	3	—	—	—	—
2M	—	—	0	0*	0	1	—	—	—	—

Note: Settings with an error of 1% or less are recommended.

Legend

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmit/receive operation not possible

Table 13.6 shows the maximum bit rates in asynchronous mode when the baud rate generator is being used. Tables 13.7 and 13.8 list the maximum rates for external clock input.

Table 13.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pφ (MHz)	**Maximum Bit Rate (Bit/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 13.7 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pφ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bit/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 13.8 Maximum Bit Rates during External Clock Input (Synchronous Mode)

Pφ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bit/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

13.3 Operation

13.3.1 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 13.9. 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 13.10.

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 on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode:

- The transmission/reception format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 13.9 Serial Mode Register Settings and SCI Communication Formats

	SCSMR Settings					SCI Communication Format			
Mode	Bit 7 C/$\overline{\text{A}}$	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 used	Not used	1 bit
					1				2 bits
			1		0		Used		1 bit
					1				2 bits
		1	0		0	7-bit	Not used		1 bit
					1				2 bits
			1		0		Used		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not used	Used	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Synchronous	1	*	*	*	*	8-bit		Not used	None

Note: Asterisks (*) indicate don't-care bits.

Table 13.10 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSMR	SCSCR Settings		SCI Transmit/Receive Clock	
Mode	Bit 7 C/$\overline{\text{A}}$	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		

13.3.2 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 13.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The 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 16 times the bit rate. Receive data is latched at the center of each bit.

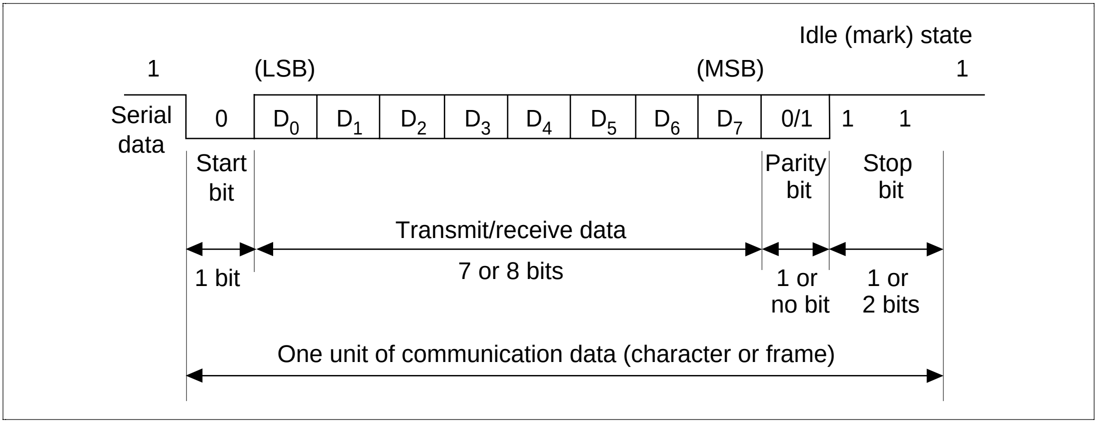


Figure 13.2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits)

Transmit/Receive Formats: Table 13.11 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 13.11 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	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	—	1	0	S	8-bit data								MPB	STOP	
0	—	1	1	S	8-bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Legend

— :	Don’t care bits
S:	Start bit
STOP:	Stop bit
P:	Parity bit
MPB:	Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{\text{A}}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 13.10).

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 shown in figure 13.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

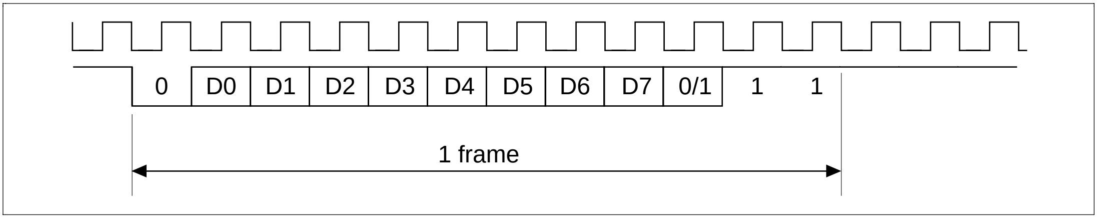


Figure 13.3 Output Clock and Serial Data Timing (Asynchronous Mode)

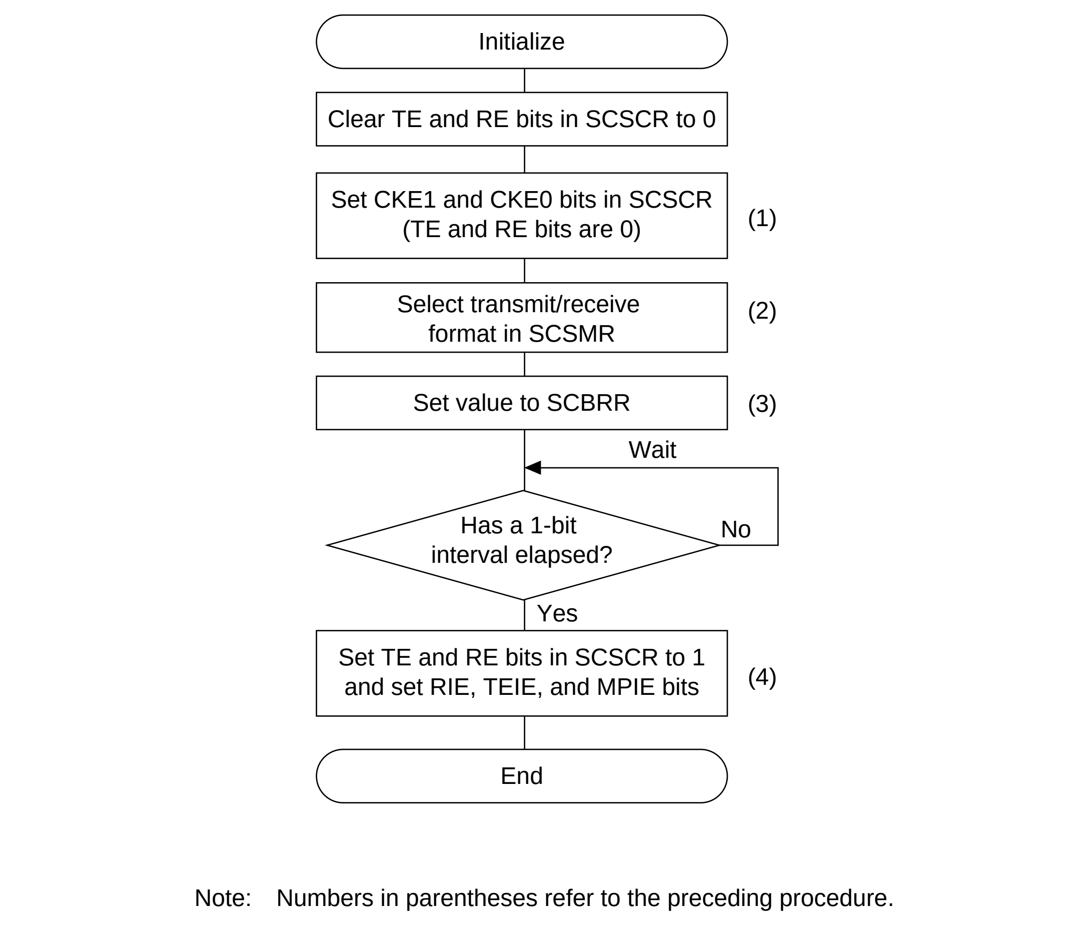
Transmitting and Receiving Data (SCI Initialization (Asynchronous Mode)): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

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 13.4 is a sample flowchart for initializing the SCI. The procedure for initializing the SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made 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).

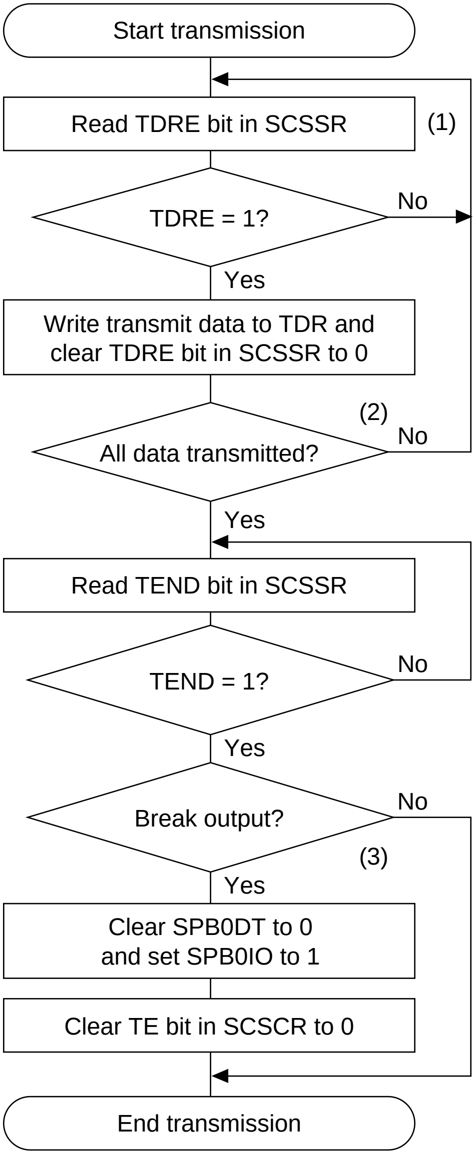

Initialize
Clear TE and RE bits in SCSCR to 0
Set CKE1 and CKE0 bits in SCSCR (TE and RE bits are 0)
(1)
Select transmit/receive format in SCSMR
(2)
Set value to SCBRR
(3)
Wait
Has a 1-bit interval elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TEIE, and MPIE bits
(4)
End

Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 13.5 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: Clear the SPB0DT bit in the SCSPTR, set the SPB0IO bit to 1 and then clear the TE bit to 0 in SCSCR.



Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.5 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 13.6 shows an example of SCI transmit operation in asynchronous mode.

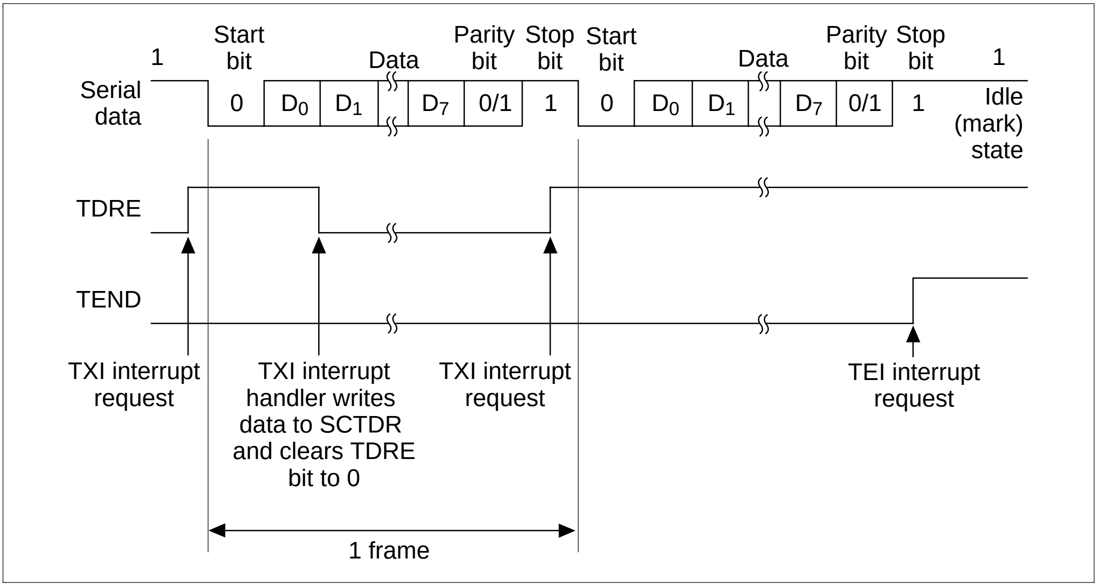
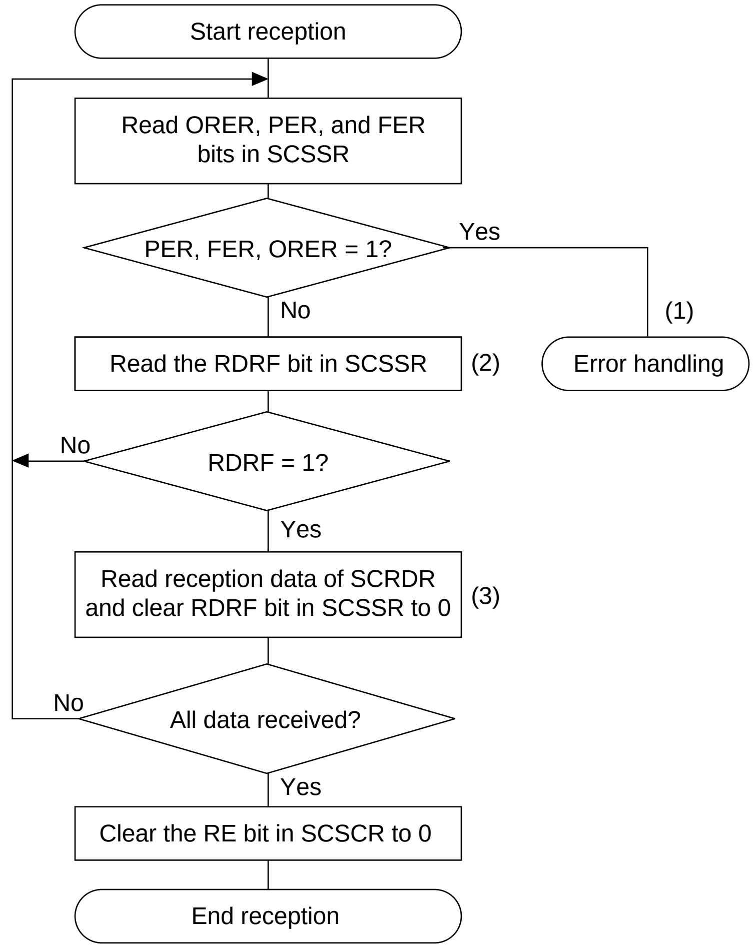


Figure 13.6 SCI Transmit Operation in Asynchronous Mode (Example: 8-Bit Data with Parity and One Stop Bit)

Receiving Serial Data (Asynchronous Mode): Figure 13.7 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 remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
2. SCI status check and receive-data read: Read the serial status register (SCSSR), check that RDRF is set to 1, then read receive data from the receive data register (SCRDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read the RDRF and SCRDR bits and clear RDRF to 0 before the stop bit of the current frame is received.


Start reception
Read ORER, PER, and FER bits in SCSSR
PER, FER, ORER = 1?
Yes
No
(1)
Read the RDRF bit in SCSSR
(2)
Error handling
No
RDRF = 1?
Yes
Read reception data of SCRDR and clear RDRF bit in SCSSR to 0
(3)
No
All data received?
Yes
Clear the RE bit in SCSCR to 0
End reception

Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.7 Sample Flowchart for Receiving Serial Data

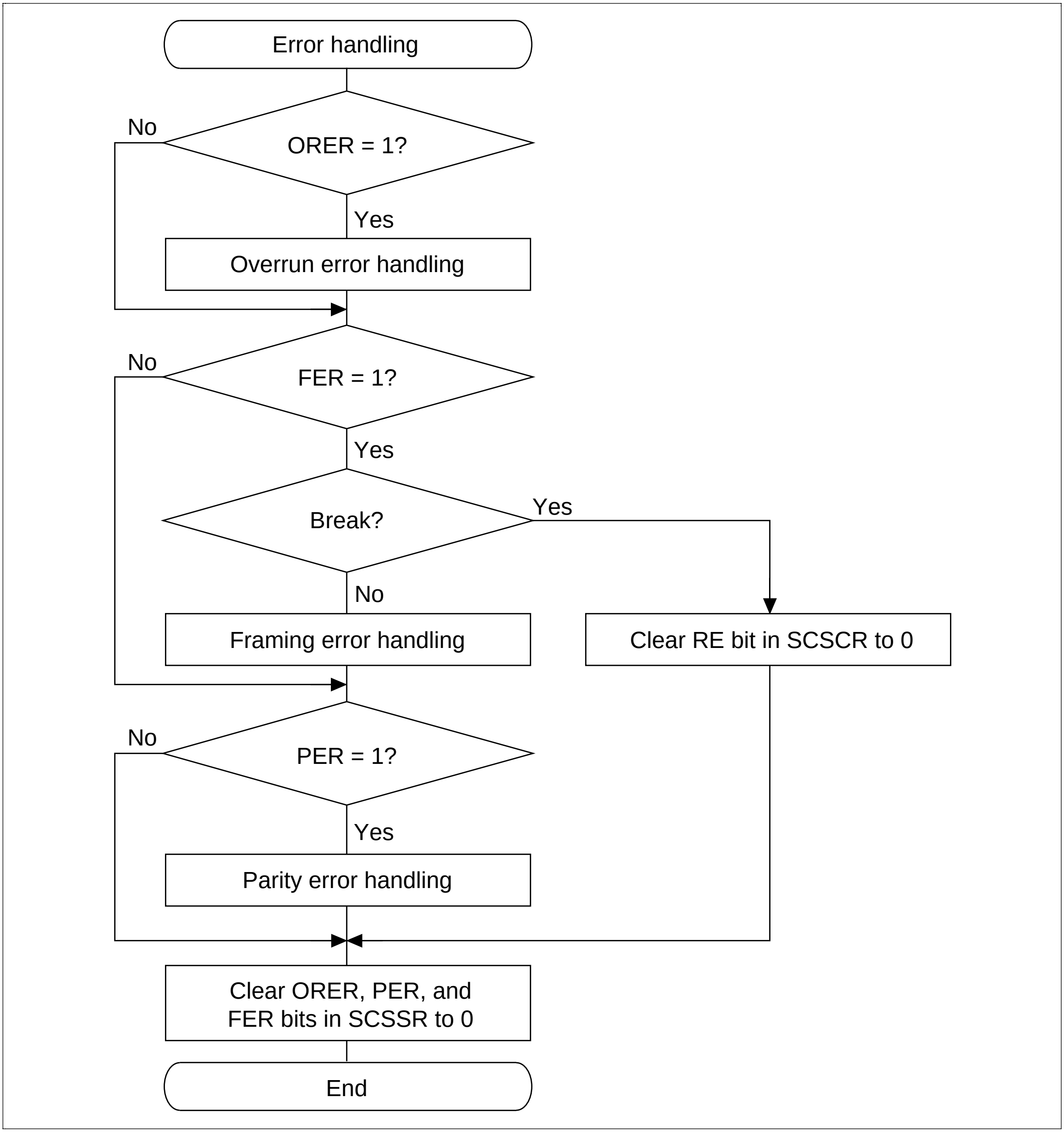


Figure 13.7 Sample Flowchart for Receiving Serial Data (cont)

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 13.12.

    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 13.12 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 13.8 shows an example of SCI receive operation in asynchronous mode.

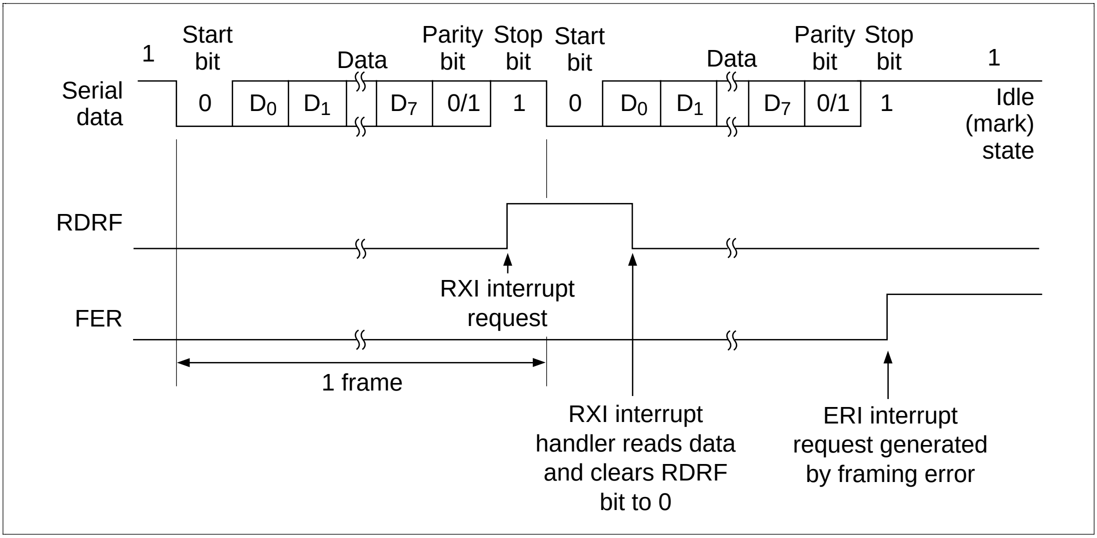


Figure 13.8 SCI Receive Operation (Example: 8-bit Data with Parity and One Stop Bit)

13.3.3 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. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 13.9 shows an example of communication among processors using the multiprocessor format.

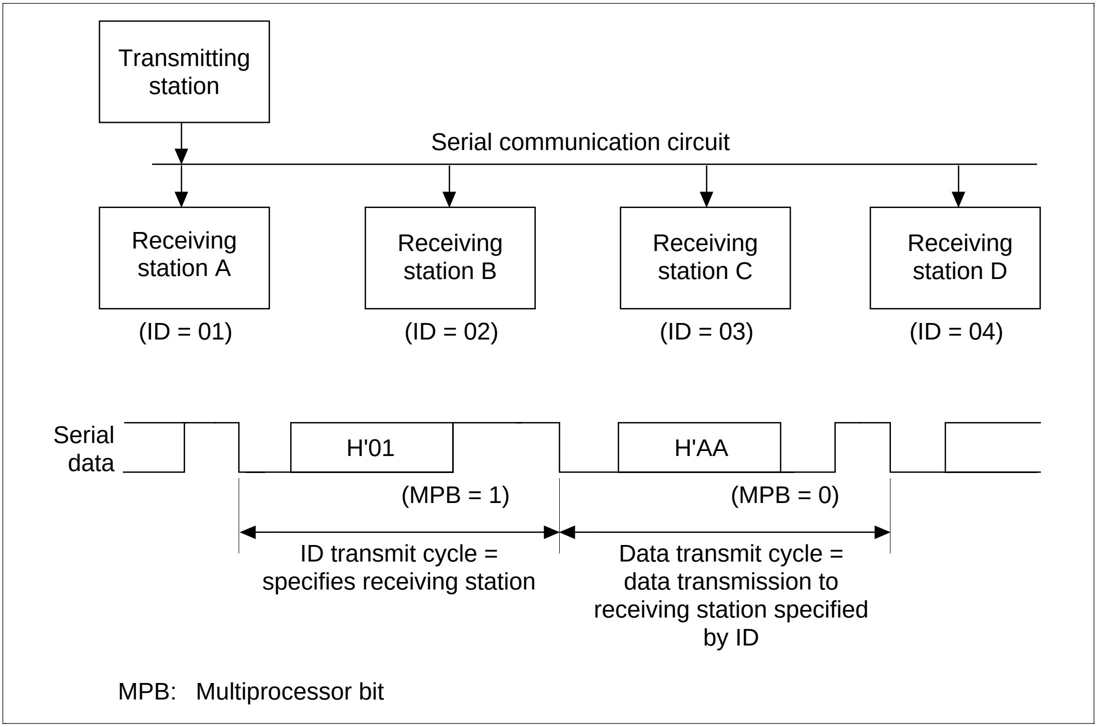


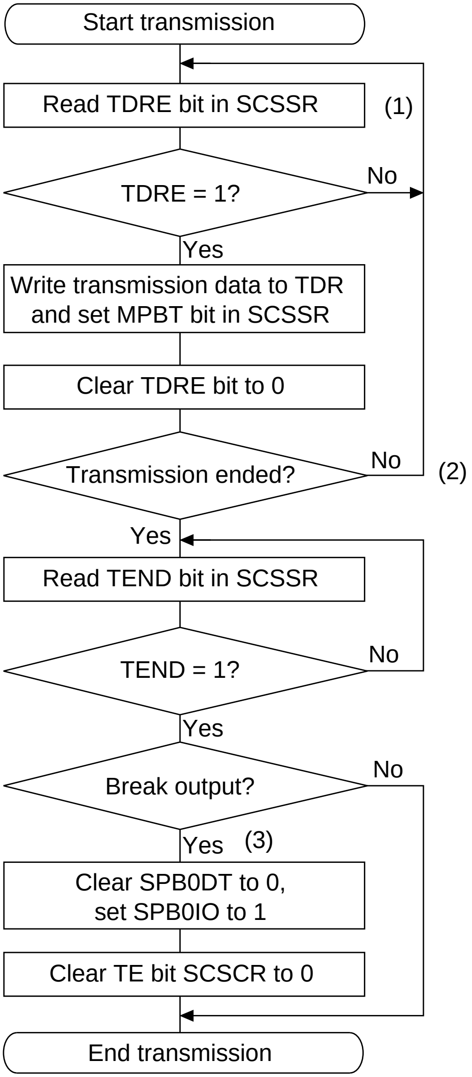
Figure 13.9 Communication Among Processors Using Multiprocessor Format (Example: Sending Data H'AA to Receiving Processor A)

Communication Formats: Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 13.11.

Clock: See the description in the asynchronous mode section.

Transmitting Multiprocessor Serial Data: Figure 13.10 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 SPB0DT bit in the SCSPTR register to 0, set SPB0IO to 1, then clear TE to 0 in SCSCR.



Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.10 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 the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits are output, LSB first.
   c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from 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 mark state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 13.11 shows SCI transmission with the multiprocessor format.

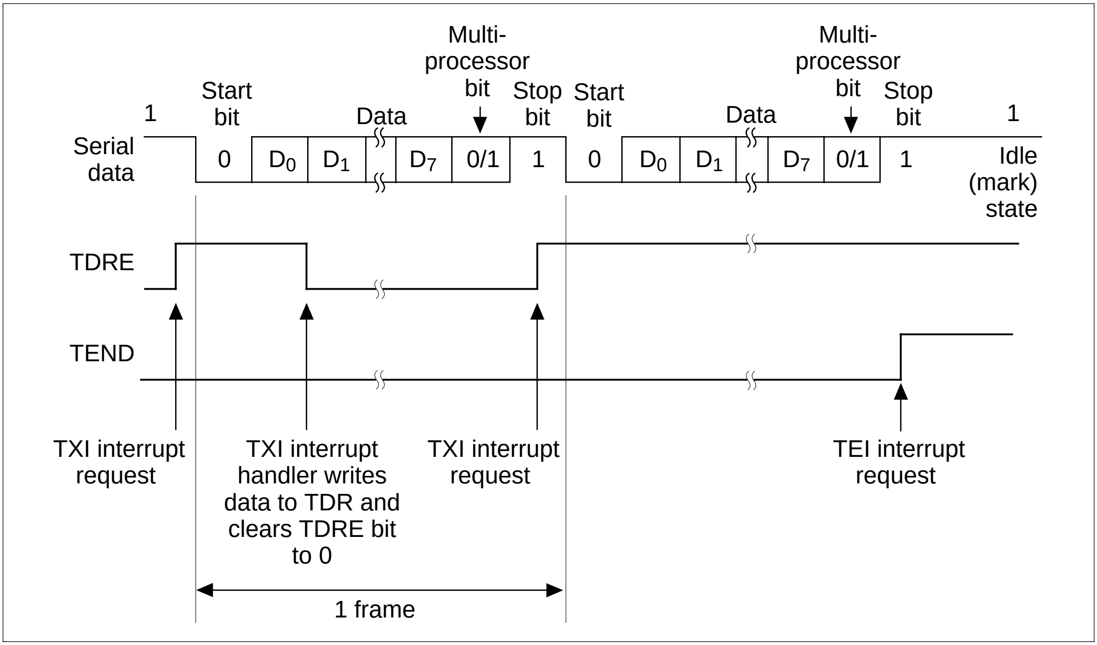
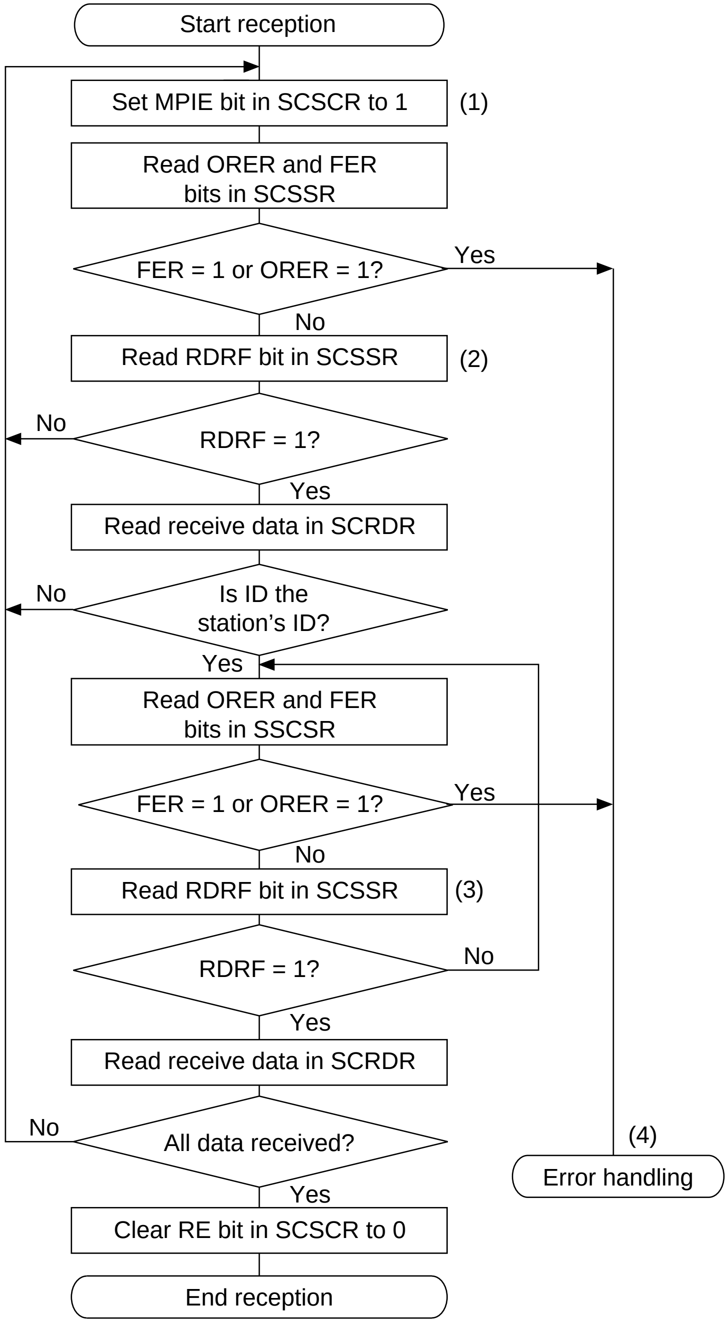


Figure 13.11 SCI Multiprocessor Transmit Operation (Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)

Receiving Multiprocessor Serial Data: Figure 13.12 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 remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.



Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data

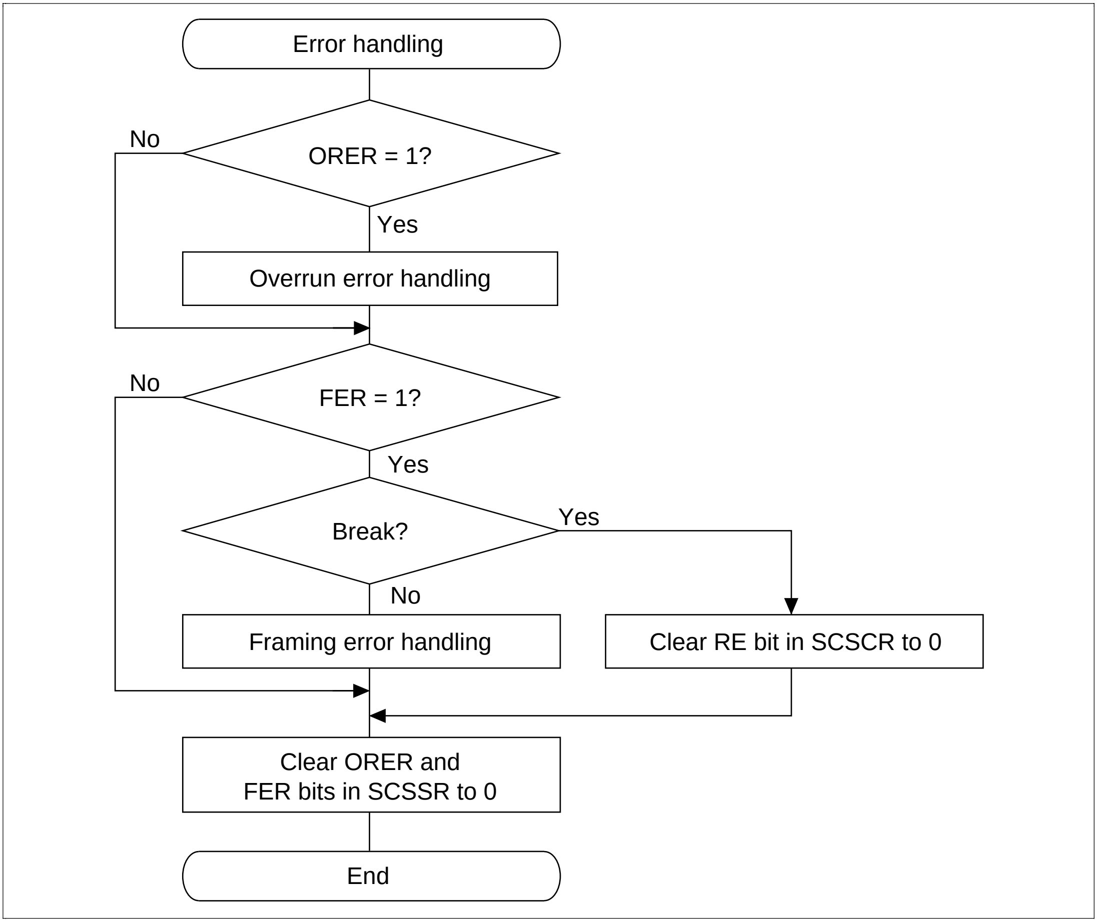


Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data (cont)

Figures 13.13 (a) and (b) show examples of SCI receive operation using a multiprocessor format.

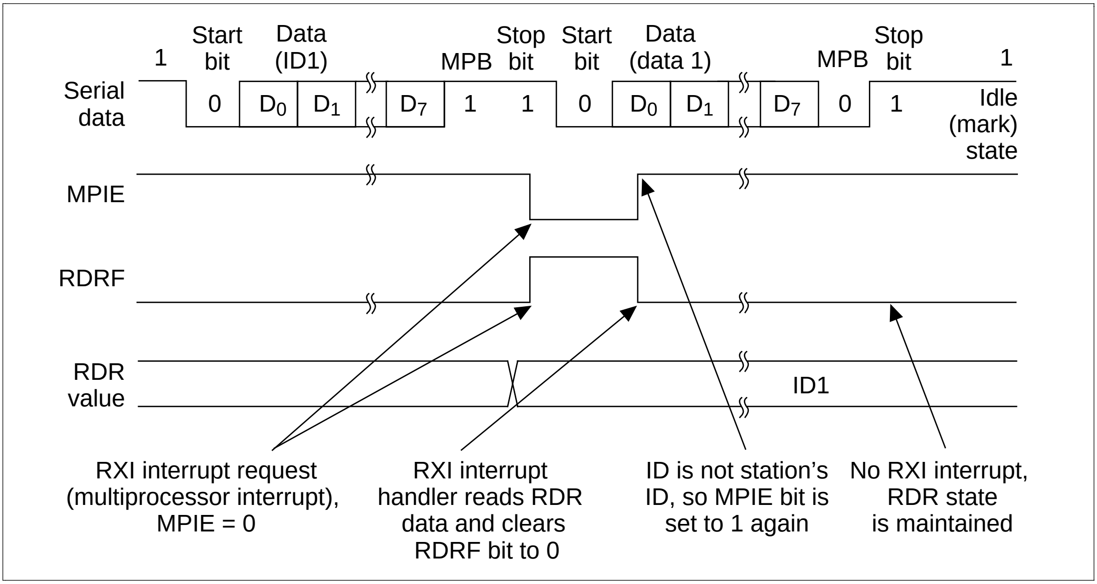

1
Start bit
Data (ID1)
MPB
Stop bit
Start bit
Data (data 1)
MPB
Stop bit
1
Serial data
0
D_0
D_1
D_7
1
1
0
D_0
D_1
D_7
0
1
Idle (mark) state
MPIE
RDRF
RDR value
ID1
RXI interrupt request (multiprocessor interrupt), MPIE = 0
RXI interrupt handler reads RDR data and clears RDRF bit to 0
ID is not station's ID, so MPIE bit is set to 1 again
No RXI interrupt, RDR state is maintained

Figure 13.13 (a) Example of SCI Receive Operation: Own ID Does Not Match Data (8-Bit Data with Multiprocessor Bit and One Stop Bit)

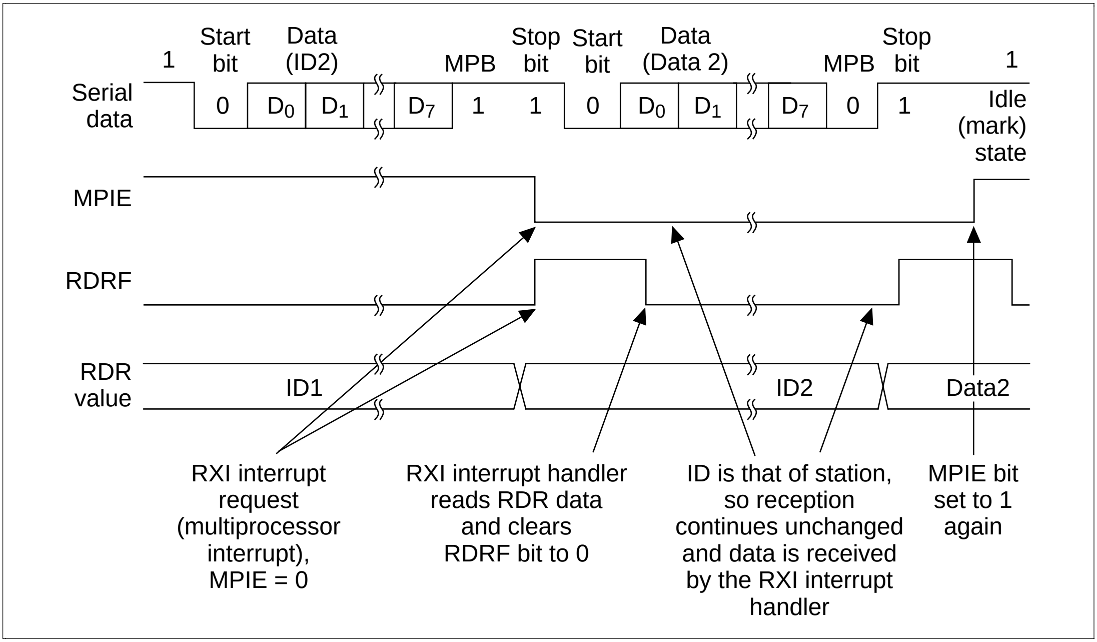

1
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data 2)
MPB
Stop bit
1
Serial data
0
D_0
D_1
D_7
1
1
0
D_0
D_1
D_7
0
1
Idle (mark) state
MPIE
RDRF
RDR value
ID1
ID2
Data2
RXI interrupt request (multiprocessor interrupt), MPIE = 0
RXI interrupt handler reads RDR data and clears RDRF bit to 0
ID is that of station, so reception continues unchanged and data is received by the RXI interrupt handler
MPIE bit set to 1 again

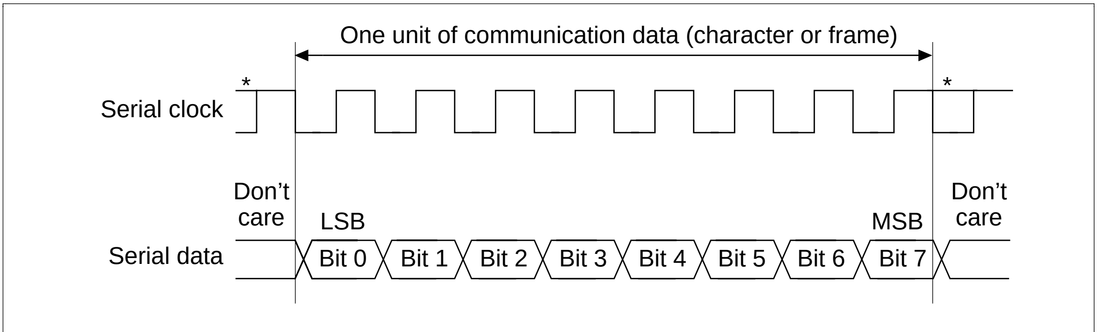
Figure 13.13 (b) Example of SCI Receive Operation: Own ID Matches Data (8-Bit Data with Multiprocessor Bit and One Stop Bit)

13.3.4 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 13.14 shows the general format in synchronous serial communication.



Note: * High except in continuous transmitting or receiving

Figure 13.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode, the SCI transmits or receives data by synchronizing with the falling edge of the serial clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the $C/\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR). See table 13.10.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the SCI receives in 2-

character units, so a 16 pulse synchronization clock is output. To receive in 1-character units, select an external clock source.

Transmitting and Receiving Data: SCI Initialization (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 13.15 is a sample flowchart for initializing the SCI. The procedure for initializing the SCI is:

1. Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE and RE cleared to 0.
2. Select the communication format in the serial mode register (SCSMR).
3. Write the value corresponding to the bit rate in the bit rate register (SCBRR) unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE and MPIE. Setting TE and RE allows use of the TxD and RxD pins.

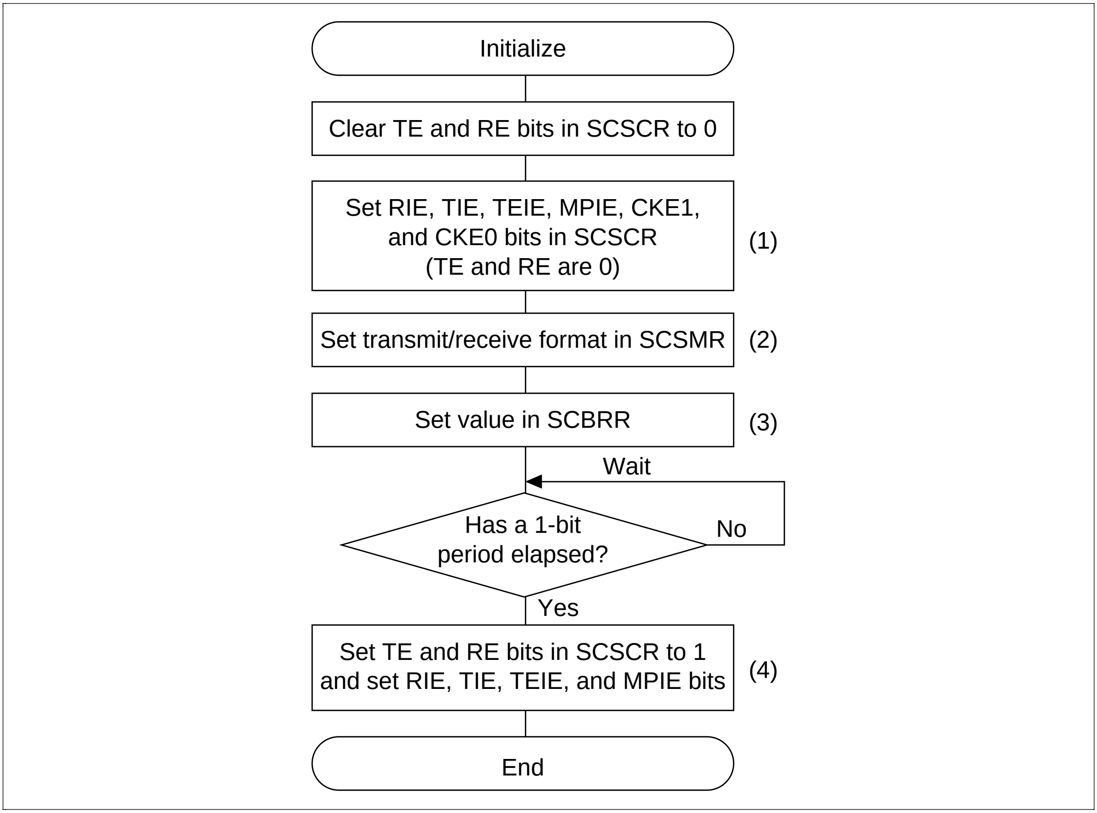
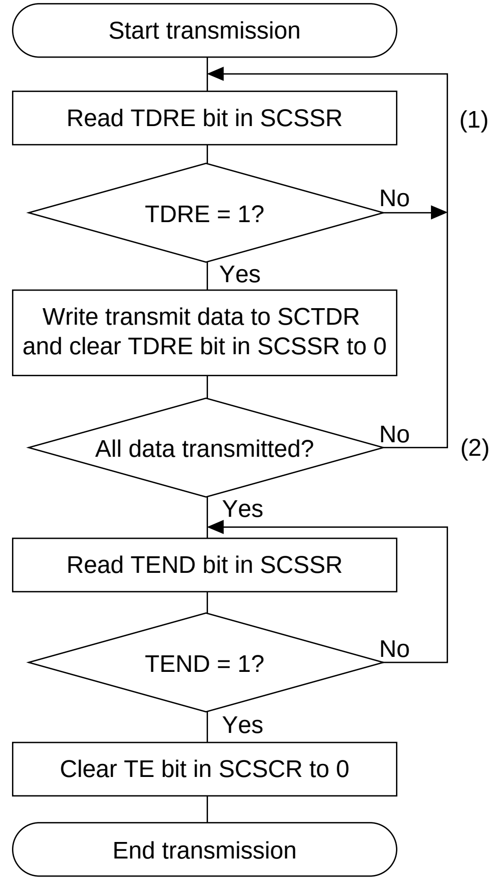

Initialize
Clear TE and RE bits in SCSCR to 0
Set RIE, TIE, TEIE, MPIE, CKE1, and CKE0 bits in SCSCR (TE and RE are 0)
(1)
Set transmit/receive format in SCSMR
(2)
Set value in SCBRR
(3)
Wait
Has a 1-bit period elapsed?
No
Yes
Set TE and RE bits in SCSCR to 1 and set RIE, TIE, TEIE, and MPIE bits
(4)
End

Figure 13.15 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 13.16 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.



Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.16 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 the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 13.17 shows an example of SCI transmit operation.

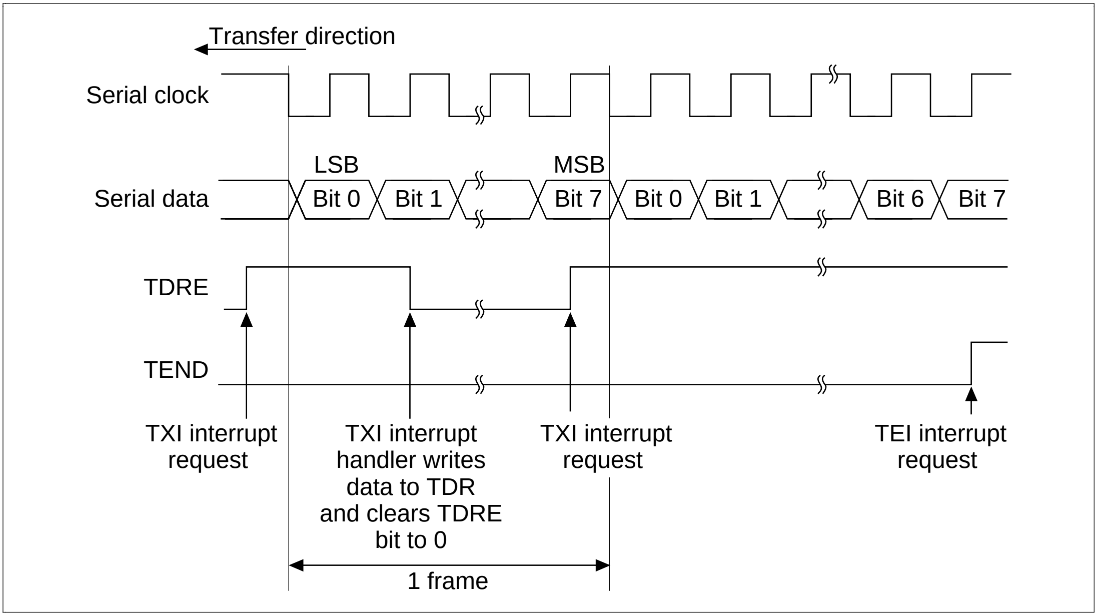
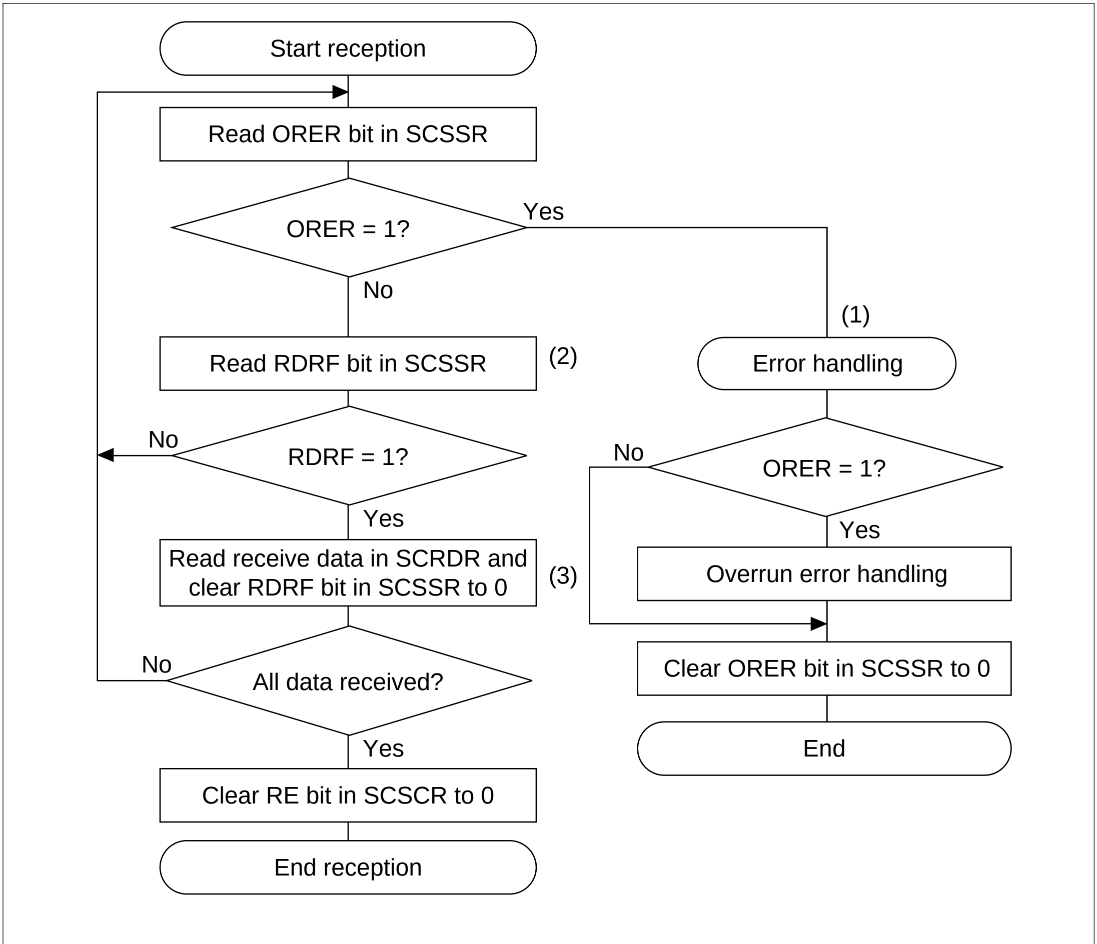


Figure 13.17 Example of SCI Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 13.18 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.



Note: Numbers in parentheses refer to the preceding procedure.

Figure 13.18 Sample Flowchart for Serial Receiving

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 13.12. This state prevents further transmission or reception. While receiving, the RDRF bit is not set to 1. Be sure to clear the error flag.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in 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 13.19 shows an example of the SCI receive operation.

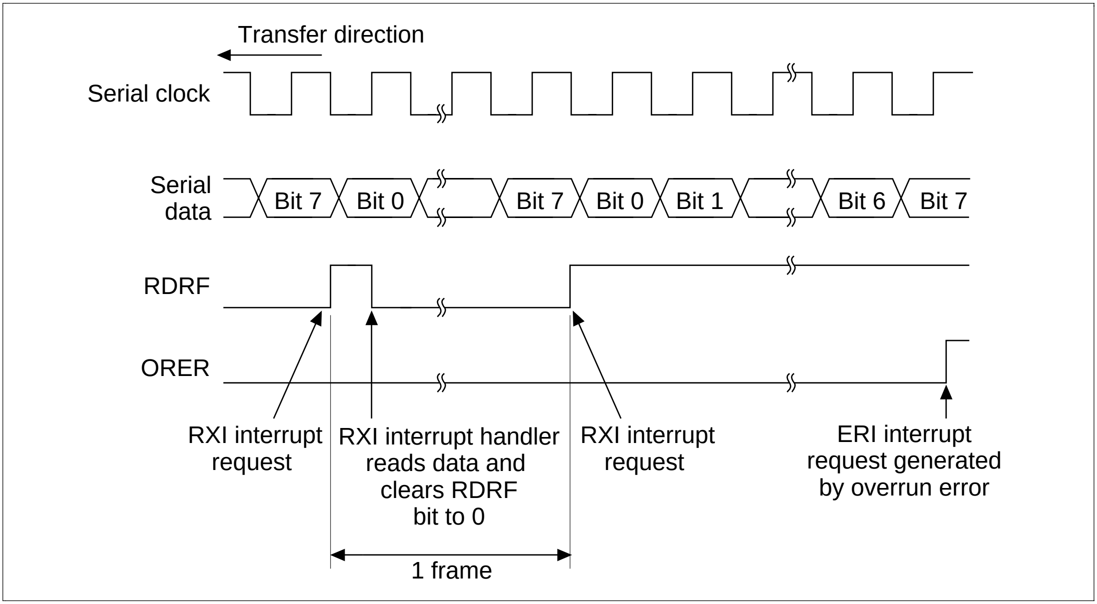
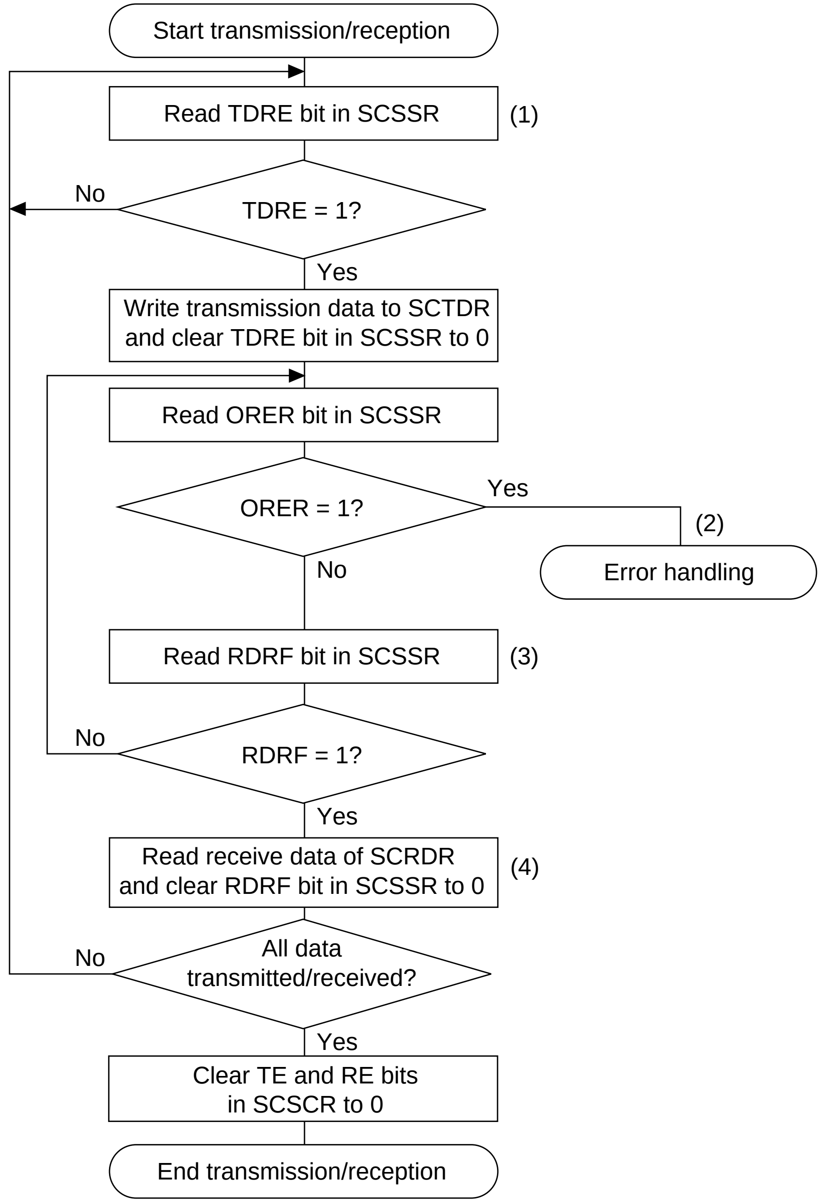


Figure 13.19 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 13.20 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 MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted.


Start transmission/reception
Read TDRE bit in SCSSR
(1)
No
TDRE = 1?
Yes
Write transmission data to SCTDR and clear TDRE bit in SCSSR to 0
Read ORER bit in SCSSR
ORER = 1?
Yes
(2)
Error handling
No
Read RDRF bit in SCSSR
(3)
No
RDRF = 1?
Yes
Read receive data of SCRDR and clear RDRF bit in SCSSR to 0
(4)
No
All data transmitted/received?
Yes
Clear TE and RE bits in SCSCR to 0
End transmission/reception

Notes: 1. When switching from transmitting or receiving to simultaneous transmitting and receiving, simultaneously clear TE and RE to 0, then simultaneously set TE and RE to 1.

2. Numbers in parentheses refer to the preceding procedure.

Figure 13.20 Sample Flowchart for Serial Transmitting

13.4 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 13.13 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 13.13 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is Cleared
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	
TXI	Transmit data empty (TDRE)	↓
TEI	Transmit end (TEND)	Low

See section 4, Exception Handling, for information on the priority order and relationship to non-SCI interrupts.

13.5 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 13.14 shows 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 13.14 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
Framing error	0	0	1	0	O
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

O: Receive data is transferred from SCRSR to SCRDR.

X: Receive data is not 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 input/output direction and level of the TxD pin can be set using the SPB0IO and SPB0DT bits in the serial port register (SCSPTR). Use these bits to send breaks. After initialization, the pin will not function as a TxD pin until the TE bit is set to 1 (enabling transmission). Through this period, the value of the SPB0DT bit substitutes for the mark state. For this reason, the SPB0IO and SPB0DT bits are initially set to 1 (output, high level). To send a break during serial transmission, clear the SPB0DT bit to 0 (low level), then clear TE to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized without regard to the current transmission status, and 0 is output from the TxD pin.

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 13.21).

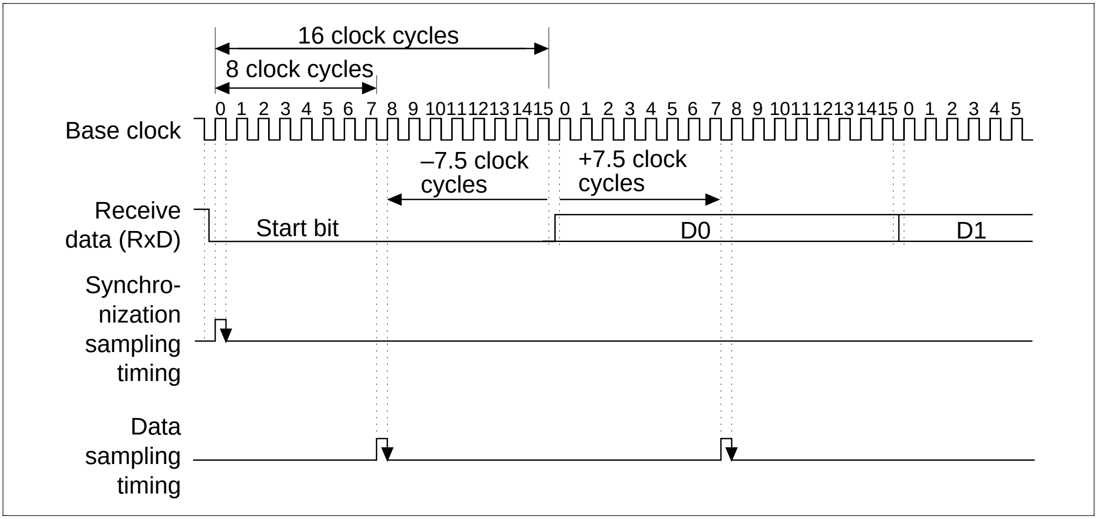


Figure 13.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in the 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 shown in equation 2.

Equation 2:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

Cautions on Use of Clock Synchronous External Clock Mode:

- Set TE = RE = 1 only when the external clock SCK is 1.
- Do not set TE = RE = 1 until at least four peripheral operating clock cycles after the external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 peripheral operating clock cycles after the rising edge of the RxD D7 bit SCK input, but it cannot be copied to SCRDR.

Caution on Use of Clock Synchronous Internal Clock Mode: When receiving, RDRF is 1 when RE is set to zero 1.5 peripheral operating clock cycles after the rising edge of the RxD D7 bit SCK output, but it cannot be copied to SCRDR.

Section 14 Smart Card Interface

14.1 Overview

As an added serial communications interface function, the SCI supports an IC card (smart card) interface that conforms to the ISO/IEC standard 7816-3 for identification of cards. Register settings are used to switch between the ordinary serial communication interface and the smart card interface.

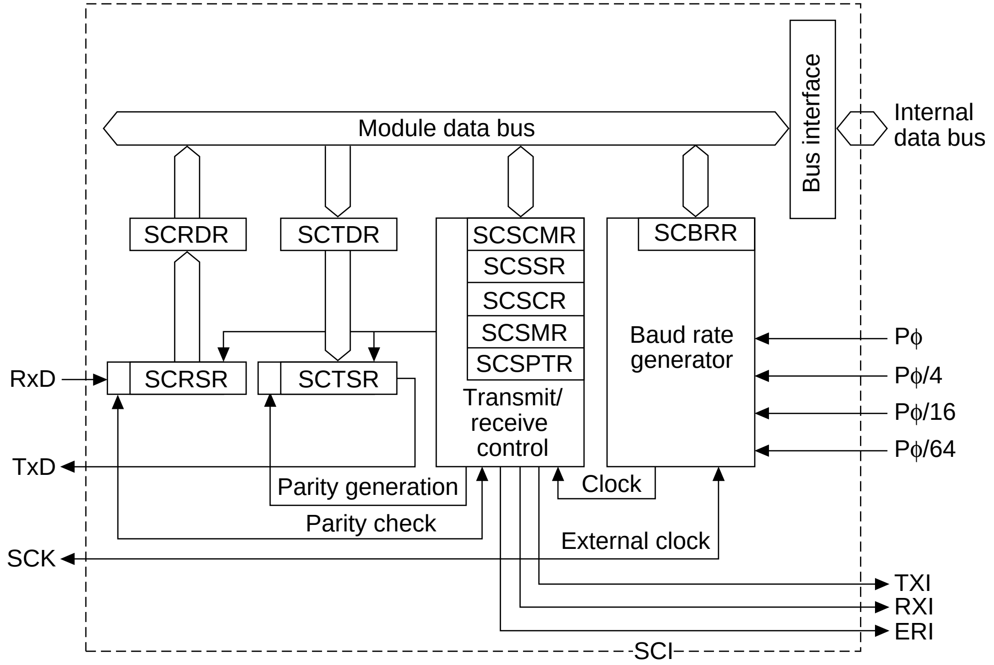
14.1.1 Features

The smart card interface has the following features:

- Asynchronous mode
  - Data length: Eight bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communication-error interrupts are requested independently.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the smart card interface.



SCSCMR:	Smart card mode register
SCRSR:	Receive data register
SCRDR:	Receive data register
SCTSR:	Transmit shift register
SCTDR:	Transmit data register
SCSMR:	Serial mode register
SCSCR:	Serial control register
SCSSR:	Serial status register
SCBRR:	Bit rate register
SCSPTR:	Serial port register

Figure 14.1 Smart Card Interface Block Diagram

14.1.3 Pin Configuration

Table 14.1 summarizes the smart card interface pins.

Table 14.1 SCI Pins

Pin Name	Abbreviation	Input/Output	Function
Serial clock pin	SCK	Output	Clock output
Receive data pin	RxD	Input	Receive data input
Transmit data pin	TxD	Output	Transmit data output

14.1.4 Register Configuration

Table 14.2 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 14.2 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	*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.

14.2 Register Descriptions

This section describes the registers added for the smart card interface and the bits whose functions are changed.

14.2.1 Smart Card Mode Register (SCSCMR)

The smart card mode register (SCSCMR) is an 8-bit read/write register that selects smart card interface functions. SCSMR bits 0, 2, and 3 are initialized to 0 by a reset and in standby mode.

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

Bits 7 to 4 and 1—Reserved: An undefined value will be returned if these bits are read.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3: SDIR	Description
0	Contents of SCTDR are transferred LSB first, receive data is stored in SCRDR LSB first. (Initial value)
1	Contents of SCTDR are transferred MSB first, receive data is stored in SCRDR MSB first.

Bit 2—Smart Card Data Inversion (SINV): Specifies whether to invert the logic level of the data. This function is used in combination with bit 3 for transmitting and receiving with an inverse convention card. SINV does not affect the logic level of the parity bit. See section 14.3.4, 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

14.2.2 Serial Status Register (SCSSR)

In the smart card interface mode, the function of SCSSR bit 4 is changed. The setting conditions for bit 2, the TEND bit, are also changed.

Bit:	7	6	5	4	3	2	1	0
Bit name:	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: Only 0 can be written, to clear the flag.

Bits 7 to 5: These bits have the same function as in the ordinary SCI. See section 13, Serial Communication Interface, for more information.

Bit 4—Error Signal Status (ERS): In the smart card interface mode, bit 4 indicates the status of the error signal returned from the receiving side during transmission. The smart card interface cannot detect framing errors.

Bit 4: ERS	Description
0	Receiving ended normally with no error signal. (Initial value) ERS is cleared to 0 when the chip is reset or enters standby mode, or when software reads ERS after it has been set to 1, then writes 0 in ERS.
1	An error signal indicating a parity error was transmitted from the receiving side. ERS is set to 1 if the error signal sampled is low.

Note: The ERS flag maintains its status even when the TE bit in SCSCR is cleared to 0.

Bits 3 to 0: These bits have the same function as in the ordinary SCI. See section 13, Serial Communication Interface, for more information. The setting conditions for bit 2, the transmit end bit (TEND), are changed as follows.

Bit 2: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE, or when data is written in SCTDR.
1	End of transmission. (Initial value) TEND is set to 1 when: • the chip is reset or enters standby mode, • the TE bit in SCSCR is 0 and the FER/ERS bit is also 0, • the C/$\overline{\text{A}}$ 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/$\overline{\text{A}}$ 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.

14.3 Operation

14.3.1 Overview

The primary functions of the smart card interface are described below.

1. Each frame consists of 8 data bits and 1 parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (elementary time units: the period for 1 bit to transfer) from the end of the parity bit to the start of the next frame.
2. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu has elapsed from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at least 2 etu from the time the error signal is sampled.
5. The specification complies with ISO/ICE7816-3, but the only type of data transmission protocol supported is protocol type T = 0 : asynchronous double-character transmission protocol.

14.3.2 Pin Connections

Figure 14.2 shows the pin connection diagram for the smart card interface. During communication with an IC card, transmission and reception are both carried out over the same data transfer line, so connect the TxD and RxD pins on the chip. Pull up the data transfer line to the power supply V_{CC} side with a resistor.

When using the clock generated by the smart card interface on an IC card, input the SCK pin output to the IC card's CLK pin. This connection is not necessary when the internal clock is used on the IC card.

Use the chip's port output as the reset signal. Apart from these pins, the power and ground pin connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed communication is possible and auto-diagnosis can be performed.

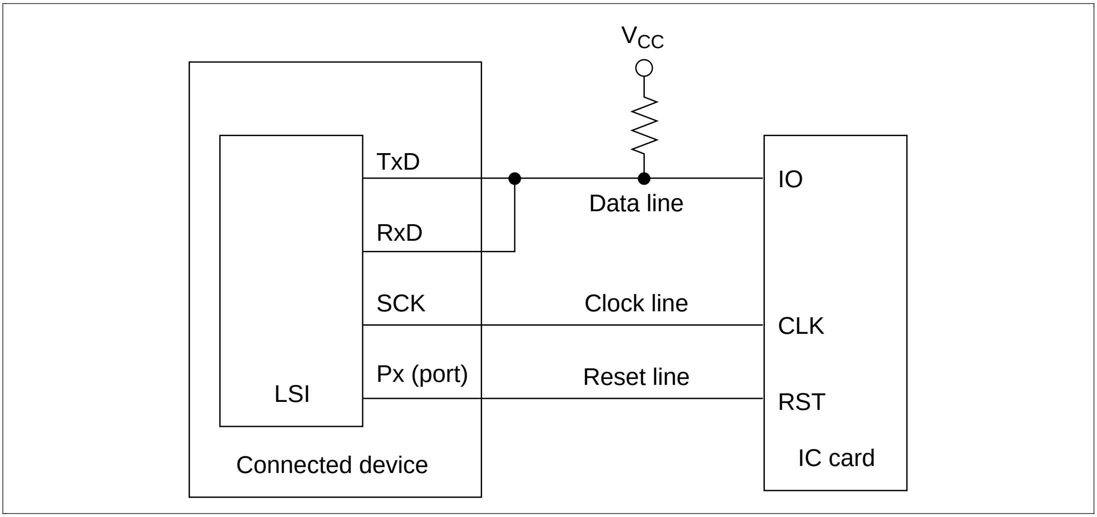


Figure 14.2 Pin Connection Diagram for the Smart Card Interface

14.3.3 Data Format

Figure 14.3 shows the data format for the smart card interface. In this mode, parity is checked every frame while receiving and error signals sent to the transmitting side whenever an error is detected so that data can be re-transmitted. During transmission, error signals are sampled and data re-transmitted whenever an error signal is detected.

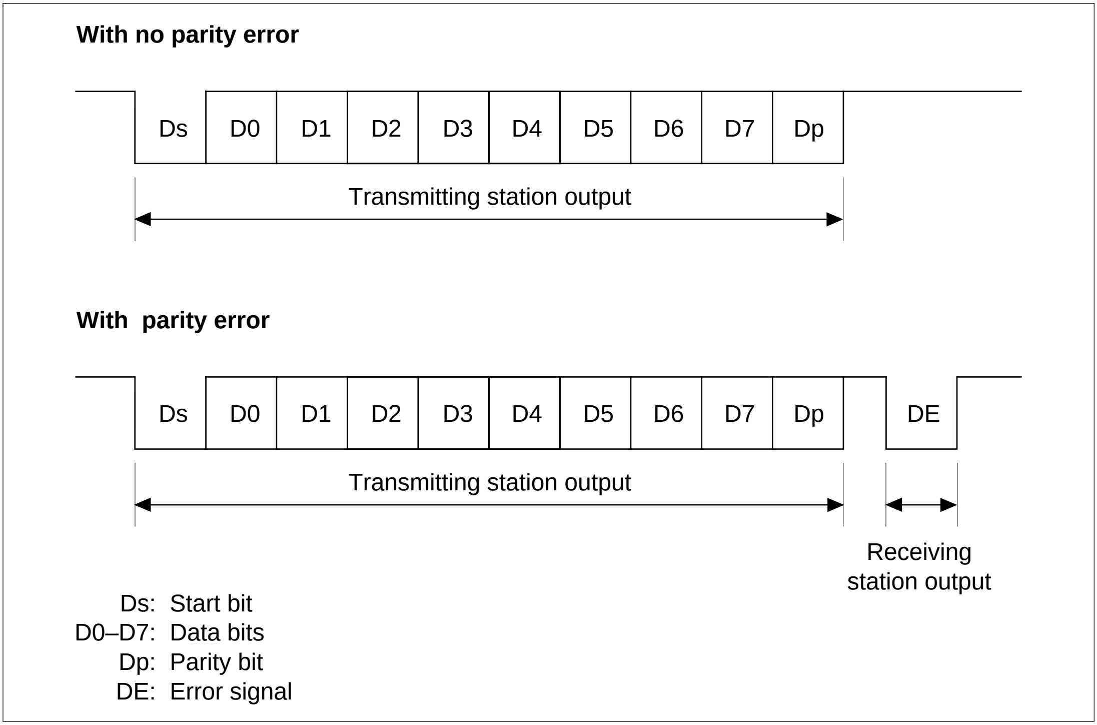


Figure 14.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is high impedance when not in use and is fixed high with a pull-up resistor.
2. The transmitting side starts one frame of data transmission. The data frame starts with a start bit (Ds, low level). The start bit is followed by eight data bits (D0–D7) and a parity bit (Dp).
3. On the smart card interface, the data line returns to high impedance after this. The data line is pulled high with a pull-up resistor.
4. The receiving side checks parity. When the data is received normally with no parity errors, the receiving side then waits to receive the next data. When a parity error occurs, the receiving side outputs an error signal (DE, low level) and requests re-transfer of data. The receiving station returns the signal line to high impedance after outputting the error signal for a specified period. The signal line is pulled high with a pull-up resistor.
5. The transmitting side transmits the next frame of data unless it receives an error signal. If it does receive an error signal, it returns to step 2 to re-transmit the erroneous data.

14.3.4 Register Settings

Table 14.3 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 14.3 Register Settings for the Smart Card Interface

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCSMR	H'FFFFFE80	$C/\overline{A}$	0	1	$O/\overline{E}$	1	0	CKS1	CKS0
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE1	CKE0
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	0	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—	SMIF

Note: Dashes indicate unused bits.

1. Setting the serial mode register (SCSMR): Set the $O/\overline{E}$ bit to 0 when the IC card uses the direct convention or to 1 when it uses the inverse convention. Select the on-chip baud rate generator clock source with the CKS1 and CKS0 bits (see section 14.3.5, Clock). When bit 7 (C/A) of the serial mode register (SCSMR) is set to 1 (default value : 1), bit 2 (TEND) of the serial status register (SCSSR) is simply set to 1 (TXI interrupt request) 1 etu (default value : 2.5 etu) after transmission of a 1-byte character, and 11 etu continuous transmission (block transfer protocol) cannot be performed.
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 14.3.5, Clock, to see how to calculate the set value.
3. Setting the serial control register (SCSCR): The TIE, RIE, TE and RE bits function as they do for the ordinary SCI. See section 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 set to 0 for IC cards that use the direct convention and both to 1 when the inverse convention is used. The SMIF bit is set to 1 for the smart card interface.

Figure 14.4 shows sample waveforms for register settings of the two types of IC cards (direct convention and inverse convention) and their start characters.

In the direct convention type, the logical 1 level is state Z, the logical 0 level is state A, and communication is LSB first. The start character data is H'3B. The parity bit is even (from the smart card standards), and thus a 1.

In the inverse convention type, the logical 1 level is state A, the logical 0 level is state Z, and communication is MSB first. The start character data is H'3F. The parity bit is even (from the smart card standards), and thus a 0, which corresponds to state Z.

Only data bits D7–D0 are inverted by the SINV bit. To invert the parity bit, set the $O/\overline{E}$ bit in SCSMR to odd parity mode. This applies to both transmission and reception.

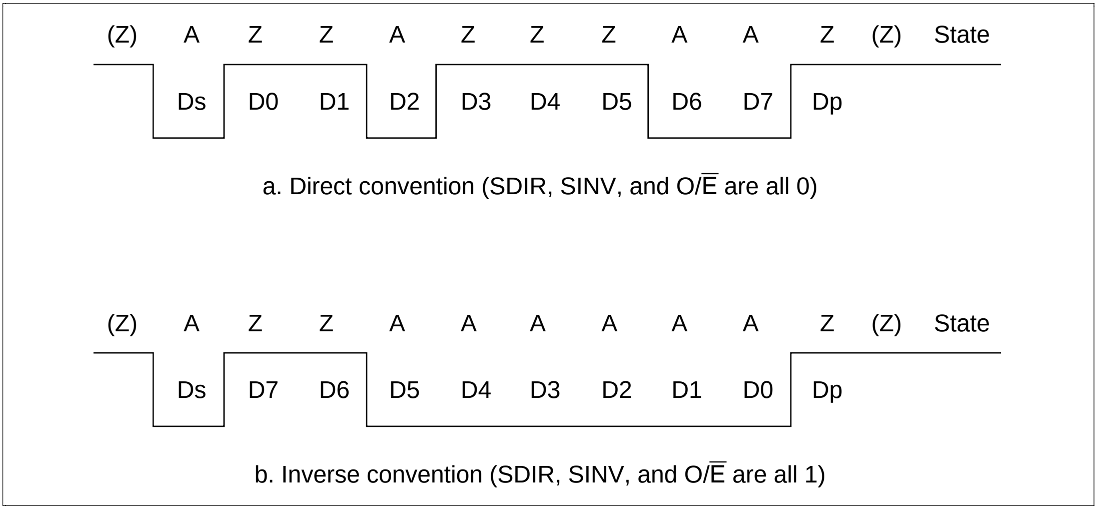

(Z)
A
Z
Z
A
Z
Z
Z
A
A
Z
(Z)
State
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp
a. Direct convention (SDIR, SINV, and O/$\overline{E}$ are all 0)
(Z)
A
Z
Z
A
A
A
A
A
A
Z
(Z)
State
Ds
D7
D6
D5
D4
D3
D2
D1
D0
Dp
b. Inverse convention (SDIR, SINV, and O/$\overline{E}$ are all 1)

Figure 14.4 Waveform of Start Character

14.3.5 Clock

Only the internal clock generated by the on-chip baud rate generator can be used as the communication clock in the smart card interface. The bit rate for the clock is set by the bit rate register (SCBRR) and the CKS1 and CKS0 bits in the serial mode register (SCSMR), and is calculated using the equation below. Table 14.5 shows sample bit rates. If clock output is then selected by setting CKE0 to 1, a clock with a frequency 372 times the bit rate is output from the SCK0 pin.

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where:

N = Value set in SCBRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
Pϕ = Peripheral module operating frequency (MHz)*
n = 0–3 (table 14.4)

Table 14.4 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 14.5 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)						
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**	**18.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: The bit rate is rounded to two decimal places.

Calculate the value to be set in the bit rate register (SCBRR) from the operating frequency and the bit rate. N is an integer in the range 0 ≤ N ≤ 255, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^{6} - 1$$

Table 14.6 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 14.7 Maximum Bit Rates for Frequencies (Smart Card Interface Mode)

Pφ (MHz)	Maximum Bit Rate (Bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error}(\%) = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N+1)} \times 10^6 - 1\right) \times 100$$

Table 14.8 shows the relationship between transmit/receive clock register set values and output states on the smart card interface.

Table 14.8 Register Set Values and SCK Pin

	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{\text{A}}$**	**CKE1**	**CKE0**	**Output**	**State**
1*1	1	0	0	0	Port	Determined by setting of port register SPB1IO and SPB1DT bits
	1	0	0	1	⎍⎍	SCK (serial clock) output state
2*2	1	1	0	0	Low output	Low output state
	1	1	0	1	⎍⎍	SCK (serial clock) output state
3*2	1	1	1	0	High output	High output state
	1	1	1	1	⎍⎍	SCK (serial clock) output state

Notes: 1. The SCK output state changes as soon as the CKE0 bit is modified. The CKE1 bit should be cleared to 0.

2. The clock duty remains constant despite stopping and starting of the clock by modification of the CKE0 bit.

14.3.6 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 14.5 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.

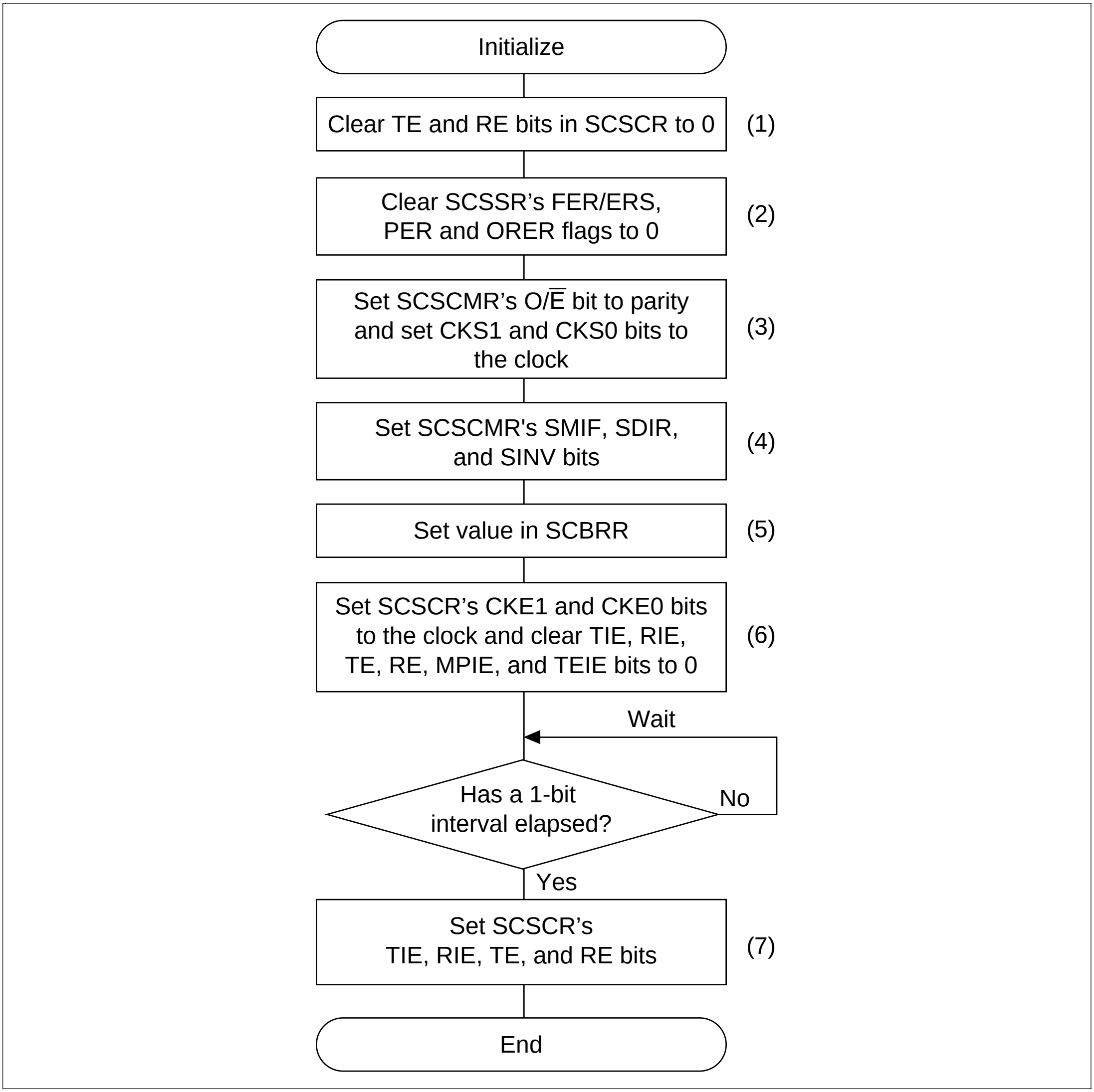

Initialize
Clear TE and RE bits in SCSCR to 0
(1)
Clear SCSSR's FER/ERS, PER and ORER flags to 0
(2)
Set SCSCMR's O/$\overline{E}$ bit to parity and set CKS1 and CKS0 bits to the clock
(3)
Set SCSCMR's SMIF, SDIR, and SINV bits
(4)
Set value in SCBRR
(5)
Set SCSCR's CKE1 and CKE0 bits to the clock and clear TIE, RIE, TE, RE, MPIE, and TEIE bits to 0
(6)
Wait
Has a 1-bit interval elapsed?
No
Yes
Set SCSCR's TIE, RIE, TE, and RE bits
(7)
End

Figure 14.5 Initialization Flowchart (Example)

Serial Data Transmission: The handling procedures in the smart card mode differ from ordinary SCI processing because data is retransmitted when an error signal is sampled during a data transmission. This results in the transmission processing flowchart shown in figure 14.6.

1. Initialize the smart card interface mode as described in initialization above.
2. Check that the FER/ERS bit in SCSSR is cleared to 0.
3. Repeat steps 2 and 3 until the TEND flag in SCSSR is set to 1.
4. Write the transmit data into SCTDR, clear the TDRE flag to 0 and start transmitting. The TEND flag will be cleared to 0.
5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

This processing can be interrupted. When the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested when the TEND flag is set to 1 at the end of the transmission. When the RIE bit is set to 1 and interrupt requests are enabled, a communication error interrupt (ERI) will be requested when the ERS flag is set to 1 when an error occurs in transmission. See Interrupt Operation below for more information.

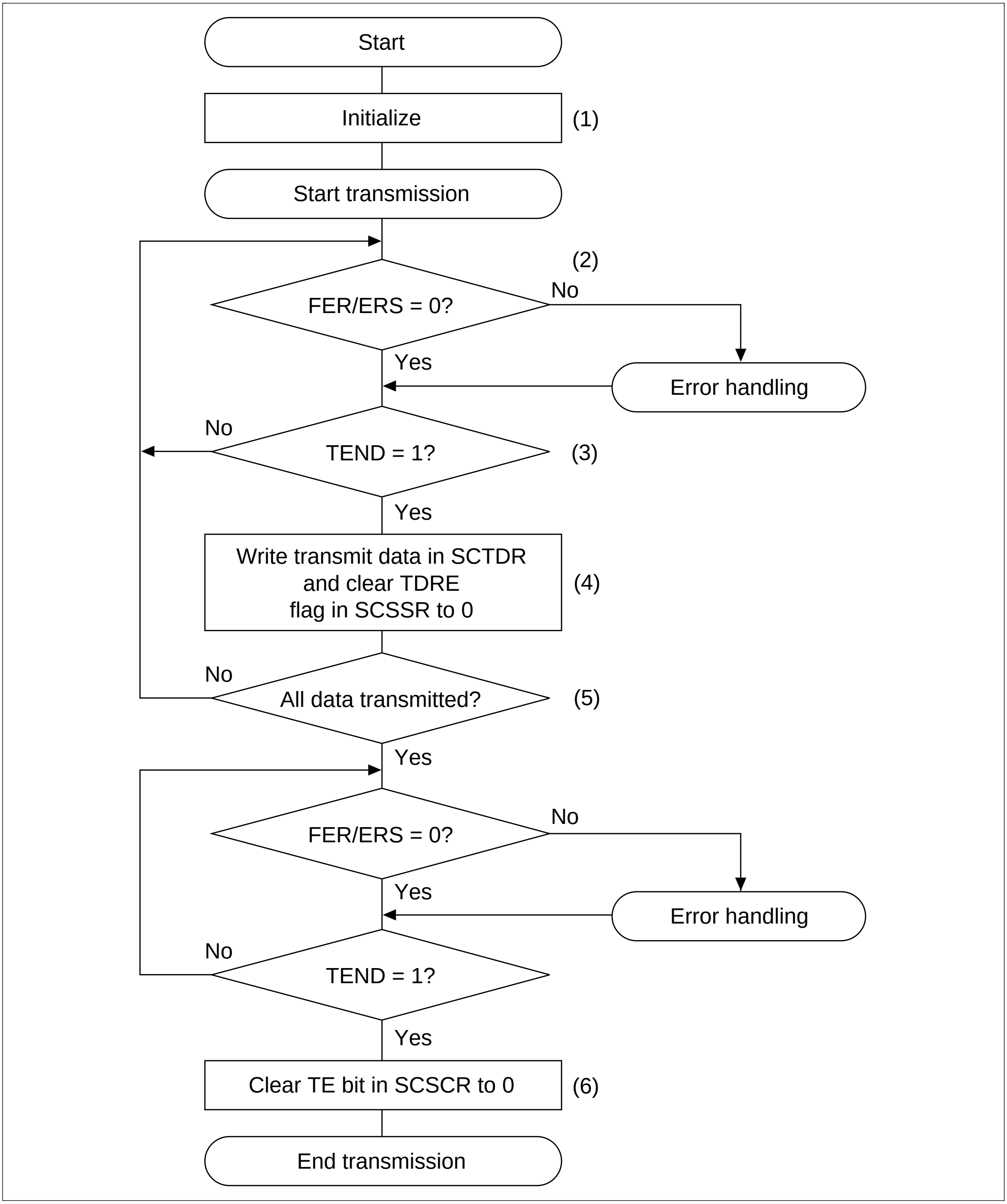


Figure 14.6 Transmission Flowchart

Serial Data Reception: The handling procedures in the smart card mode are the same as in ordinary SCI processing. The reception processing flowchart is shown in figure 14.7.

1. Initialize the smart card interface mode as described above in Initialization and in figure 14.5.
2. Check that the ORER and PER flags in SCSSR are cleared to 0. If either flag is set, clear both to 0 after performing the appropriate error handling procedures.
3. Repeat steps 2 and 3 until the RDRF flag is set to 1.
4. Read the receive data from SCRDR.
5. To receive more data, clear the RDRF flag to 0 and return to step 2.
6. To end reception, clear the RE bit to 0.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at the end of the reception. When an error occurs during reception and either the ORER or PER flag is set to 1, a communication error interrupt (ERI) will be requested. See Interrupt Operation, below, for more information.

The received data will be transferred to SCRDR even when a parity error occurs during reception and PER is set to 1, so this data can still be read.

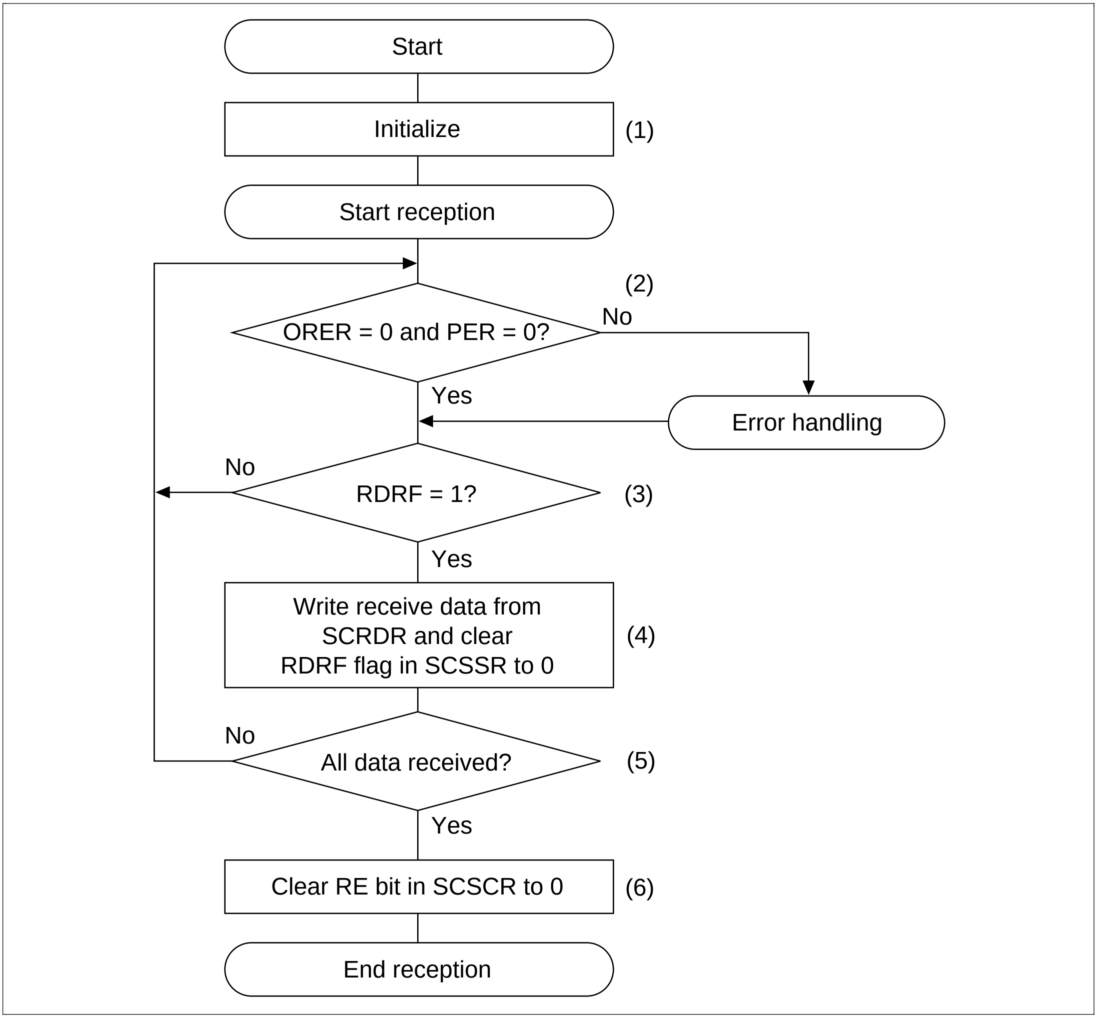


Figure 14.7 Reception Flowchart (Example)

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 14.9).

Table 14.9 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

14.4 Usage Notes

When the SCI is used as a smart card interface, be sure that all criteria in sections 14.4.1 and 14.4.2 are applied.

14.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI runs on a basic clock with a frequency of 372 times the transfer rate. During reception, the SCI samples the fall of the start bit using the base clock to achieve internal synchronization. Receive data is latched internally on the rising edge of the 186th basic clock cycle (figure 14.8).

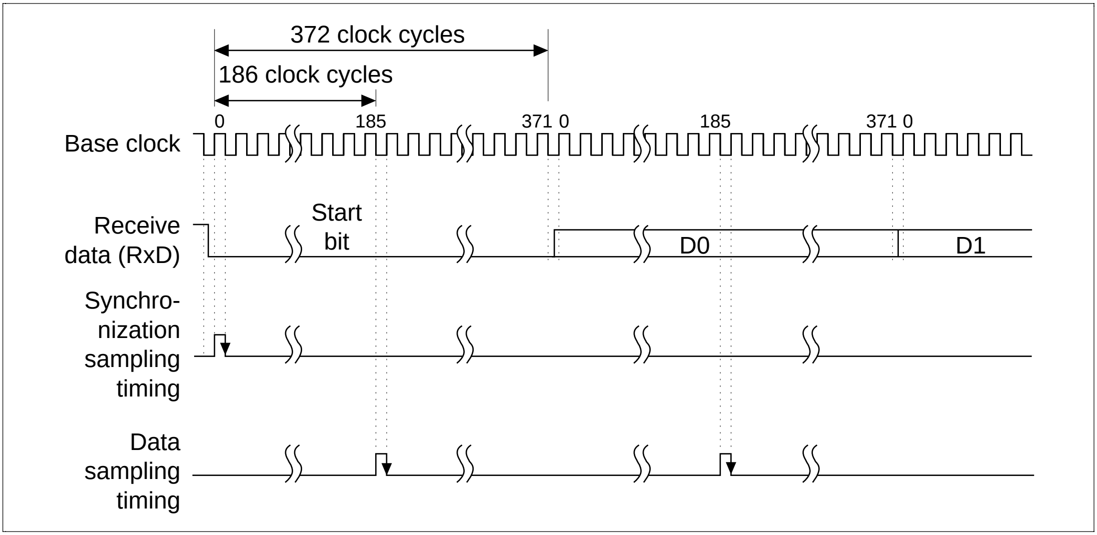


Figure 14.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Where:

M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

14.4.2 Retransmission (Receive and Transmit Modes)

Retransmission by the SCI in Receive Mode: Figure 14.9 shows the retransmission operation in the SCI receive mode.

1. When the received parity bit is checked and an error is found, the PER bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the PER bit before the next parity bit is sampled.
2. The RDRF bit in SCSSR is not set in the frame that caused the error.
3. When the received parity bit is checked and no error is found, the PER bit in SCSSR is not set.
4. When the received parity bit is checked and no error is found, reception is considered to have been completed normally and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state status when it transmits the error signal.

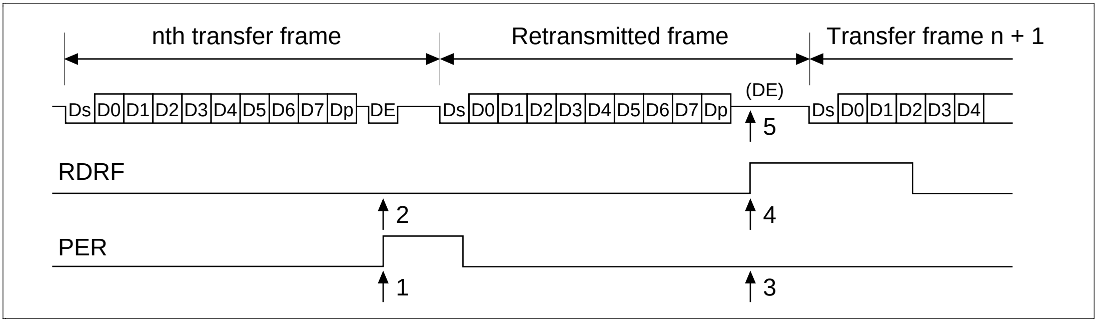


Figure 14.9 Retransmission in SCI Receive Mode

Retransmission by the SCI in Transmit Mode: Figure 14.10 shows the retransmission operation in the SCI transmit mode.

1. After transmission of one frame is completed, the FER/ERS bit in SCSSR is set to 1 when a error signal is returned from the receiving side. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the FER/ERS bit before the next parity bit is sampled.
2. The TEND bit in SCSSR is not set in the frame that received the error signal that indicated the error.
3. The FER/ERS bit in SCSR is not set when no error signal is returned from the receiving side.
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is set to 1 when the transmission of the frame that includes the retransmission is considered completed. If the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.

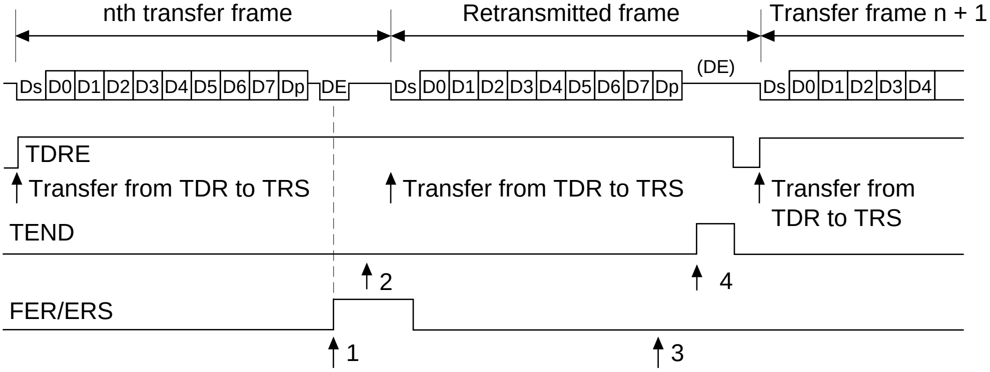


Figure 14.10 Retransmission in SCI Transmit Mode

15.1 Overview

The has an on-chip 8-bit general-purpose I/O port and an on-chip I/O port for the serial communication interface (SCI).

15.1.1 Features

The general-purpose I/O port has the following features:

- Direction of each bit of the 8-bit I/O port can be set independently
- When each bit is set for input mode, it is possible to set each bit for independent pull-up
- Ports can be used as I/O ports or as data bus lines, for a maximum data bus width of 32 bits, by setting the PORTEN bit in bus control register 2 (BCR2)

The SCI I/O port has the following features:

- When the I/O port is set to output and the SCI is not enabled, data can be output. This allows transmission of the break status. SCK pin control is also possible.
- The value of the RxD pin can be read at any time. This enables break detection.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the 8-bit general-purpose I/O port.

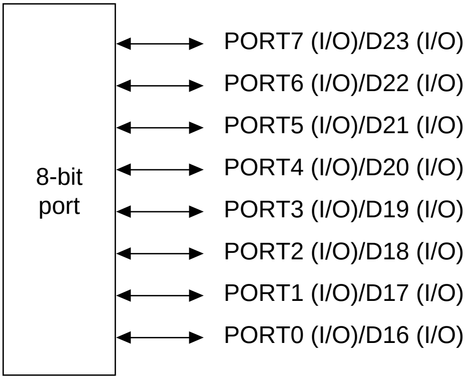
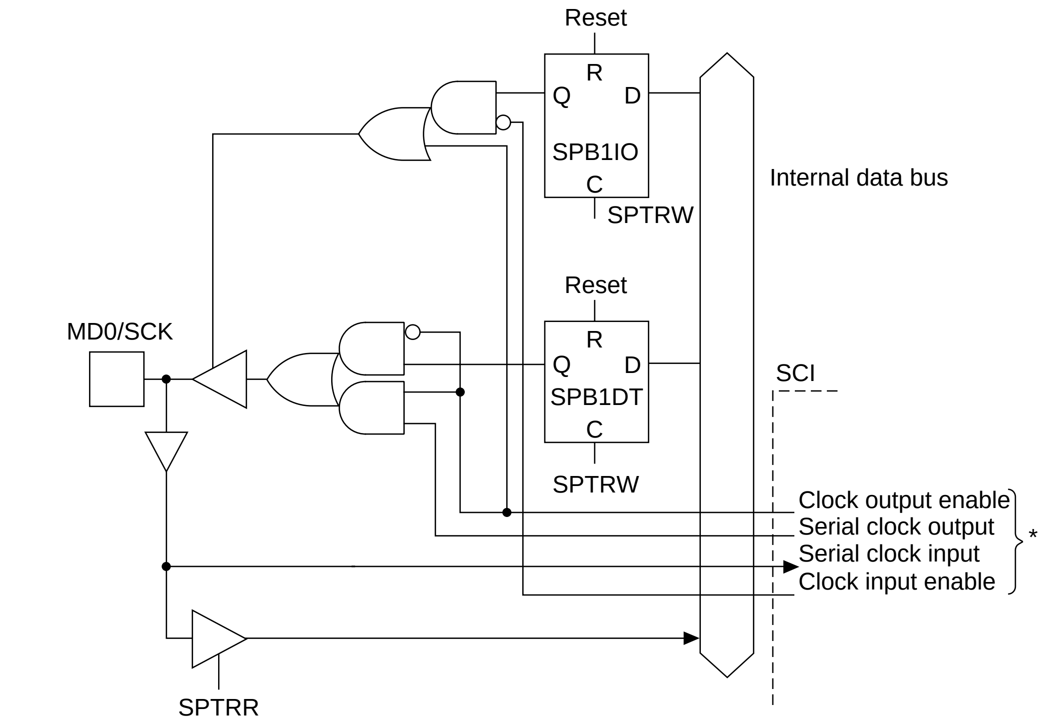


Figure 15.1 8-Bit I/O Port

Figures 15.2 to 15.4 show block diagrams of the SCI I/O port.



SPTRW: SPTR write
SPTRR: SPTR read

Note: * Signals that set the SCK pin function to internal clock output or external clock input as specified by the CKE0 and CKE1 bits in SCSCR, and the C/$\overline{A}$ bit in SCSMR.

Figure 15.2 SCI I/O Port: MD0/SCK Pin

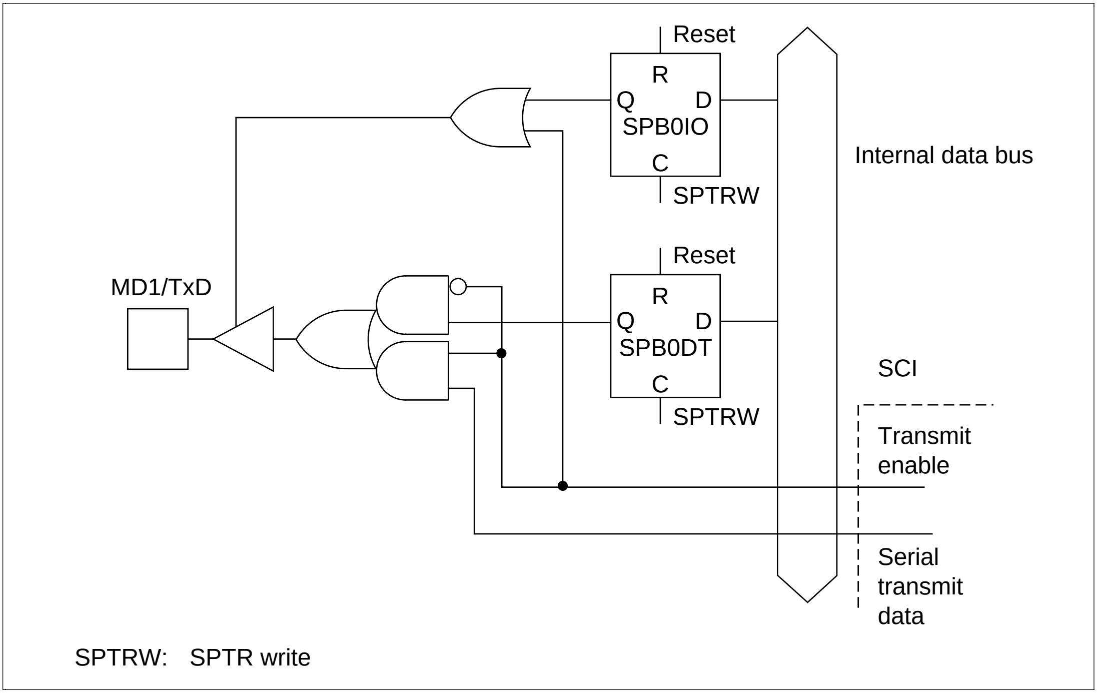


Figure 15.3 SCI I/O Port: MD1/TxD Pin

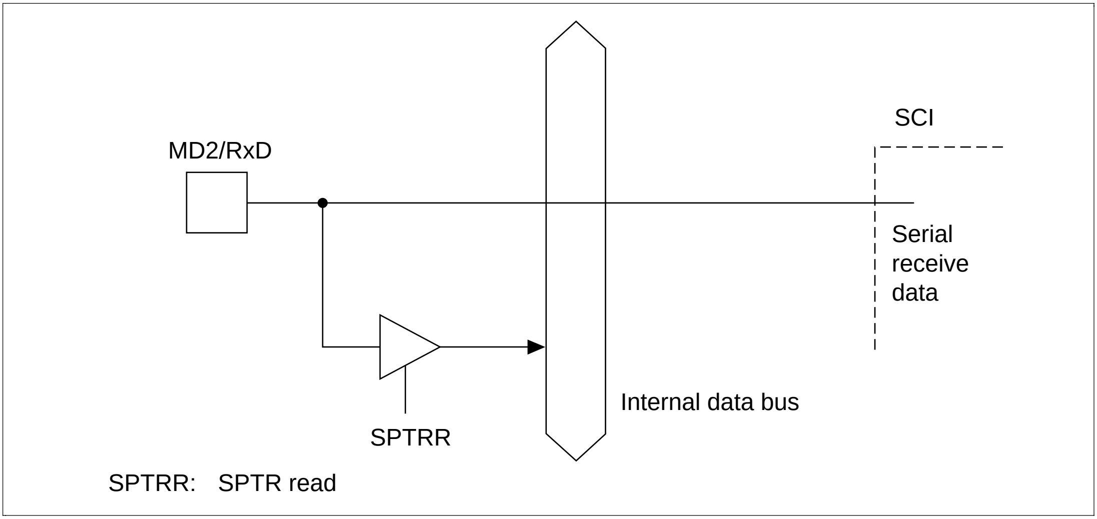


Figure 15.4 SCI I/O Port: MD2/RxD Pin

15.1.3 Pin Configuration

Table 15.1 shows the pin configuration of the 8-bit general-purpose I/O port.

Table 15.1 Pin Configuration

Pin	Signal	I/O	Function
Port 7	PORT7	I/O	I/O port
Port 6	PORT6	I/O	I/O port
Port 5	PORT5	I/O	I/O port
Port 4	PORT4	I/O	I/O port
Port 3	PORT3	I/O	I/O port
Port 2	PORT2	I/O	I/O port
Port 1	PORT1	I/O	I/O port
Port 0	PORT0	I/O	I/O port

Table 15.2 shows the pin configuration of the SCI I/O port.

Table 15.2 Pin Configuration

Pin	Signal	I/O	Function
Serial transmission	TxD	O	Serial data transmission and break status transmission
Serial reception	RxD	I	Serial data reception and break status detection
Serial clock	SCK	I/O	Serial clock input/output and I/O port

Note: These pins function as mode input pins MD0–MD2 after a power-on reset. They are made to function as serial pins by performing SCI operation settings with the TE, RE, CKEI, and CKE0 bits in SCSCR and the C/$\overline{\text{A}}$ bit in SCSMR. Break status transmission and detection can be performed by means of the SCI's SCSPTR register.

15.1.4 Register Configuration

Table 15.3 shows the configuration of the two registers of the 8-bit general-purpose I/O port (PCTR and PDTR) and the one register of the SCI I/O port (SCSPTR).

Table 15.3 Register Configuration

Register	Symbol	R/W	Initial Value*[1]	Address	Access Size
Port control register	PCTR	R/W	H'0000	H'FFFFFF76	16
Port data register	PDTR	R/W	Undefined	H'FFFFFF78	8
Serial port register	SCSPTR	R/W	*[2]	H'FFFFFF7C	8

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

2. Bits other than 2 and 0 are initialized to H'00. Bits 2 and 0 are undefined.

15.2 Register Descriptions

15.2.1 Port Control Register (PCTR)

Bit:	15	14	13	12	11	10	9	8
Bit name:	PB7 PUP	PB7 IO	PB6 PUP	PB6 IO	PB5 PUP	PB5 IO	PB4 PUP	PB4 IO
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:	PB3 PUP	PB3 IO	PB2 PUP	PB2 IO	PB1 PUP	PB1 IO	PB0 PUP	PB0 IO
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 control register (PCTR) is a 16-bit read/write register that controls the input/output direction and pull-up for each bit in the 8-bit port. As the initial value of the port data register (PDR) is undefined, all the bits in the 8-bit port should be set to output with PCTR after writing a value to the PDTR register.

PCTR is initialized to H'0000 by a power-on reset. It is not initialized by a manual reset or in standby mode, and retains its contents.

Bit 2n + 1 (n = 0–7): Port Pull-Up Control (PBnPUP): Controls the pull-up of each bit in the 8-bit port by means of built-in resistors. This setting is valid even if the port pin is set to output by the PBnIO bit. Therefore, to avoid unnecessary power consumption and ensure the reliability of the chip, a pull-up setting should not be made when the corresponding port pin has been set to output.

Bit 2n + 1: PBnPUP	Description	
0	Bit n (n = 0–7) of the 8-bit port is pulled up.	(Initial value)
1	Bit n (n = 0–7) of the 8-bit port is not pulled up.	

Bit 2n (n = 0–7)—Port I/O Control (PBnDIR): Controls whether each bit of 8-bit port is an input or an output.

Bit 2n: PBnIO	Description	
0	Bit n (n = 0–7) of the 8-bit port is an input.	(Initial value)
1	Bit n (n = 0–7) of the 8-bit port is an output.	

15.2.2 Port Data Register (PDTR)

The port data register (PDTR) is an 8-bit read/write register used as data latches for each bit of the 8-bit port. When a bit is set to be used as an output, the value written into PDTR is output from the external pin. When a value is read from PDTR, the external pin value sampled on the external bus clock is returned.

PDTR is not initialized by a power-on reset or manual reset, or in standby mode, and it retains its contents. However, if PDTR is read when a bus release request is issued (when $\overline{\text{BREQ}}$ is asserted), its value may not be read correctly. Therefore, $\overline{\text{BREQ}}$ should not be asserted when reading PDTR.

Bit:	7	6	5	4	3	2	1	0
Bit name:	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.2.3 Serial Port Register (SCSPTR)

The serial port register (SCSPTR) is an 8-bit register that the CPU can always read and write. It controls I/O and data of the port multiplexed with the serial communication interface (SCI) pins. Input data can be read from the RxD pin and output data can be transmitted to the TxD pin; this controls breaks for serial transmission and reception. In addition, SCK pin data reading and output data writing can be performed by means of bits 3 and 2.

All SCSPTR bits except bits 2 and 0 are initialized to 0 by a power-on reset; the value of bits 2 and 0 is undefined. SCSPTR is not initialized by a manual reset or in standby mode, and it retains its contents.

Bit:	7	6	5	4	3	2	1	0
Bit name:	—	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT
Initial value:	0	0	0	0	0	—	0	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits always read 0. The write value should always be 0.

Bit 3—Serial Port Clock Port I/O (SPB1IO): Specifies serial port SCK pin input/output. When the SCK pin is actually set as a port output pin and outputs the value set by the SPB1DT bit, the $C/\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR should be cleared to 0.

Bit 3: SPB1IO	Description	
0	The SPB1DT value is not output to the SCK pin.	(Initial value)
1	The SPB1DT bit value is output to the SCK pin.	

Bit 2—Serial Port Clock Port Data (SPB1DT): Specifies the serial port SCK pin input/output data. Input or output is specified by the SPB1IO bit (see the description of SPB1IO for details). When output is specified, the value of the SPB1DT bit is output to the SCK pin. The SCK pin value is read from the SPB1DT bit regardless of the value of the SPB1IO bit. The initial value of this bit after a power-on reset is undefined.

Bit 2: SPB1DT	Description	
0	I/O data level is low.	(Initial value)
1	I/O data level is high.	

Bit 1—Serial Port Break I/O (SPB0IO): Specifies the serial port TxD pin output condition. When the TxD pin is actually set as a port output pin and outputs the value set by the SPB0DT bit, the TE bit in SCSCR should be cleared to 0.

Bit 1: SPB0IO	Description	
0	The SPB0DT bit value is not output to the TxD pin.	(Initial value)
1	The SPB0DT bit value is output to the TxD pin.	

Bit 0–Serial Port Break Data (SPB0DT): Specifies serial port RxD pin input data and TxD pin output data. The TxD pin output condition is set with the SPB0IO bit (see the description of SPB0IO above). When the TxD pin is set as an output, the value of the SPB0DT bit is output to the TxD pin. The RxD pin value is always read from the SPB0DT bit, regardless of the value of the SPB0IO bit. The initial value of this bit after a power-on reset is undefined.

Bit 0 : SPB0DT	Description	
0	I/O data level is low.	(Initial value)
1	I/O data level is high.	

Section 16 Electrical Characteristics (-SH7708, SH7708S-)

16.1 Absolute Maximum Ratings

Table 16.1 Absolute Maximum Ratings

Item	Symbol	Ratings	Units
Power supply voltage	V_{CC}	–0.3 to 4.6	V
Input voltage	V_{in}	–0.3 to V_{CC} + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{str}	–55 to 125	°C

Note: Operating the SH7708 Series above maximum ratings can damage or destroy it.

16.2 DC Characteristics

Table 16.2 DC Characteristics (Ta = –20 to 75°C)

Item		Symbol	Min	Type	Max	Unit	Remarks
Power supply voltage		V_{CC}	3.0	3.3	3.6	V	In normal operation, sleep mode, and standby mode
			2.0	3.3	3.6		RTC operating voltage in standby mode
Current	Normal operation	I_{CC}	—	—	195*1	mA	V_{CC} = 3.3 V
			—	—	100*2		*1: 60 MHz (Pø = 30 MHz)
			—	—	50*3		*2: 30 MHz
	In sleep mode		—	75*1	95*1		*3: 15 MHz
			—	40*2	50*2		
			—	20*3	25*3		
	In standby mode		—	15	30	µA	Ta = 25°C (RTC on)
			—	—	400		Ta > 50°C (RTC on)
			—	5	15		Ta = 25°C (RTC off)
			—	—	300		Ta > 50°C (RTC off)
Input voltage	RESET, NMI	VIH	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	BREQ, IRL3–IRL0, MD5–MD0		$V_{CC} - 0.5$	—	$V_{CC} + 0.3$		Standby mode
			$V_{CC} - 0.7$	—	$V_{CC} + 0.3$		Normal operation
	EXTAL, CKIO		$V_{CC} - 0.7$	—	$V_{CC} + 0.3$		
	Other input pins		2.0	—	$V_{CC} + 0.3$		
	RESET, NMI	VIL	–0.3	—	$V_{CC} \times 0.1$		
	BREQ, IRL3–IRL0, MD5–MD0		–0.3	—	0.5		Standby mode
			–0.3	—	$V_{CC} \times 0.2$		Normal operation
	Other input pins		–0.3	—	$V_{CC} \times 0.2$		
Input leak current	All input pins	|lin|	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Three-state leak current	I/O, output, all pins (off condition)	|lsti|	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V

Table 16.2 DC Characteristics (Ta = –20 to 75°C) (cont)

Item		Symbol	Min	Type	Max	Unit	Remarks
Output voltage	All output pins	VOH	2.4	—	—	V	V_{CC} = 3.0 V, IOH = –200 µA
			2.0	—	—		V_{CC} = 3.0 V, IOH = –2 mA
		VOL	—	—	0.55		V_{CC} = 3.6 V, IOL = 1.6 mA
Pull-up resistance	Port pins	Rpull	30	60	120	kΩ	
Terminal capacitance	All pins	C	—	—	20	pF	

Notes: 1. Regardless of whether PLL or RTC is used, connect $PLLV_{CC}$, $RTCV_{CC}$ to V_{CC}, and PLLGND, RTCGND to GND.

2. With VIH min = V_{CC} – 0.5 V, VIL max = 0.5 V, and all output pins unloaded.

Table 16.3 Permissible Output Current Values (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	IOL	—	—	2.0	mA
Permissible output low current (total)	ΣIOL	—	—	120	
Permissible output high current (per pin)	–IOH	—	—	2.0	
Permissible output high current (total)	Σ (–IOH)	—	—	40	

Note: To ensure reliability, output current must not exceed the maximum values listed.

16.3 AC Characteristics

Input for the SH7708 Series should, as a rule, be clock synchronous. Keep to the setup and hold times for each input signal unless otherwise directed.

Table 16.4 LSI Clock Values (Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit
Operating frequency	CPU, cache, TLB	f	1	—	60	MHz
	External bus		1	—	60	
	Peripheral modules		0.25	—	30	

16.3.1 Clock Timing

Table 16.5 Clock Timing (V_{CC} = 3.3 ± 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	16.1
EXTAL clock input cycle time	t_{EXcyc}	33.3	500	ns	
EXTAL clock input low level pulse width	t_{EXL}	8*1 or 12*2	—	ns	
EXTAL clock input high level 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 frequency (input)	f_{CKI}	8	15	MHz	16.2
CKIO clock cycle time (input)	t_{CKIcyc}	66.7	125	ns	
CKIO clock low-level pulse width (input)	t_{CKIL}	8	—	ns	
CKIO clock high-level pulse width (input)	t_{CKIH}	8	—	ns	
CKIO clock rise time (input)	t_{CKIr}	—	4	ns	
CKIO clock fall time (input)	t_{CKIf}	—	4	ns	
CKIO clock output frequency (output)	f_{OP}	1	15	MHz	16.3
CKIO clock cycle time (output)	t_{cyc}	66.7	1000	ns	
CKIO clock low-level pulse width (output)	t_{CKOL}	20	—	ns	
CKIO clock high-level pulse width (output)	t_{CKOH}	20	—	ns	
CKIO clock rise time (output)	t_{CKOr}	—	7	ns	
CKIO clock fall time (output)	t_{CKOf}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	16.4
Power-on oscillation settling time/mode setting	t_{OSCMD}	10	—	ms	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	0	—	ns	
$\overline{\text{RESET}}$ set-up time	t_{RESS}	20	—	ns	
$\overline{\text{BREQ}}$ set-up time	t_{BREQS}	20	—	ns	
MD reset hold time	t_{MDRH}	20	—	ns	
Reset assert time	t_{RESW}	20	—	tcyc	16.4, 16.5, 16.11
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	16.5

Table 16.5 Clock Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to +75°C, Maximum External Bus Operating Frequency: 15 MHz) (cont)

Item	Symbol	Min	Max	Unit	Figure
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	16.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	16.7
PLL synchronization settling time	t_{PLL}	100	—	µs	16.8, 16.9, 16.10
IRL interrupt decision time (using RTC and in standby mode)	t_{IRLSTB}	100	—	µs	16.10

Notes: 1. PLL circuit 2 in operation.

2. PLL circuit 2 not in operation.

Table 16.6 Clock Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –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	16.1
EXTAL clock input cycle time	t_{EXcyc}	33.3	500	ns	
EXTAL clock input low level pulse width	t_{EXL}	7*1 or 10*2	—	ns	
EXTAL clock input high level 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 frequency (input)	f_{CKI}	8	30	MHz	16.2
CKIO clock cycle time (input)	t_{CKIcyc}	33.3	125	ns	
CKIO clock low-level pulse width (input)	t_{CKIL}	7	—	ns	
CKIO clock high-level pulse width (input)	t_{CKIH}	7	—	ns	
CKIO clock rise time (input)	t_{CKIr}	—	3	ns	
CKIO clock fall time (input)	t_{CKIf}	—	3	ns	
CKIO clock output frequency (output)	f_{OP}	1	130	MHz	16.3
CKIO clock cycle time (output)	t_{cyc}	33.3	1000	ns	
CKIO clock low-level pulse width (output)	t_{CKOL}	8	—	ns	
CKIO clock high-level pulse width (output)	t_{CKOH}	8	—	ns	
CKIO clock rise time (output)	t_{CKOr}	—	6	ns	
CKIO clock fall time (output)	t_{CKOf}	—	6	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	16.4
Power-on oscillation settling time/mode setting	t_{OSCMD}	10	—	ms	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	0	—	ns	
$\overline{\text{RESET}}$ set-up time	t_{RESS}	20	—	ns	
$\overline{\text{BREQ}}$ set-up time	t_{BREQS}	20	—	ns	
MD reset hold time	t_{MDRH}	20	—	ns	
Reset assert time	t_{RESW}	20	—	tcyc	16.4, 16.5, 16.11
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	16.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	16.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	16.7

Table 16.6 Clock Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 30 MHz) (cont)

Item	Symbol	Min	Max	Unit	Figure
PLL synchronization settling time	t_{PLL}	100	—	µs	16.8, 16.9, 16.10
IRL interrupt decision time (using RTC and in standby mode)	t_{IRLSTB}	100	—	µs	16.10

Notes: 1. PLL circuit 2 in operation.

2. PLL circuit 1 not in operation.

Table 16.7 Clock Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 60 MHz)

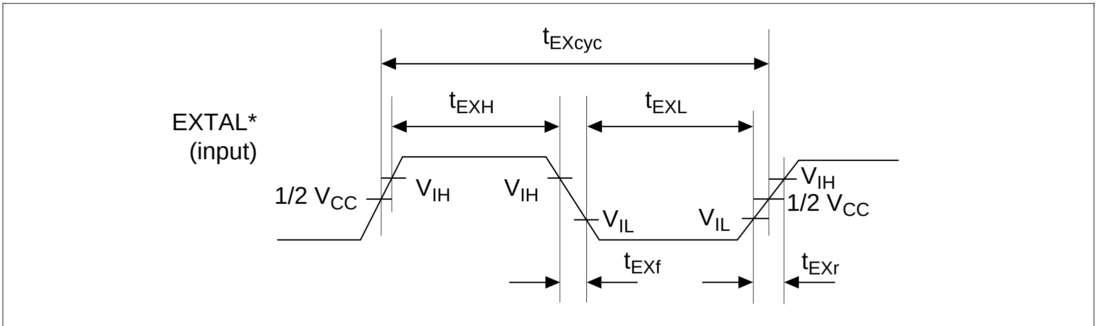
Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	2	60	MHz	16.1
EXTAL clock input cycle time	t_{EXcyc}	16.7	500	ns	
EXTAL clock input low-level pulse width	t_{EXL}	4*[1] or 10*[2]	—	ns	
EXTAL clock input high-level pulse width	t_{EXH}	4*[1] or 10*[2]	—	ns	
EXTAL clock input rise time	t_{EXr}	—	2	ns	
EXTAL clock input fall time	t_{EXf}	—	2	ns	
CKIO clock frequency (input)	f_{CKI}	8	60	MHz	16.2
CKIO clock cycle time (input)	t_{CKIcyc}	16.7	125	ns	
CKIO clock low-level pulse width (input)	t_{CKIL}	4	—	ns	
CKIO clock high-level pulse width (input)	t_{CKIH}	4	—	ns	
CKIO clock rise time (input)	t_{CKIr}	—	2	ns	
CKIO clock fall time (input)	t_{CKIf}	—	2	ns	
CKIO clock output frequency (output)	f_{OP}	1	60	MHz	16.3
CKIO clock cycle time (output)	t_{cyc}	16.7	1000	ns	
CKIO clock low-level pulse width (output)	t_{CKOL}	3	—	ns	
CKIO clock high-level pulse width (output)	t_{CKOH}	3	—	ns	
CKIO clock rise time (output)	t_{CKOr}	—	5	ns	
CKIO clock fall time (output)	t_{CKOf}	—	5	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	16.4
Power-on oscillation settling time/mode setting	t_{OSCMD}	10	—	ms	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	0	—	ns	
$\overline{\text{RESET}}$ set-up time	t_{RESS}	20	—	ns	
$\overline{\text{BREQ}}$ set-up time	t_{BREQS}	20	—	ns	
MD reset hold time	t_{MDRH}	20	—	ns	
Reset assert time	t_{RESW}	20	—	tcyc	16.4, 16.5, 16.11
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	16.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	16.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	16.7

Table 16.7 Clock Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 60 MHz) (cont)

Item	Symbol	Min	Max	Unit	Figure
PLL synchronization settling time	t_{PLL}	100	—	µs	16.8, 16.9, 16.10
IRL interrupt decision time (using RTC and in standby mode)	t_{IRLSTB}	100	—	µs	16.10

Notes: 1. PLL circuit 2 in operation.

2. IPLL circuit 2 not in operation.



Note: * The clock input from the EXTAL pin.

Figure 16.1 EXTAL Clock Input Timing

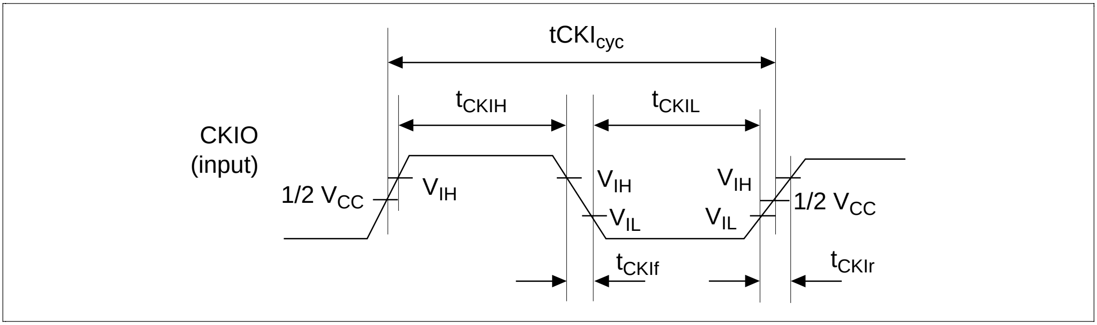


Figure 16.2 CKIO Clock Input Timing

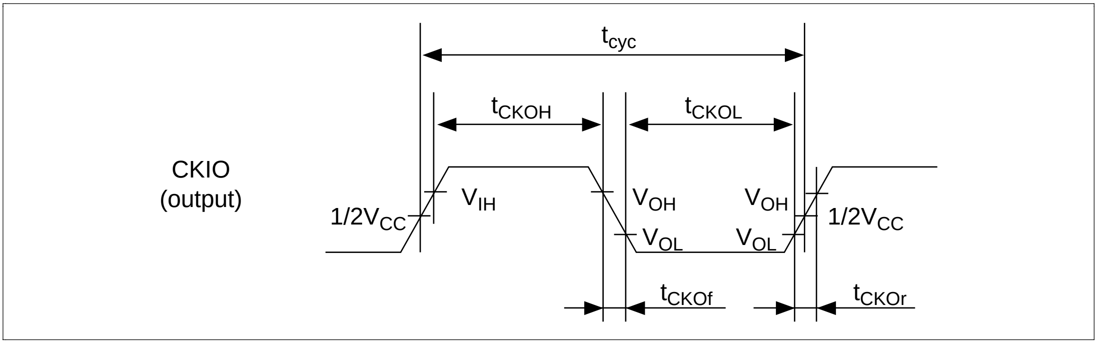
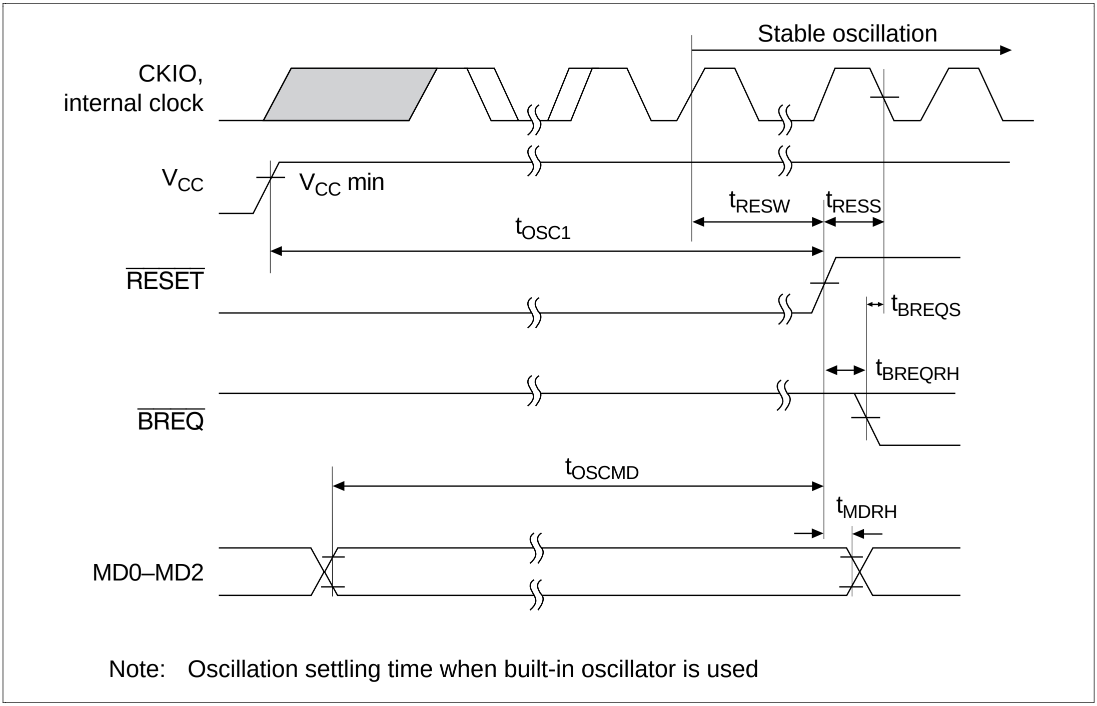
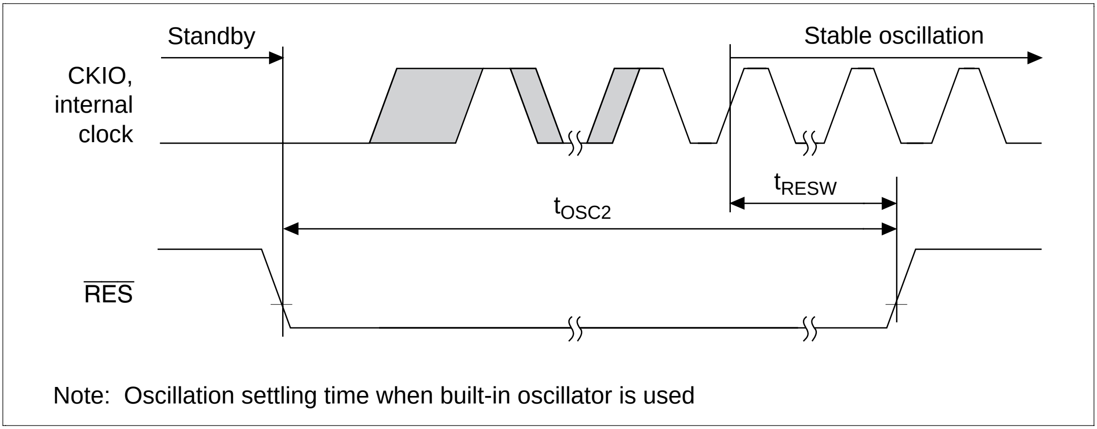


Figure 16.3 CKIO Clock Output Timing



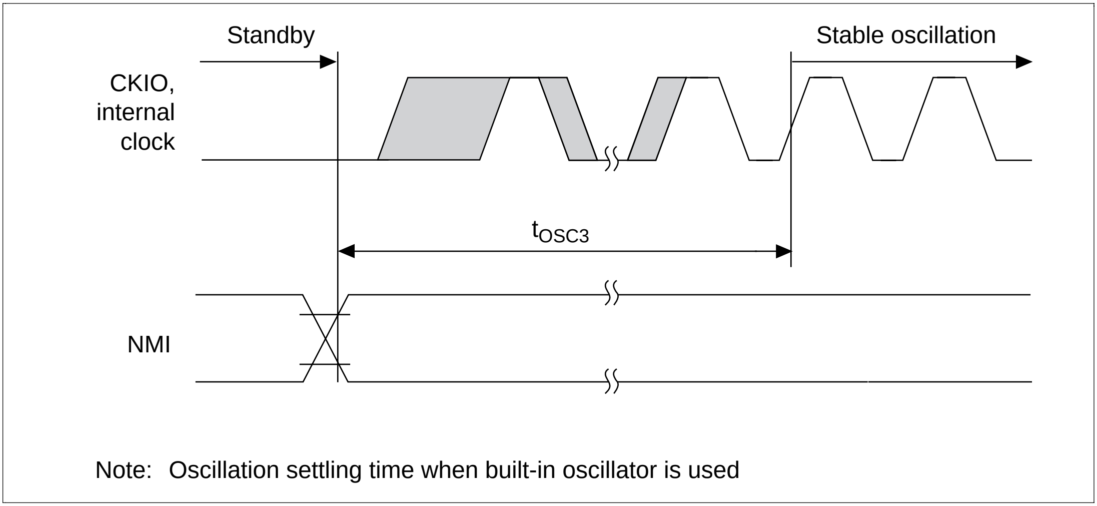
Note: Oscillation settling time when built-in oscillator is used

Figure 16.4 Power-On Oscillation Settling Time



Note: Oscillation settling time when built-in oscillator is used

Figure 16.5 Standby Return Oscillation Settling Time (Return by RESET)



Note: Oscillation settling time when built-in oscillator is used

Figure 16.6 Standby Return Oscillation Settling Time (Return by NMI)

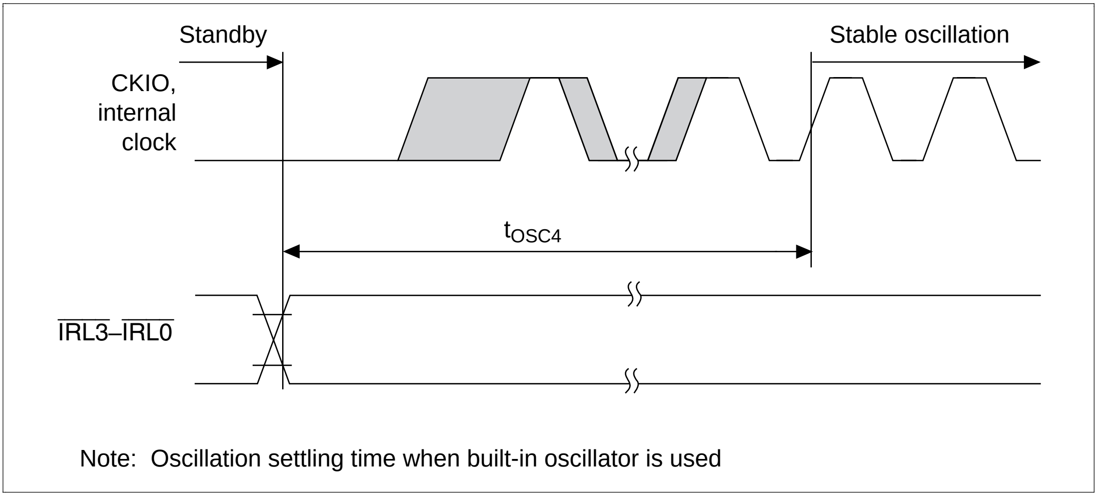

Standby
Stable oscillation
CKIO,
internal
clock
t_{OSC4}
$\overline{IRL3}$–$\overline{IRL0}$
Note: Oscillation settling time when built-in oscillator is used

Figure 16.7 Standby Return Oscillation Settling Time (Return by $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$)

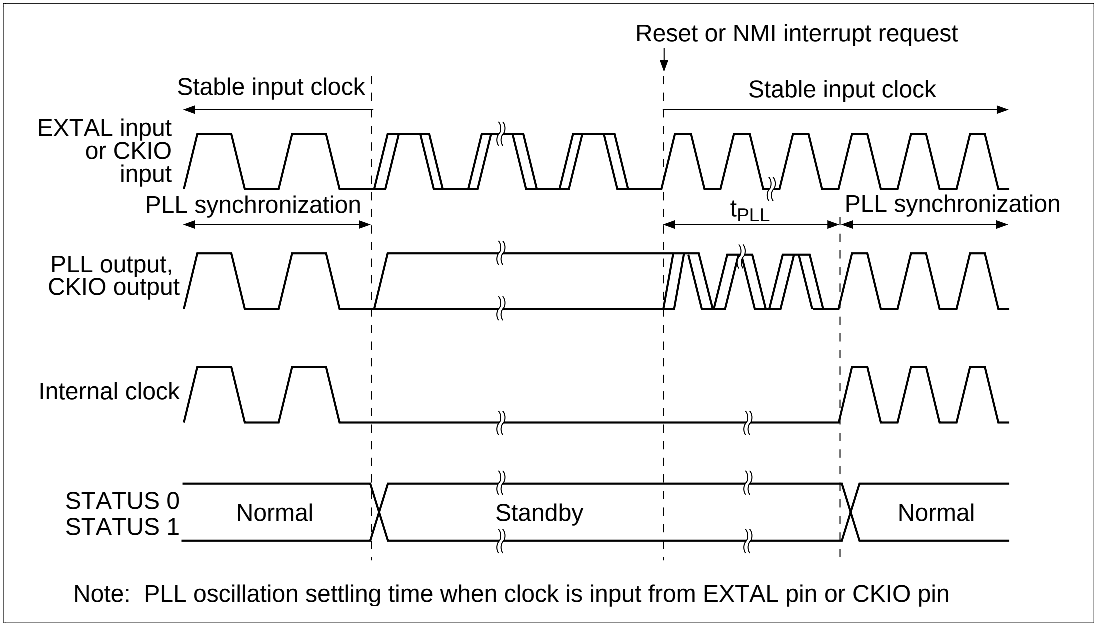
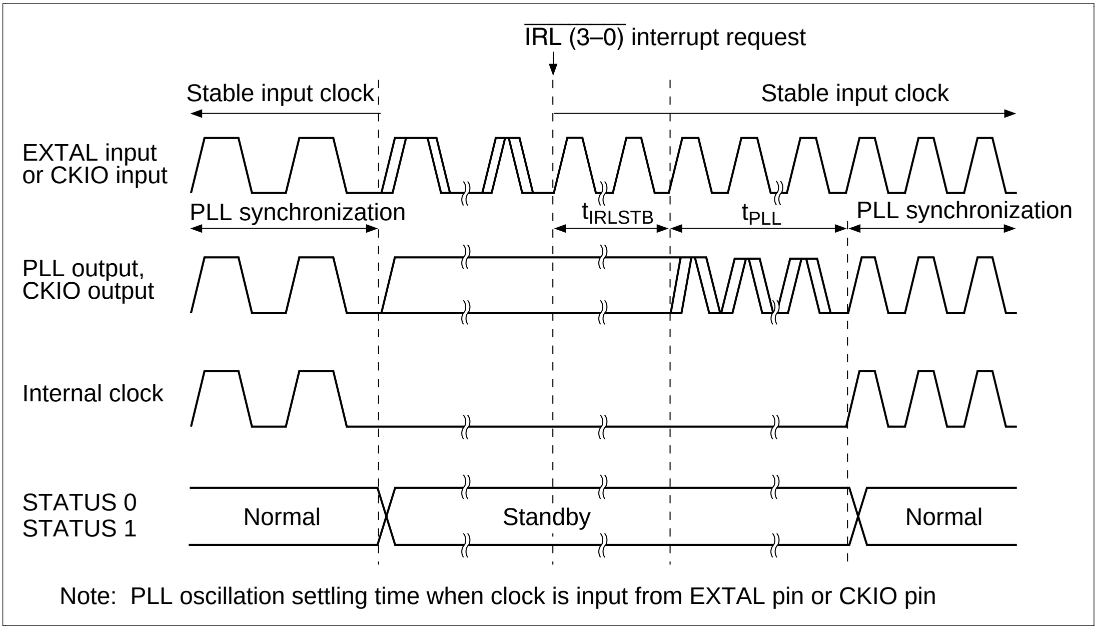
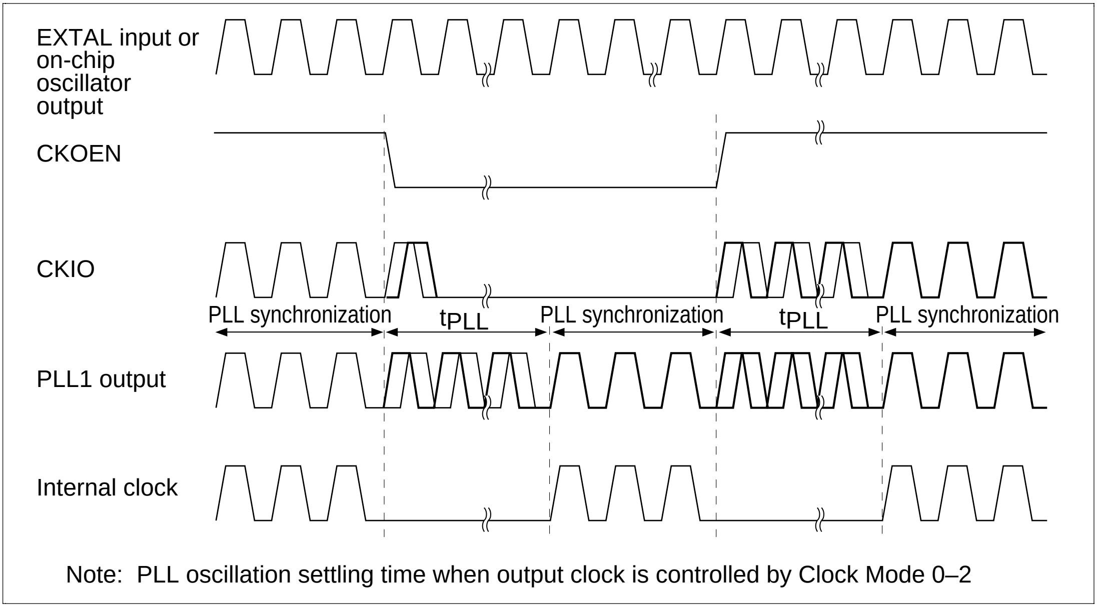


Figure 16.8 PLL Synchronization Settling Time in Case of Reset or NMI Interrupt



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 16.9 PLL Synchronization Settling Time in Case of IRL Interrupt



Note: PLL oscillation settling time when output clock is controlled by Clock Mode 0–2

Figure 16.10 PLL Synchronization Settling Time in Case of CKOEN Bit Manipulation

16.3.2 Control Signal Timing

Table 16.8 Control Signal Timing (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to + 75°C)

		–60*[2]			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RESET}}$ pulse width	t_{RESW}	20	—	tcyc	16.11, 16.13, 16.15
$\overline{\text{RESET}}$ setup time	t_{RESS}	23	—	ns	
$\overline{\text{RESET}}$ hold time	t_{RESH}	2	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	12	—	ns	
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—	ns	
$\overline{\text{BREQ}}$ reset setup time	t_{BREQRS}	17	—	ns	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	16	—	ns	
MD reset setup time	t_{MDRS}	20	—	tcyc	16.12
MD reset hold time	t_{MDRH}	16	—	ns	
NMI setup time*[1]	t_{NMIS}	15	—	ns	16.13
$\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ setup time*[1]	t_{IRLS}	10	—	ns	
NMI hold time	t_{NMIH}	4	—	ns	
$\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ hold time	t_{IRLH}	4	—	ns	
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	12	ns	16.14
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	12	ns	16.15, 16.16
STATUS1, STATUS0 delay time	t_{STD}	—	16	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	16	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	16	ns	
Bus buffer on time 1	t_{BON1}	0	16	ns	
Bus buffer on time 2	t_{BON2}	0	16	ns	

Notes: 1. $\overline{\text{RESET}}$, NMI, and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ 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 fall.

2. Upper limit of external bus clock is 60 MHz.

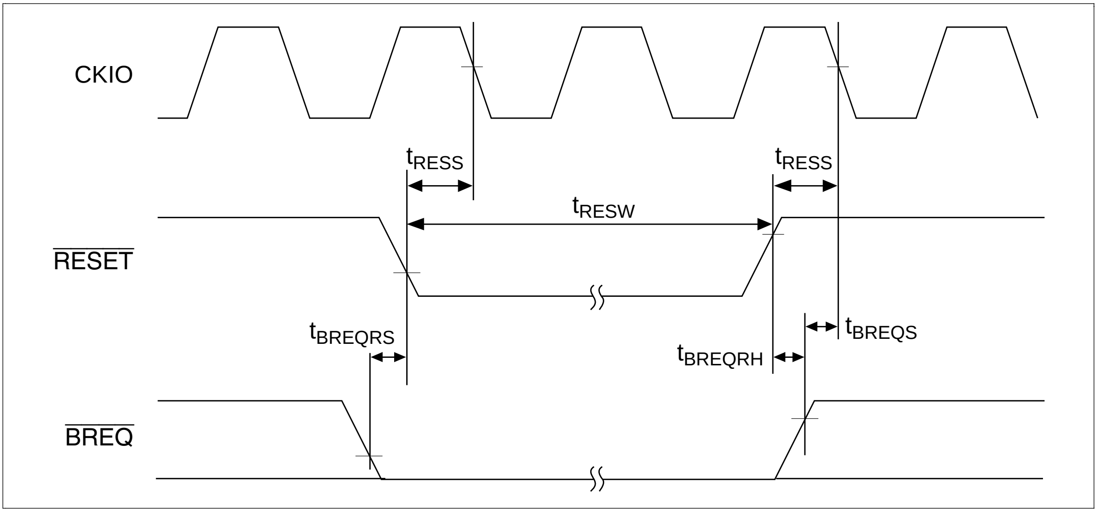


Figure 16.11 Manual Reset Input Timing

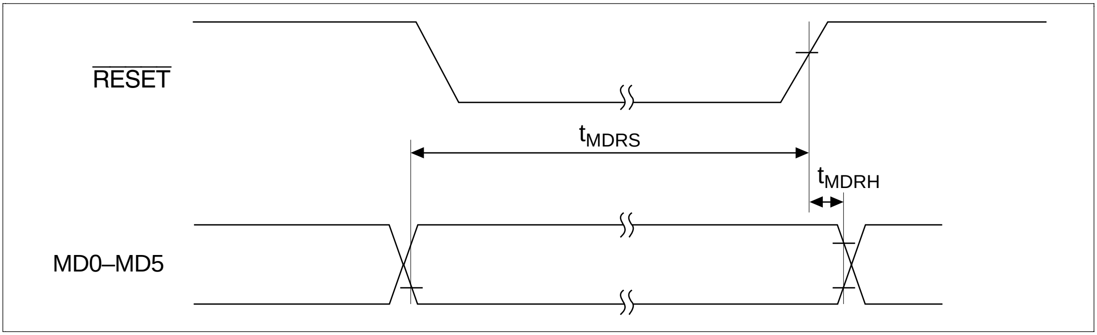


Figure 16.12 Mode Input Timing

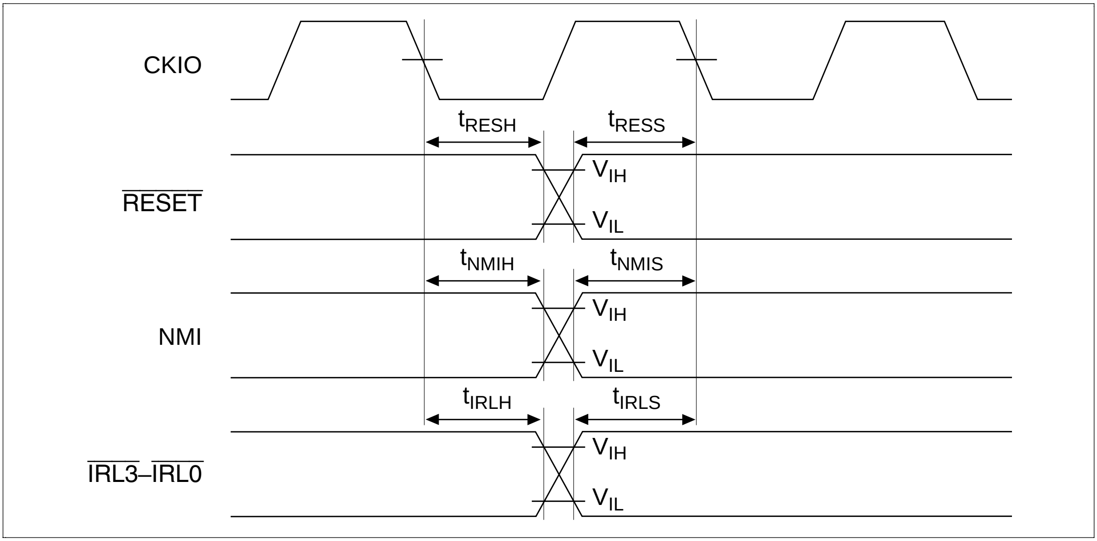


Figure 16.13 Interrupt Signal Input Timing

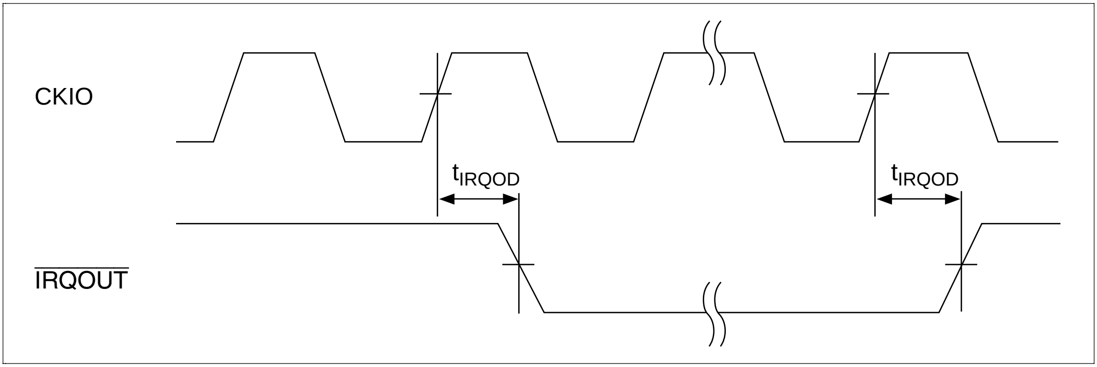


Figure 16.14 $\overline{\text{IRQOUT}}$ Timing

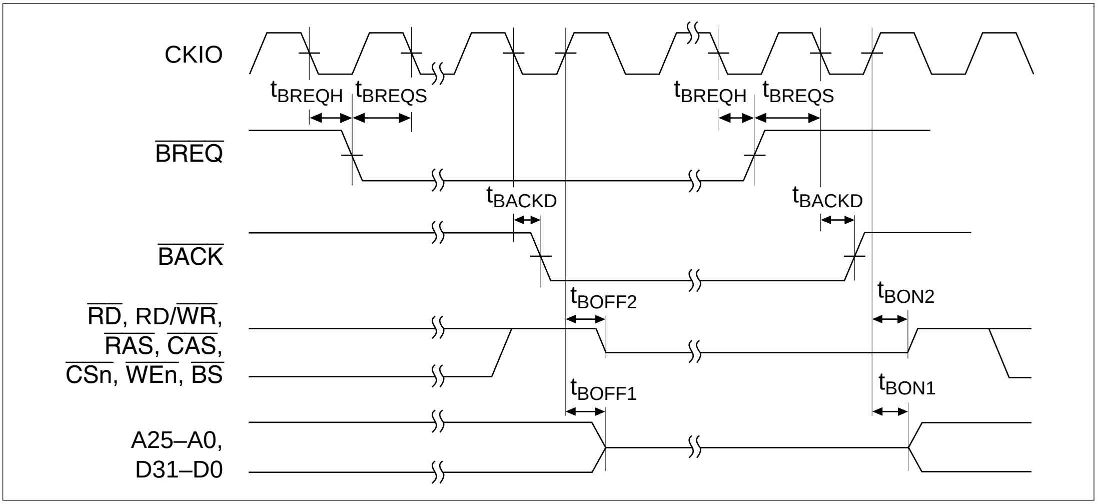


Figure 16.15 Bus Release Timing

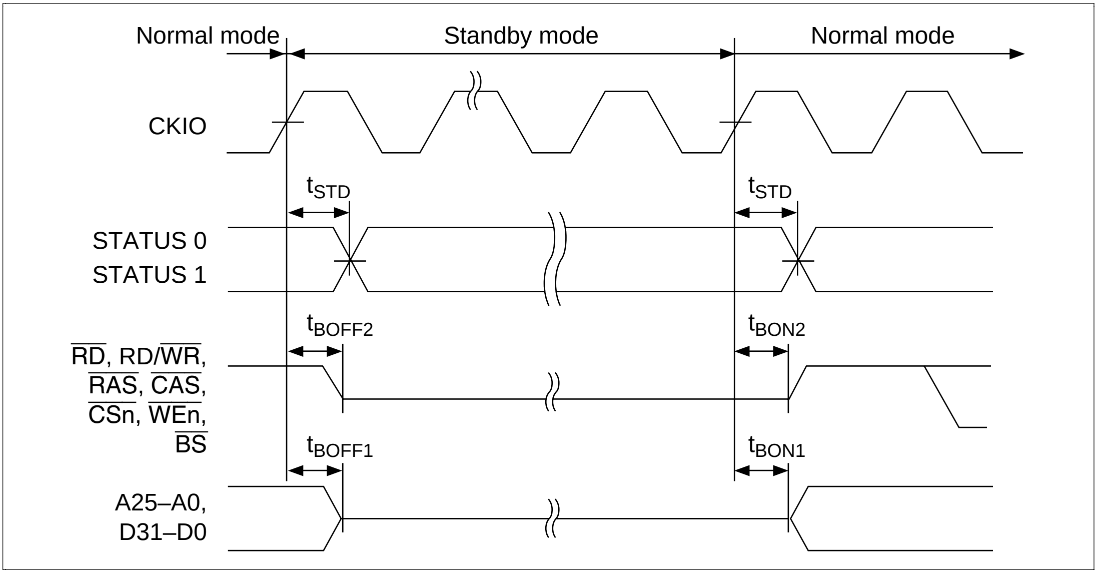


Figure 16.16 Pin Drive Timing for Standby Mode

16.3.3 AC Bus Timing Specifications

Table 16.9 Bus Timing (Conditions: Clock Mode 0/1/2/7, V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C)

		–60*[1]			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
Address delay time	t_{AD}	—	13	ns	16.17–16.58
Address setup time	t_{AS}	0	—	ns	
Address hold time	t_{AH}	0	—	ns	
$\overline{\text{BS}}$ delay time	t_{BSD}	—	12	ns	
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	—	12	ns	
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—	12	ns	
Read write delay time	t_{RWD}	—	12	ns	
Read write setup time	t_{RWS}	0	—	ns	
Read write hold time	t_{RWH}	0	—	ns	
Read strobe delay time	t_{RSD}	—	12	ns	
Read data setup time 1	t_{RDS1}	12	—	ns	
Read data setup time 2	t_{RDS2}	8	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	3	—	ns	
Write enable delay time	t_{WED}	—	12	ns	
Write data delay time 1	t_{WDD1}	—	15	ns	
Write data delay time 2	t_{WDD2}	—	15	ns	
Write data setup time	t_{WDS}	0	—	ns	
Write data hold time 1	t_{WDH1}	0	—	ns	
Write data hold time 2	t_{WDH2}	0	—	ns	
Write data hold time 3	t_{WDH3}	0	—	ns	
Write data hold time 4	t_{WDH4}	0	—	ns	
$\overline{\text{WAIT}}$ setup time*[2]	t_{WTS}	12	—	ns	
$\overline{\text{WAIT}}$ hold time*[2]	t_{WTH}	4	—	ns	

Table 16.9 Bus Timing (Conditions: Clock Mode 0/1/2/7, V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C) (cont)

		–60*[1]			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	—	13	ns	16.23–16.44
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	—	13	ns	
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	13	ns	
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	—	13	ns	
DQM delay time	t_{DQMD}	—	12	ns	
CKE delay time	t_{CKED}	—	12	ns	
$\overline{\text{CE}}$ delay time	t_{CED}	—	13	ns	16.45–16.51
$\overline{\text{OE}}$, $\overline{\text{RFSH}}$ delay time	t_{OED}	—	13	ns	
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	12	ns	16.56–16.58
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	12	ns	
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	12	—	ns	
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	4	—	ns	

Notes: 1. Upper limit of external bus clock is 60 MHz.

2. $\overline{\text{WAIT}}$ is a synchronous signal. Operation cannot be guaranteed if the setup and hold times shown here are not observed.

Table 16.10 Bus Timing (Conditions: Clock Mode 3/4/5/6, V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	—	20	ns	16.17–16.58
Address setup time	t_{AS}	0	—	ns	
Address hold time	t_{AH}	20	—	ns	
$\overline{\text{BS}}$ delay time	t_{BSD}	—	19	ns	
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	—	19	ns	
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—	20	ns	
Read write delay time	t_{RWD}	—	19	ns	
Read write setup time	t_{RWS}	0	—	ns	
Read write hold time	t_{RWH}	0	—	ns	
Read strobe delay time	t_{RSD}	—	20	ns	
Read data setup time 1	t_{RDS1}	12	—	ns	
Read data setup time 2	t_{RDS2}	12	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	8	—	ns	
Write enable delay time	t_{WED}	—	20	ns	
Write data delay time 1	t_{WDD1}	—	25	ns	
Write data delay time 2	t_{WDD2}	—	19	ns	
Write data setup time	t_{WDS}	0	—	ns	
Write data hold time 1	t_{WDH1}	0	—	ns	
Write data hold time 2	t_{WDH2}	0	—	ns	
Write data hold time 3	t_{WDH3}	0	—	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	12	—	ns	
$\overline{\text{WAIT}}$ hold time	t_{WTH}	8	—	ns	

Table 16.10 Bus Timing (Conditions: Clock Mode 3/4/5/6, V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C) (cont)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	—	20	ns	16.23–16.44
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	—	19	ns	
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	20	ns	
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	—	19	ns	
DQM delay time	t_{DQMD}	—	19	ns	
CKE delay time	t_{CKED}	—	19	ns	
$\overline{\text{CE}}$ delay time	t_{CED}	—	20	ns	16.45–16.51
$\overline{\text{OE}}$, $\overline{\text{RFSH}}$ delay time	t_{OED}	—	20	ns	
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	20	ns	16.56–16.58
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	20	ns	
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	12	—	ns	
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	8	—	ns	

16.3.4 Basic Timing

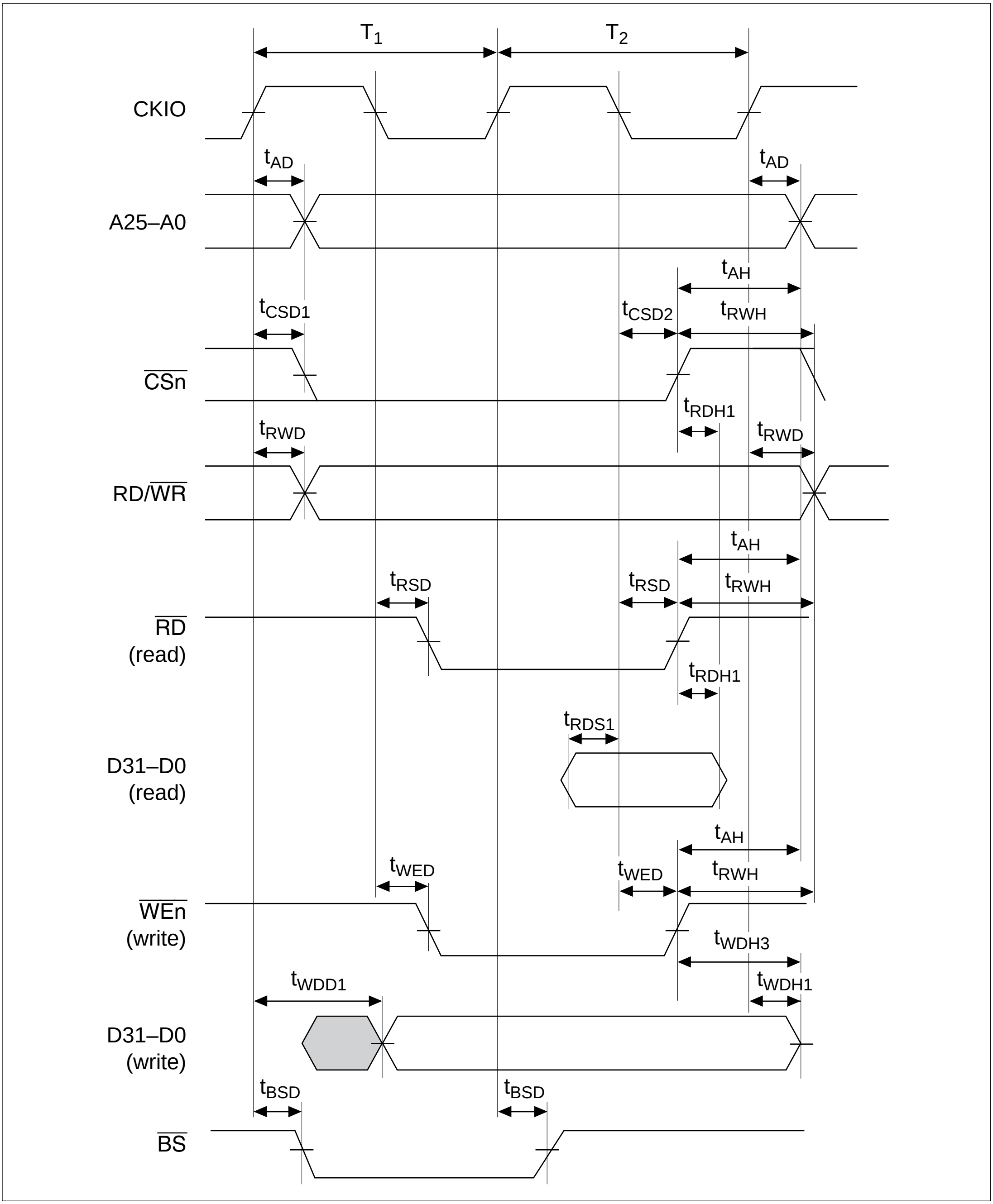


Figure 16.17 Basic Bus Cycle (No Wait)

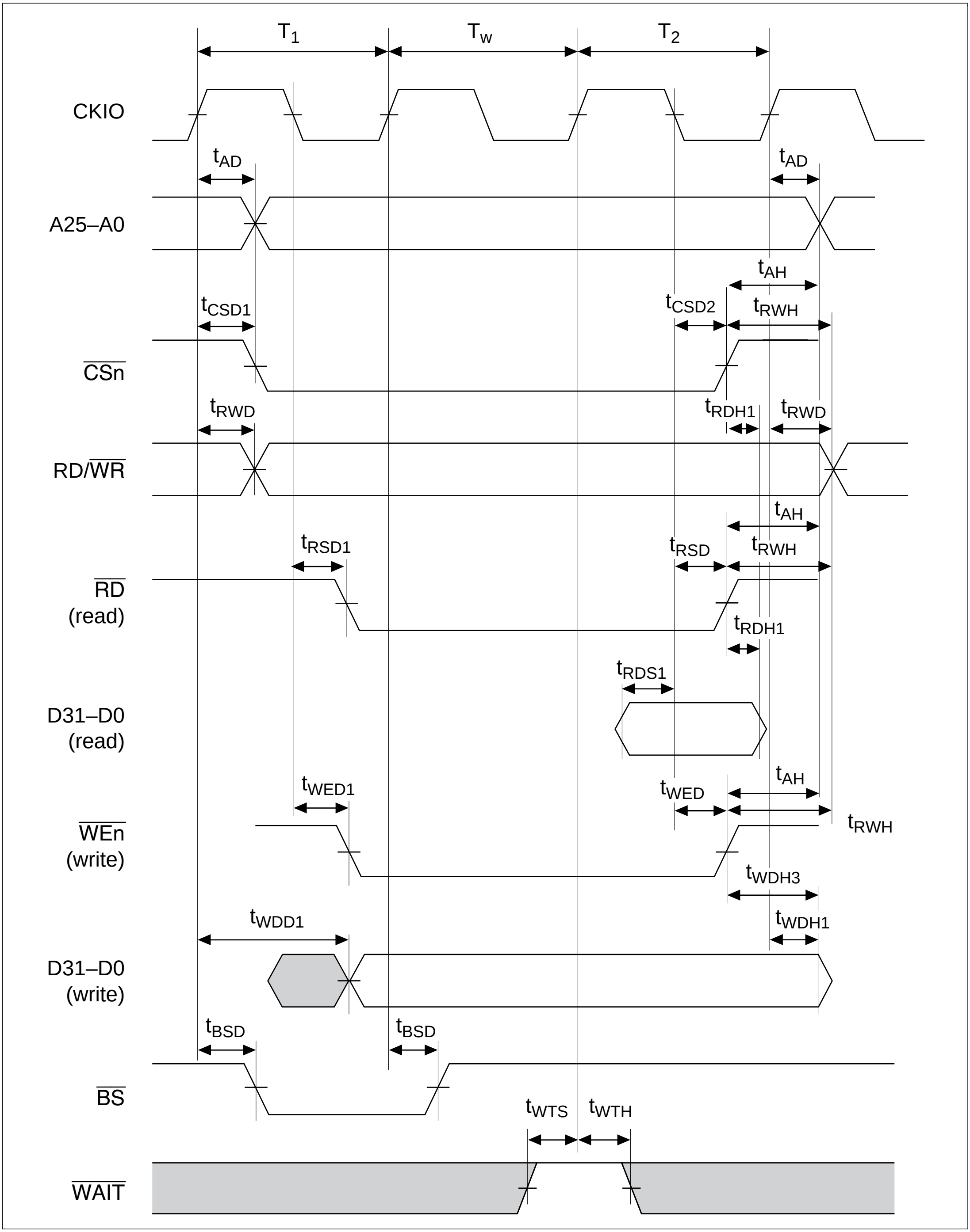


Figure 16.18 Basic Bus Cycle (1 Wait)

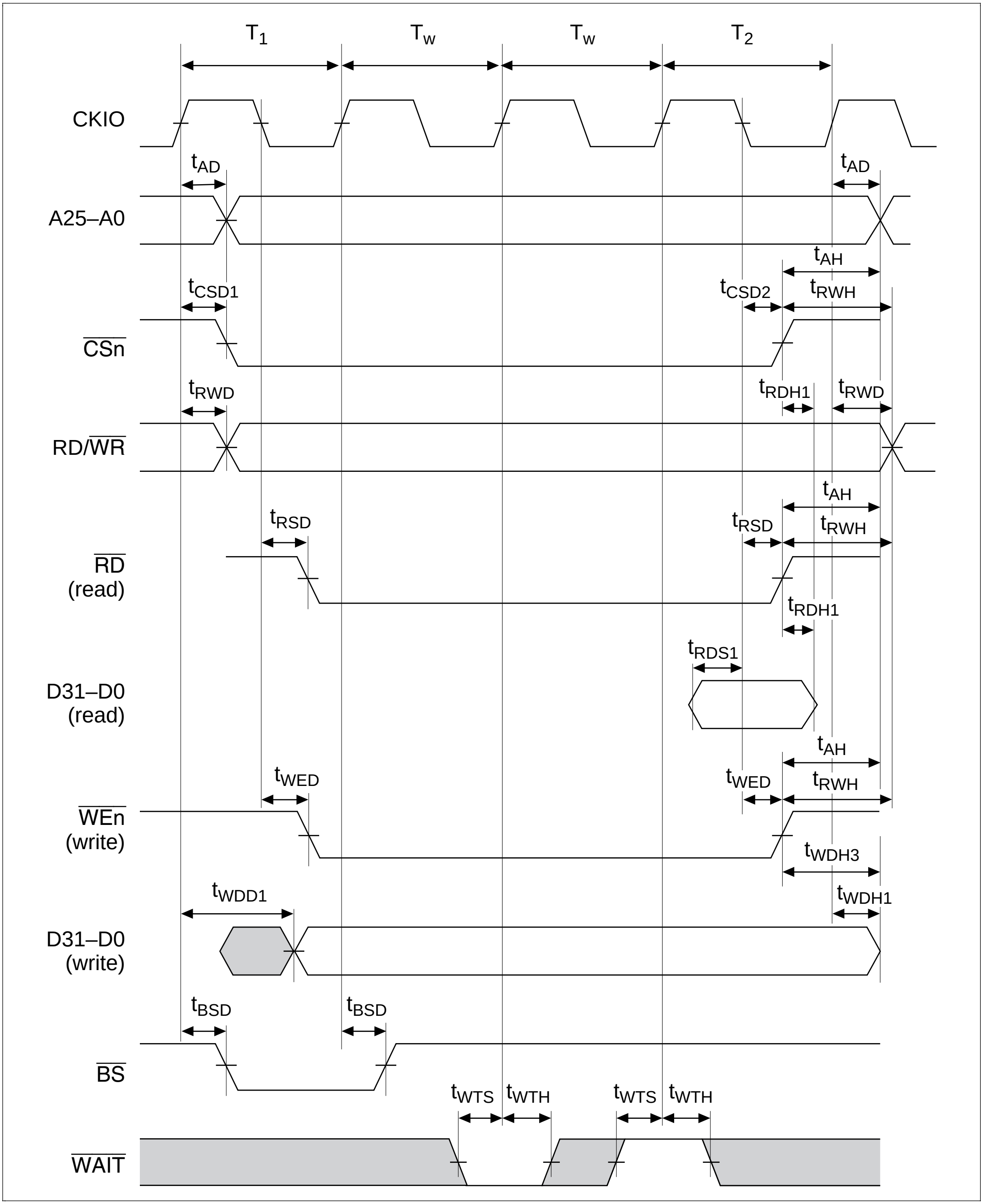
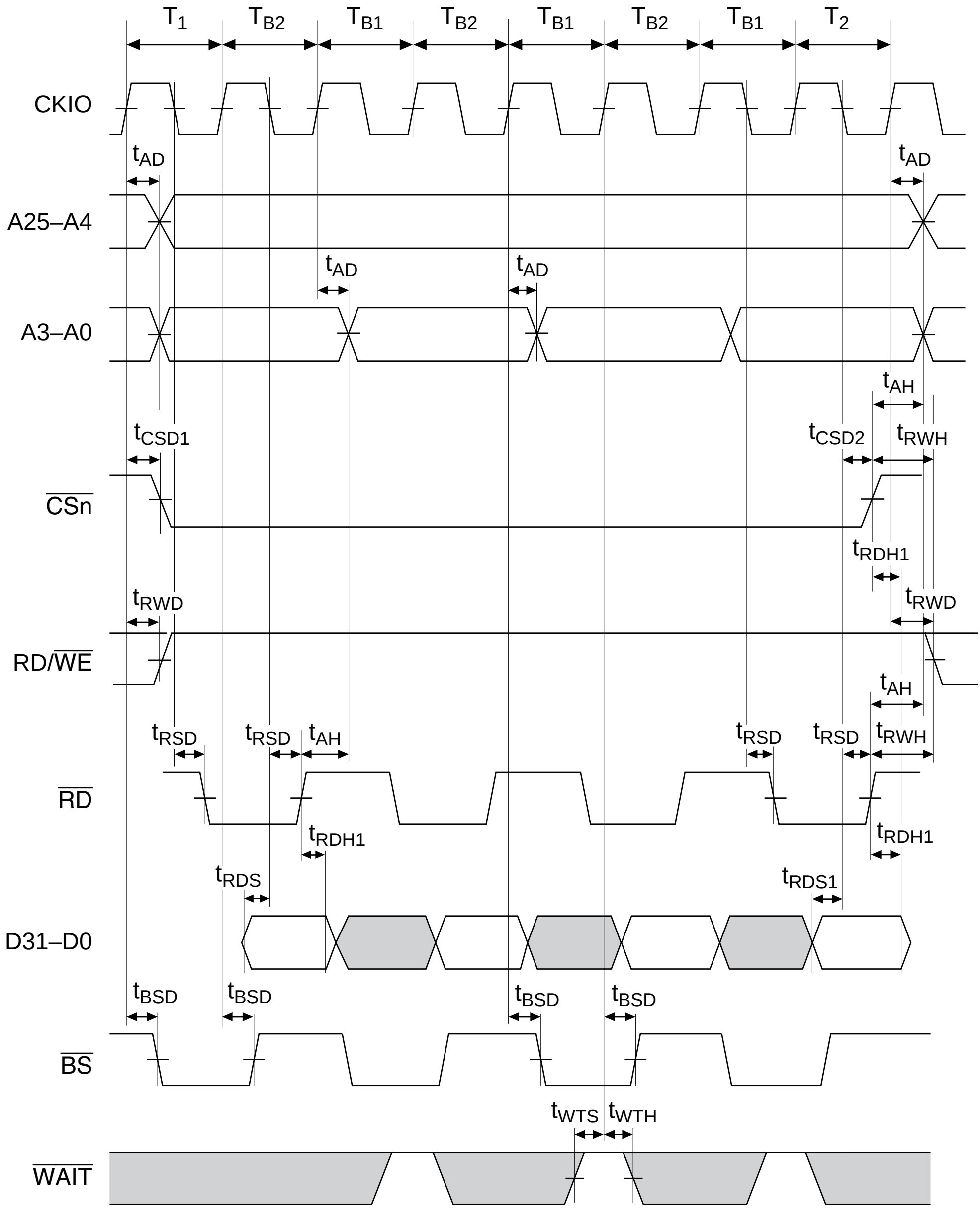


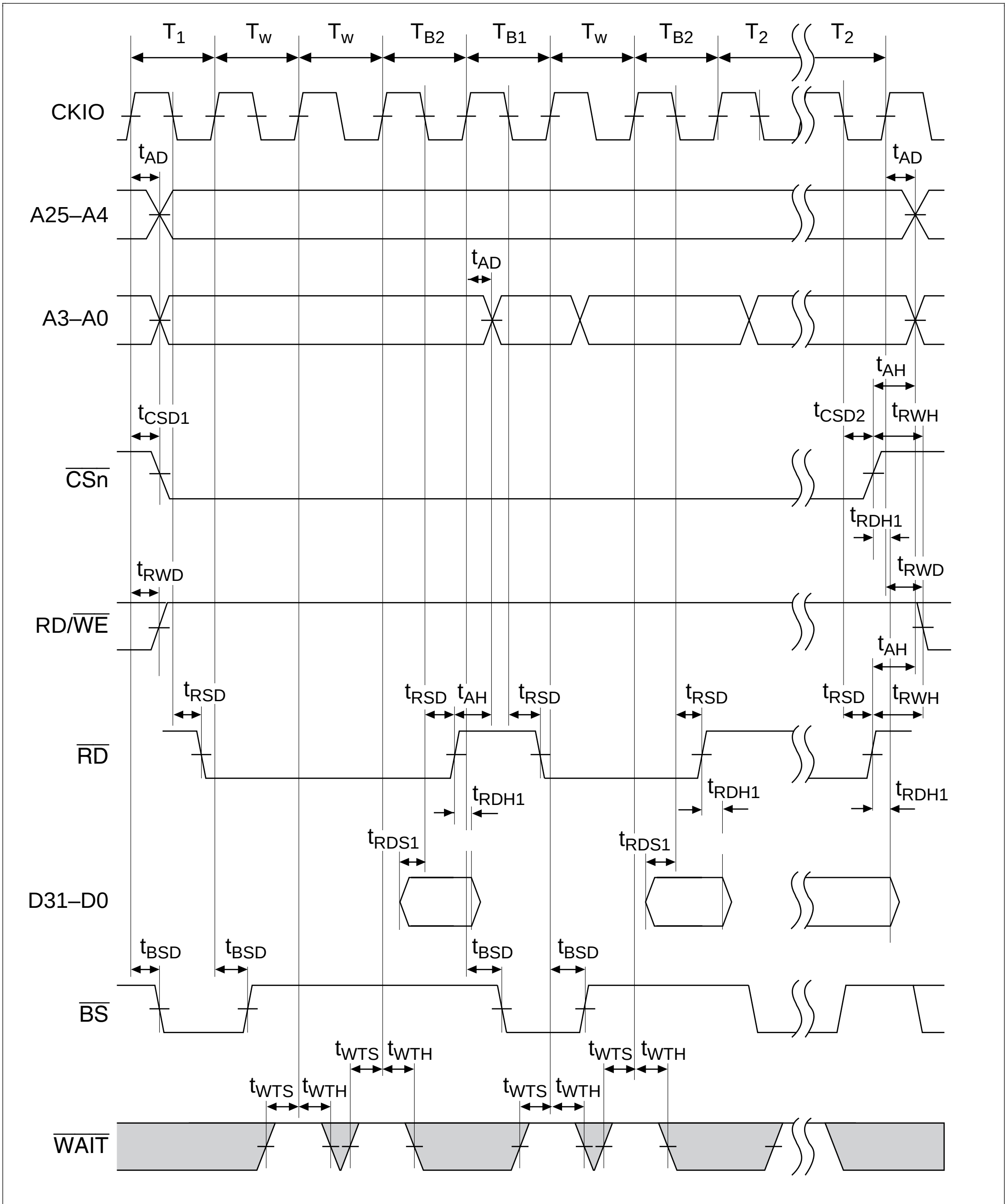
Figure 16.19 Basic Bus Cycle (External Wait)

16.3.5 Burst ROM Timing



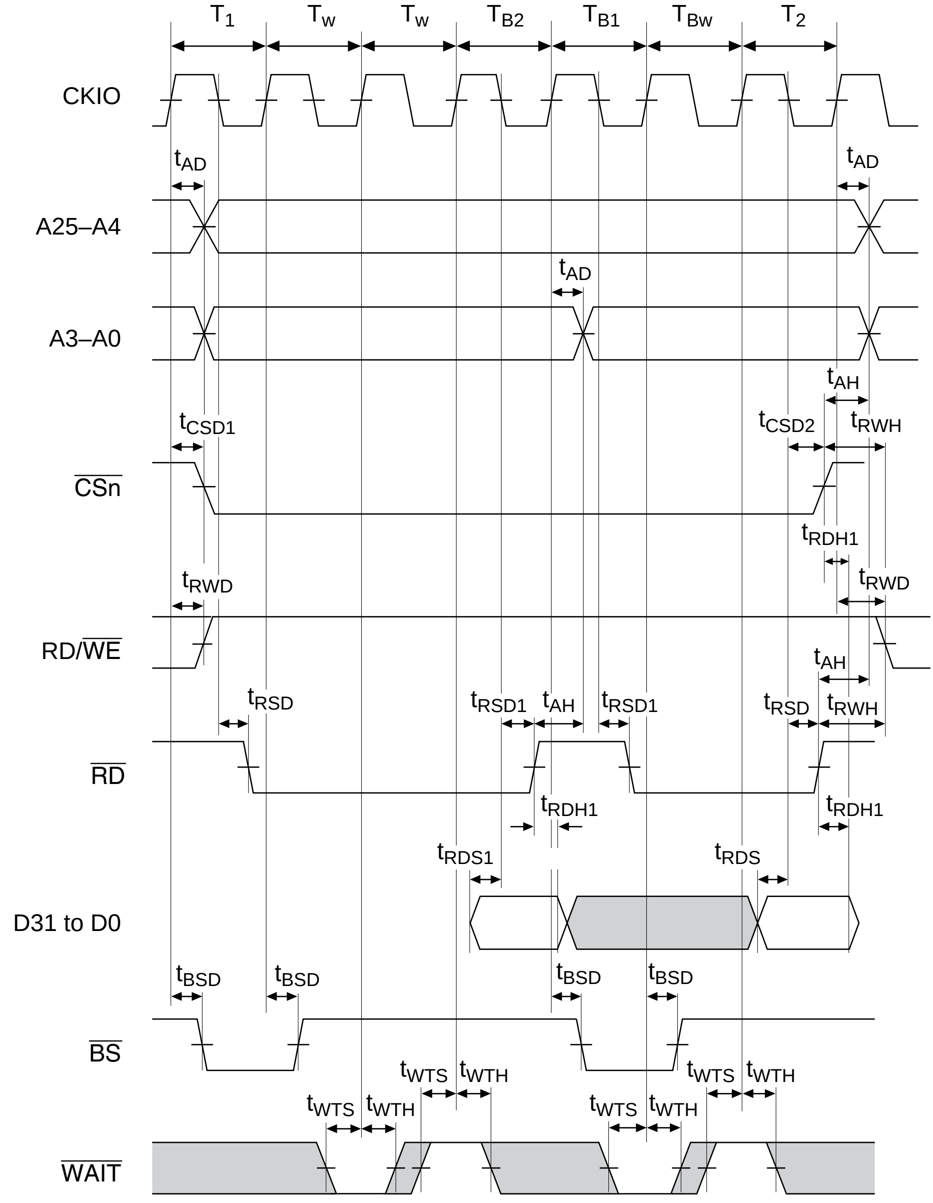
Note: In the write cycle, the basic bus cycle is performed.

Figure 16.20 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 16.21 Burst ROM Bus Cycle (2 Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 16.22 Burst ROM Bus Cycle (External Wait Input)

16.3.6 DRAM Timing

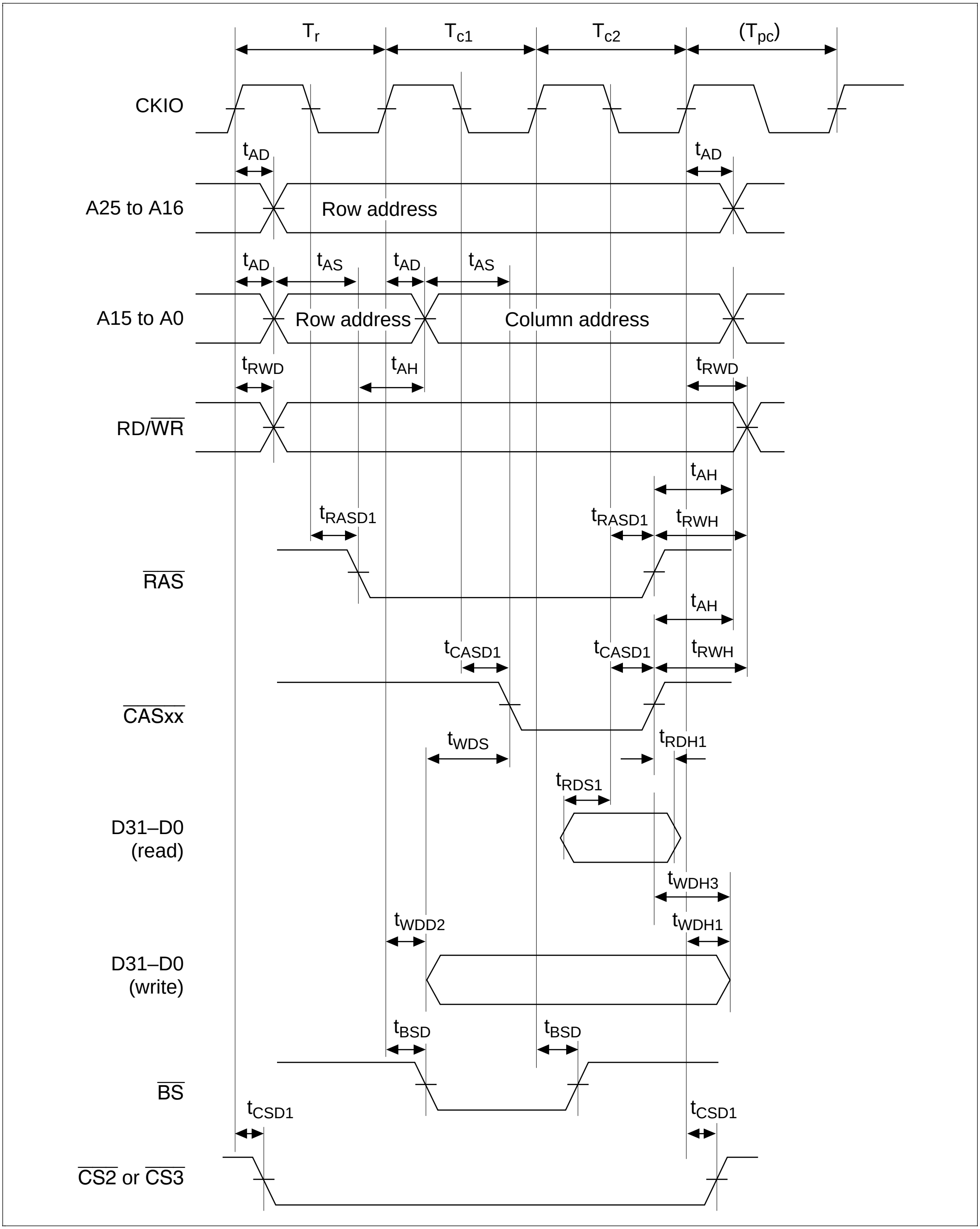


Figure 16.23 DRAM Bus Cycle (RCD = 0, AnW = 1, TPC = 0)

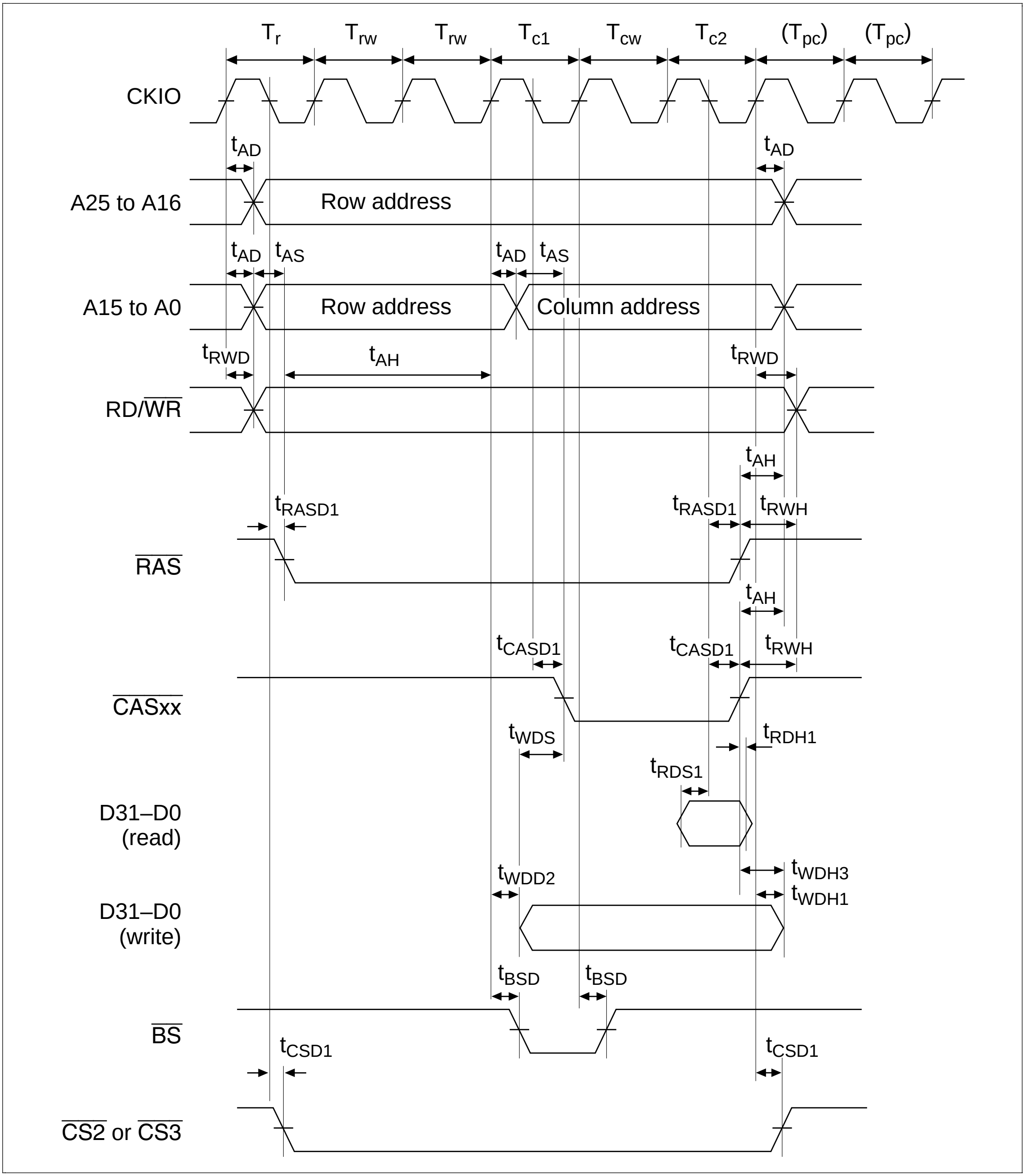


Figure 16.24 DRAM Bus Cycle (RCD = 2, AnW = 2, TPC = 1)

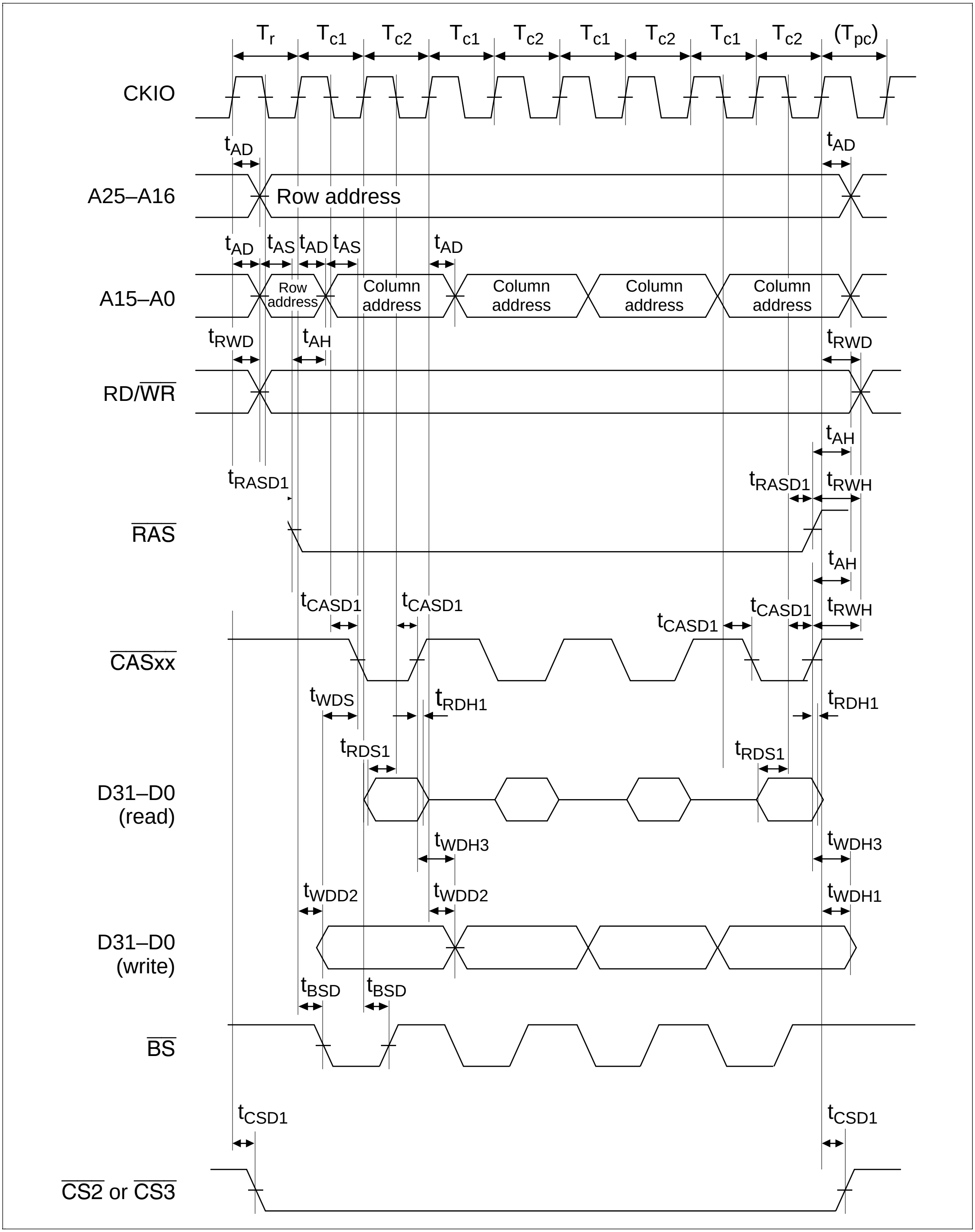


Figure 16.25 DRAM Burst Bus Cycle (RCD = 0, AnW = 1, TPC = 0)

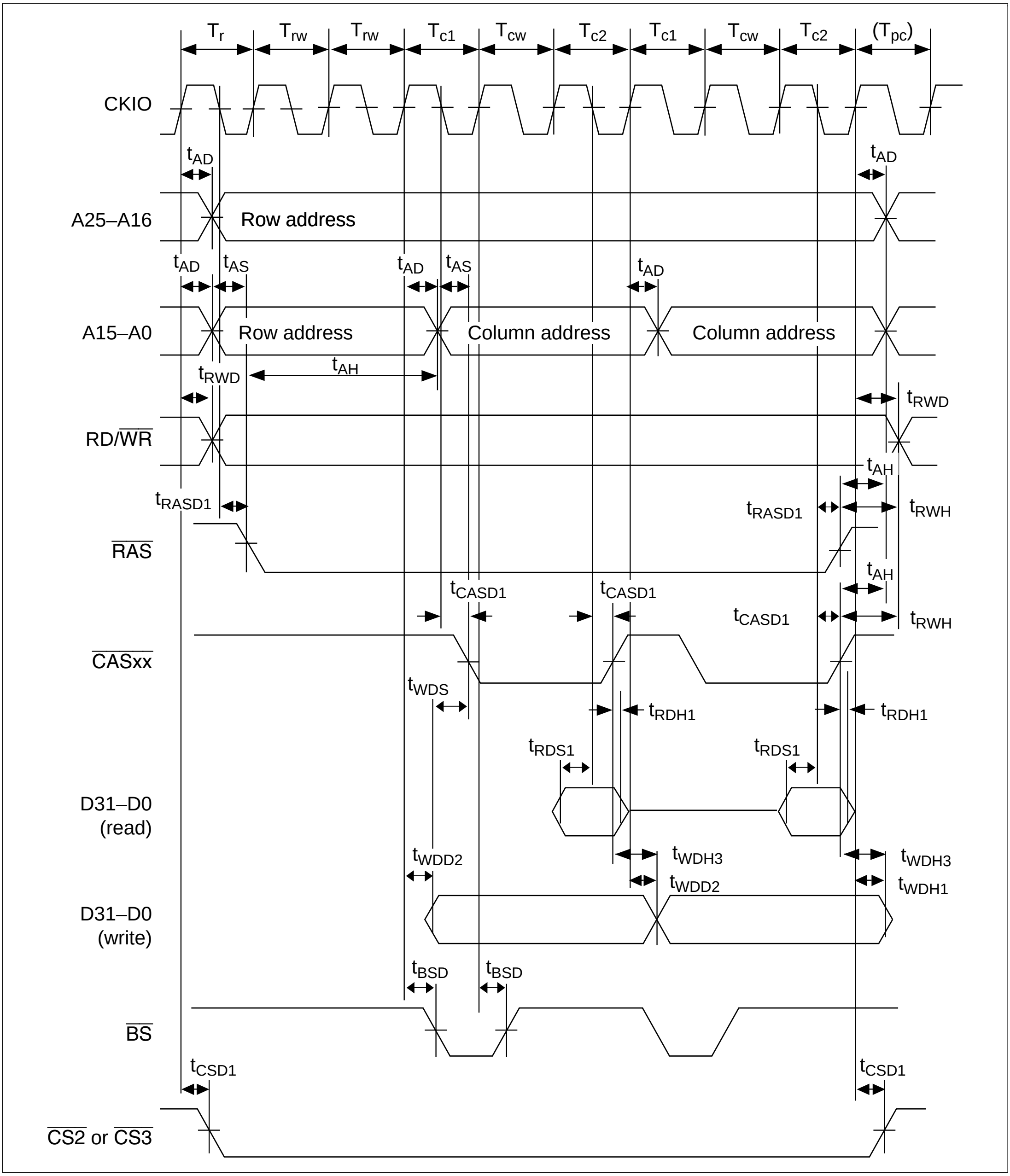


Figure 16.26 DRAM Burst Bus Cycle (RCD = 2, AnW = 2, TPC = 0)

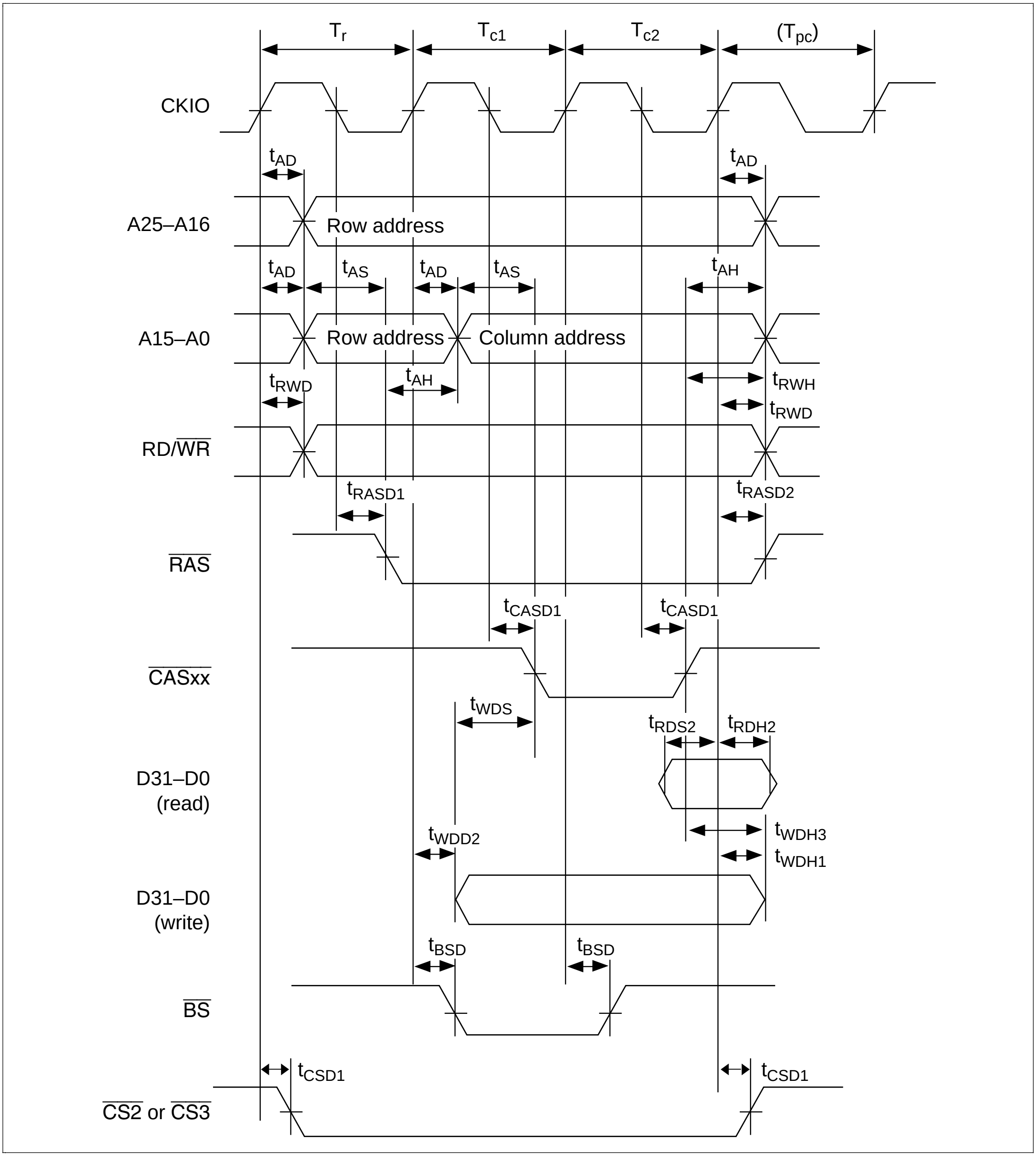


Figure 16.27 DRAM Bus Cycle (EDO Mode, RCD = 0, AnW = 1, TPC = 0)

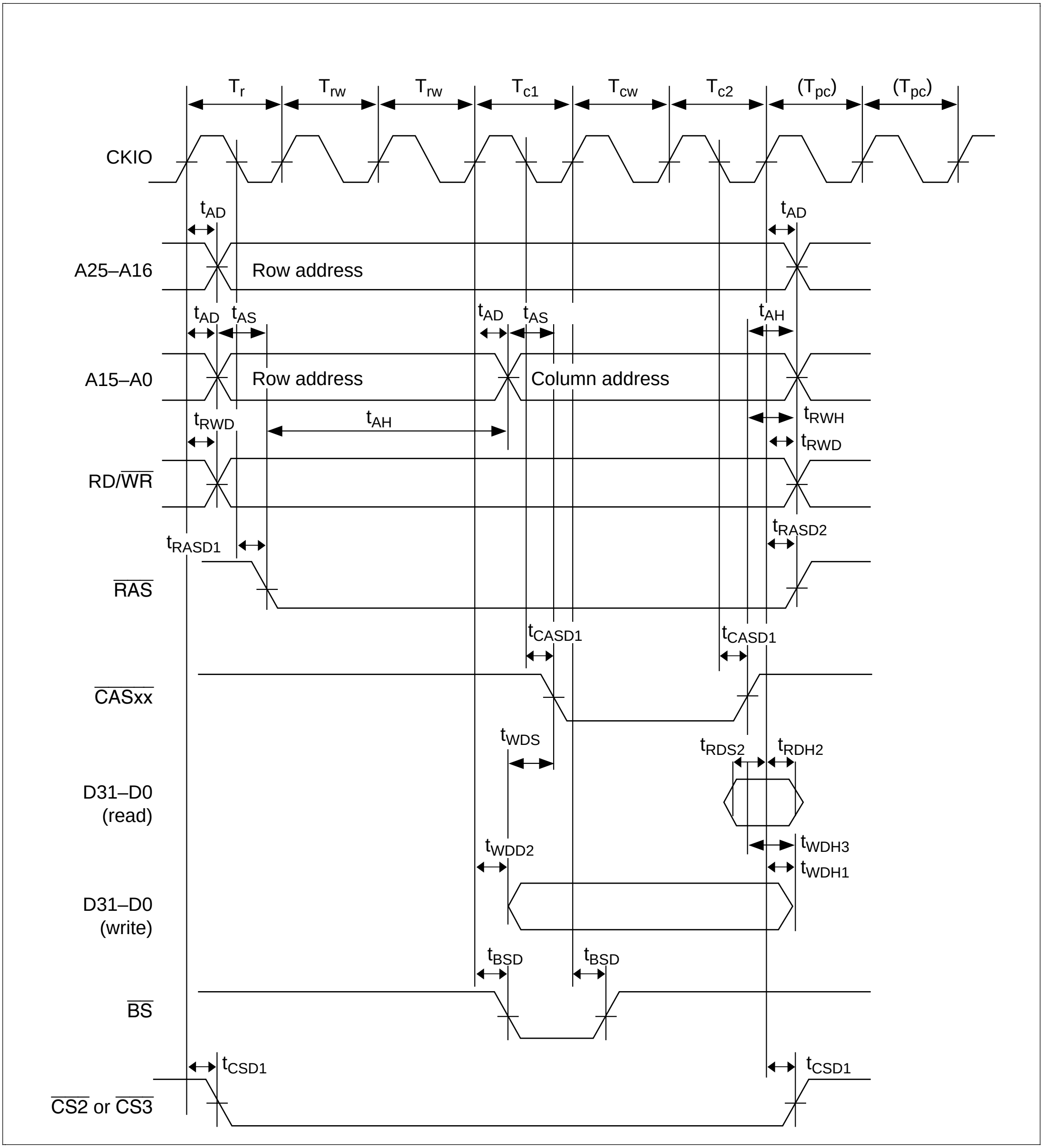


Figure 16.28 DRAM Bus Cycle (EDO Mode, RCD = 2, AnW = 2, TPC = 1)

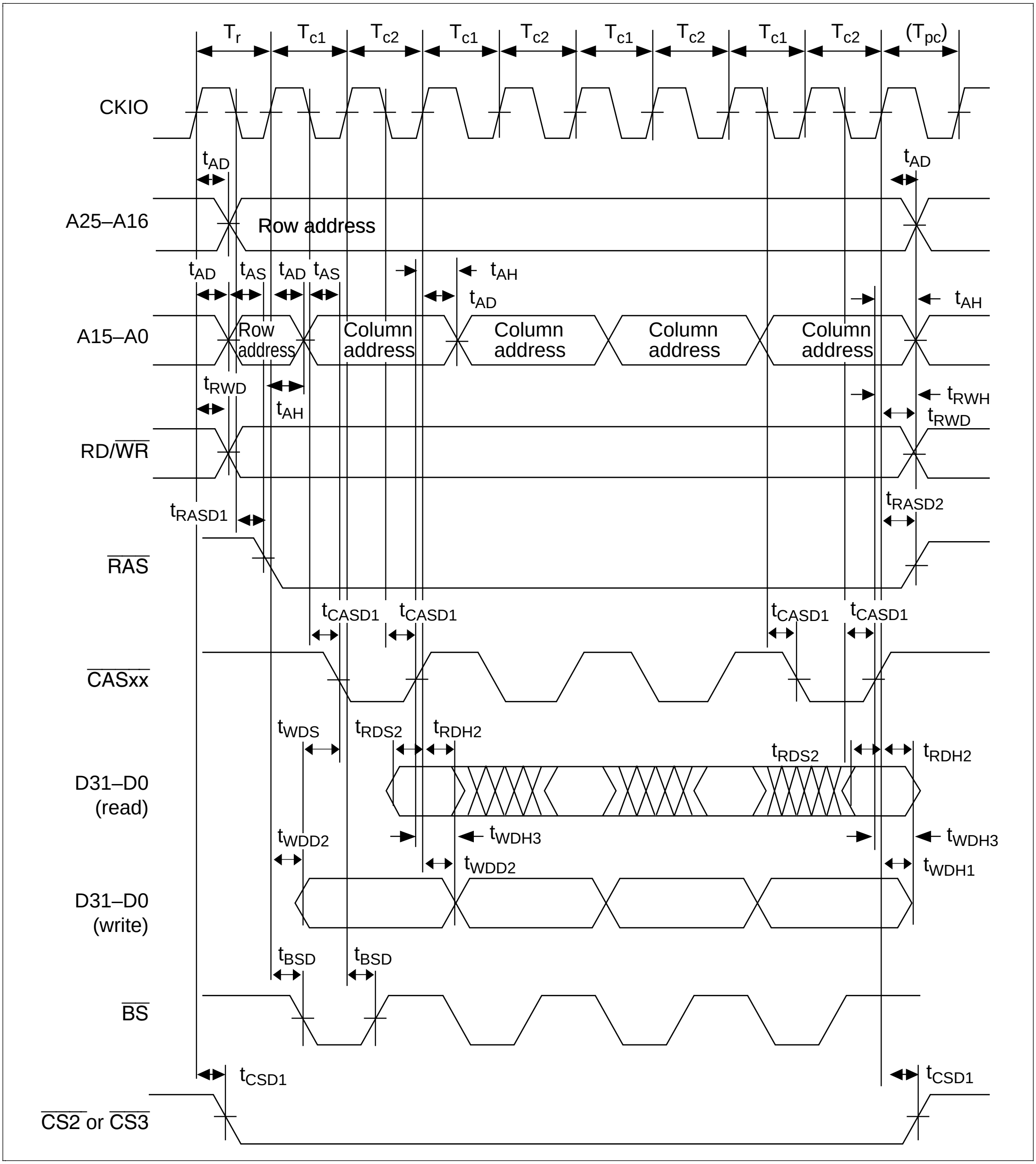


Figure 16.29 DRAM Burst Bus Cycle (EDO Mode, RCD = 0, AnW = 1, TPC = 0)

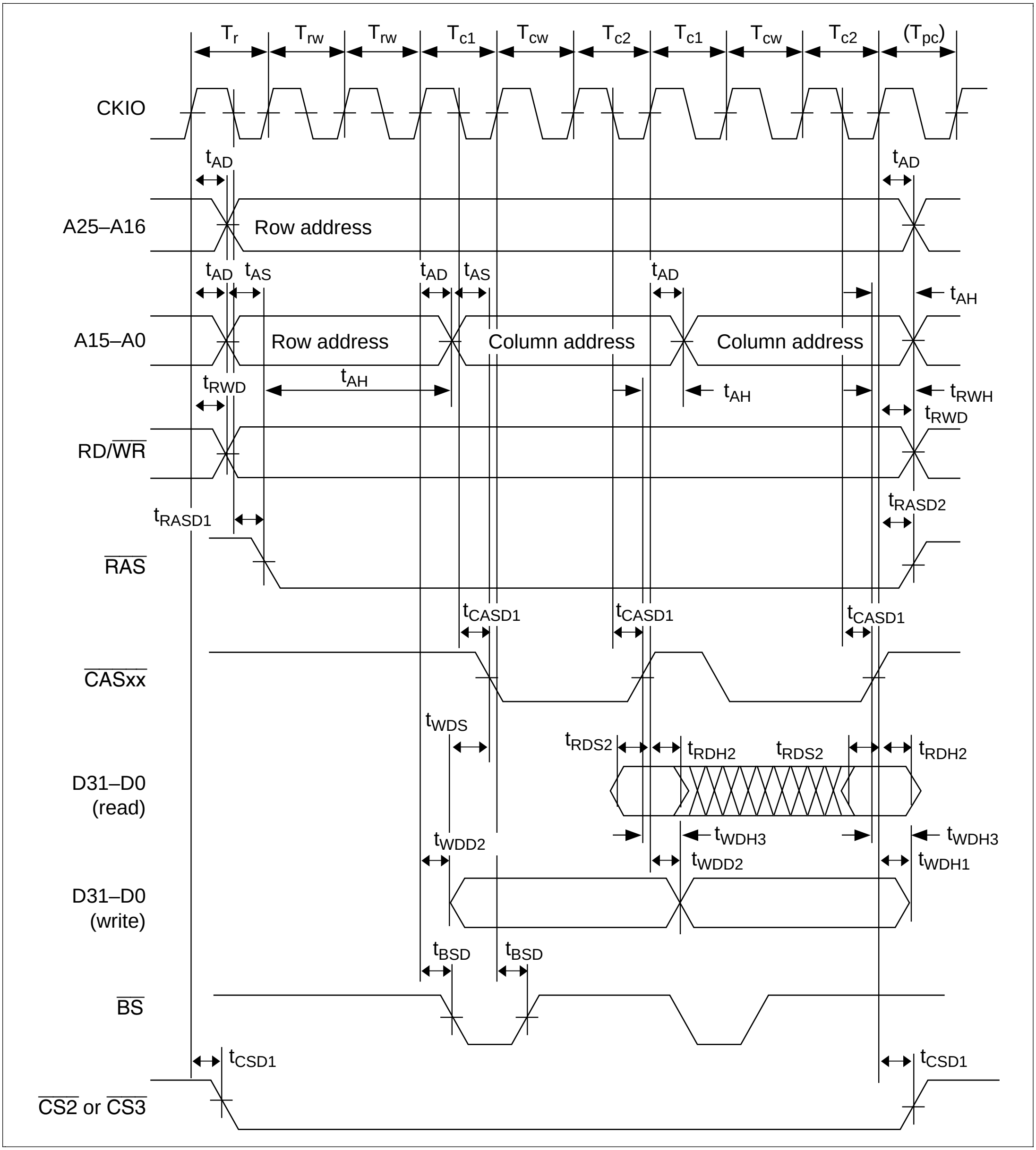


Figure 16.30 DRAM Burst Bus Cycle (EDO Mode, RCD = 2, AnW = 2, TPC = 0)

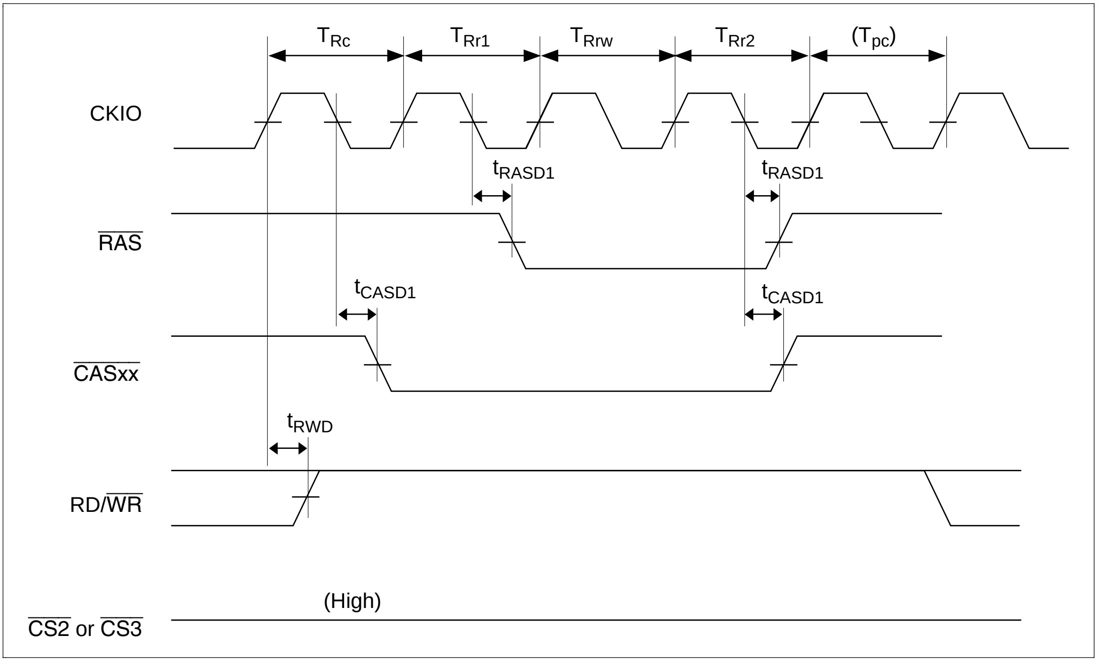


Figure 16.31 DRAM CAS-Before-RAS Refresh Cycle (TRAS = 0, TPC = 0)

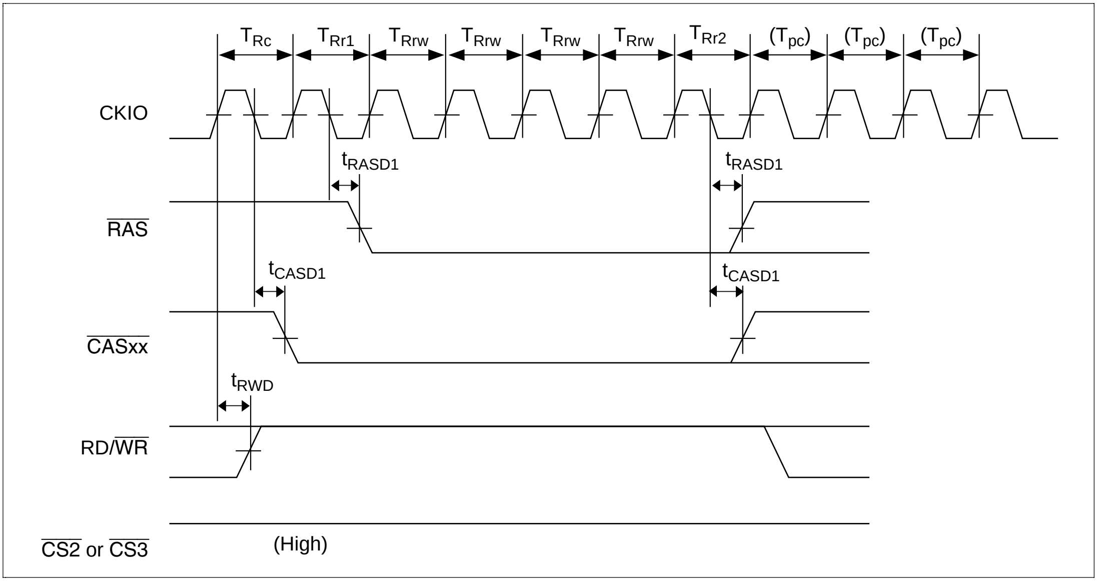


Figure 16.32 DRAM CAS-Before-RAS Refresh Cycle (TRAS = 3, TPC = 2)

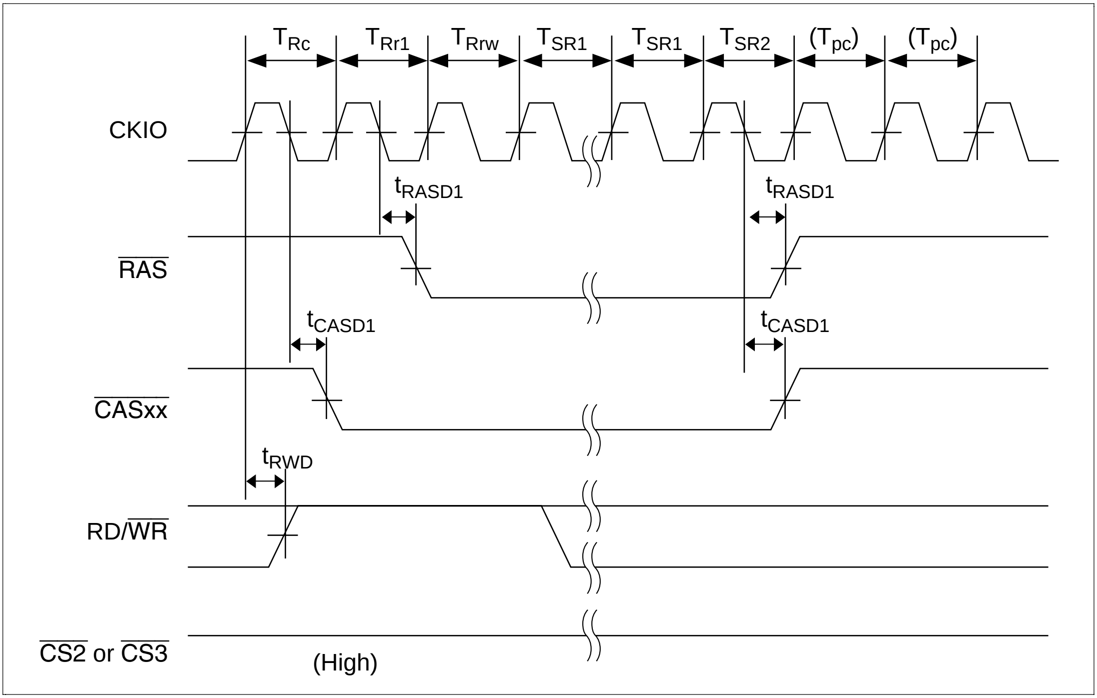


Figure 16.33 DRAM Self-Refresh Cycle (TPC = 0)

16.3.7 Synchronous DRAM Timing

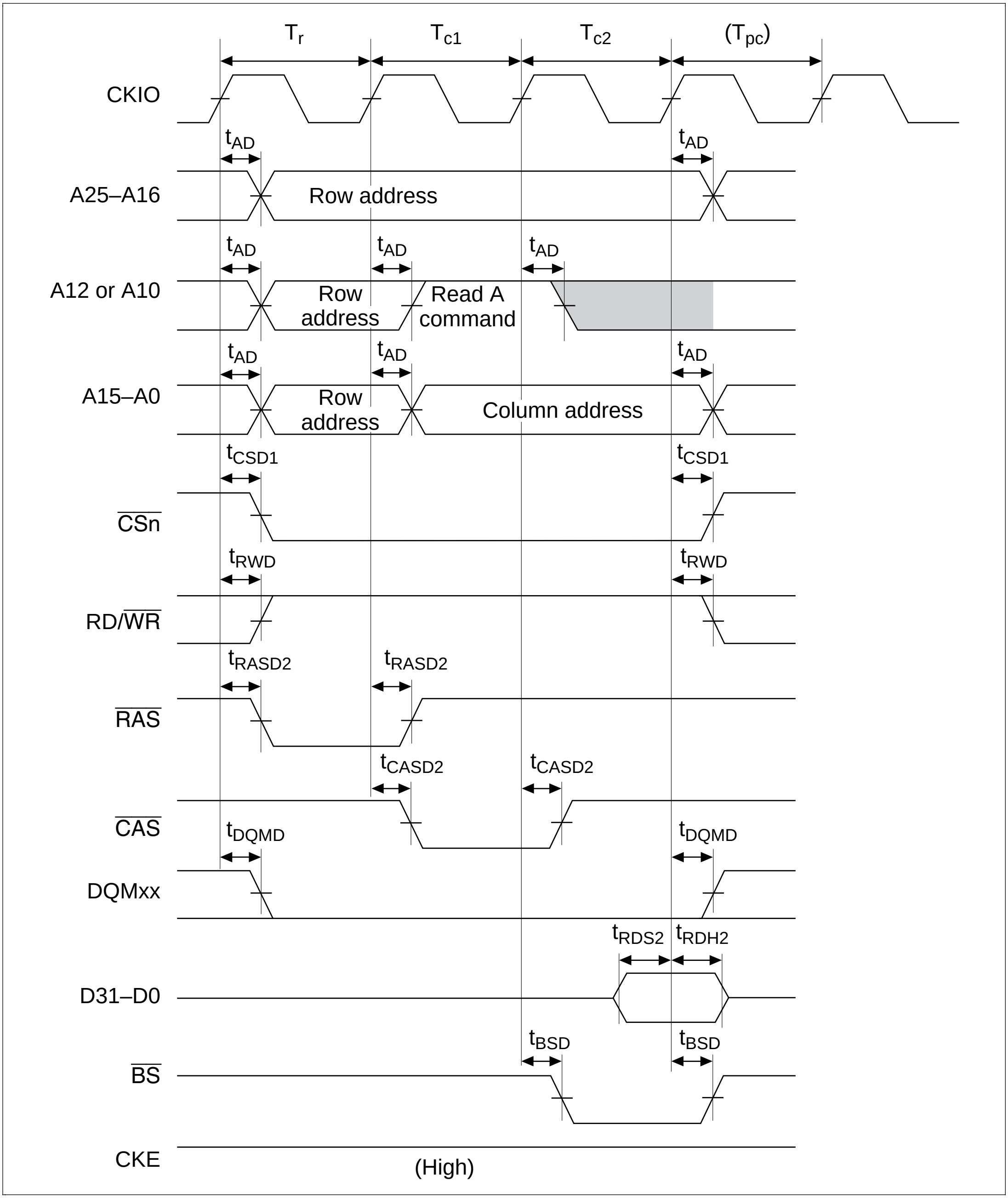


Figure 16.34 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0)

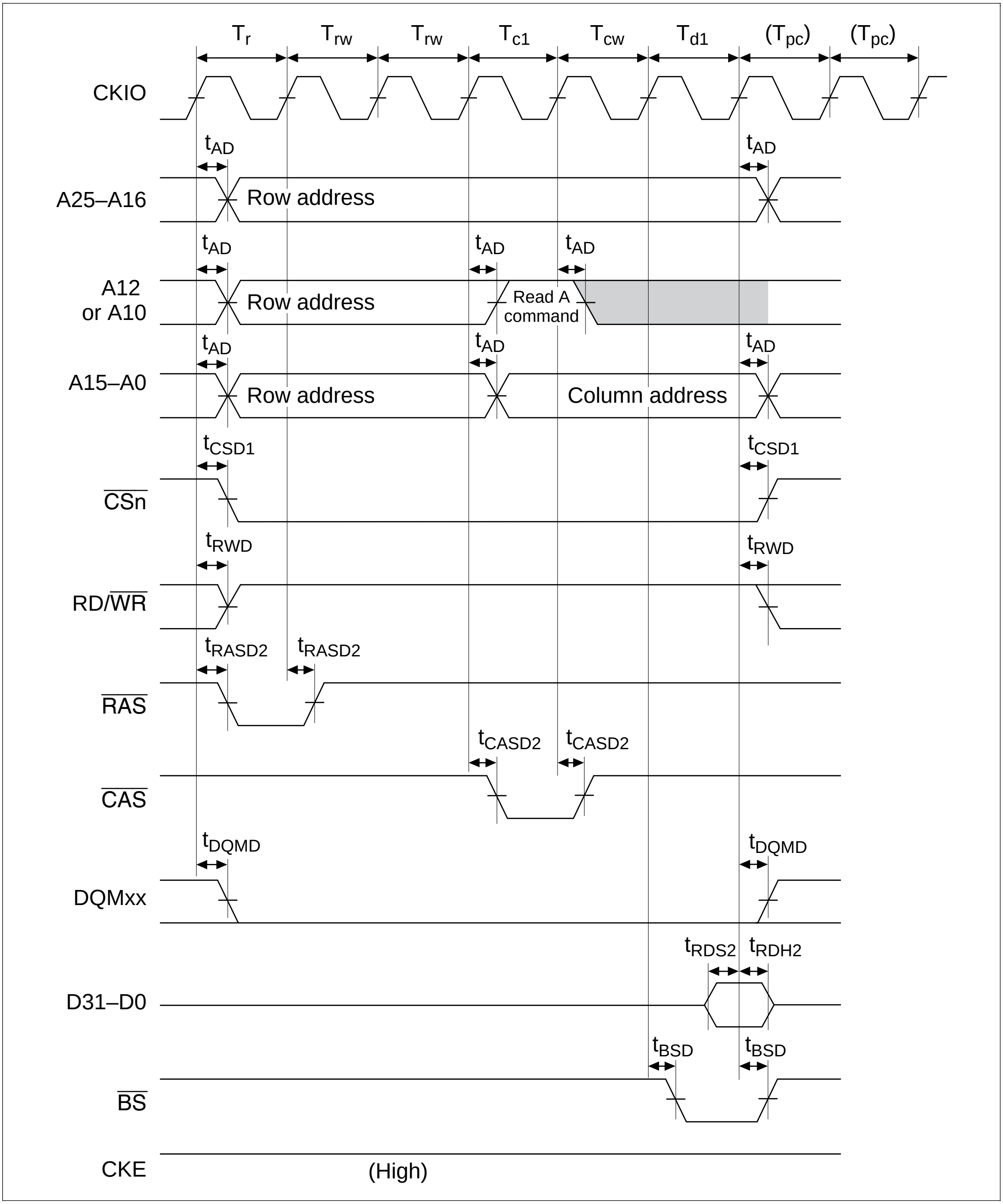


Figure 16.35 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1)

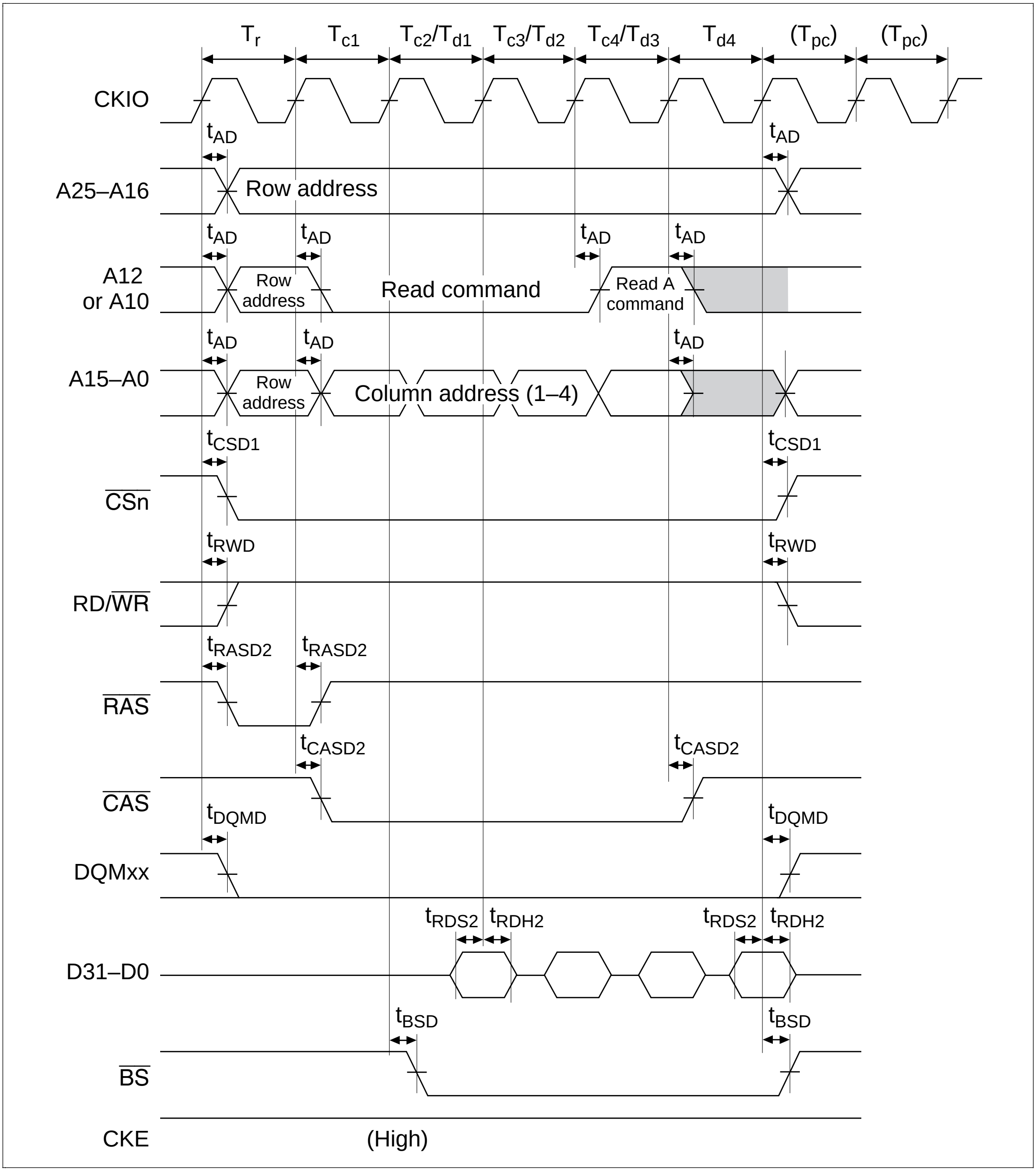


Figure 16.36 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

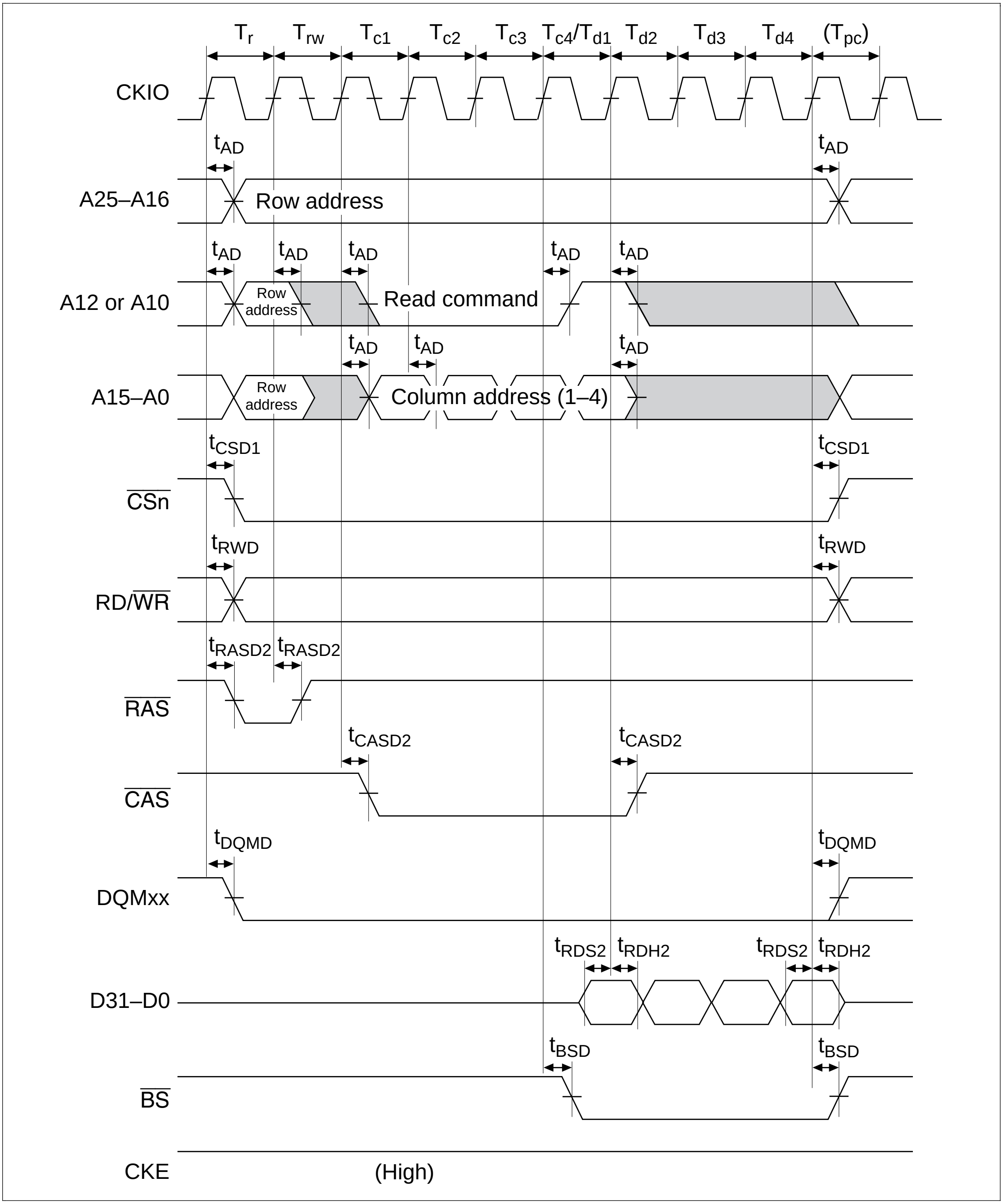


Figure 16.37 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

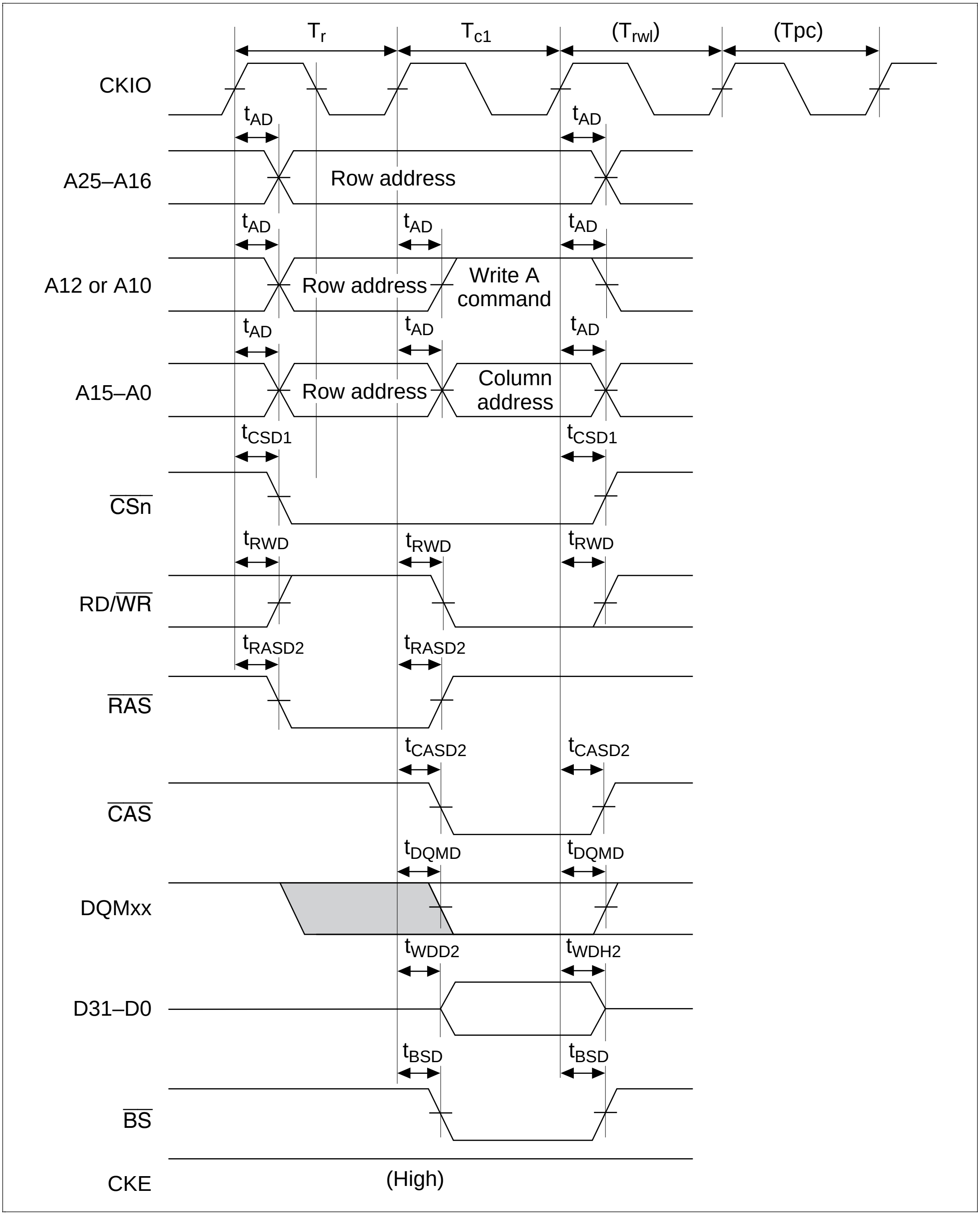


Figure 16.38 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0)

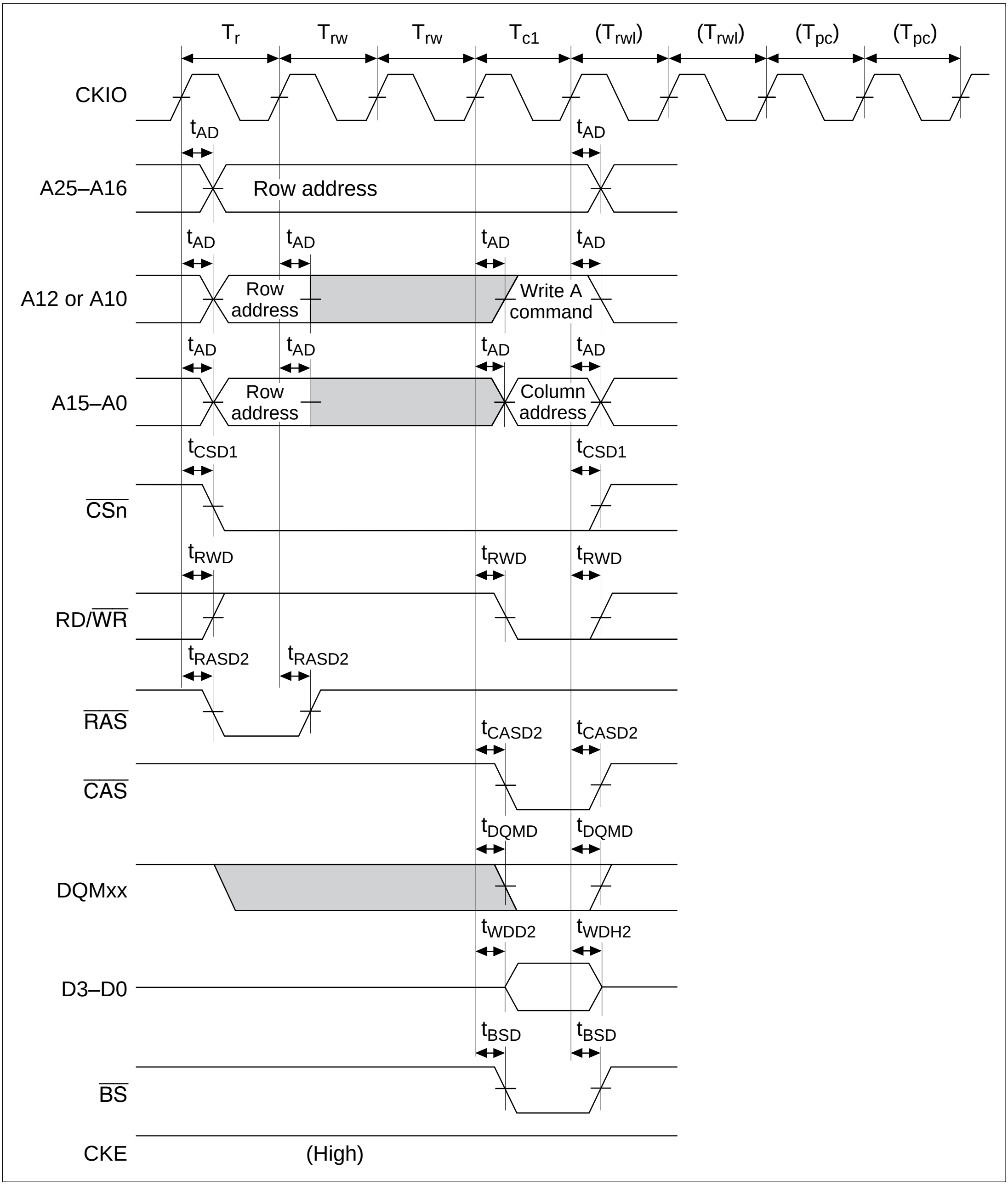


Figure 16.39 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1)

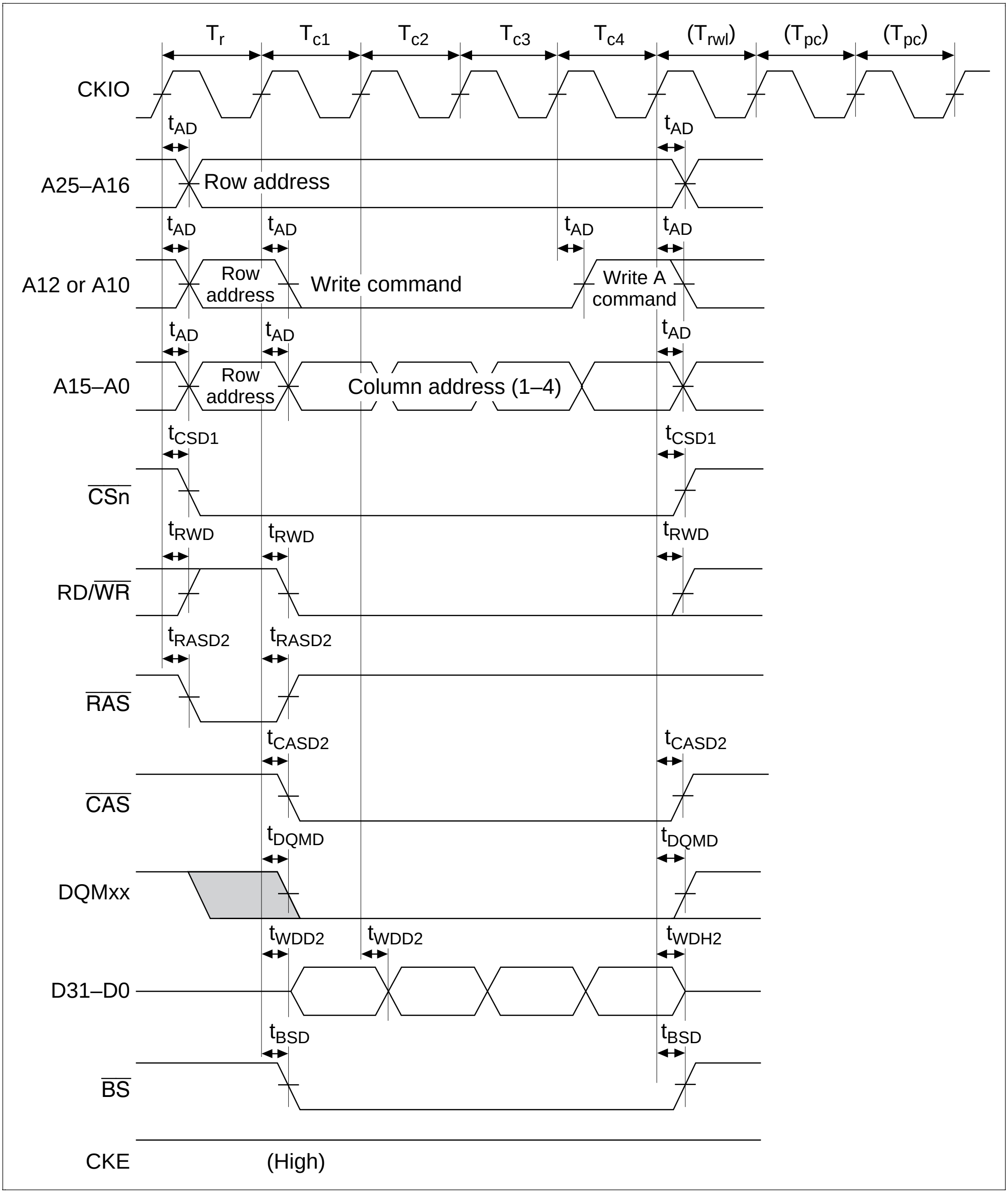


Figure 16.40 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

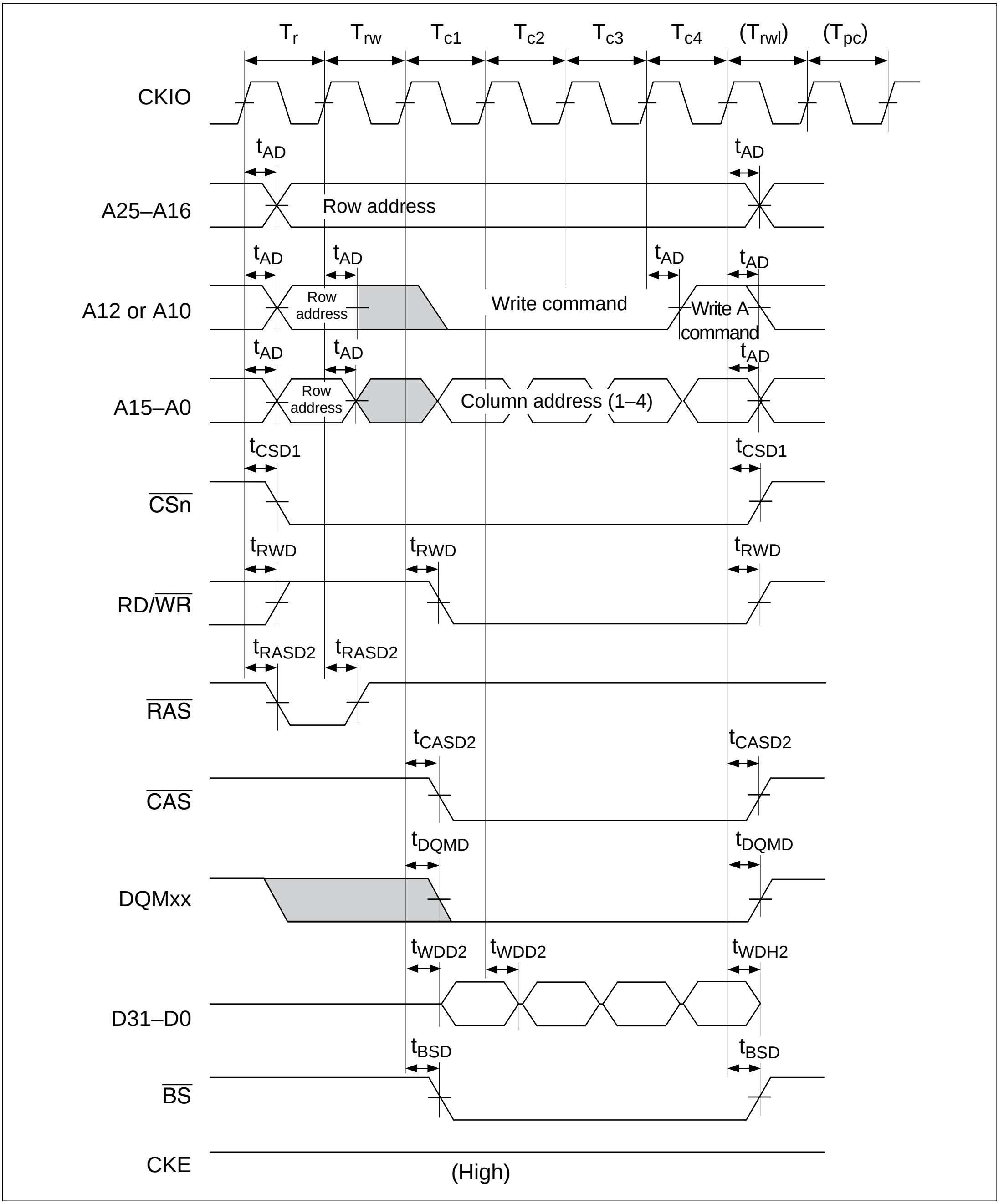


Figure 16.41 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

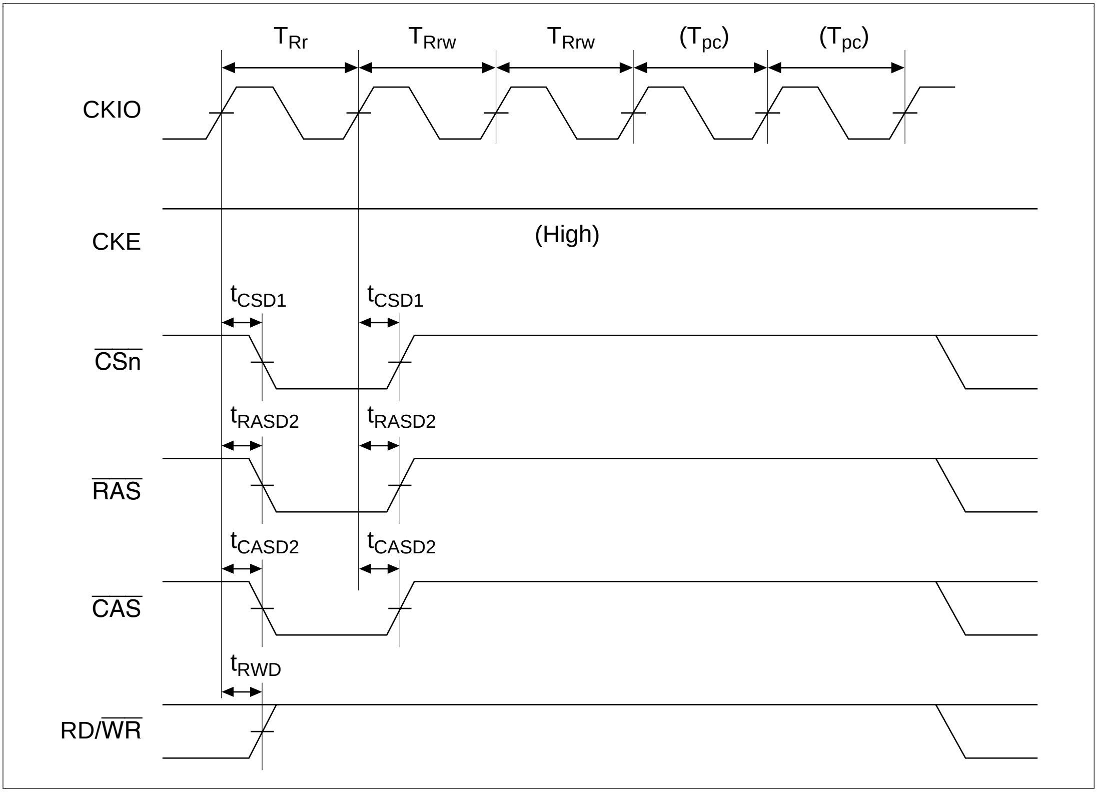


Figure 16.42 Synchronous DRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

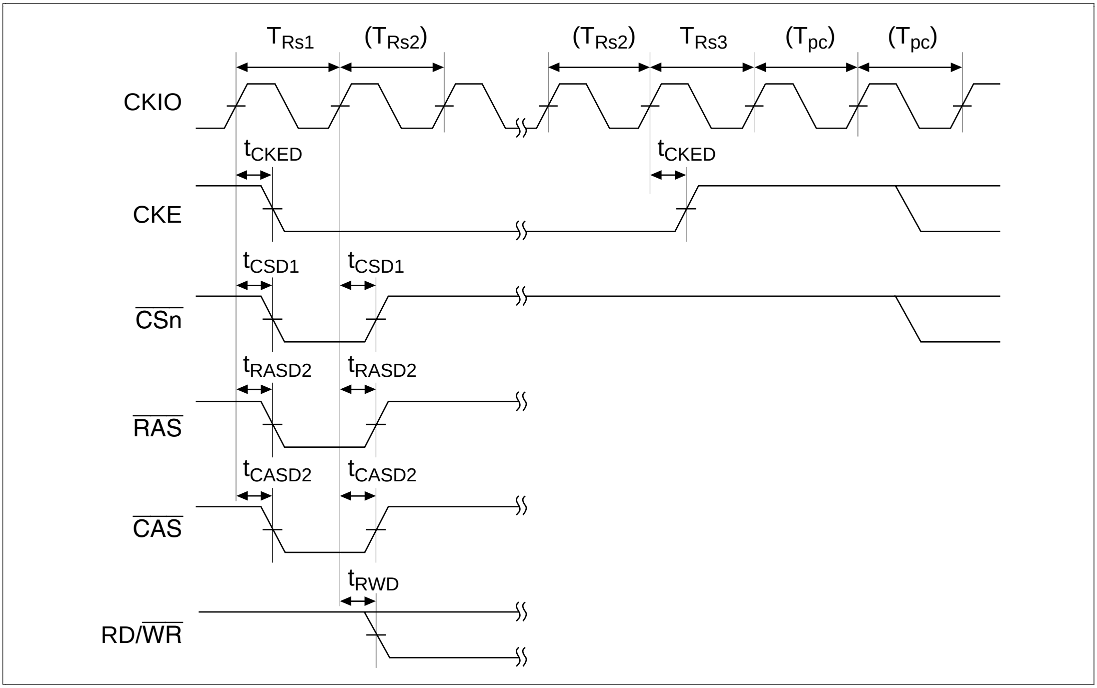


Figure 16.43 Synchronous DRAM Self-Refresh Cycle (TPC = 0)

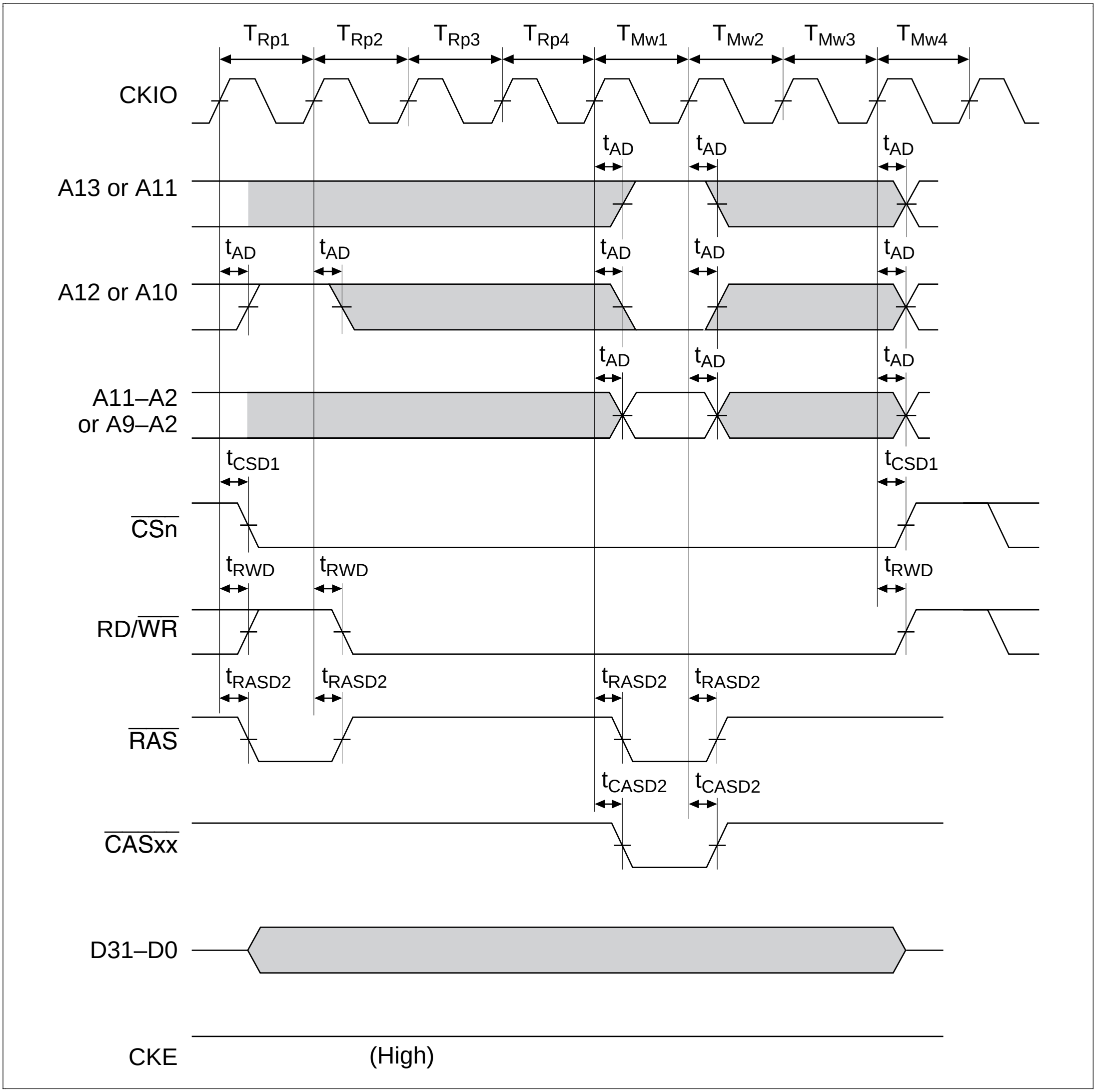


Figure 16.44 Synchronous DRAM Mode Register Write Cycle

16.3.8 Pseudo-SRAM Timing

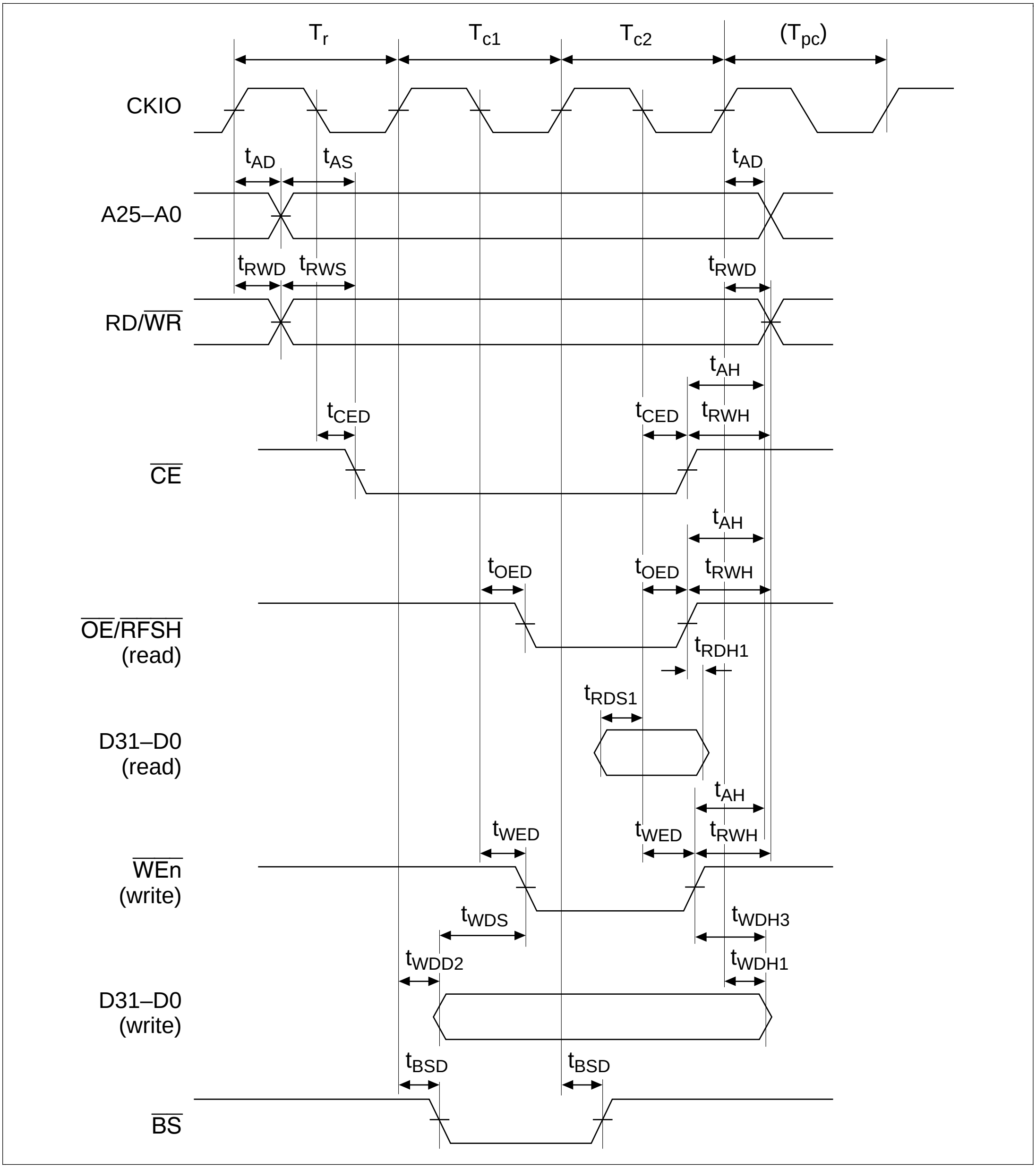


Figure 16.45 Pseudo-SRAM Bus Cycle (RCD = 0, A3W = 1, TPC = 0)

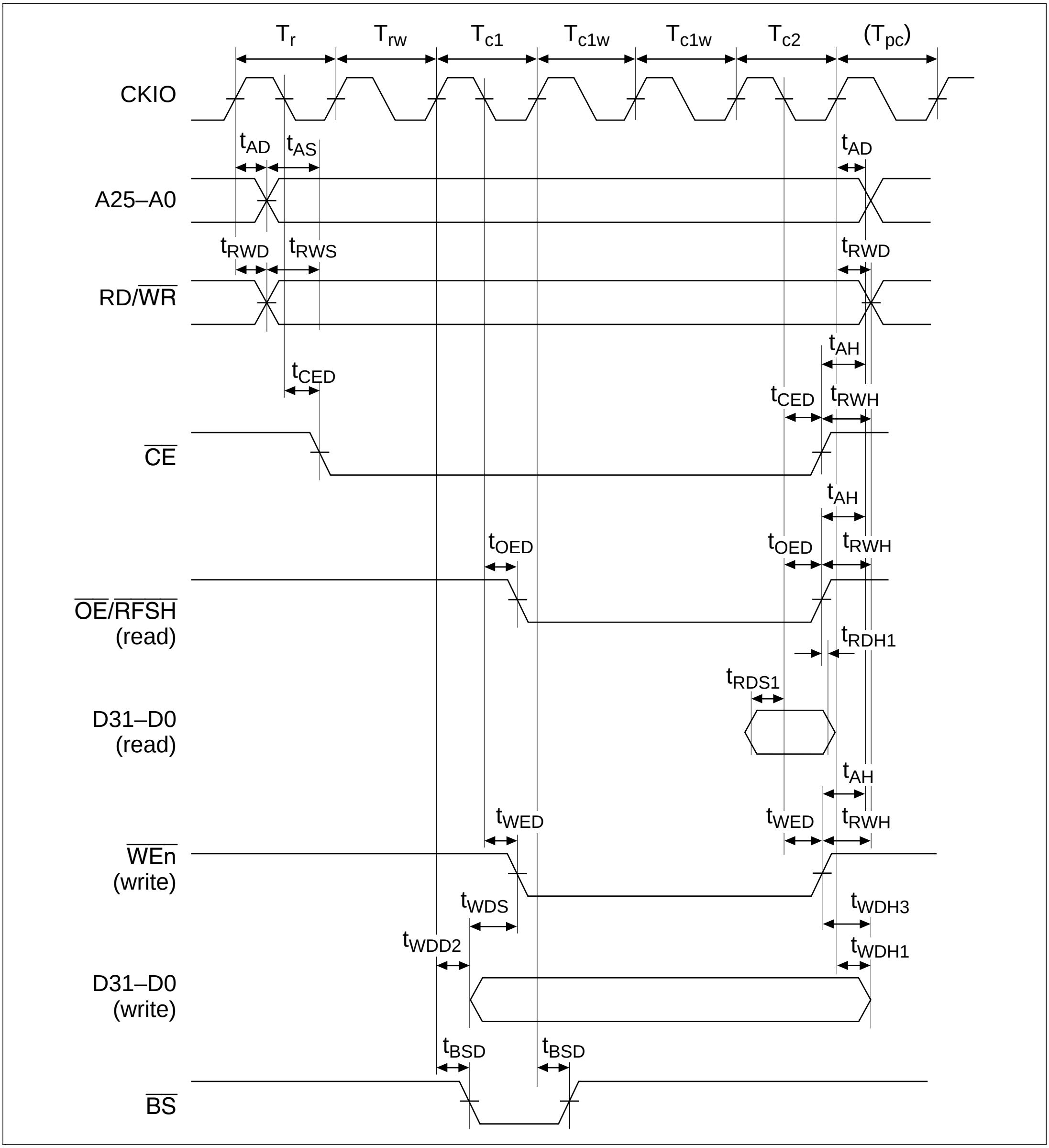


Figure 16.46 Pseudo-SRAM Read Cycle (RCD = 1, A3W = 3, TPC = 0)

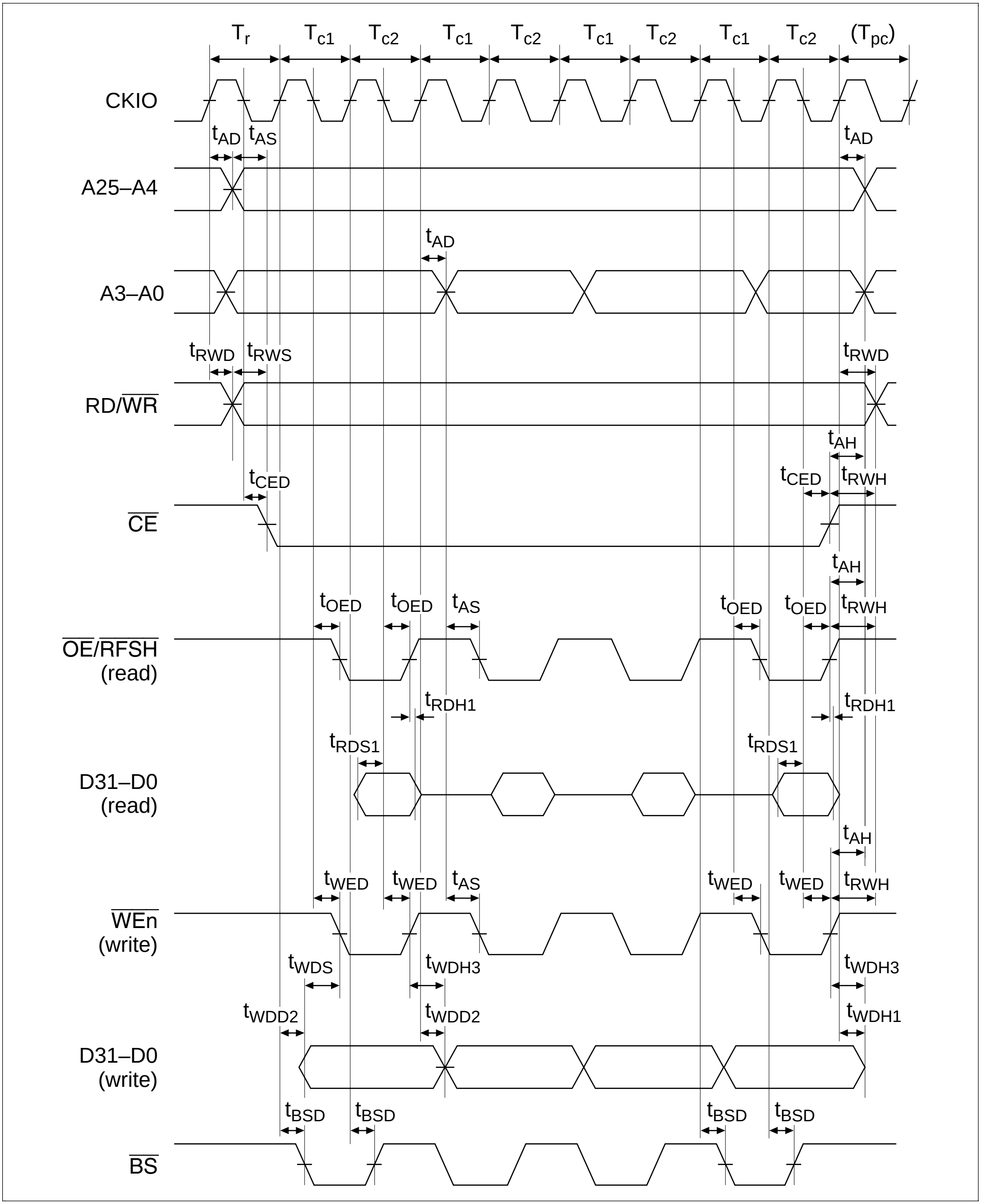


Figure 16.47 Pseudo-SRAM Bus Cycle (Static Column Mode, RCD = 0, A3W = 1, TPC = 0)

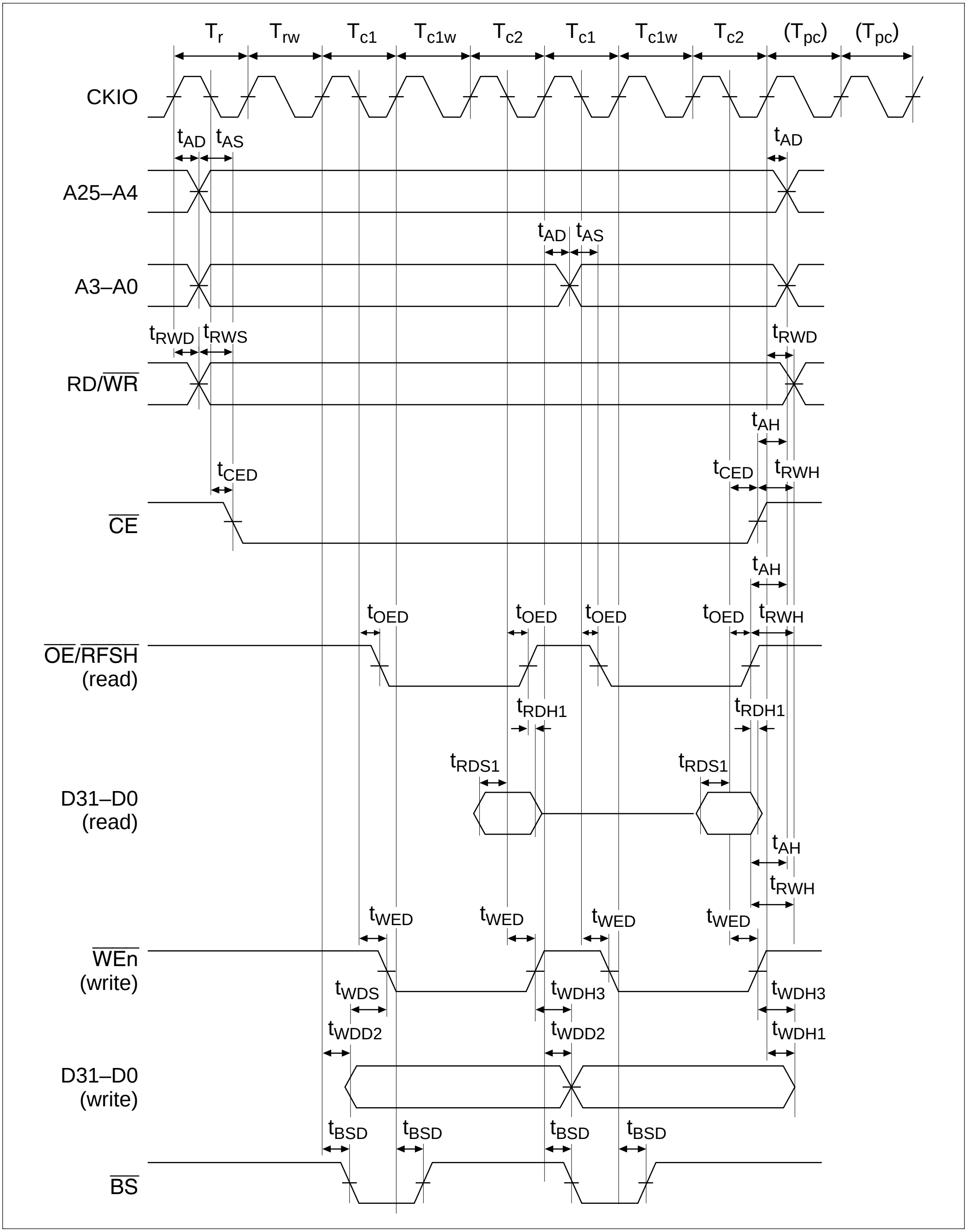


Figure 16.48 Pseudo-SRAM Bus Cycle (Static Column Mode, RCD = 1, A3W = 2, TPC = 1)

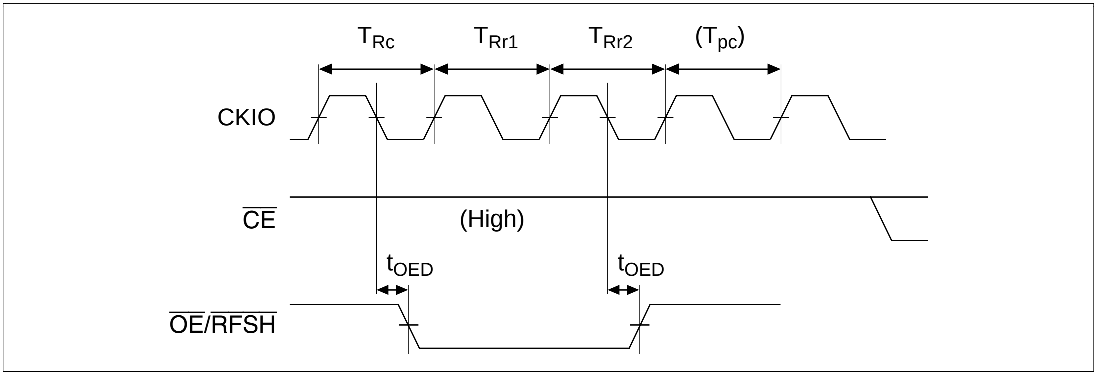


Figure 16.49 Pseudo-SRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

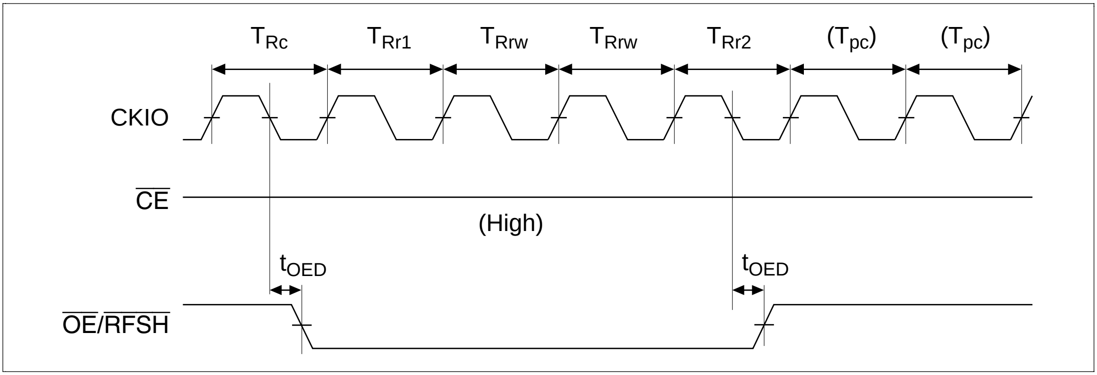


Figure 16.50 Pseudo-SRAM Auto-Refresh Cycle (TRAS = 2, TPC = 1)

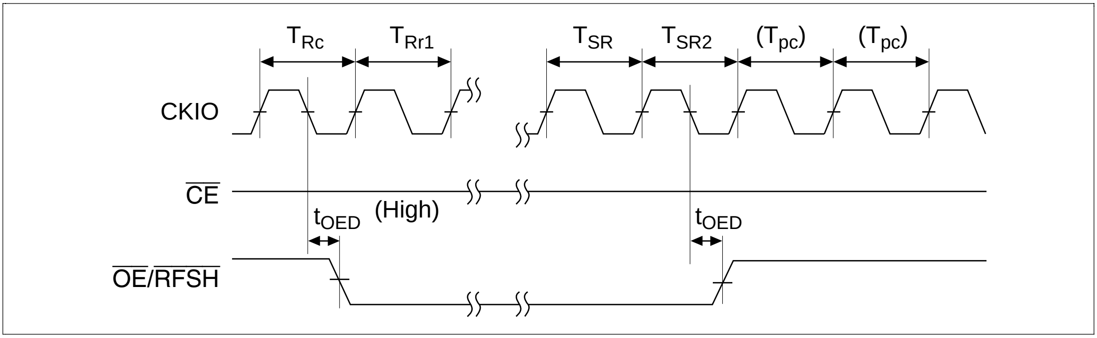


Figure 16.51 Pseudo-SRAM Self-Refresh Cycle (TPC = 0)

16.3.9 PCMCIA Timing

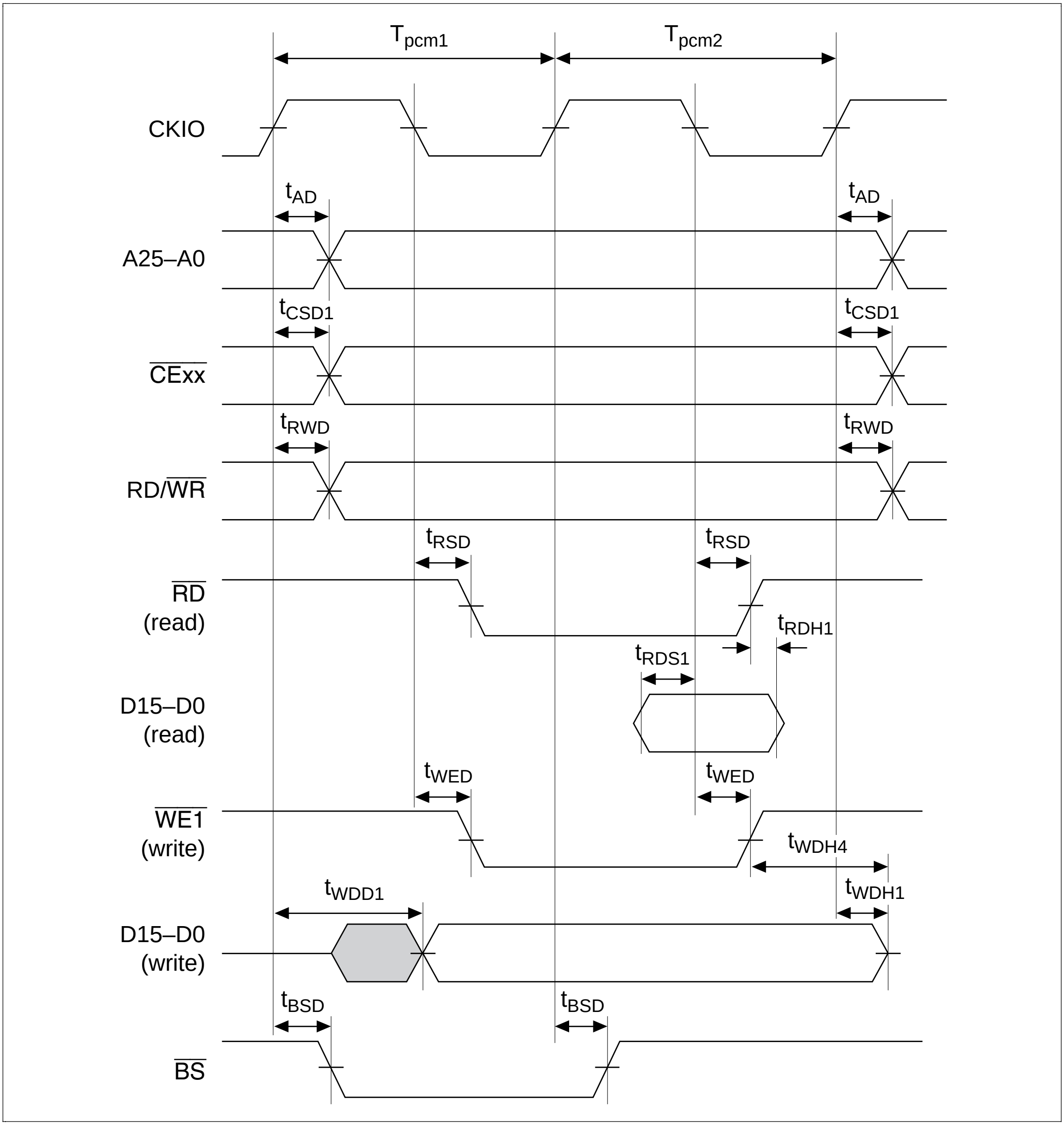


Figure 16.52 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

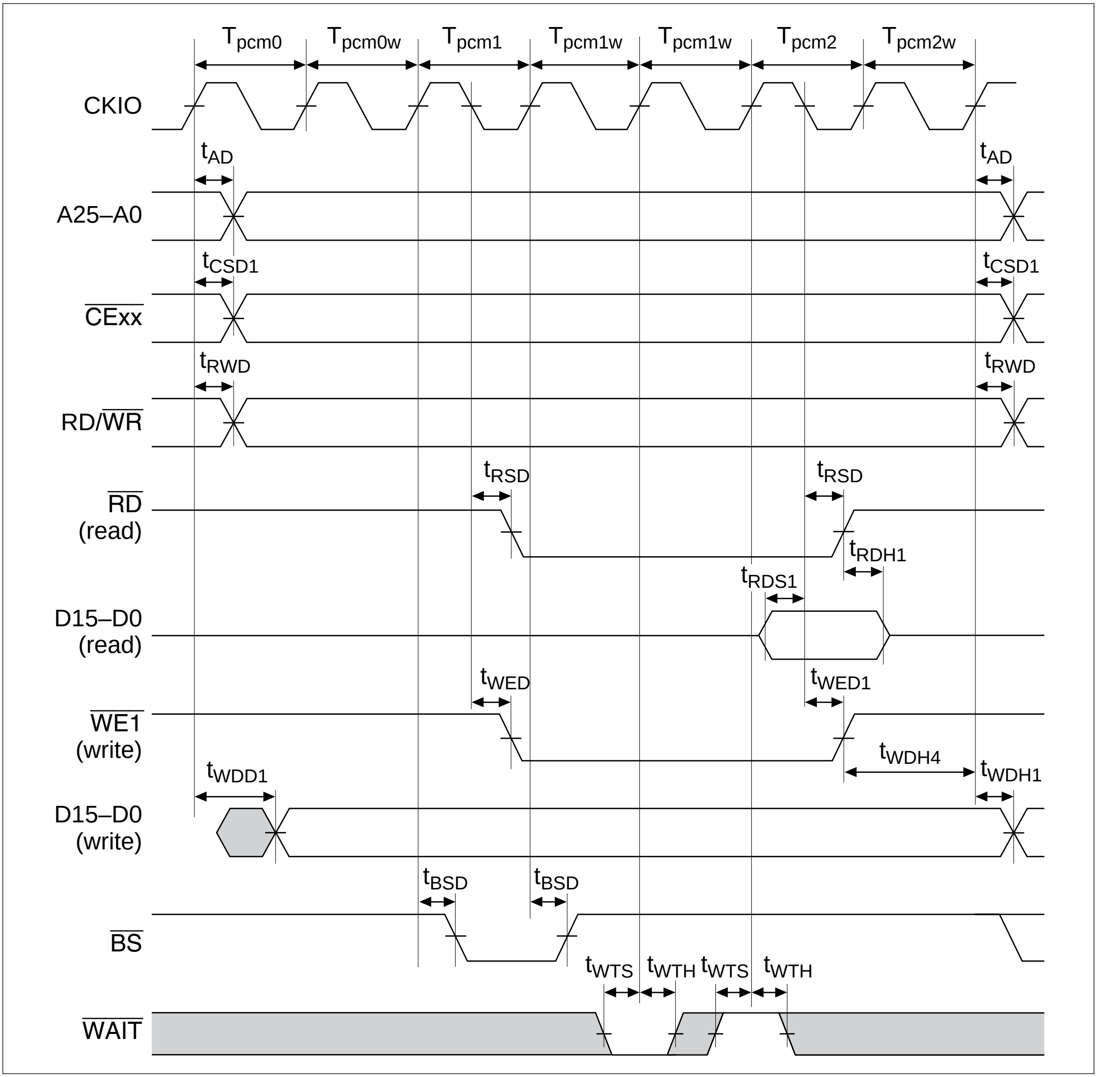
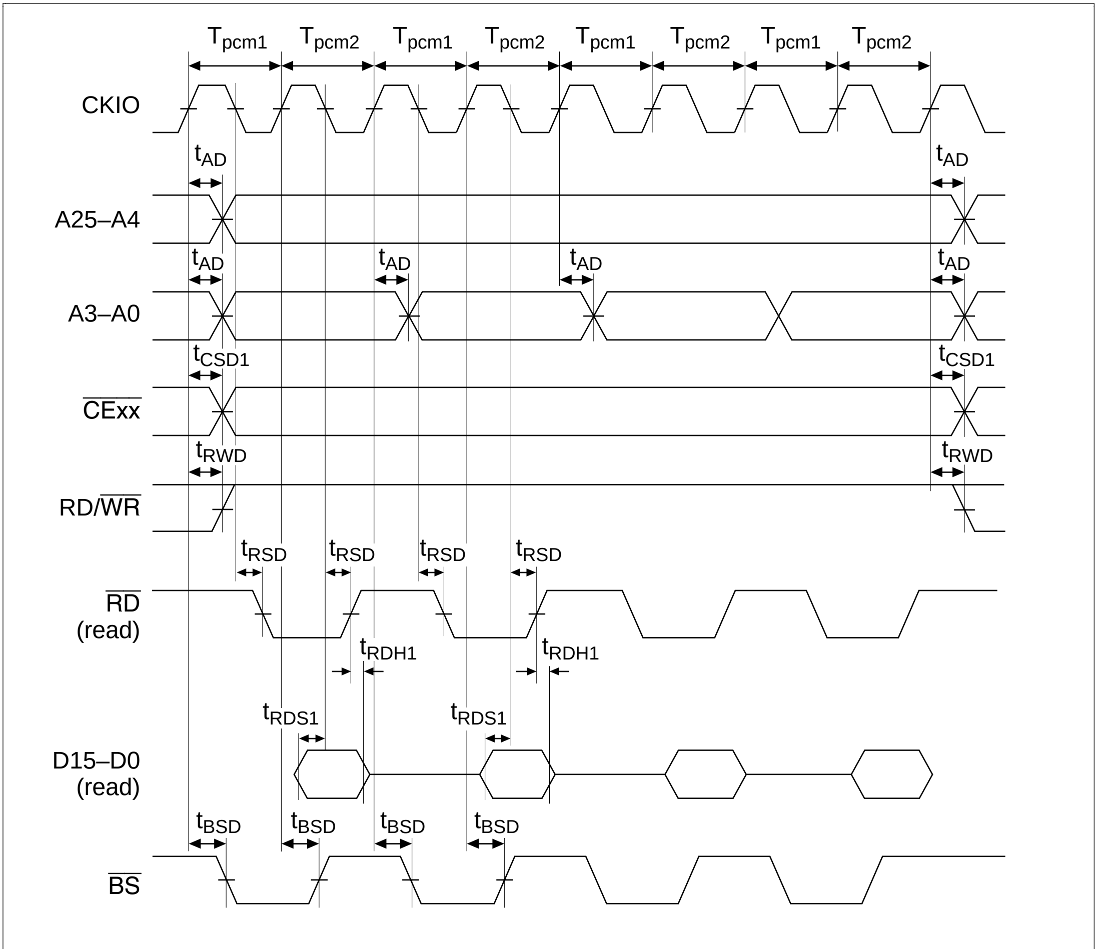
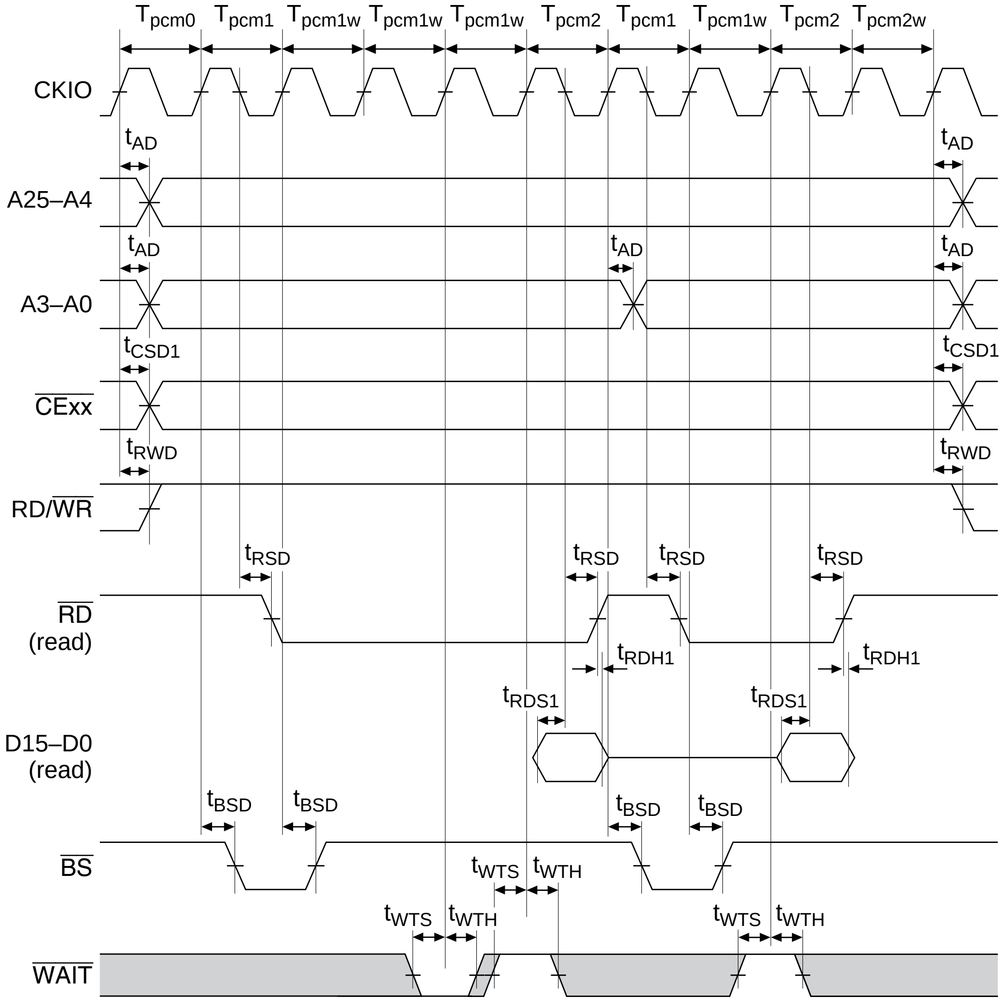


Figure 16.53 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, 1 Wait, External Wait)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 16.54 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 16.55 PCMCIA Memory Bus Cycle (Burst Read, TED = 1, TEH = 1, 2 Waits, Burst Pitch = 3)

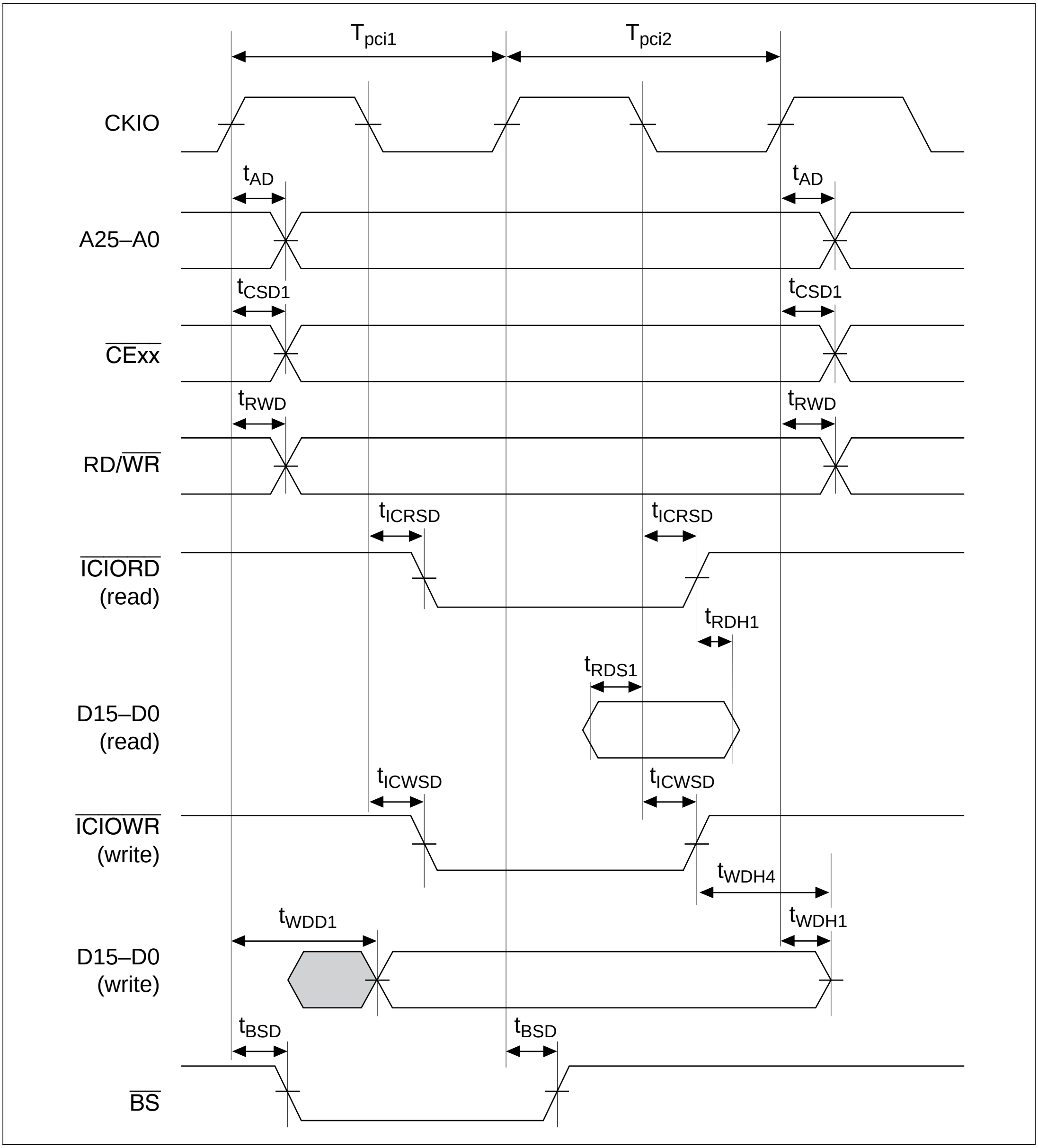


Figure 16.56 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

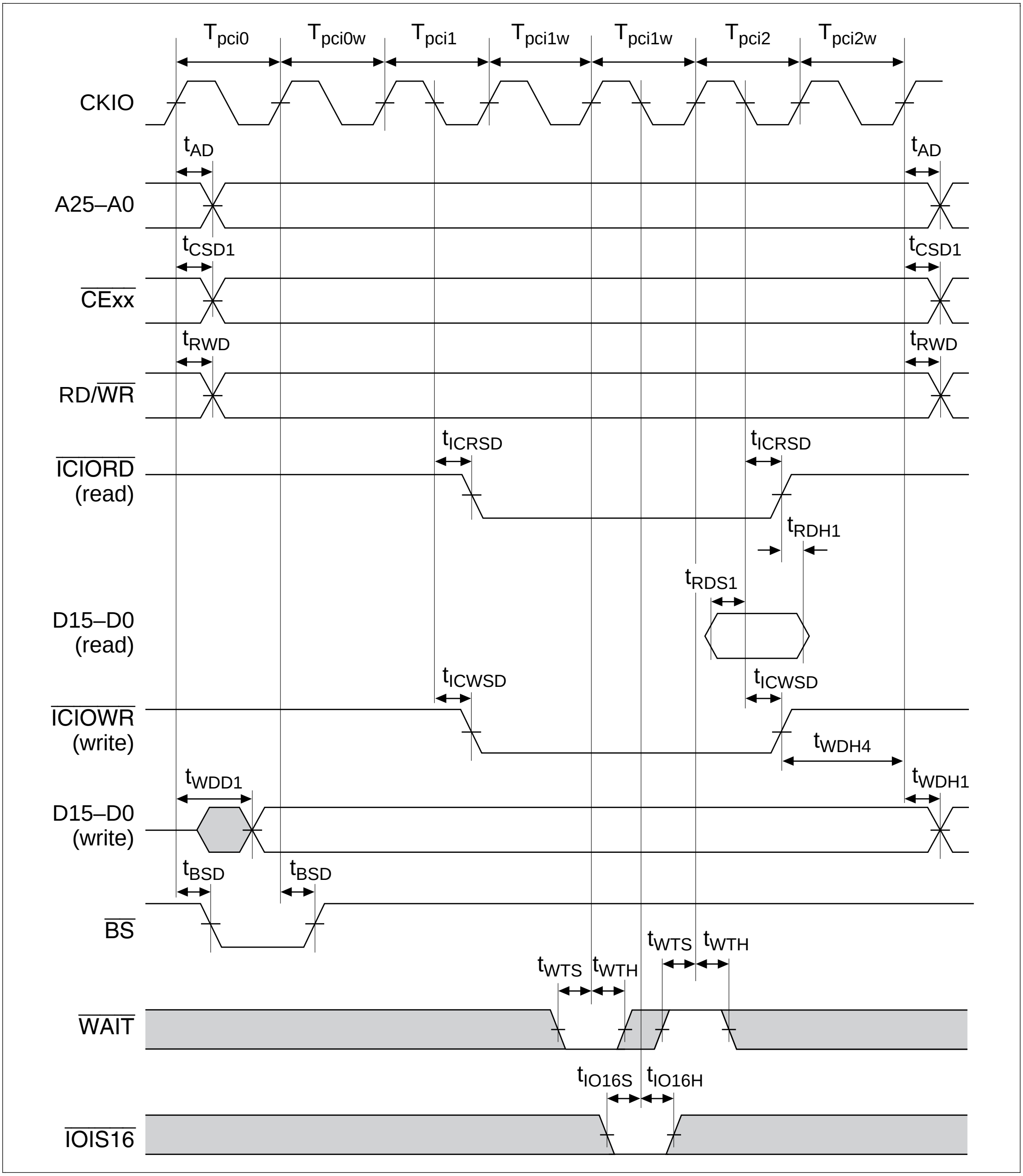


Figure 16.57 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, 1 Wait, External Wait)

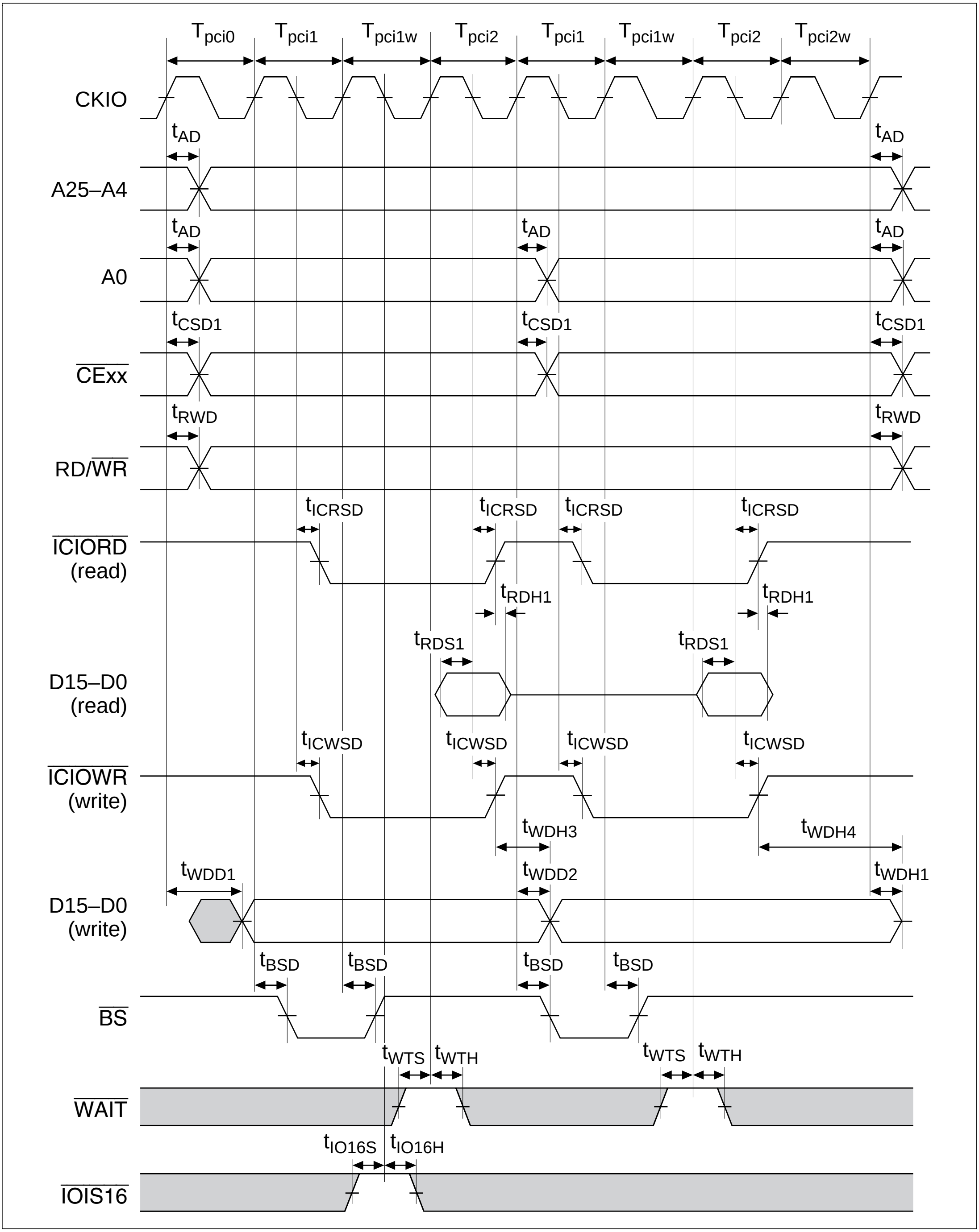


Figure 16.58 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, 1 Wait, Bus Sizing)

16.3.10 Peripheral Module Signal Timing

Table 16.11 Peripheral Module Signal Timing (Conditions: V_{CC} = 3.3 ± 0.3 V, Ta = 0 to 75°C)

				–15		–30		–60			
Module	**Item**		**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Figure**
TMU, RTC	Timer input setup time		t_{CLKS1}	20	—	15	—	12	—	ns	16.54, 16.55
	Timer clock input setup time		t_{CKS}	20	—	15	—	12	—	ns	
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	1.5	—	tcyc	
		Both edges	t_{TCKWL}	2.5	—	2.5	—	2.5	—	tcyc	
	Oscillation settling time		t_{ROSC}	—	3	—	3	—	3	S	16.61
SCI	Input clock cycle	Asyn-chronous	t_{Scyc}	4	—	4	—	4	—	tcyc	16.62–16.64
		syn-chronous		6	—	6	—	6	—	tcyc	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	—	1.5	tcyc	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5	—	1.5	tcyc	
	Input clock pulse width		t_{SCKw}	0.4	0.6	0.4	0.6	0.4	0.6	tscyc	
	Transmit data delay time		t_{TXD}	—	100	—	100	—	100	ns	
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—	100	—	ns	
	Receive data hold time (syn-chronous)	Clock input	t_{RXH}	100	—	100	—	100	—	ns	
		Clock output	t_{RXH}	0	—	0	—	0	—	ns	
Port	Output data delay time		t_{PORTD}	—	20	—	17	—	15	ns	
	Input data setup time		t_{PORTS}	20	—	15	—	12	—	ns	
	Input data hold time		t_{PORTH}	10	—	8	—	5	—	ns	

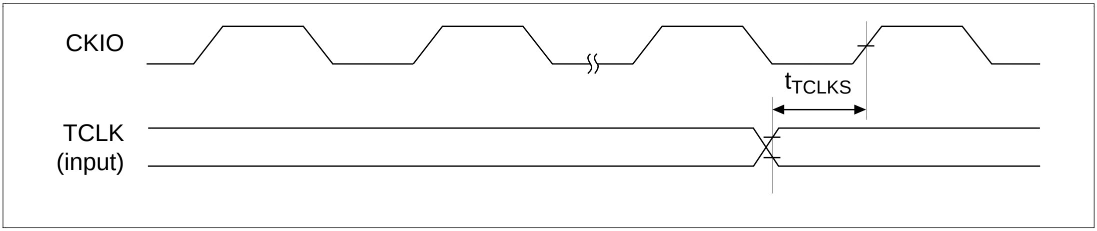


Figure 16.59 TCLK Input Timing (1)

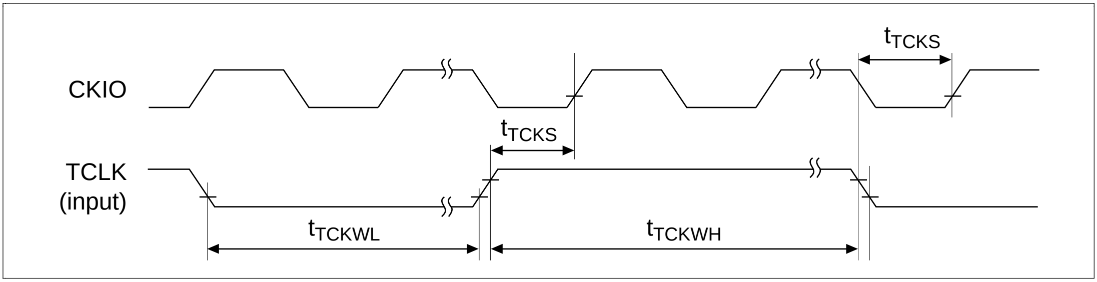


Figure 16.60 TCLK Input Timing (2)

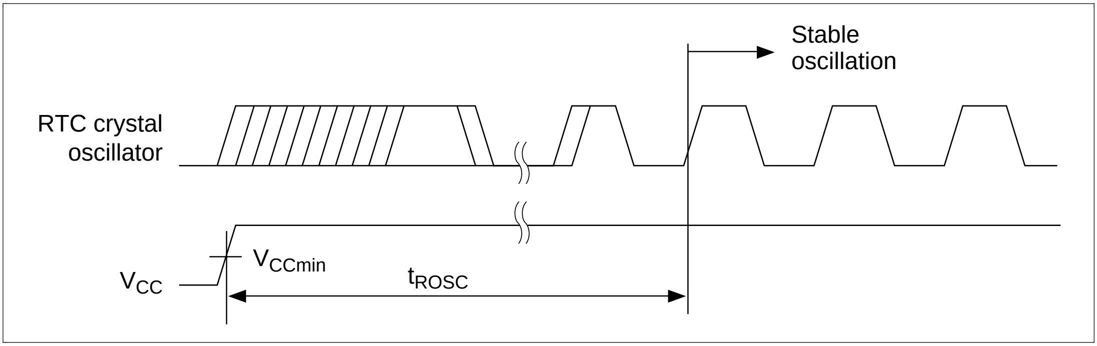


Figure 16.61 RTC Crystal Oscillator Power-On Oscillation Settling Time

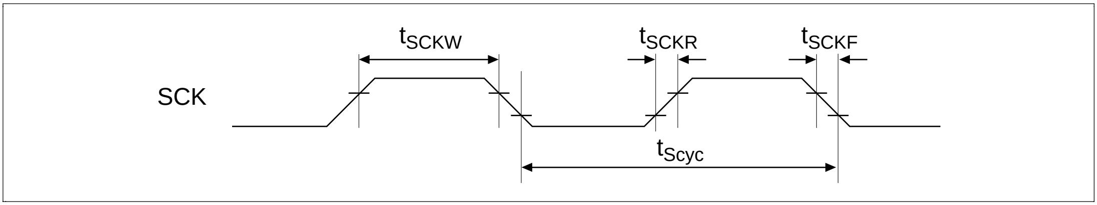


Figure 16.62 SCK Input Clock Timing

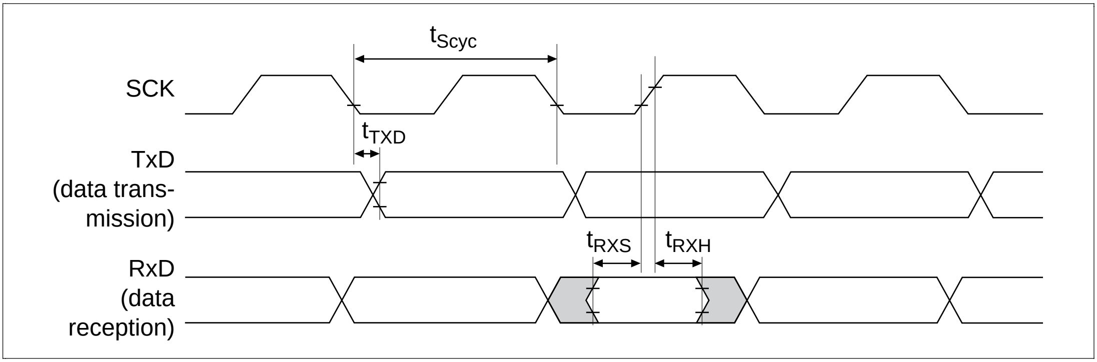


Figure 16.63 Synchronous Mode SCI Input/Output Timing

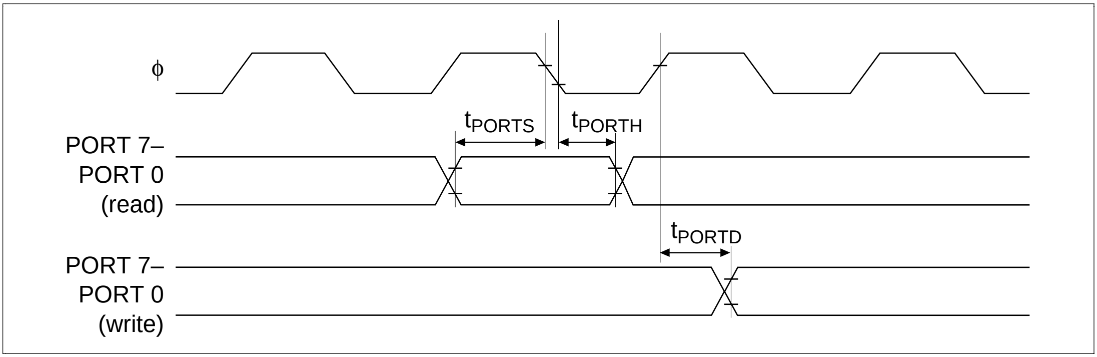
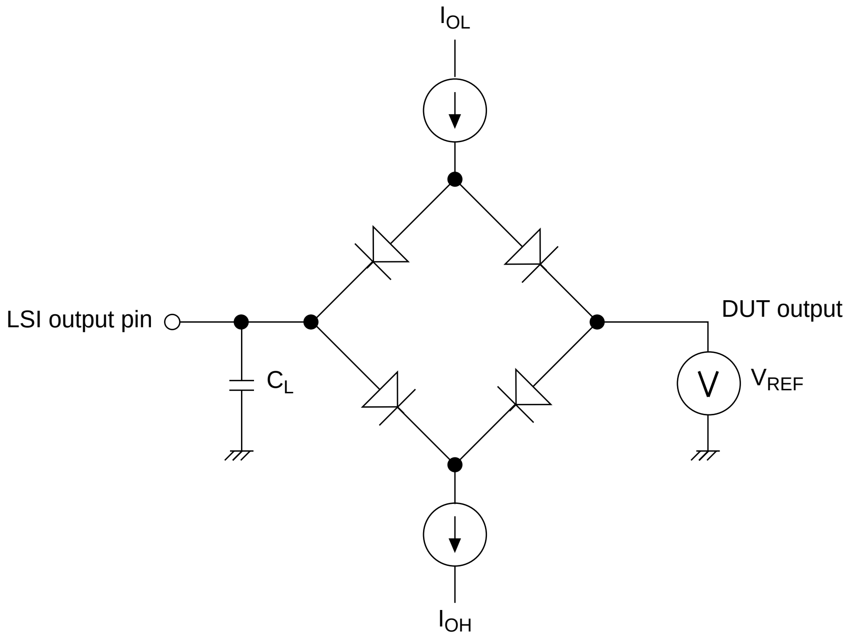


Figure 16.64 I/O Port Input/Output Timing

16.3.11 AC Characteristics Test Conditions

- Input/output signal reference level: 1.5 V (V_{CC} = 3.3 ± 0.3 V).
- Input pulse level: V_{SS} to 3.0 V (when $\overline{\text{RESET}}$, $\overline{\text{BREQ}}$, NMI, $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$, CKIO, MD5–MD0 are V_{SS} to V_{CC}).
- Input rise/fall time: 1 ns



Notes: 1. C_L is the total capacitance including the test jig probe, as shown below:

2. 30 pF: CKIO, $\overline{\text{RAS}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$–$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
I_{OL} and I_{OH} are the values shown in table 17.3

Figure 16.65 Output Load Circuit

Section 17 Electrical Characteristics (-SH7708R-)

17.1 Absolute Maximum Ratings

Table 17.1 Absolute Maximum Ratings

Item	Symbol	Ratings	Units
Power supply voltage	V_{CC}	–0.3 to 4.6	V
Input voltage	V_{in}	–0.3 to V_{CC} + 0.3	V
Operating temperature	T_{opr}	–20 to 75	°C
Storage temperature	T_{str}	–55 to 125	°C

Note: Operating the SH7708 Series above maximum ratings can damage or destroy it.

17.2 DC Characteristics

Table 17.2 DC Characteristics (Ta = –20 to 75°C)

Item		Symbol	Min	Typ	Max	Unit	Remarks
Power supply voltage		V_{CC}	3.15	3.3	3.6	V	In normal operation, sleep mode, and standby mode
Current	Normal operation	I_{CC}	—	120*1	200 *1	mA	V_{CC} = 3.3 V
							*1 Iø = 100 MHz
							Bø = 50 MHz
	In sleep mode		—	75*2	100 *2		*2 Bø = 60 MHz
							Pø = 30 MHz
	In standby mode		—	0.1*3	1*3	mA	*3 V_{CC} = 3.3 V/ Ta= 30 °C
Input voltage	RESET, NMI	VIH	V_{CC} × 0.9	—	V_{CC} + 0.3	V	
	BREQ, IRL3–IRL0, MD5–MD0		V_{CC} – 0.5	—	V_{CC} + 0.3		Standby mode
			V_{CC} – 0.7	—	V_{CC} + 0.3		Normal operation
	EXTAL, CKIO		V_{CC} – 0.7	—	V_{CC} + 0.3		
	Other input pins		2.0	—	V_{CC} + 0.3		
	RESET, NMI	VIL	–0.3	—	V_{CC} × 0.1		
	BREQ, IRL3–IRL0, MD5–MD0		–0.3	—	0.5		Standby mode
			–0.3	—	V_{CC} × 0.2		Normal operation
	Other input pins		–0.3	—	V_{CC} × 0.2		
Input leak current	All input pins	|lin|	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Three-state leak current	I/O, output, all pins (off condition)	|lsti|	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V

Table 17.2 DC Characteristics (Ta = –20 to 75°C) (cont)

Item		Symbol	Min	Typ	Max	Unit	Remarks
Output voltage	All output pins	VOH	2.4	—	—	V	V_{CC} = 3.15 V, IOH = –200 µA
			2.0	—	—		V_{CC} = 3.15 V, IOH = –2 mA
		VOL	—	—	0.55		V_{CC} = 3.6 V, IOL = 1.6 mA
Pull-up resistance	Port pins	Rpull	30	60	120	kΩ	
Terminal capacitance	All pins	C	—	—	20	pF	

Notes: 1. Regardless of whether PLL or RTC is used, connect V_{CC}(PLL), V_{CC}(RTC) to V_{CC}, and V_{SS}(PLL), V_{SS}(RTC) to V_{SS}.

2. Current condition is VIH min = V_{CC} – 0.5 V, VIL max = 0.5 V, and all output pins unloaded.

Table 17.3 Permissible Output Current Values (V_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C)

Item	Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	IOL	—	—	2.0	mA
Permissible output low current (total)	ΣIOL	—	—	120	
Permissible output high current (per pin)	–IOH	—	—	2.0	
Permissible output high current (total)	Σ (–IOH)	—	—	40	

Note: To ensure reliability, output current must not exceed the maximum values listed.

17.3 AC Characteristics

Input for the LSI should, as a rule, be clock synchronous. Keep to the setup and hold times for each input signal unless otherwise directed.

Table 17.4 LSI Clock Values (Ta = –20 to 75°C)

Item		Symbol		Unit
Operating frequency	CPU, cache, TLB	f	100	MHz
	External bus		60	
	Peripheral modules		33.3	

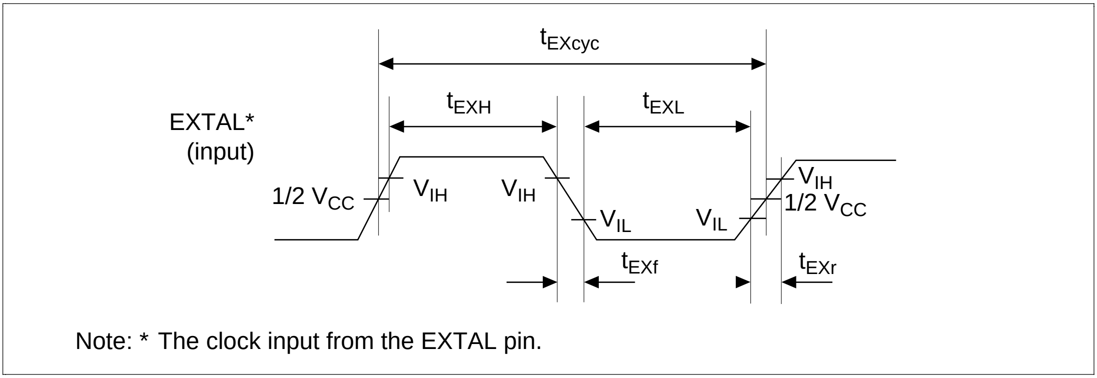
17.3.1 Clock Timing

Table 17.5 Clock Timing (V_{CC} = 3.15Å–3.6 V, Ta = –20 to 75°C, Maximum External Bus Operating Frequency: 60 MHz)

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	5	60	MHz	17.1
EXTAL clock input cycle time	t_{EXcyc}	16.7	200	ns	
EXTAL clock input low-level pulse width	t_{EXL}	4*[1] or 10*[2]	—	ns	
EXTAL clock input high-level pulse width	t_{EXH}	4*[1] or 10*[2]	—	ns	
EXTAL clock input rise time	t_{EXr}	—	2	ns	
EXTAL clock input fall time	t_{EXf}	—	2	ns	
CKIO clock frequency (input)	f_{CKI}	16	60	MHz	17.2
CKIO clock cycle time (input)	t_{CKIcyc}	16.7	62.5	ns	
CKIO clock low-level pulse width (input)	t_{CKIL}	4	—	ns	
CKIO clock high-level pulse width (input)	t_{CKIH}	4	—	ns	
CKIO clock rise time (input)	t_{CKIr}	—	2	ns	
CKIO clock fall time (input)	t_{CKIf}	—	2	ns	
CKIO clock output frequency (output)	f_{OP}	16	60	MHz	17.3
CKIO clock cycle time (output)	t_{cyc}	16.7	62.5	ns	
CKIO clock low-level pulse width (output)	t_{CKOL}	3	—	ns	
CKIO clock high-level pulse width (output)	t_{CKOH}	3	—	ns	
CKIO clock rise time (output)	t_{CKOr}	—	5	ns	
CKIO clock fall time (output)	t_{CKOf}	—	5	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	17.4
Power-on oscillation settling time/mode setting	t_{OSCMD}	10	—	ms	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	0	—	ns	
$\overline{\text{RESET}}$ set-up time	t_{RESS}	20	—	ns	
$\overline{\text{BREQ}}$ set-up time	t_{BREQS}	20	—	ns	
MD reset hold time	t_{MDRH}	20	—	ns	
Reset assert time	t_{RESW}	20	—	tcyc	17.4, 17.5, 17.11
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	17.5
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	17.6
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	17.7
PLL synchronization settling time	t_{PLL}	100	—	µs	17.8, 17.9, 17.10
IRL interrupt decision time (using RTC and in standby mode)	t_{IRLSTB}	100	—	µs	17.10

Notes: 1. PLL circuit 2 in operation.

2. IPLL circuit 2 not in operation.



Note: * The clock input from the EXTAL pin.

Figure 17.1 EXTAL Clock Input Timing

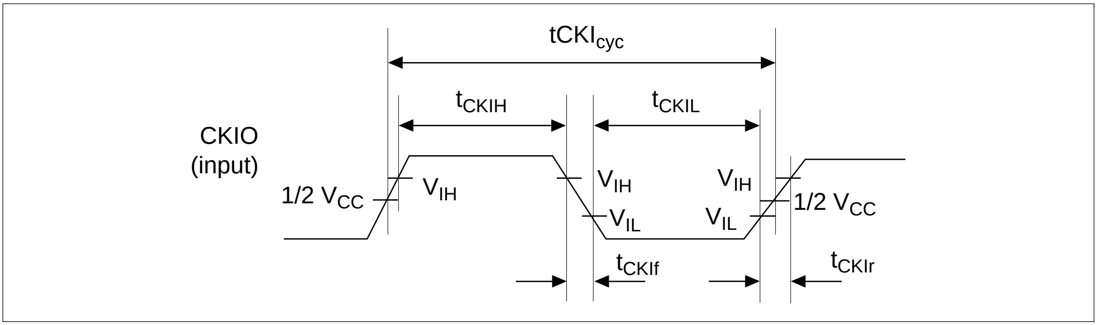


Figure 17.2 CKIO Clock Input Timing

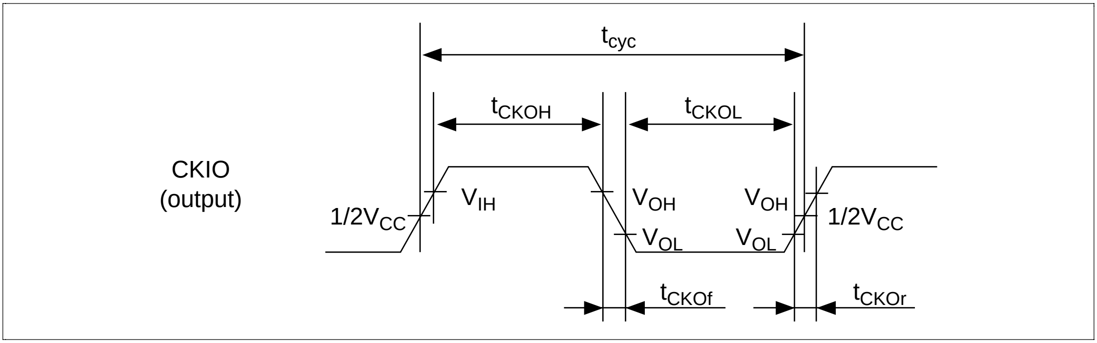
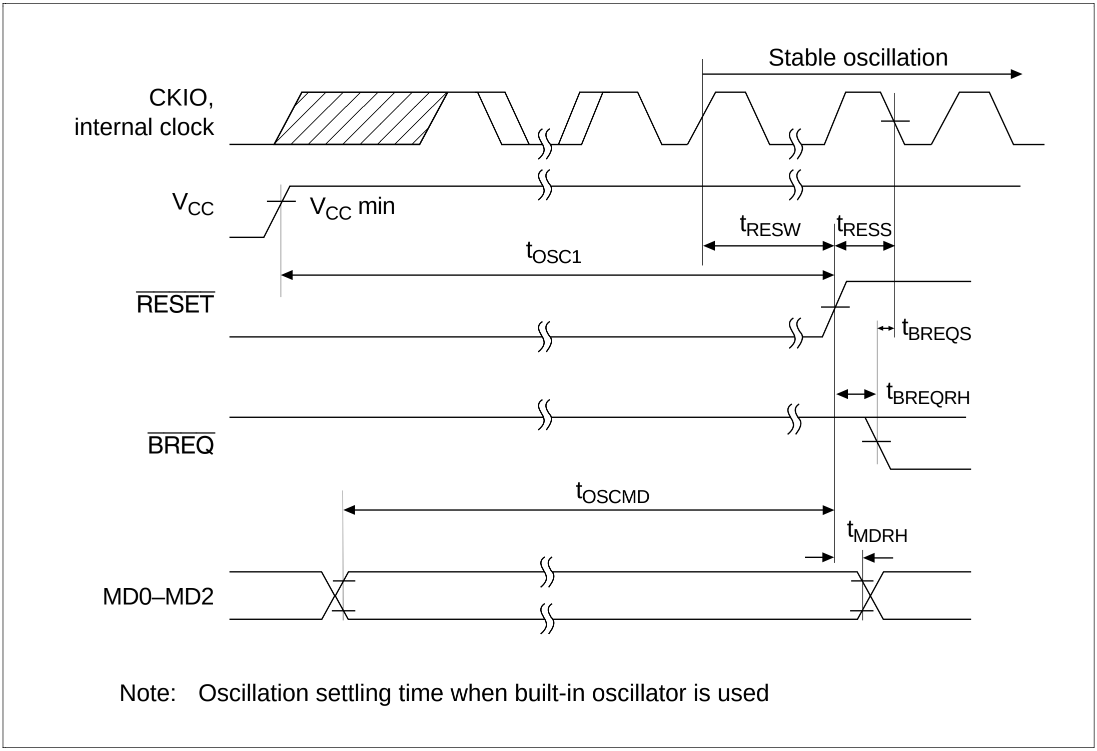
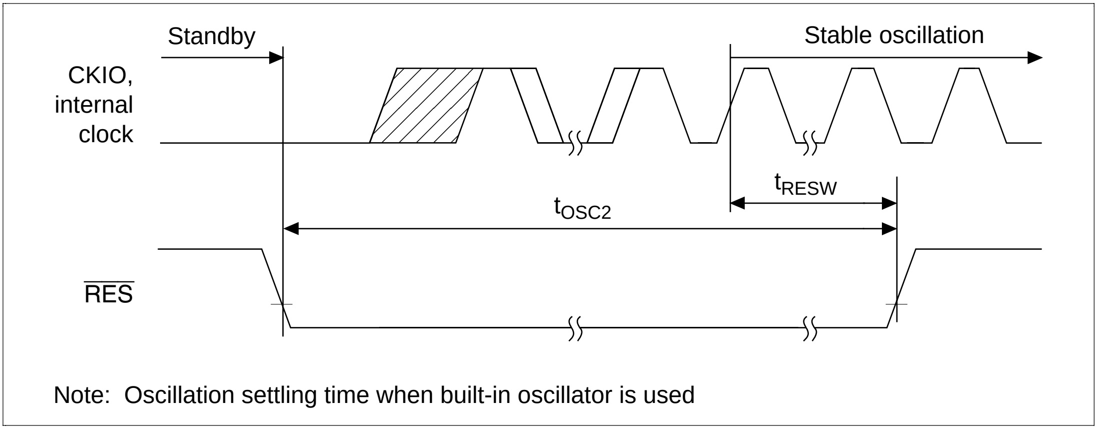


Figure 17.3 CKIO Clock Output Timing



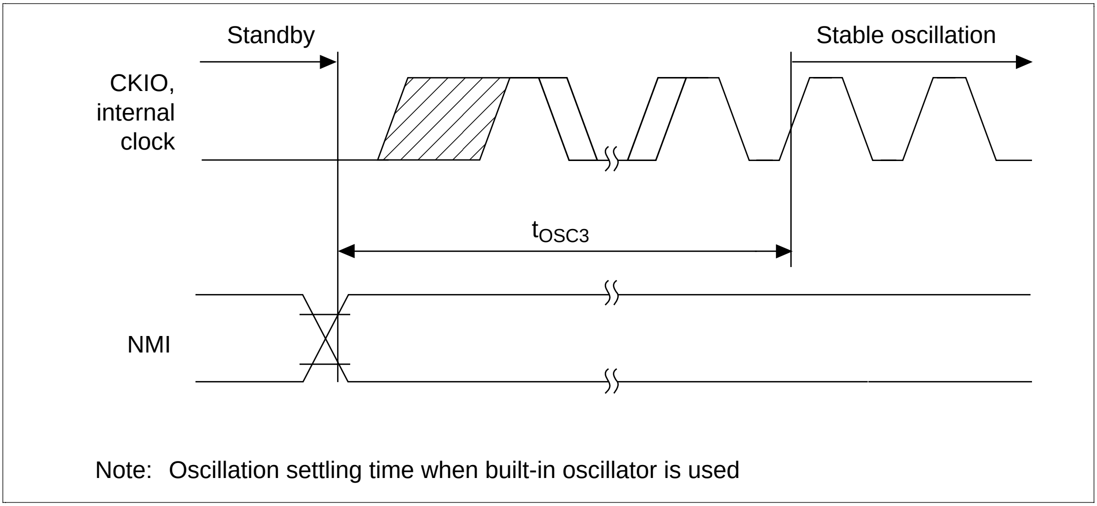
Note: Oscillation settling time when built-in oscillator is used

Figure 17.4 Power-On Oscillation Settling Time



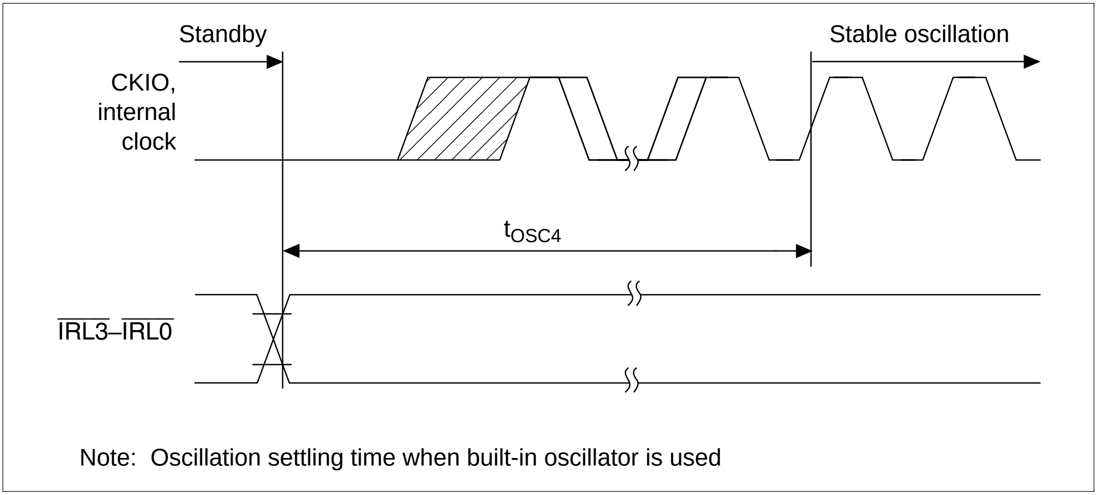
Note: Oscillation settling time when built-in oscillator is used

Figure 17.5 Standby Return Oscillation Settling Time (Return by RESET)



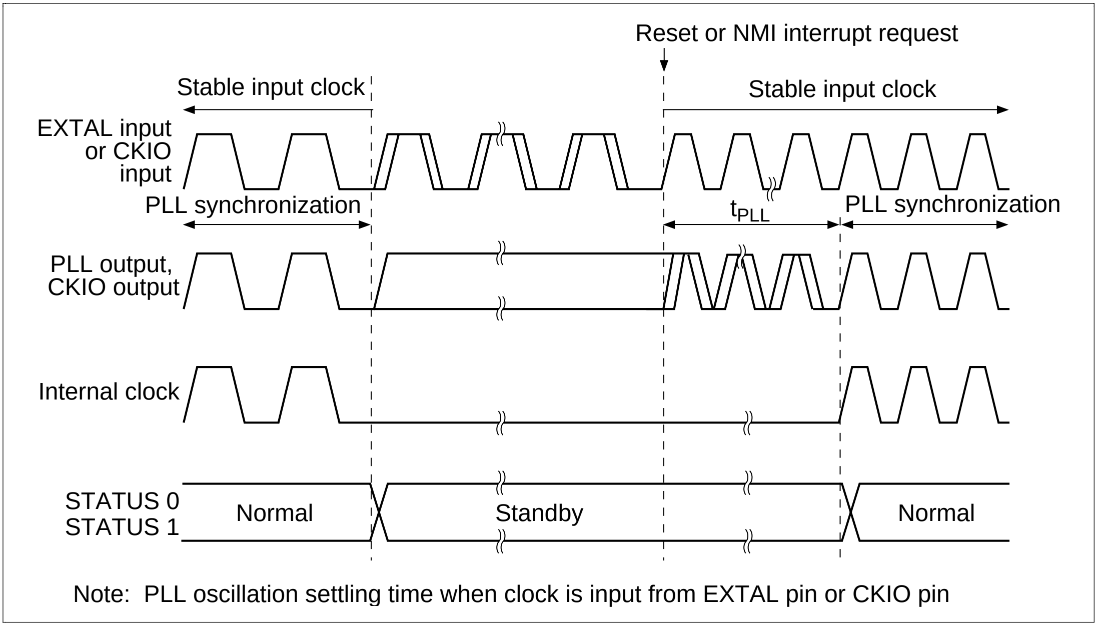
Note: Oscillation settling time when built-in oscillator is used

Figure 17.6 Standby Return Oscillation Settling Time (Return by NMI)



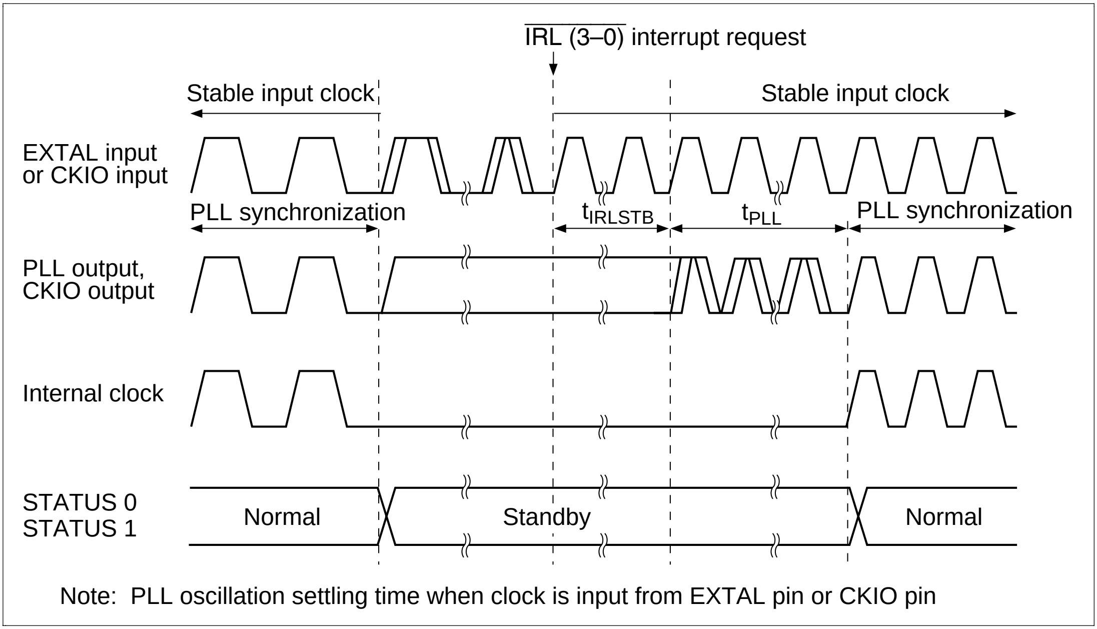
Note: Oscillation settling time when built-in oscillator is used

Figure 17.7 Standby Return Oscillation Settling Time (Return by $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$)



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 17.8 PLL Synchronization Settling Time in Case of Reset or NMI Interrupt



Note: PLL oscillation settling time when clock is input from EXTAL pin or CKIO pin

Figure 17.9 PLL Synchronization Settling Time in Case of IRL Interrupt

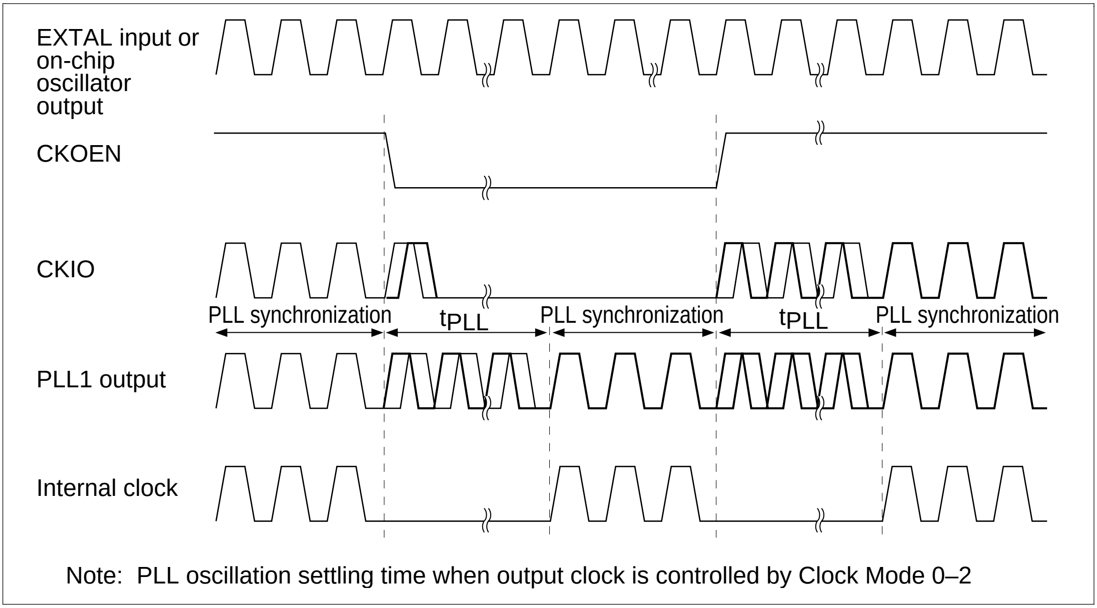

EXTAL input or on-chip oscillator output
CKOEN
CKIO
PLL synchronization
t_{PLL}
PLL synchronization
t_{PLL}
PLL synchronization
PLL1 output
Internal clock
Note: PLL oscillation settling time when output clock is controlled by Clock Mode 0–2

Figure 17.10 PLL Synchronization Settling Time in Case of CKOEN Bit Manipulation

17.3.2 Control Signal Timing

Table 17.6 Control Signal Timing (V_{CC} = 3.15–3.6 V, Ta = –20 to + 75°C)

		–60*2			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RESET}}$ pulse width	t_{RESW}	20	—	tcyc	17.11, 17.13, 17.15
$\overline{\text{RESET}}$ setup time	t_{RESS}	23	—	ns	
$\overline{\text{RESET}}$ hold time	t_{RESH}	2	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	12	—	ns	
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—	ns	
$\overline{\text{BREQ}}$ reset setup time	t_{BREQRS}	17	—	ns	
$\overline{\text{BREQ}}$ reset hold time	t_{BREQRH}	16	—	ns	
MD reset setup time	t_{MDRS}	20	—	tcyc	17.12
MD reset hold time	t_{MDRH}	16	—	ns	
NMI setup time*1	t_{NMIS}	15	—	ns	17.13
$\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ setup time*1	t_{IRLS}	10	—	ns	
NMI hold time	t_{NMIH}	4	—	ns	
$\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$ hold time	t_{IRLH}	4	—	ns	
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	12	ns	17.14
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	12	ns	17.15, 17.16
STATUS1, STATUS0 delay time	t_{STD}	—	16	ns	
Bus tri-state delay time 1	t_{BOFF1}	0	16	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	16	ns	
Bus buffer on time 1	t_{BON1}	0	16	ns	
Bus buffer on time 2	t_{BON2}	0	16	ns	

Notes: 1. $\overline{\text{RESET}}$, NMI, and $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ 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 fall.

2. Upper limit of external bus clock is 60 MHz.

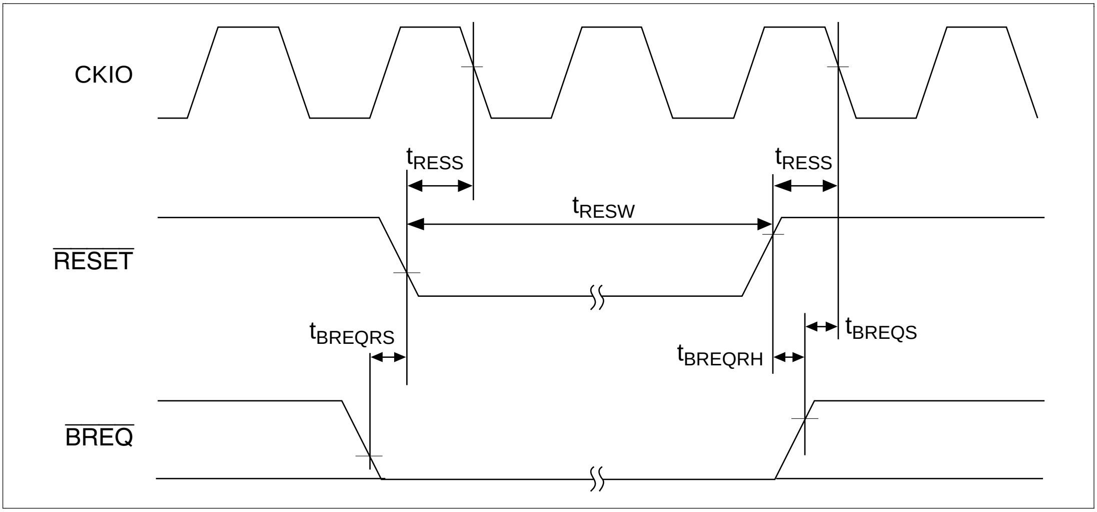


Figure 17.11 Manual Reset Input Timing

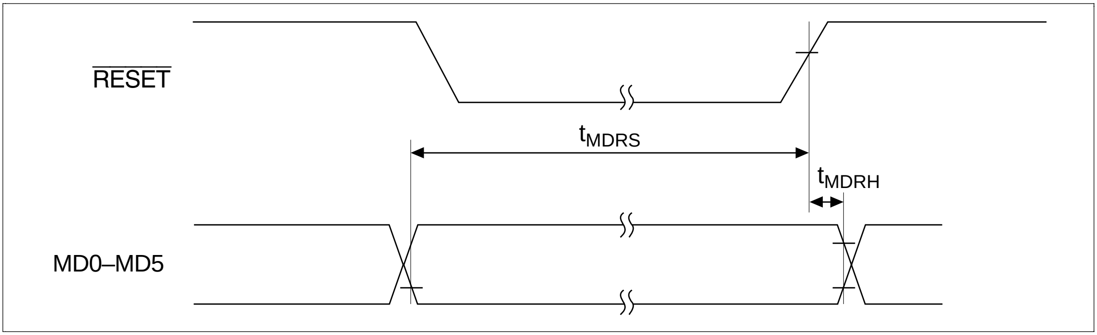


Figure 17.12 Mode Input Timing

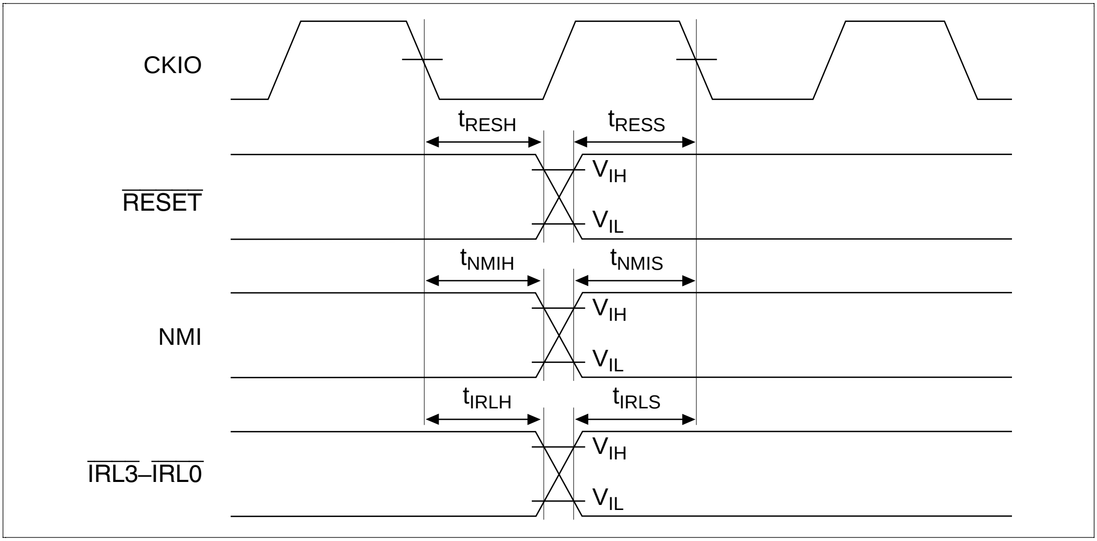


Figure 17.13 Interrupt Signal Input Timing

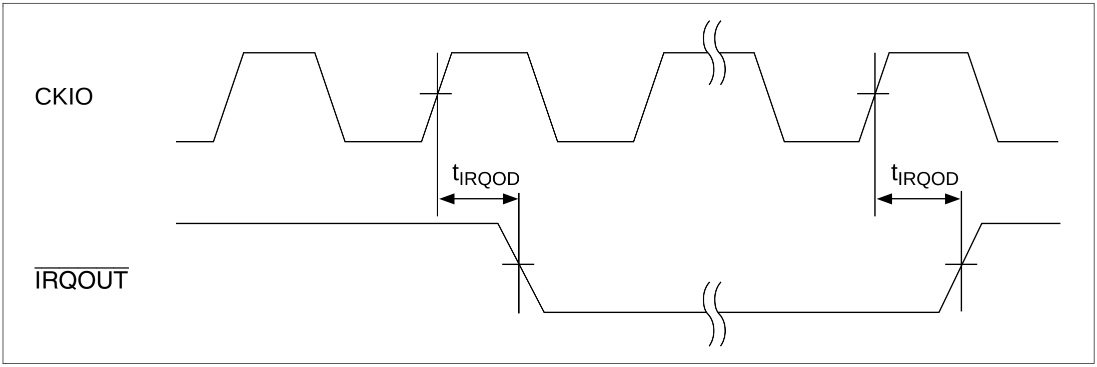


Figure 17.14 $\overline{\text{IRQOUT}}$ Timing

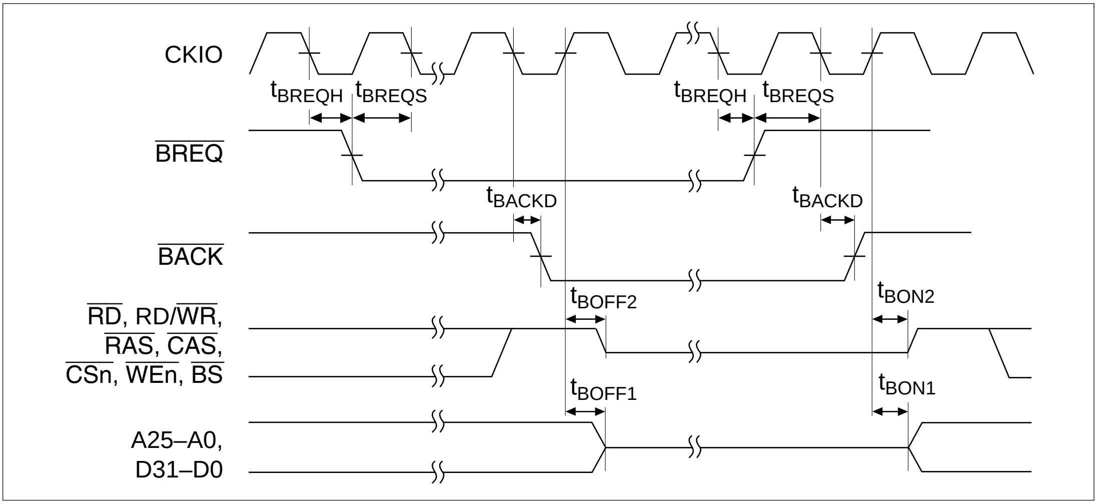


Figure 17.15 Bus Release Timing

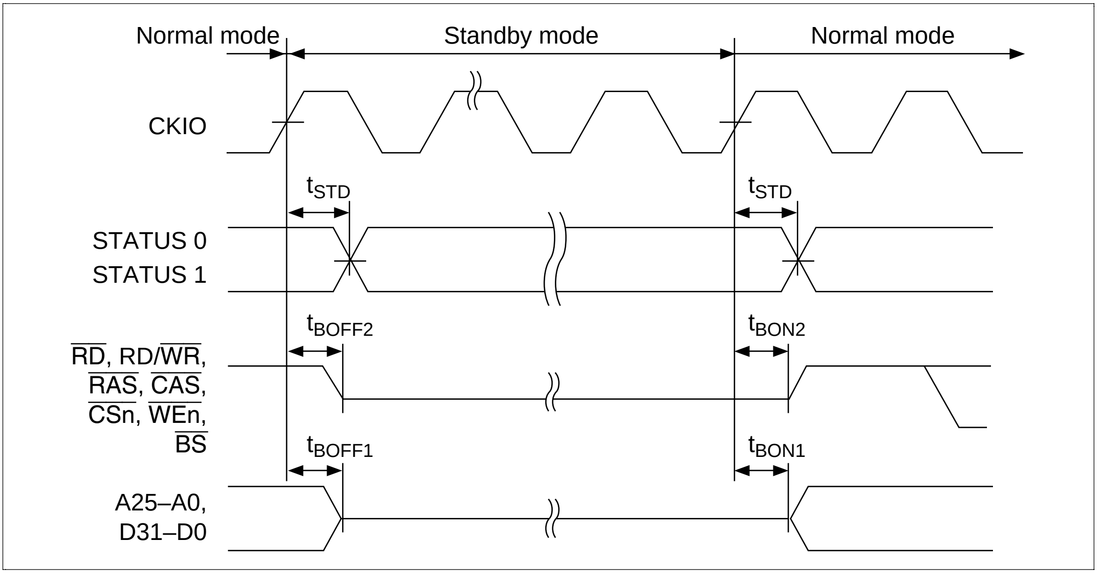


Figure 17.16 Pin Drive Timing for Standby Mode

17.3.3 AC Bus Timing Specifications

Table 17.7 Bus Timing (Conditions: Clock Mode 0/1/2/7, V_{CC} = 3.15–3.6 V, Ta = –20 to 75°C)

		–60*[1]			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
Address delay time	t_{AD}	1.5	13	ns	17.17–17.58
Address setup time	t_{AS}	0	—	ns	
Address hold time	t_{AH}	0	—	ns	
$\overline{\text{BS}}$ delay time	t_{BSD}	—	12	ns	
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	1.5	12	ns	
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—	12	ns	
Read write delay time	t_{RWD}	1.5	12	ns	
Read write setup time	t_{RWS}	0	—	ns	
Read write hold time	t_{RWH}	0	—	ns	
Read strobe delay time	t_{RSD}	—	12	ns	
Read data setup time 1	t_{RDS1}	12	—	ns	
Read data setup time 2	t_{RDS2}	8	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	3	—	ns	
Write enable delay time	t_{WED}	—	12	ns	
Write data delay time 1	t_{WDD1}	—	15	ns	
Write data delay time 2	t_{WDD2}	—	13	ns	
Write data setup time	t_{WDS}	0	—	ns	
Write data hold time 1	t_{WDH1}	0	—	ns	
Write data hold time 2	t_{WDH2}	1.5	—	ns	
Write data hold time 3	t_{WDH3}	0	—	ns	
Write data hold time 4	t_{WDH4}	0	—	ns	
$\overline{\text{WAIT}}$ setup time*[2]	t_{WTS}	12	—	ns	
$\overline{\text{WAIT}}$ hold time*[2]	t_{WTH}	4	—	ns	

Table 17.7 Bus Timing (Conditions: Clock Mode 0/1/2/7, V_{CC} = 3.15–3.6 V, Ta = –20 to 75°C) (cont)

		–60*[1]			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	—	13	ns	17.23–17.44
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	1.5	13	ns	
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	13	ns	
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	1.5	13	ns	
DQM delay time	t_{DQMD}	1.5	12	ns	
CKE delay time	t_{CKED}	—	12	ns	
$\overline{\text{CE}}$ delay time	t_{CED}	—	13	ns	17.45–17.51
$\overline{\text{OE}}$, $\overline{\text{RFSH}}$ delay time	t_{OED}	—	13	ns	
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	12	ns	17.56–17.58
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	12	ns	
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	12	—	ns	
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	4	—	ns	

Notes 1. Upper limit of external bus clock is 60 MHz.

2. $\overline{\text{WAIT}}$ is a synchronous signal. Operation cannot be guaranteed if the setup and hold times shown here are not observed.

Table 17.8 Bus Timing (Conditions: Clock Mode 3/4, V_{CC} = 3.15–3.6 V, Ta = –20 to 75°C)

Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	1.5	20	ns	17.17–17.58
Address setup time	t_{AS}	0	—	ns	
Address hold time	t_{AH}	20	—	ns	
$\overline{\text{BS}}$ delay time	t_{BSD}	—	19	ns	
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	1.5	19	ns	
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	—	20	ns	
Read write delay time	t_{RWD}	1.5	19	ns	
Read write setup time	t_{RWS}	0	—	ns	
Read write hold time	t_{RWH}	0	—	ns	
Read strobe delay time	t_{RSD}	—	20	ns	
Read data setup time 1	t_{RDS1}	12	—	ns	
Read data setup time 2	t_{RDS2}	12	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	8	—	ns	
Write enable delay time	t_{WED}	—	20	ns	
Write data delay time 1	t_{WDD1}	—	25	ns	
Write data delay time 2	t_{WDD2}	—	19	ns	
Write data setup time	t_{WDS}	0	—	ns	
Write data hold time 1	t_{WDH1}	0	—	ns	
Write data hold time 2	t_{WDH2}	1.5	—	ns	
Write data hold time 3	t_{WDH3}	0	—	ns	
Write data hold time 4	t_{WDH4}	0	—	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	12	—	ns	
$\overline{\text{WAIT}}$ hold time	t_{WTH}	8	—	ns	

Table 17.8 Bus Timing (Conditions: Clock Mode 3/4, V_{CC} = 3.15–3.6 ±V, Ta = –20 to 75°C) (cont)

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	—	20	ns	17.23–17.44
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	1.5	19	ns	
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	—	20	ns	
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	1.5	19	ns	
DQM delay time	t_{DQMD}	1.5	19	ns	
CKE delay time	t_{CKED}	—	19	ns	
$\overline{\text{CE}}$ delay time	t_{CED}	—	20	ns	17.45–17.51
$\overline{\text{OE}}$, $\overline{\text{RFSH}}$ delay time	t_{OED}	—	20	ns	
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	—	20	ns	17.56–17.58
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	—	20	ns	
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	12	—	ns	
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	8	—	ns	

17.3.4 Basic Timing

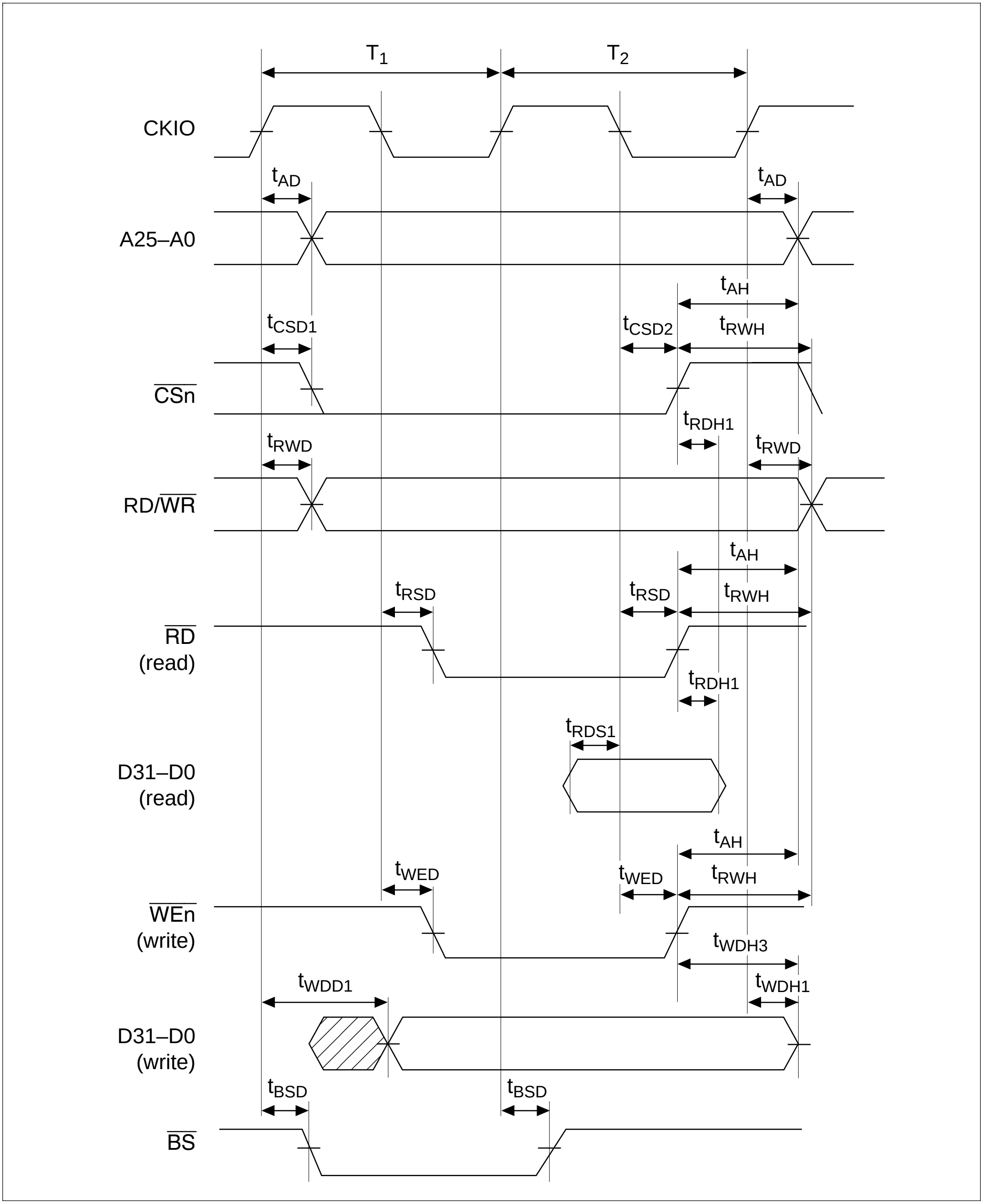


Figure 17.17 Basic Bus Cycle (No Wait)

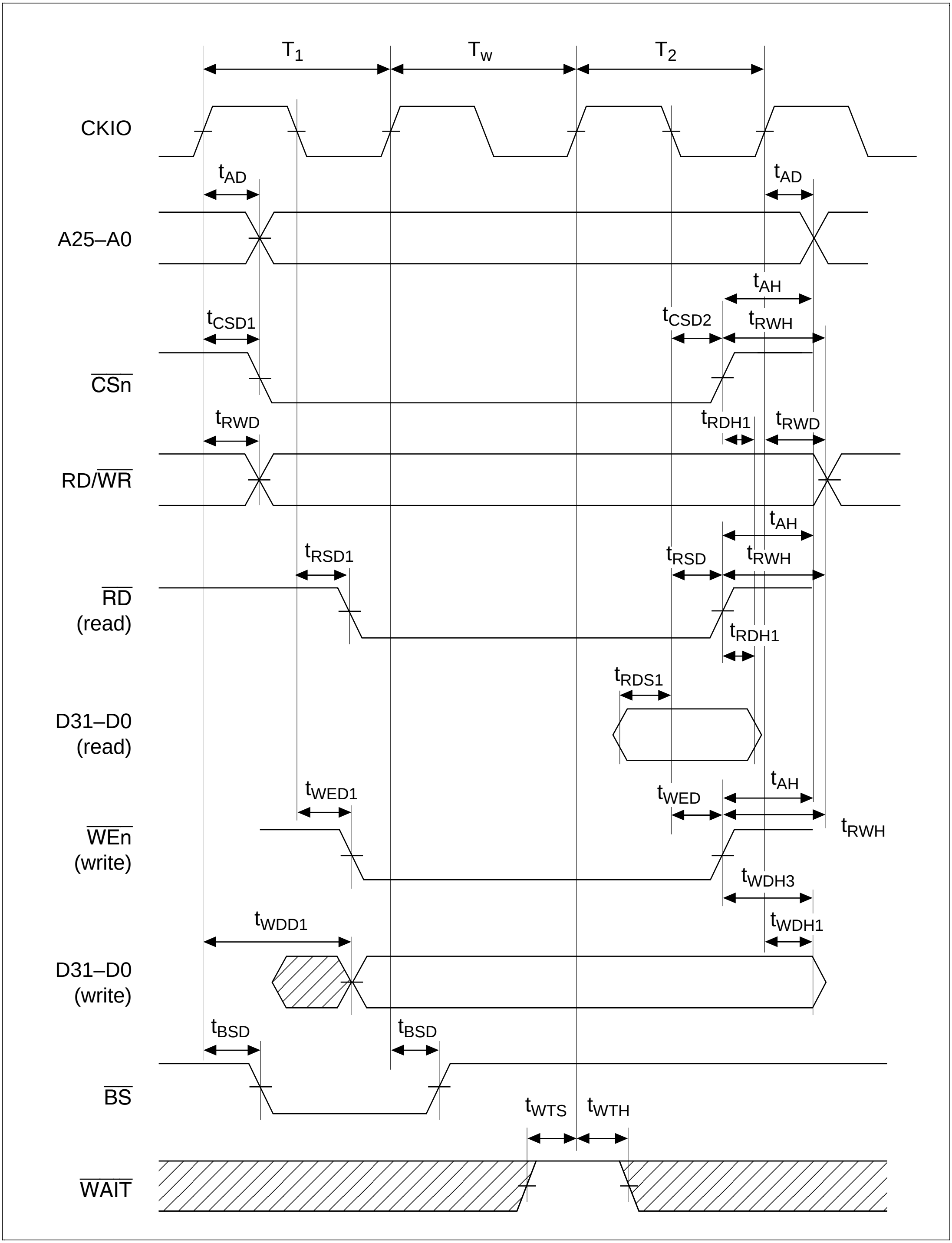


Figure 17.18 Basic Bus Cycle (1 Wait)

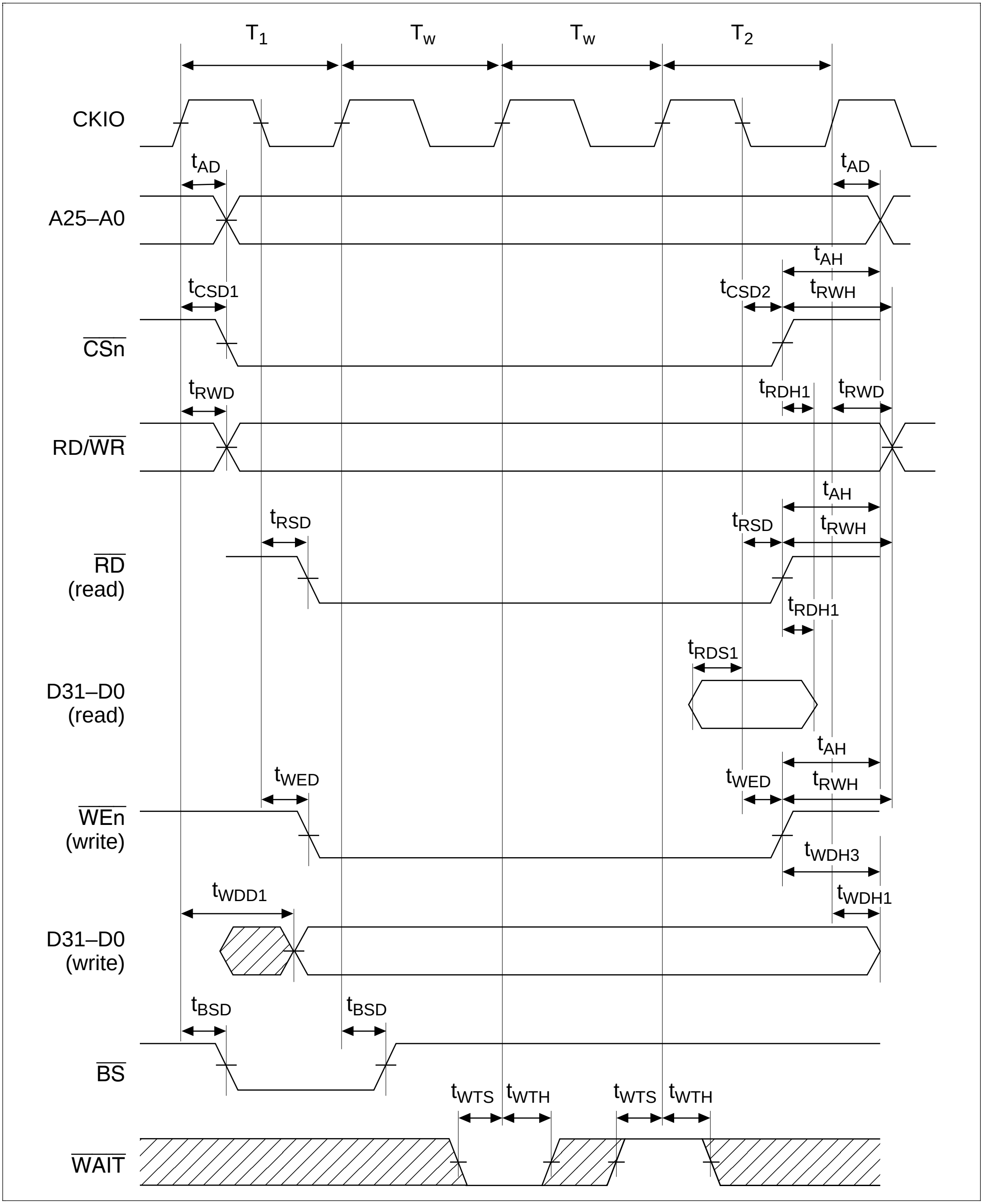
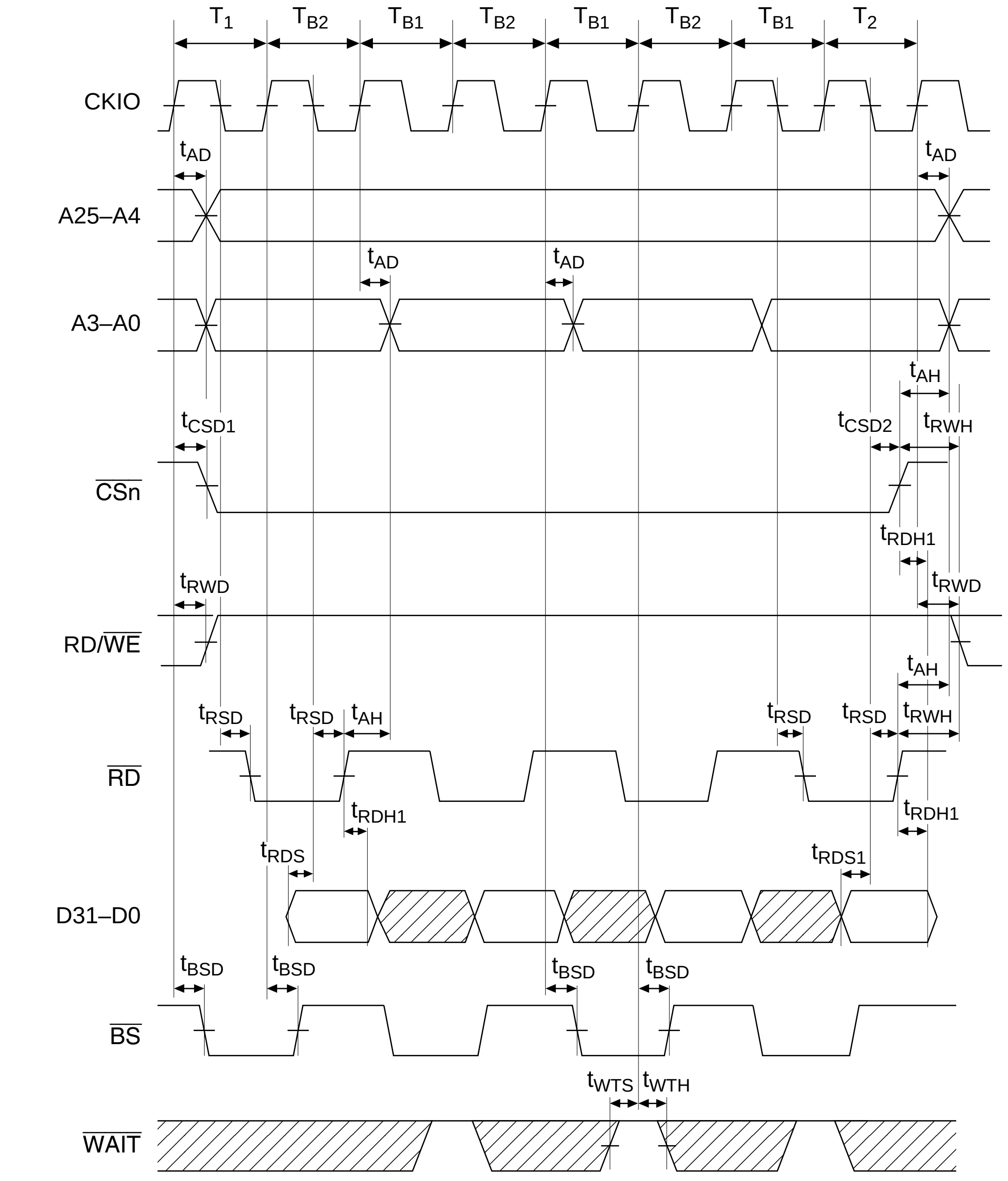


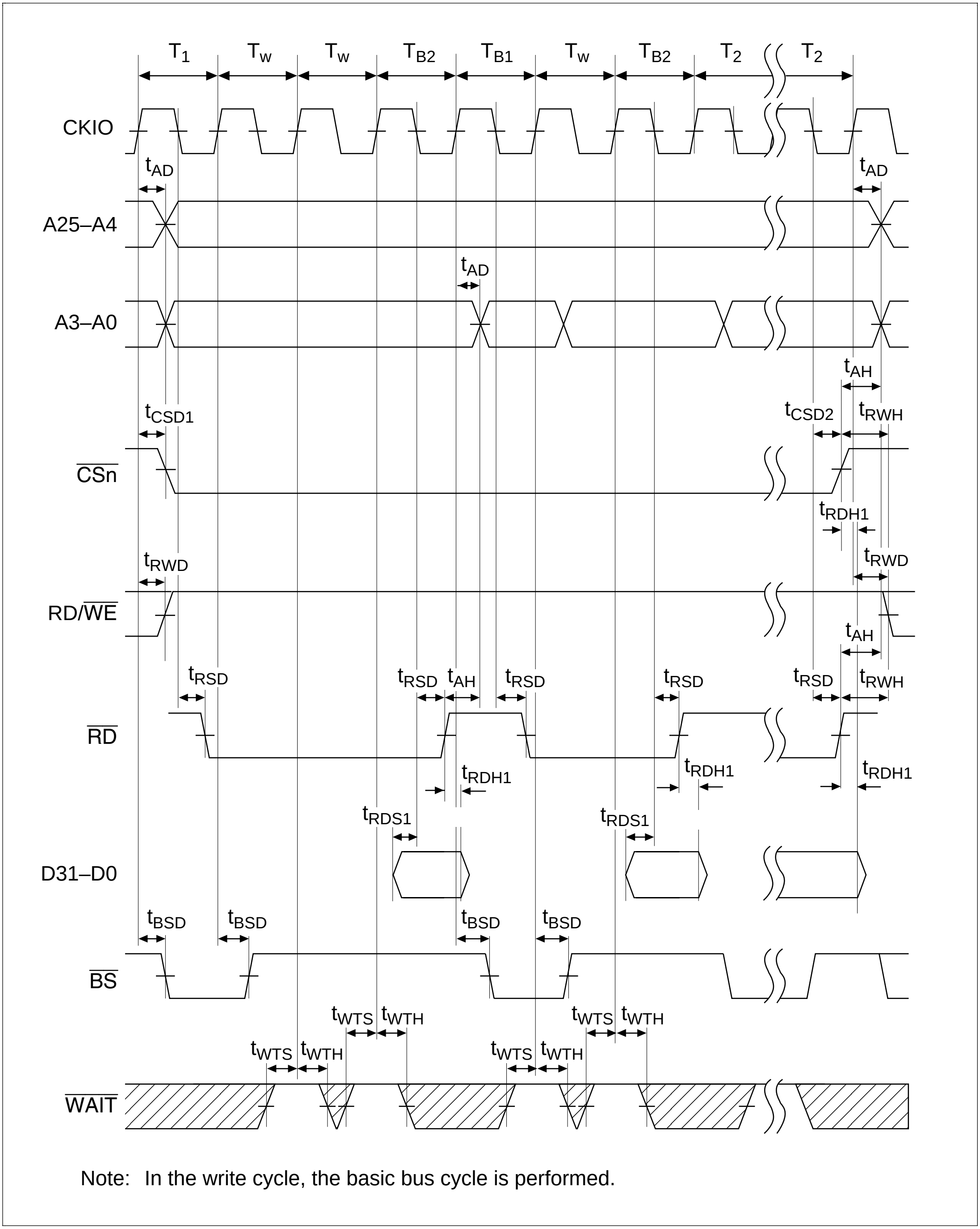
Figure 17.19 Basic Bus Cycle (External Wait)

17.3.5 Burst ROM Timing



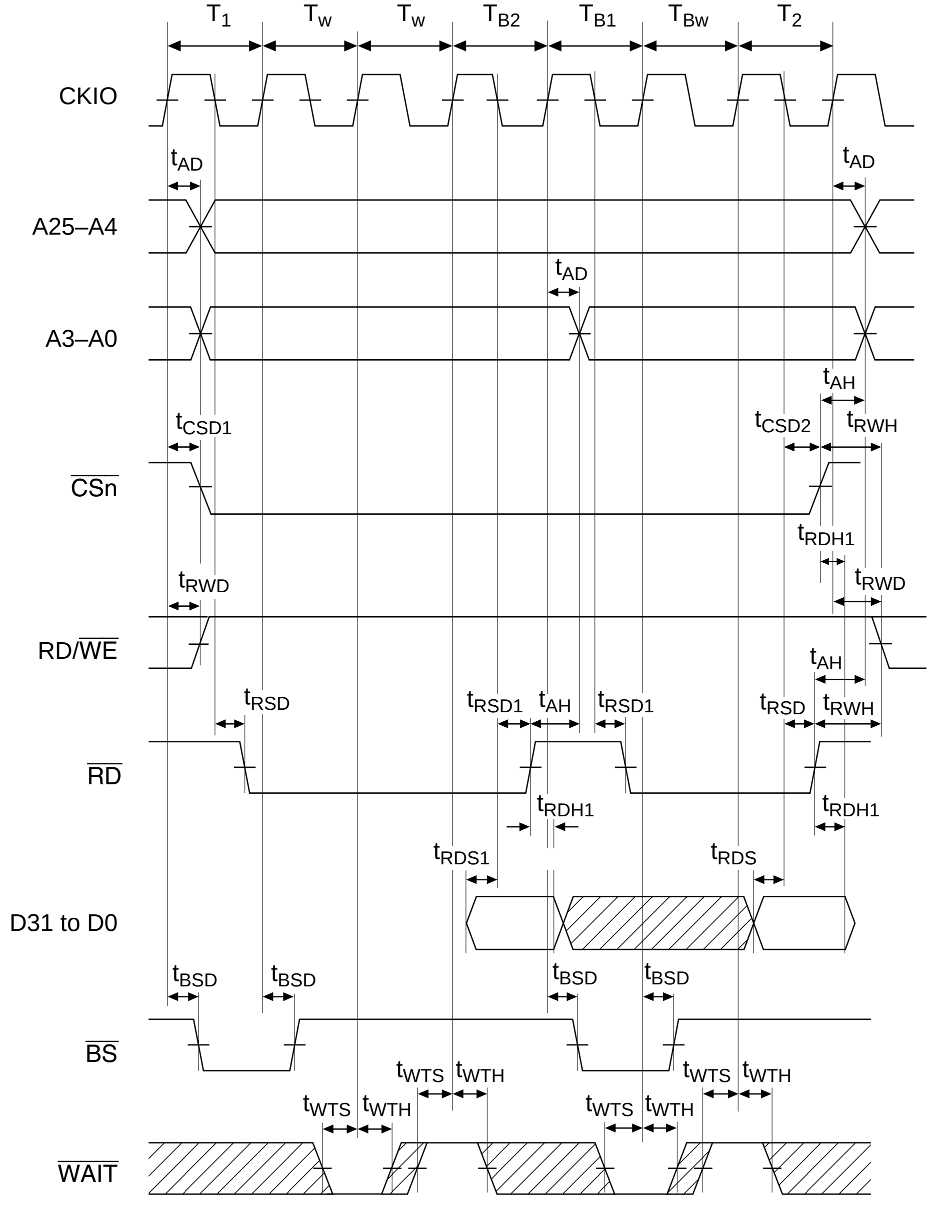
Note: In the write cycle, the basic bus cycle is performed.

Figure 17.20 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 17.21 Burst ROM Bus Cycle (2 Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 17.22 Burst ROM Bus Cycle (External Wait Input)

17.3.6 DRAM Timing

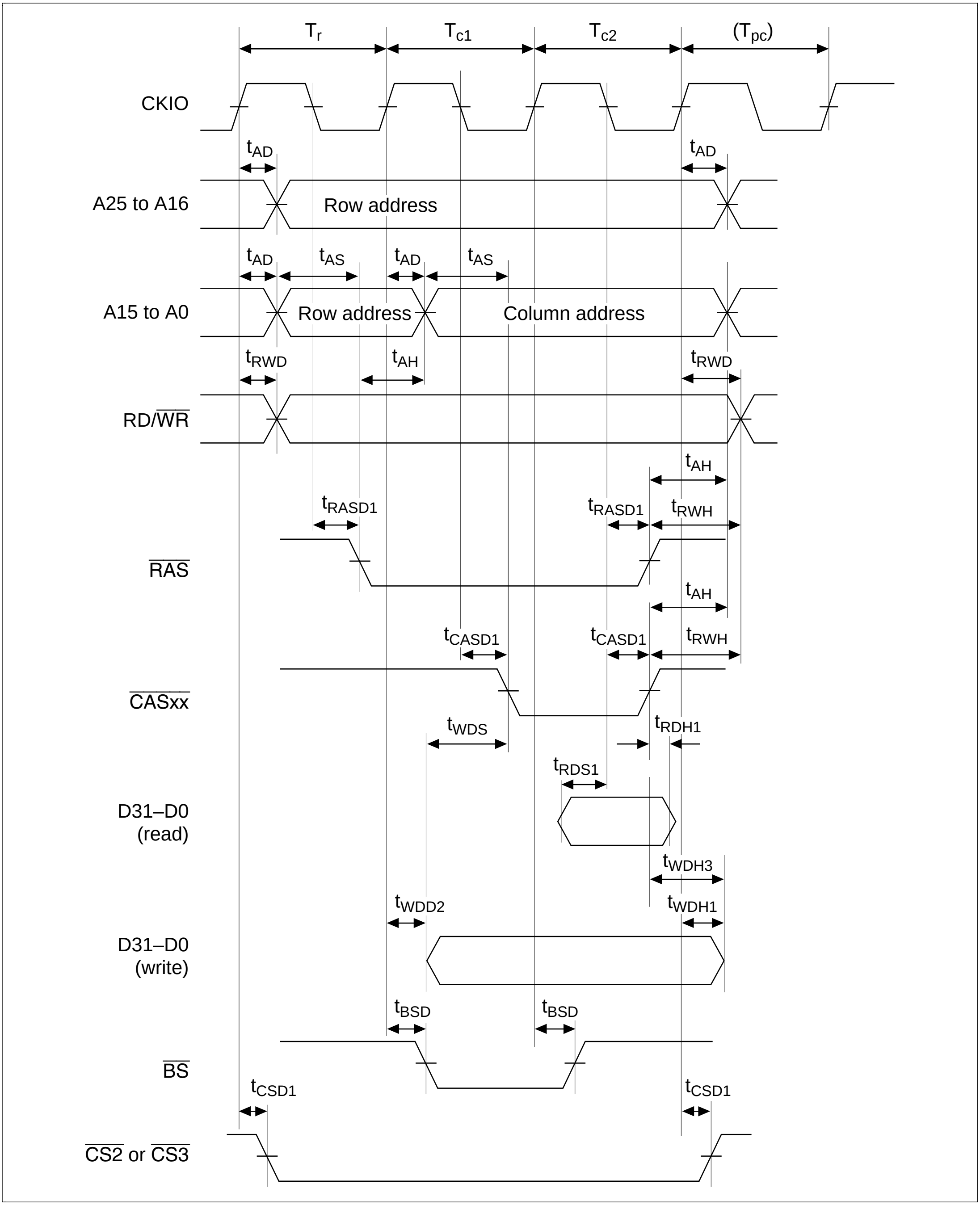


Figure 17.23 DRAM Bus Cycle (RCD = 0, AnW = 1, TPC = 0)

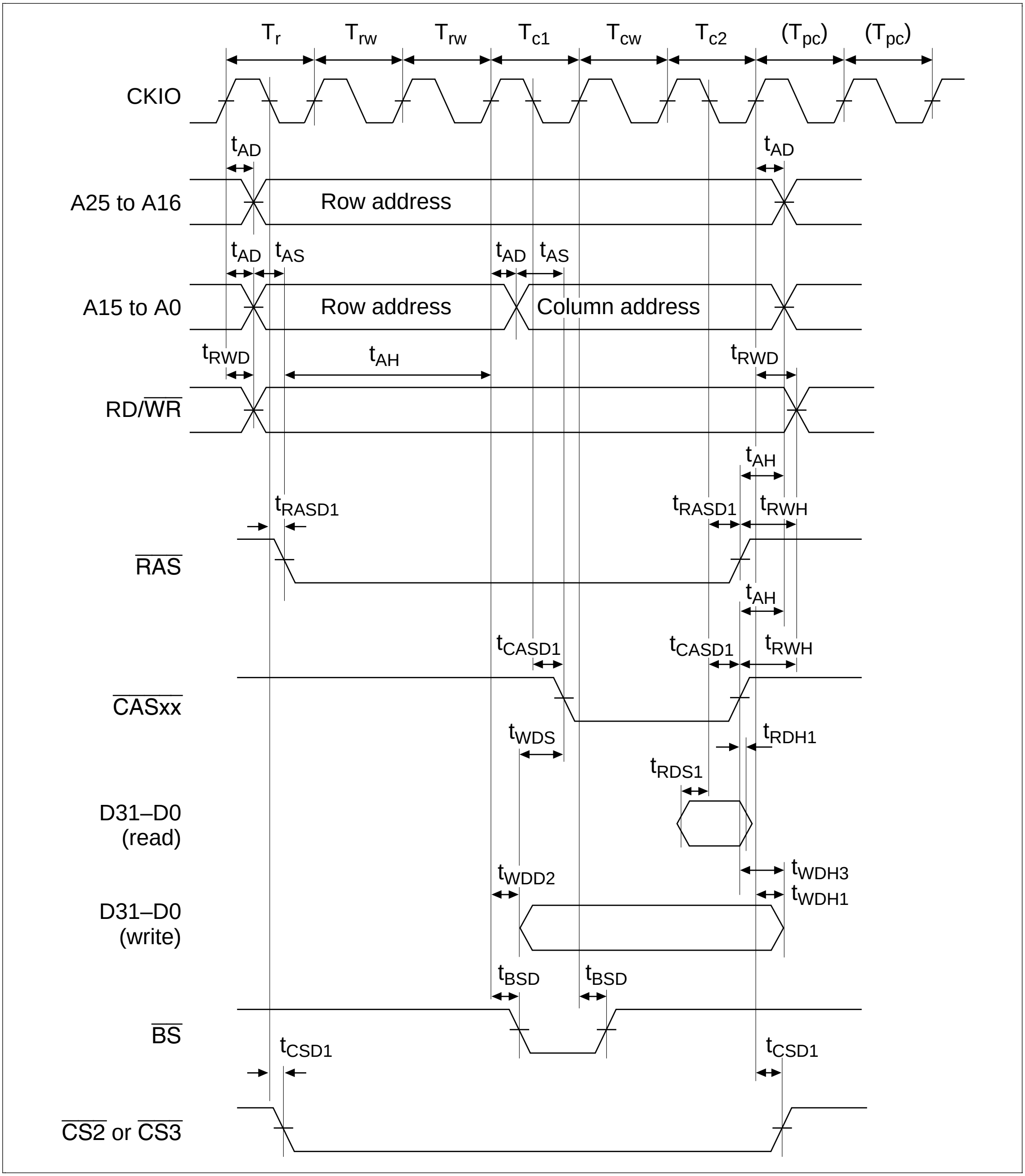


Figure 17.24 DRAM Bus Cycle (RCD = 2, AnW = 2, TPC = 1)

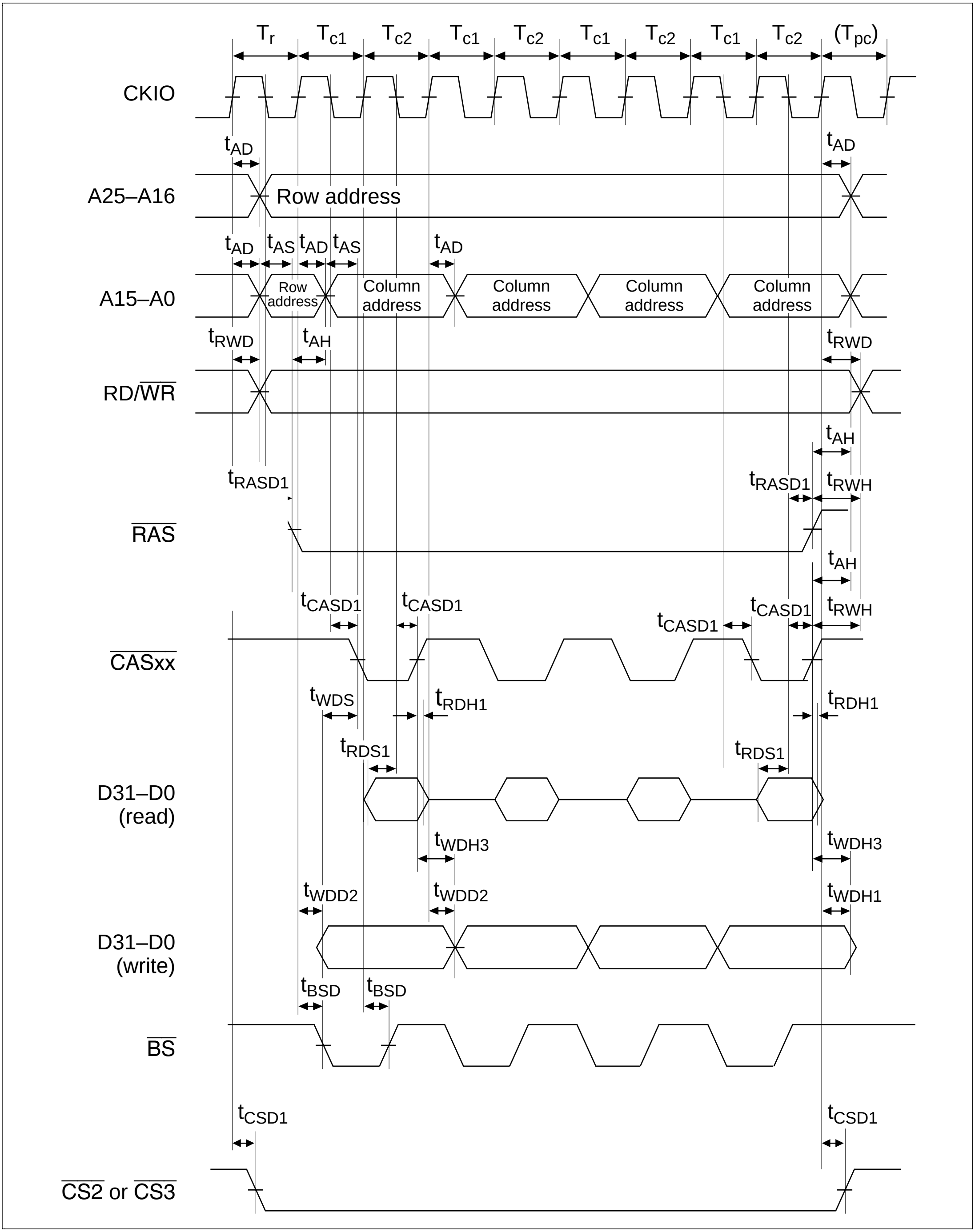


Figure 17.25 DRAM Burst Bus Cycle (RCD = 0, AnW = 1, TPC = 0)

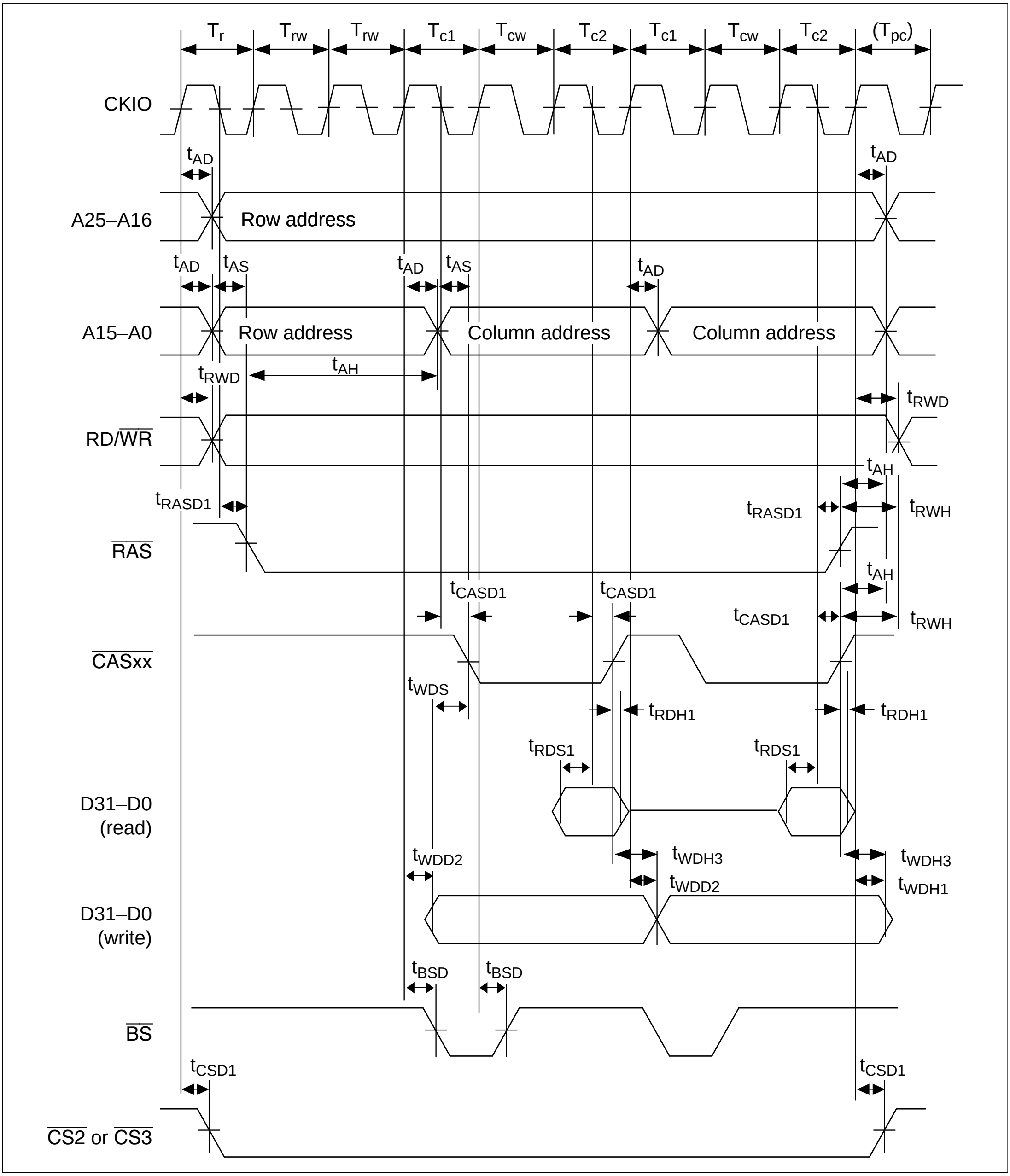


Figure 17.26 DRAM Burst Bus Cycle (RCD = 2, AnW = 2, TPC = 0)

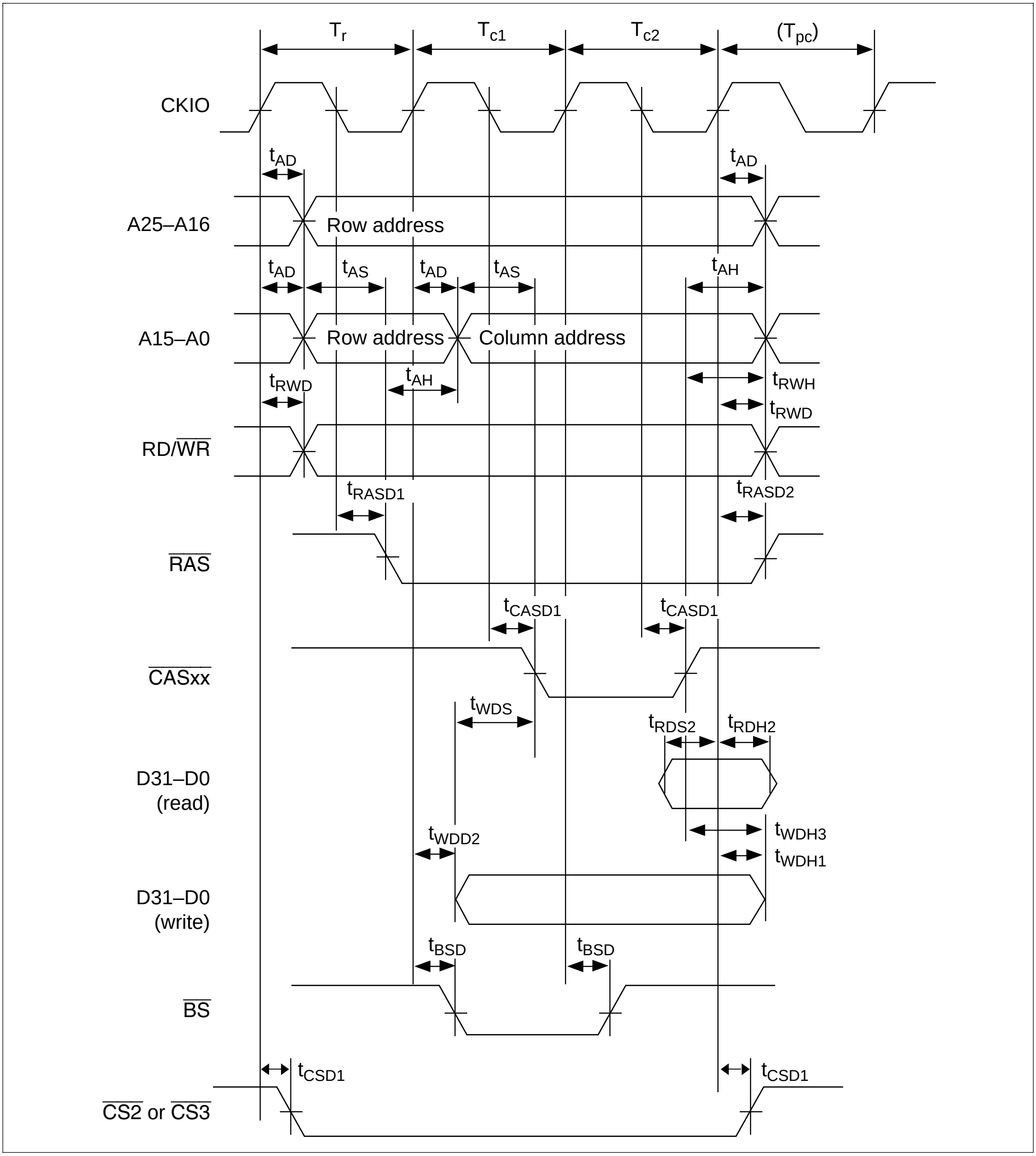


Figure 17.27 DRAM Bus Cycle (EDO Mode, RCD = 0, AnW = 1, TPC = 0)

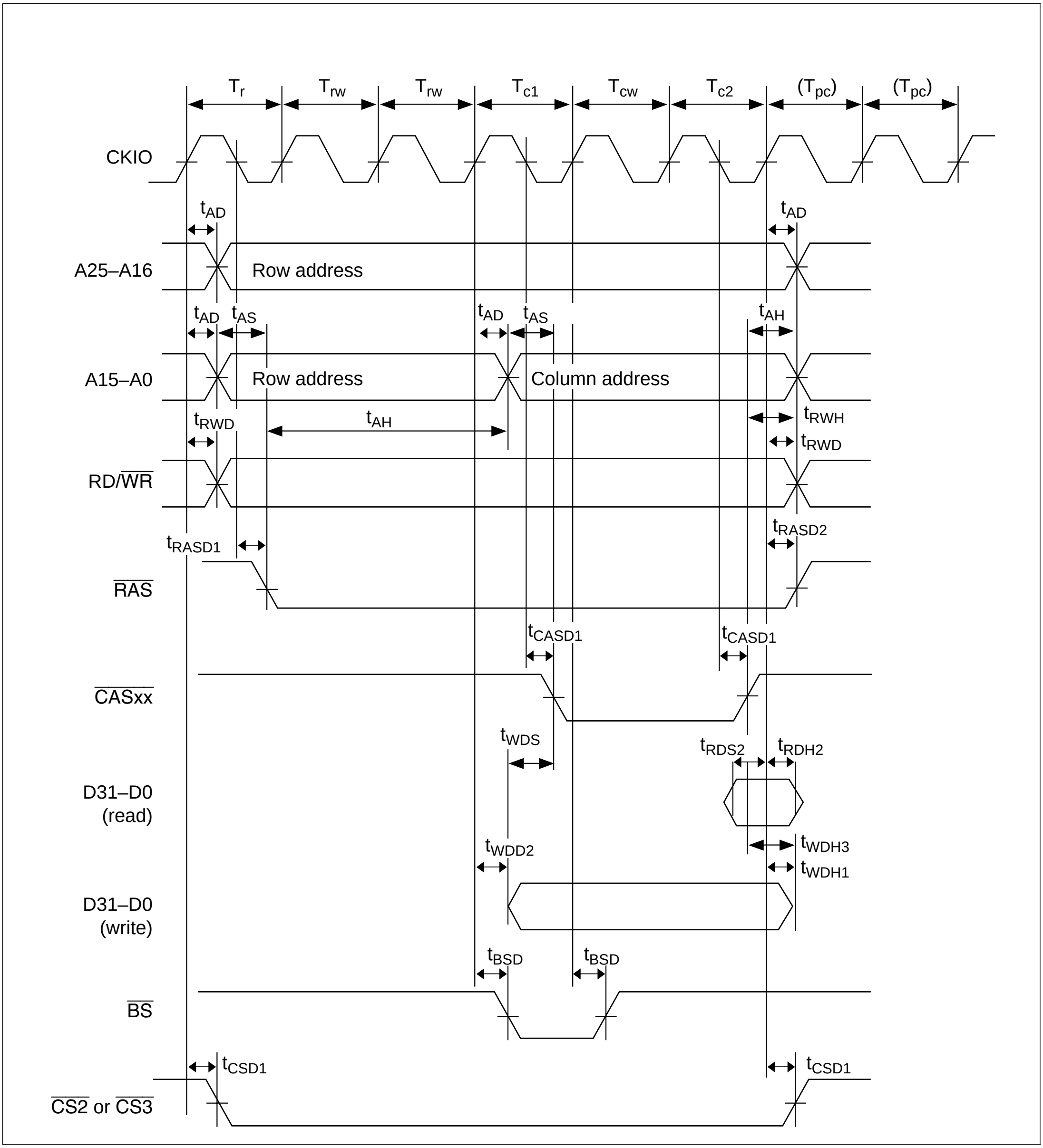


Figure 17.28 DRAM Bus Cycle (EDO Mode, RCD = 2, AnW = 2, TPC = 1)

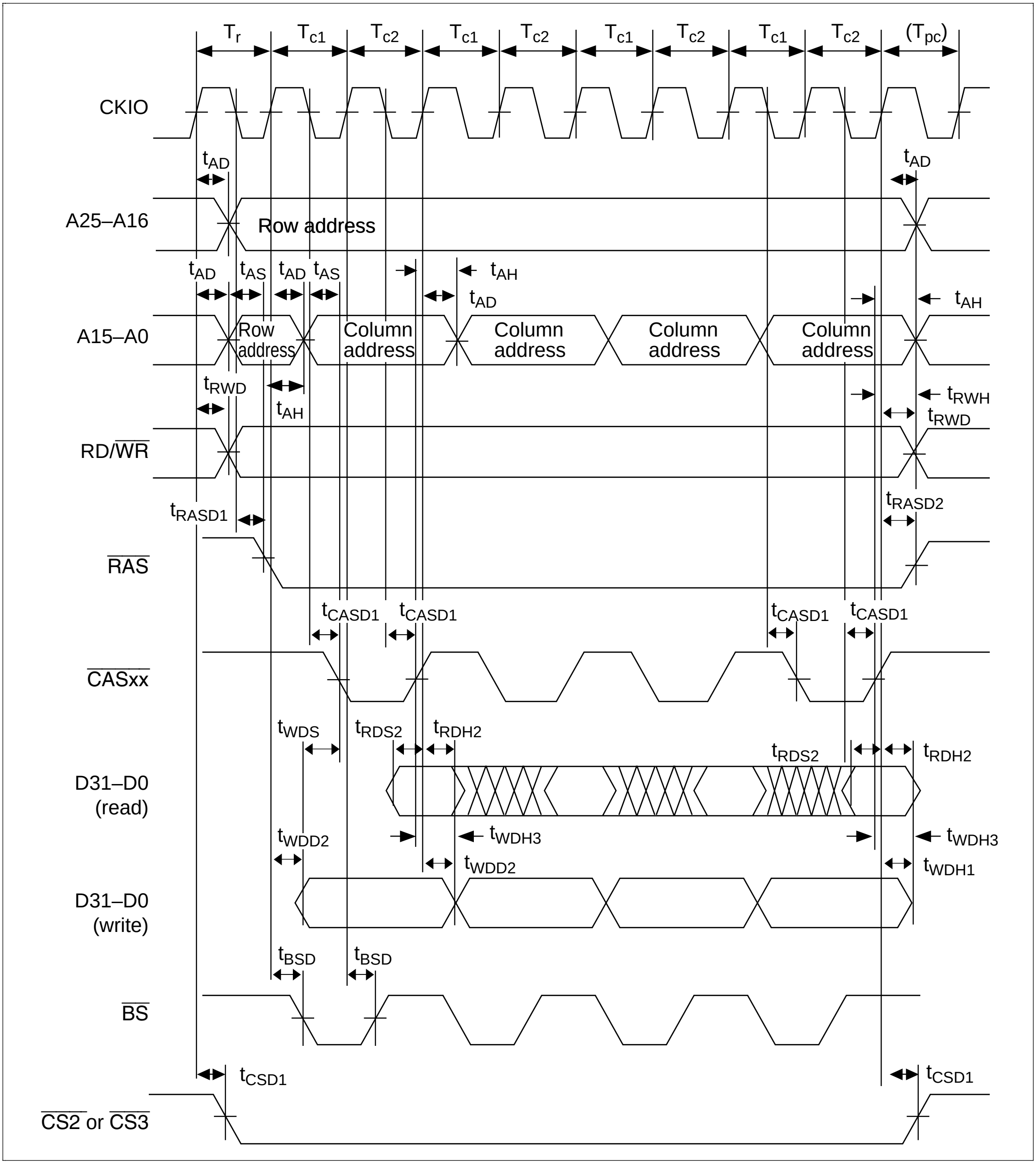


Figure 17.29 DRAM Burst Bus Cycle (EDO Mode, RCD = 0, AnW = 1, TPC = 0)

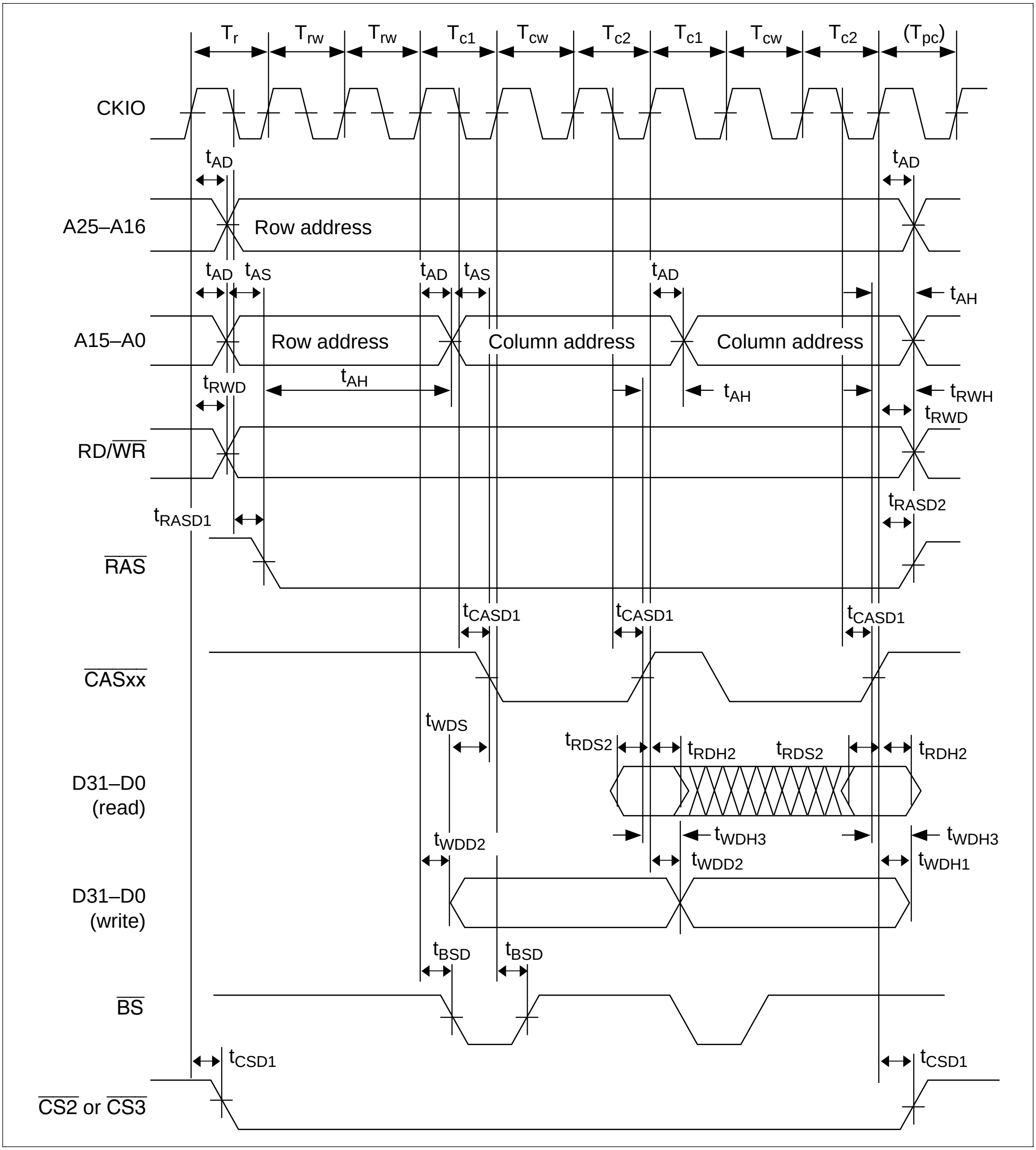


Figure 17.30 DRAM Burst Bus Cycle (EDO Mode, RCD = 2, AnW = 2, TPC = 0)

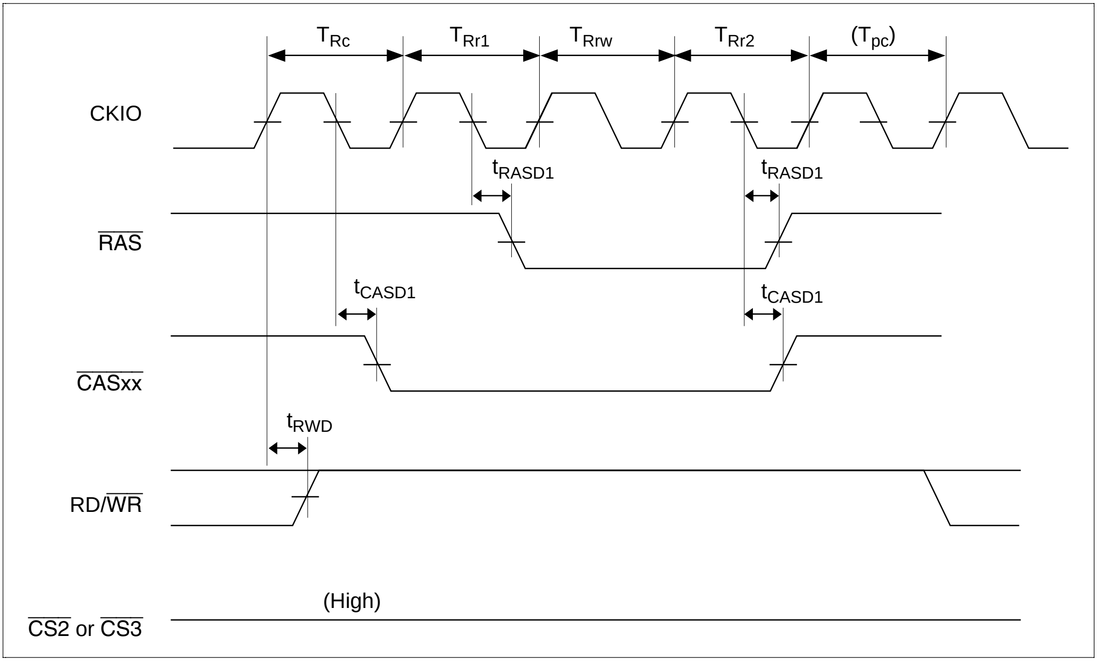


Figure 17.31 DRAM CAS-Before-RAS Refresh Cycle (TRAS = 0, TPC = 0)

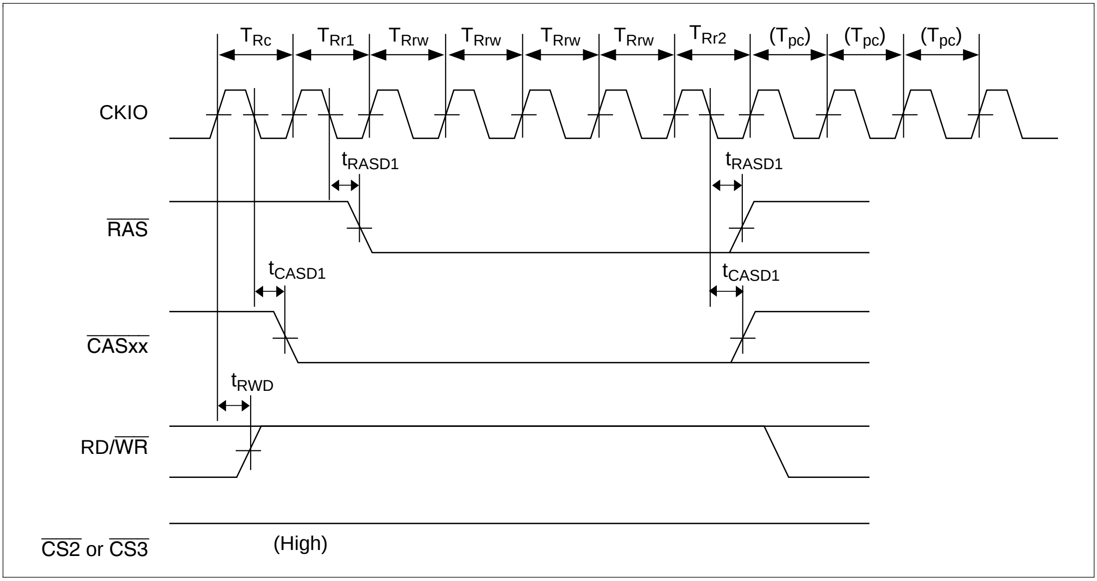


Figure 17.32 DRAM CAS-Before-RAS Refresh Cycle (TRAS = 3, TPC = 2)

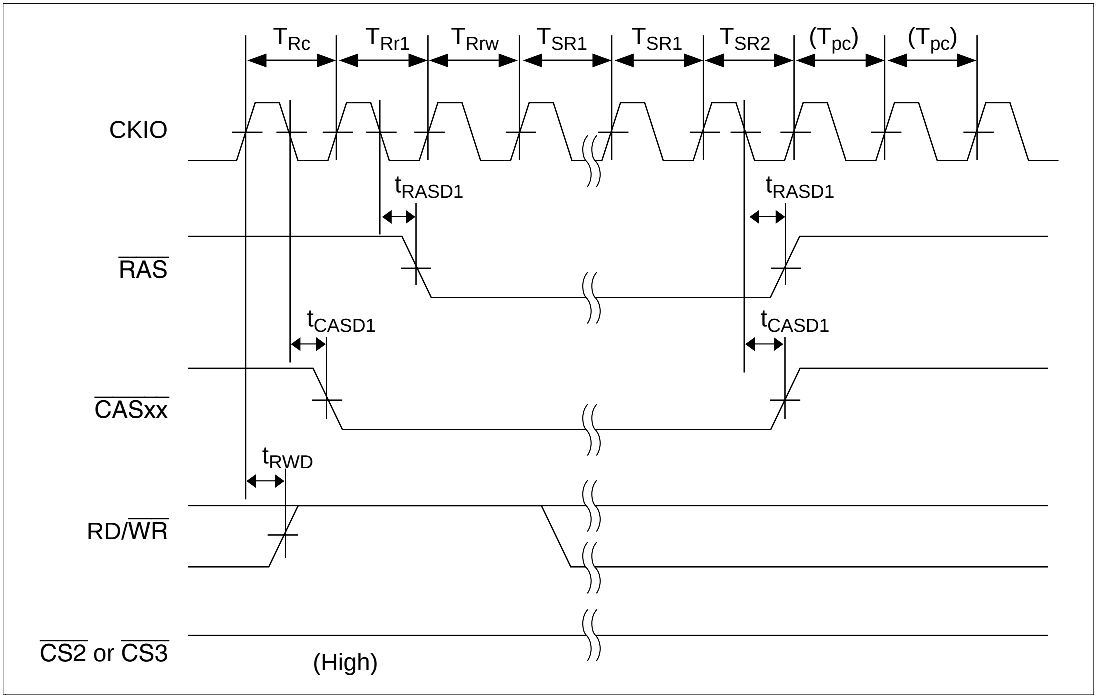


Figure 17.33 DRAM Self-Refresh Cycle (TPC = 0)

17.3.7 Synchronous DRAM Timing

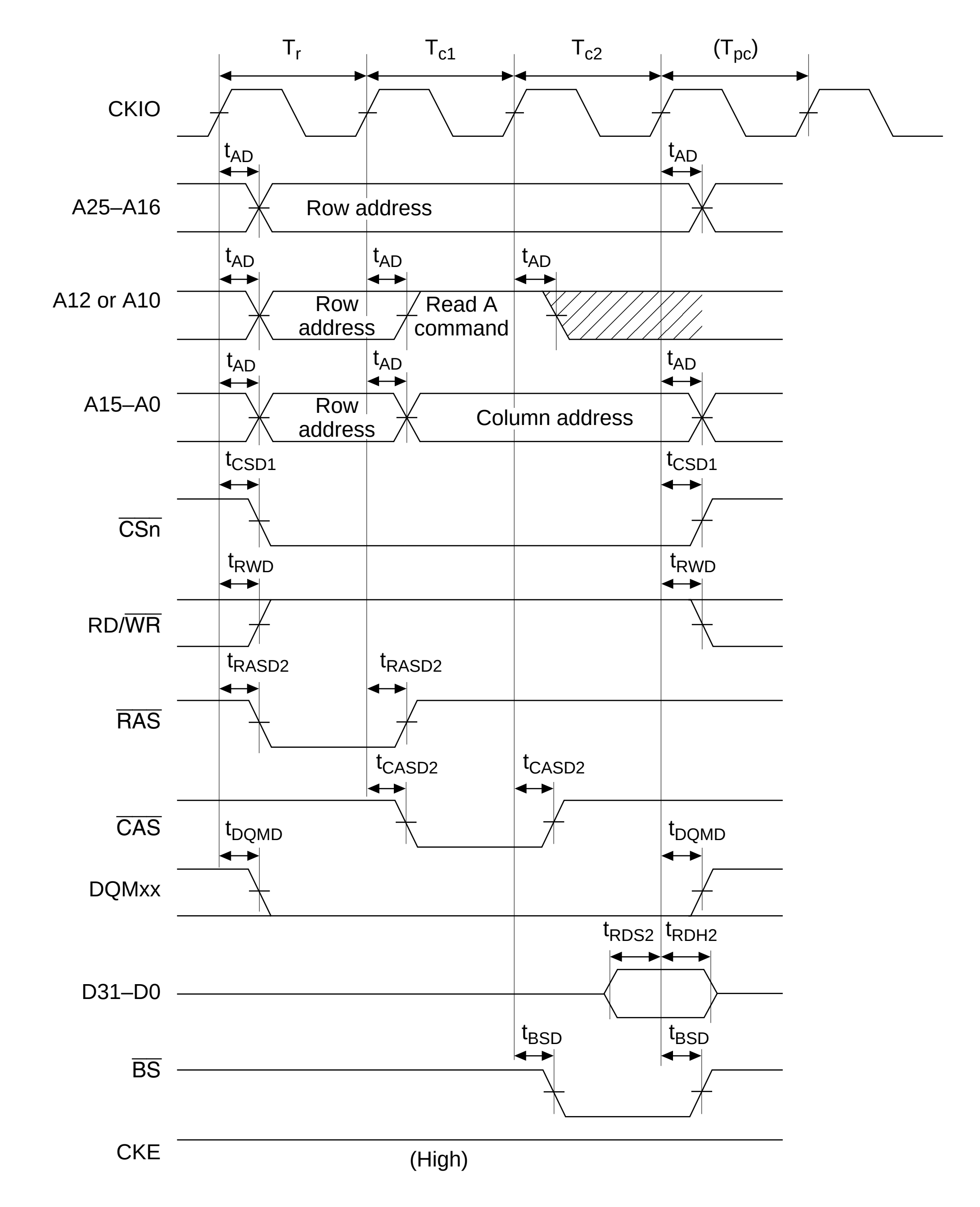


Figure 17.34 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0)

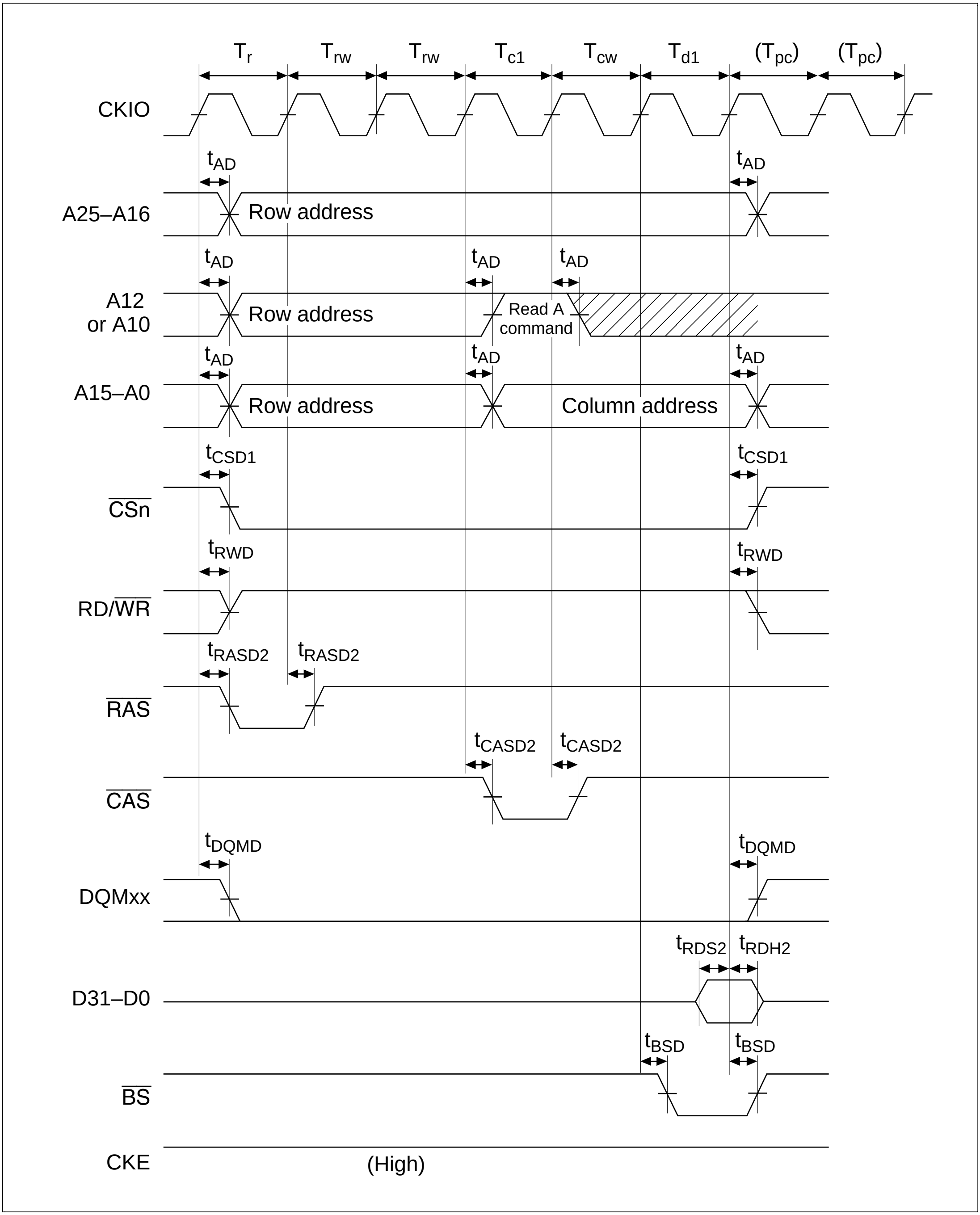


Figure 17.35 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1)

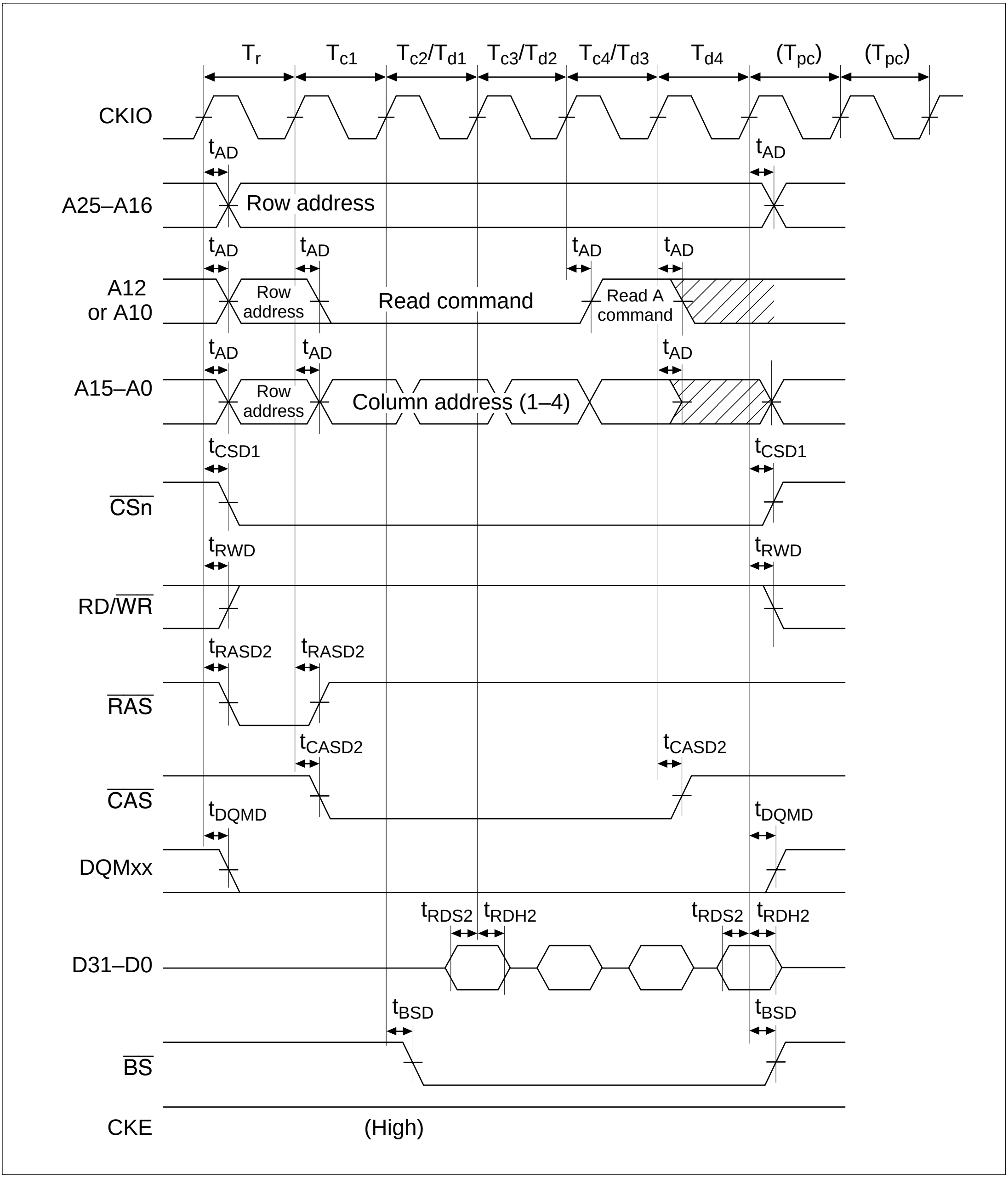


Figure 17.36 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

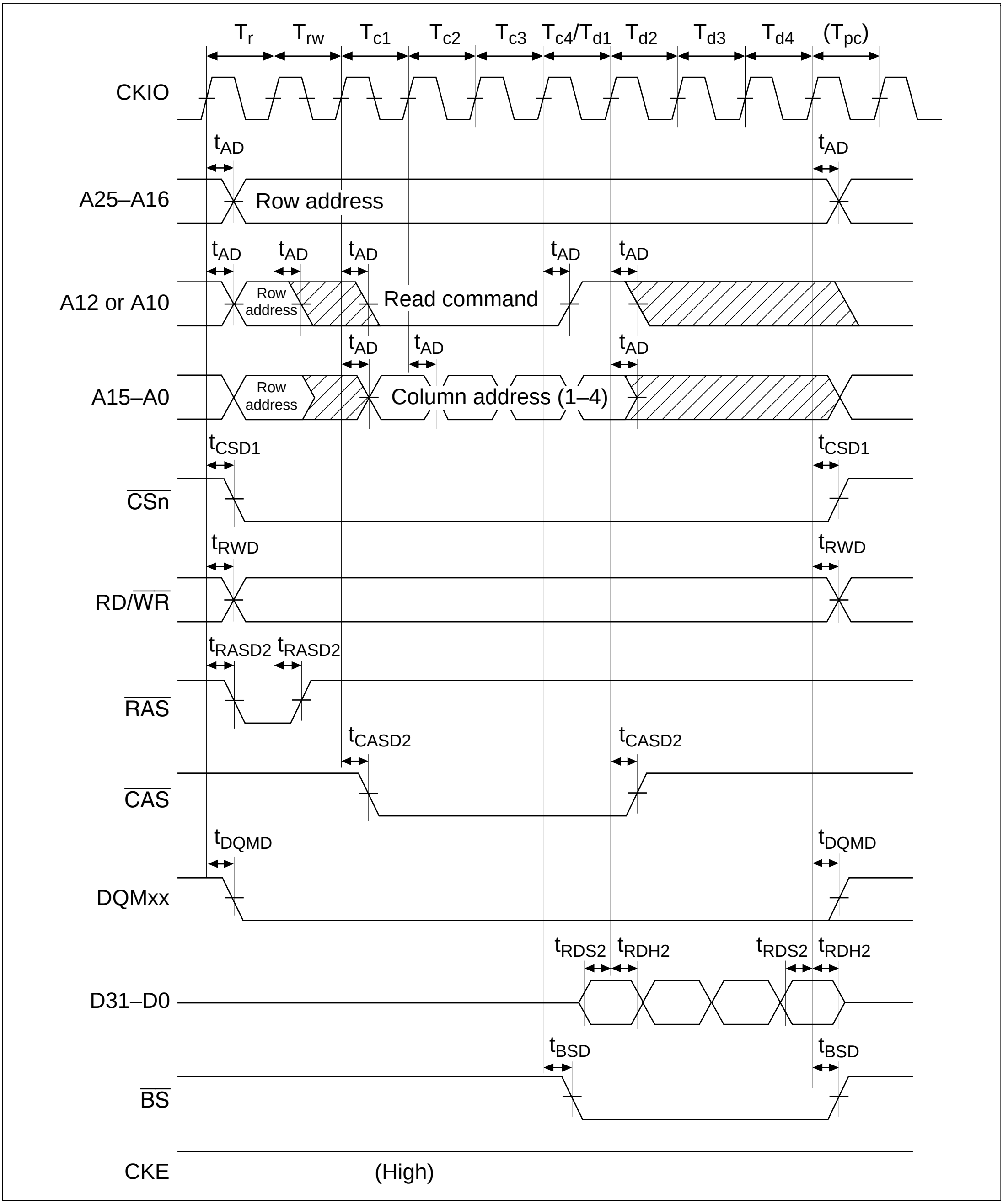


Figure 17.37 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

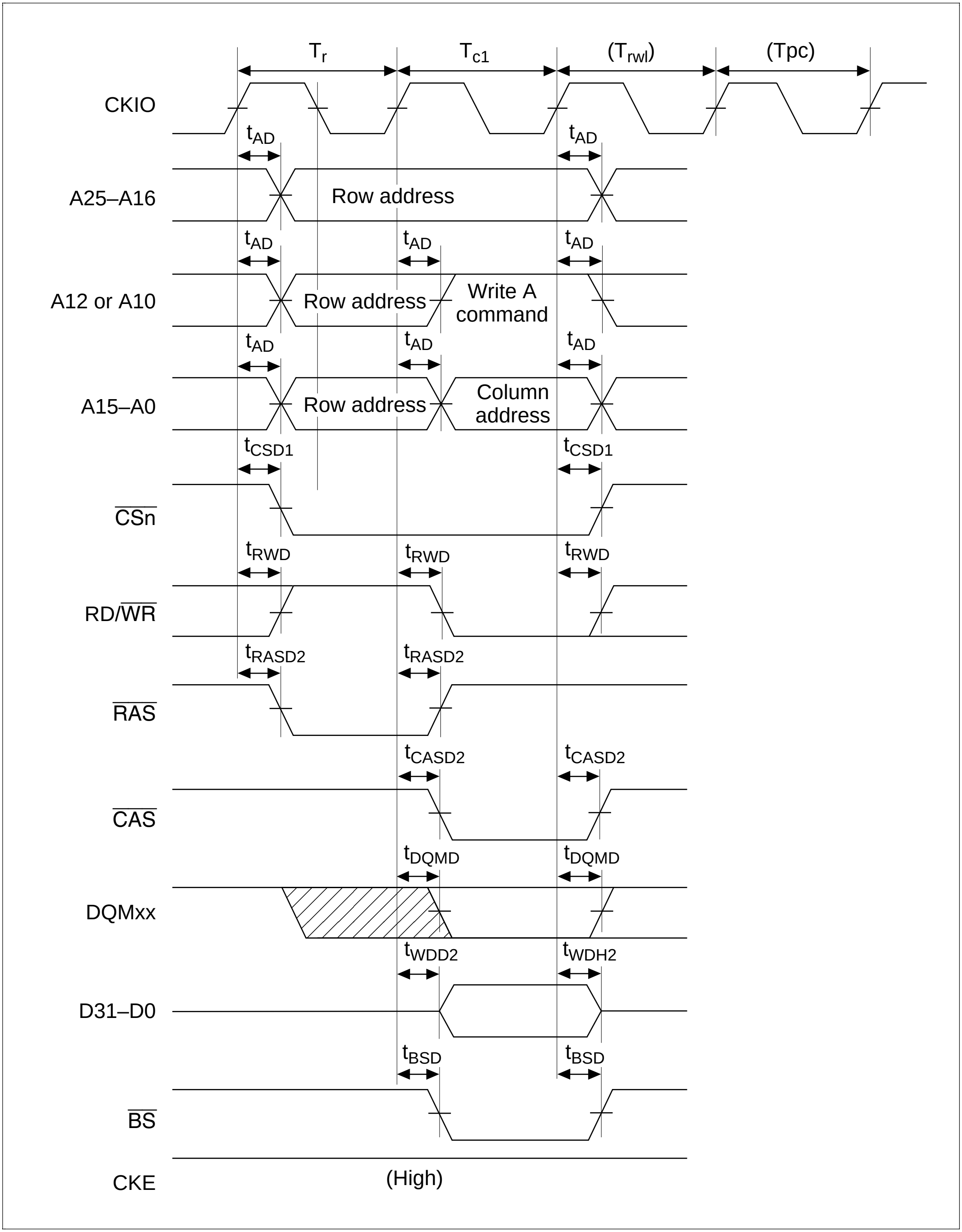


Figure 17.38 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0)

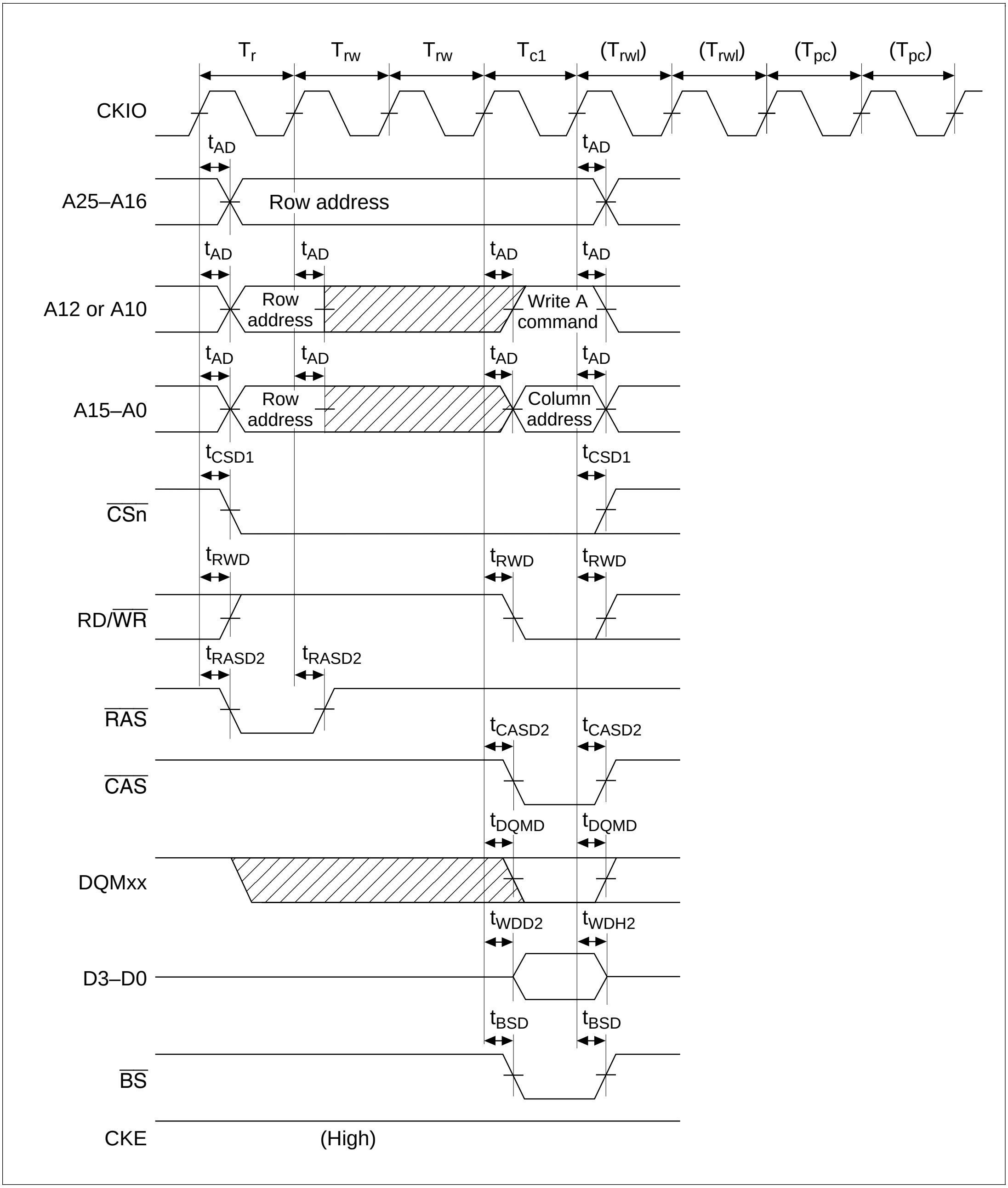


Figure 17.39 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1)

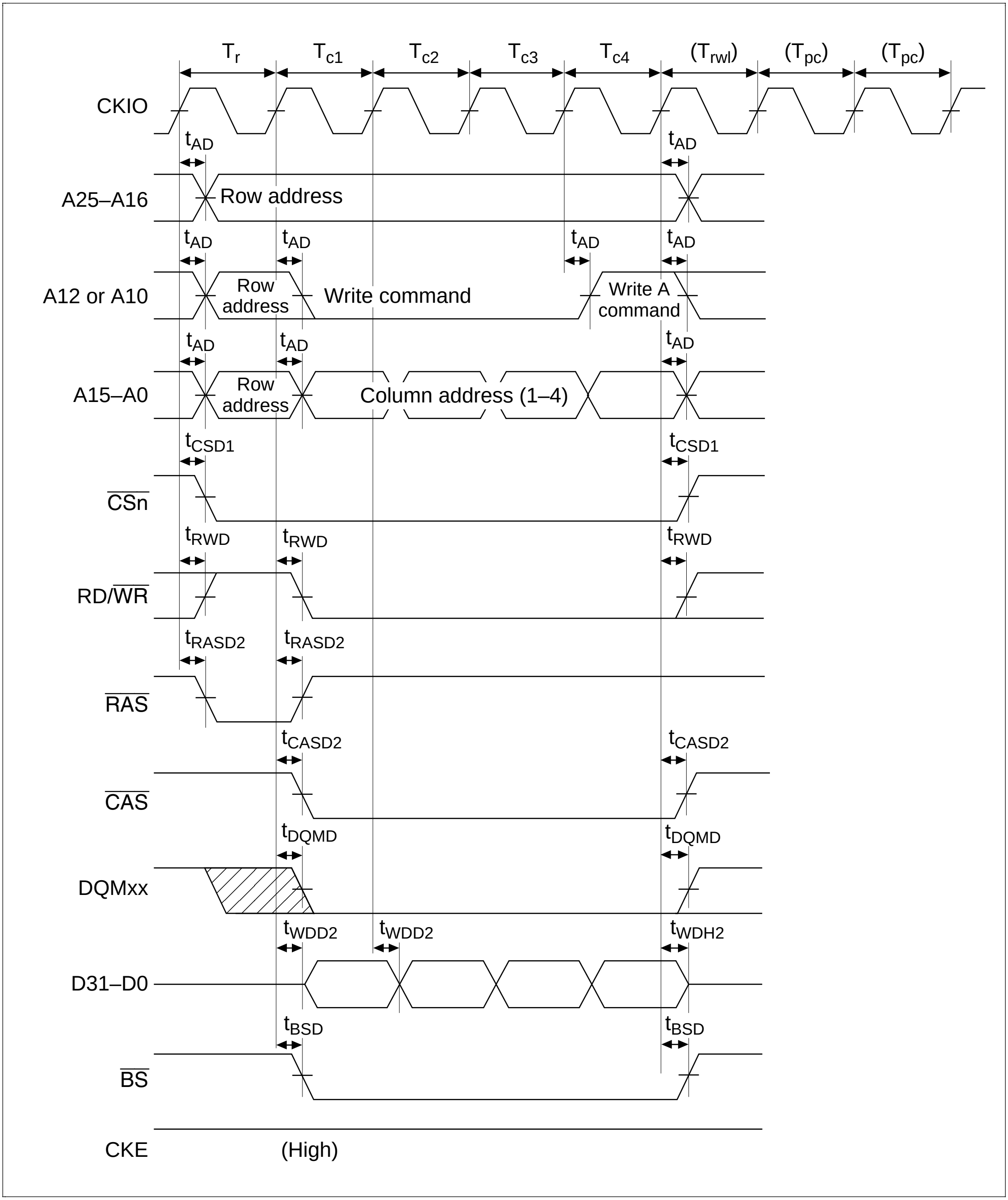


Figure 17.40 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

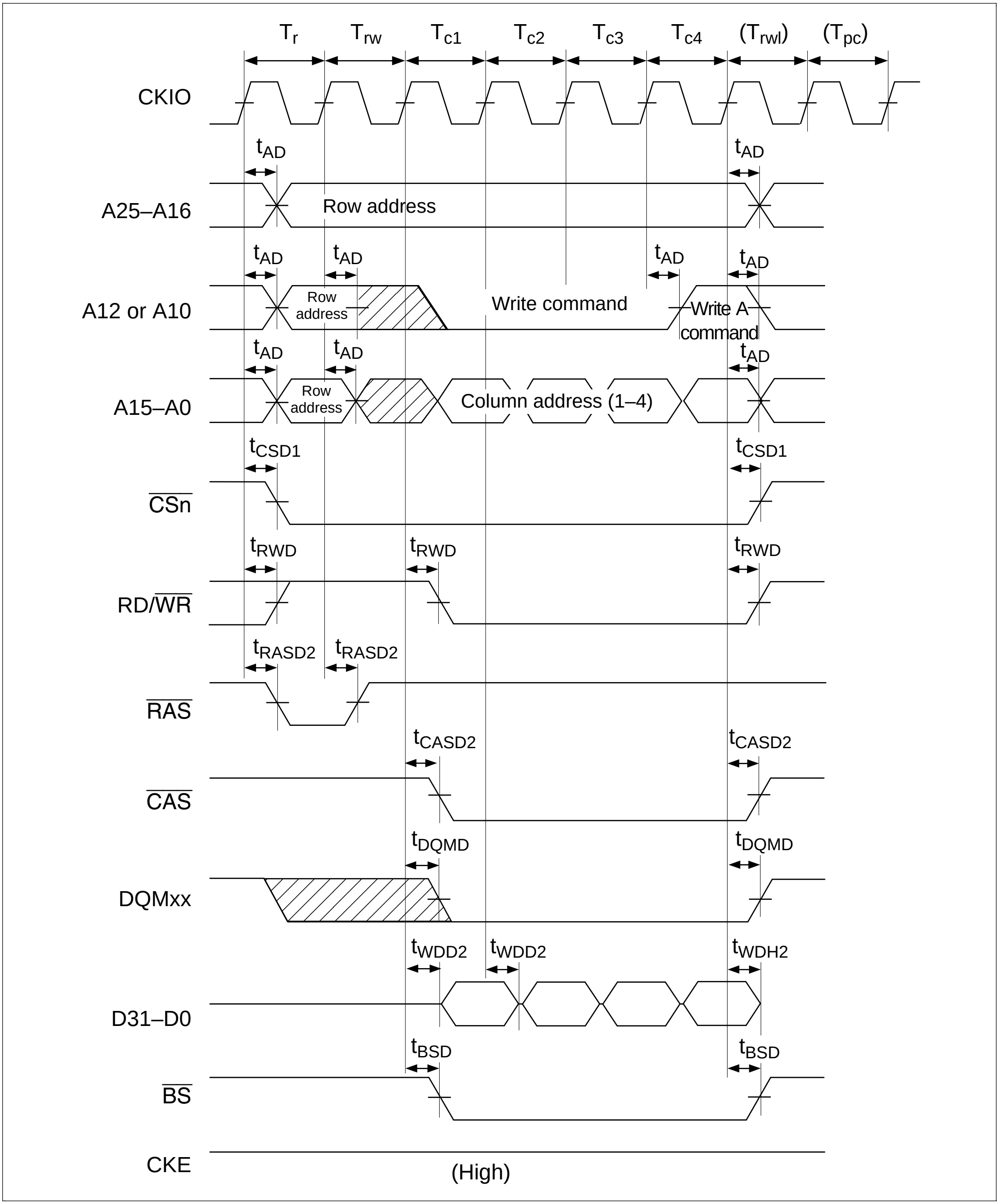


Figure 17.41 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

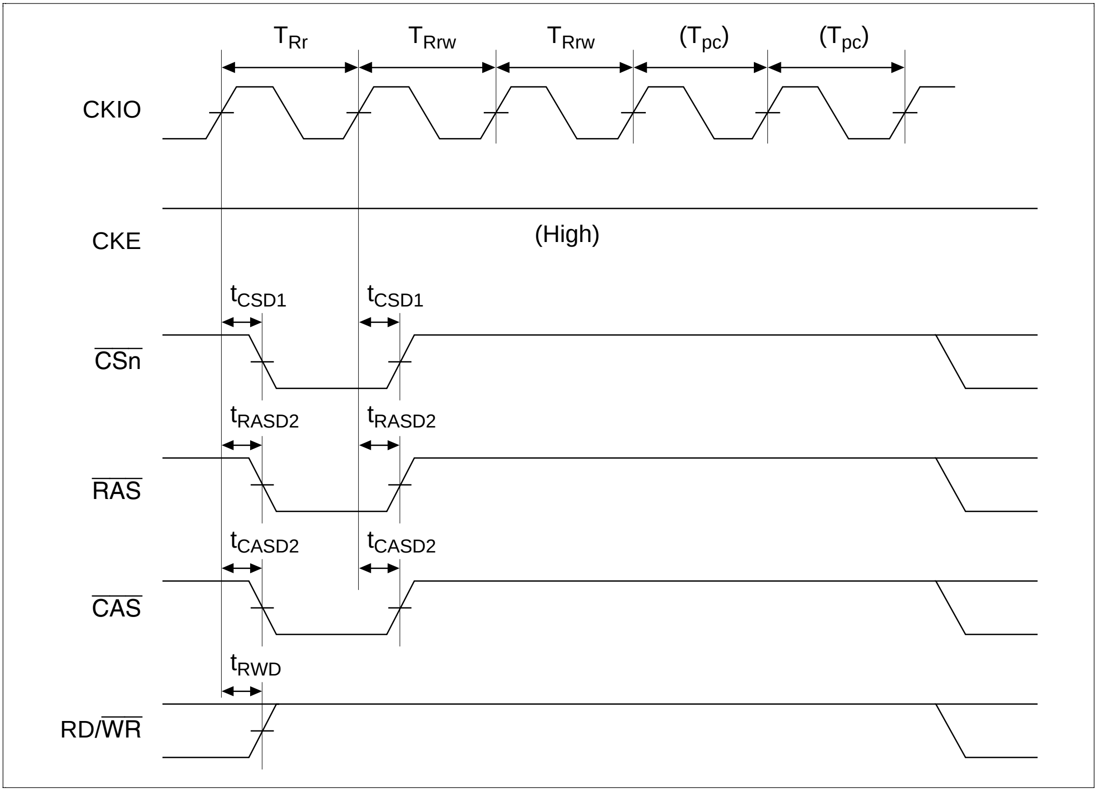


Figure 17.42 Synchronous DRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

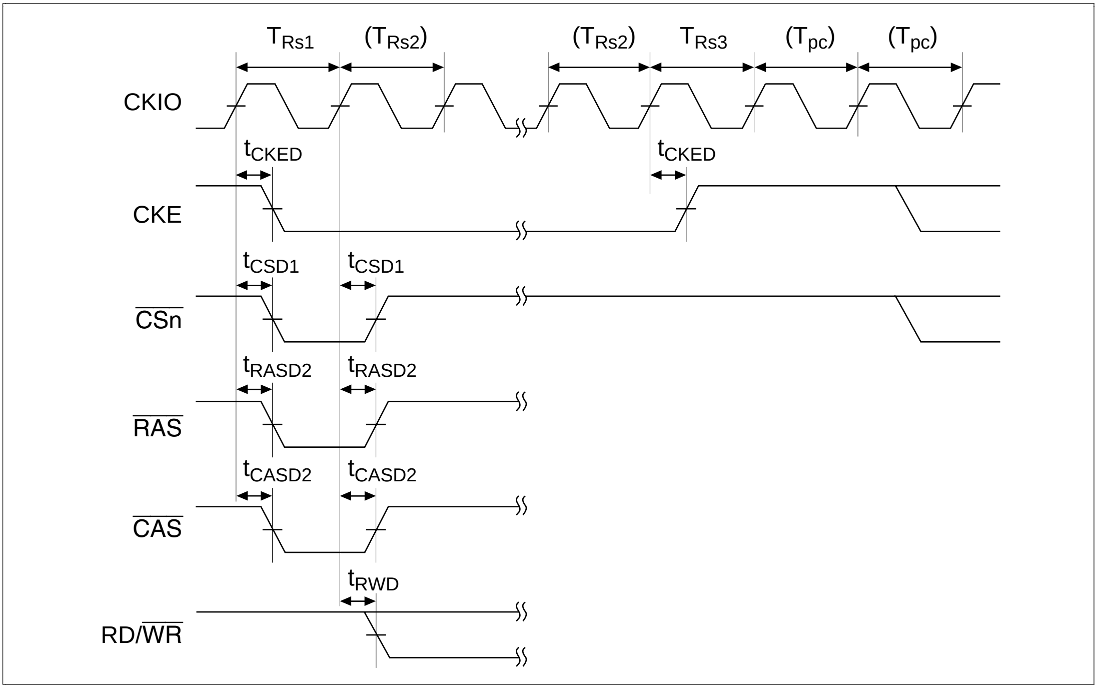


Figure 17.43 Synchronous DRAM Self-Refresh Cycle (TPC = 0)

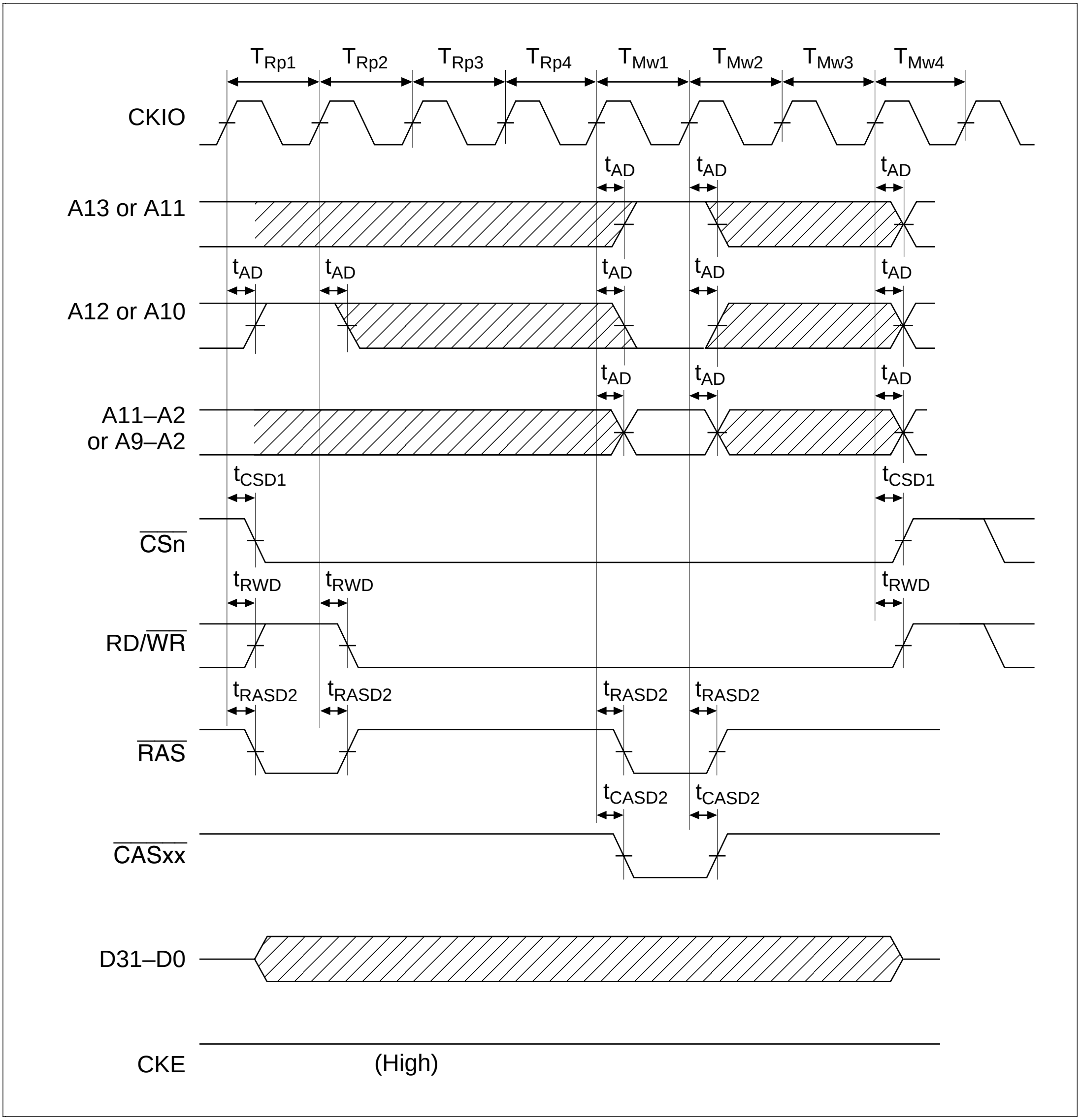


Figure 17.44 Synchronous DRAM Mode Register Write Cycle

17.3.8 Pseudo-SRAM Timing

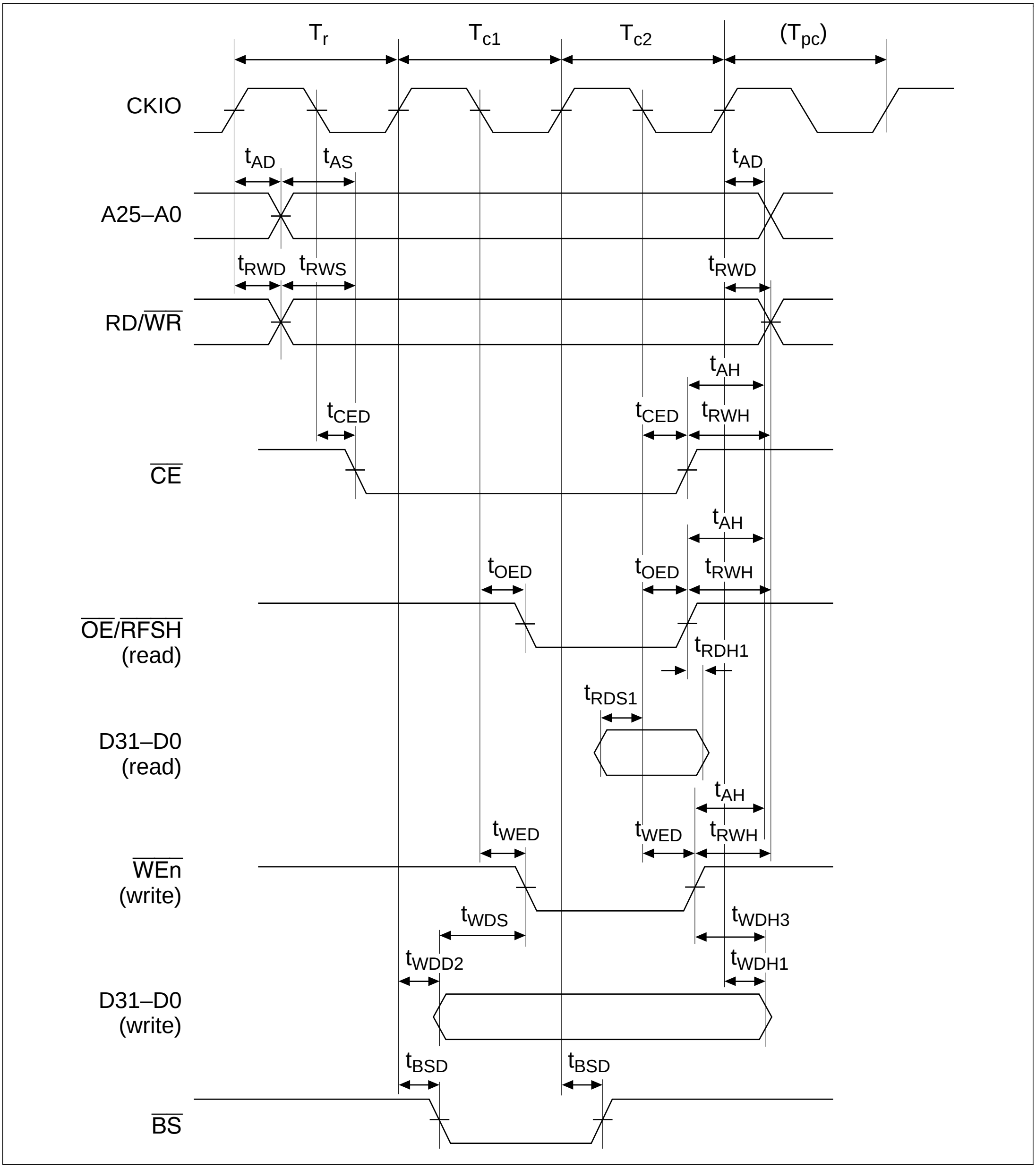


Figure 17.45 Pseudo-SRAM Bus Cycle (RCD = 0, A3W = 1, TPC = 0)

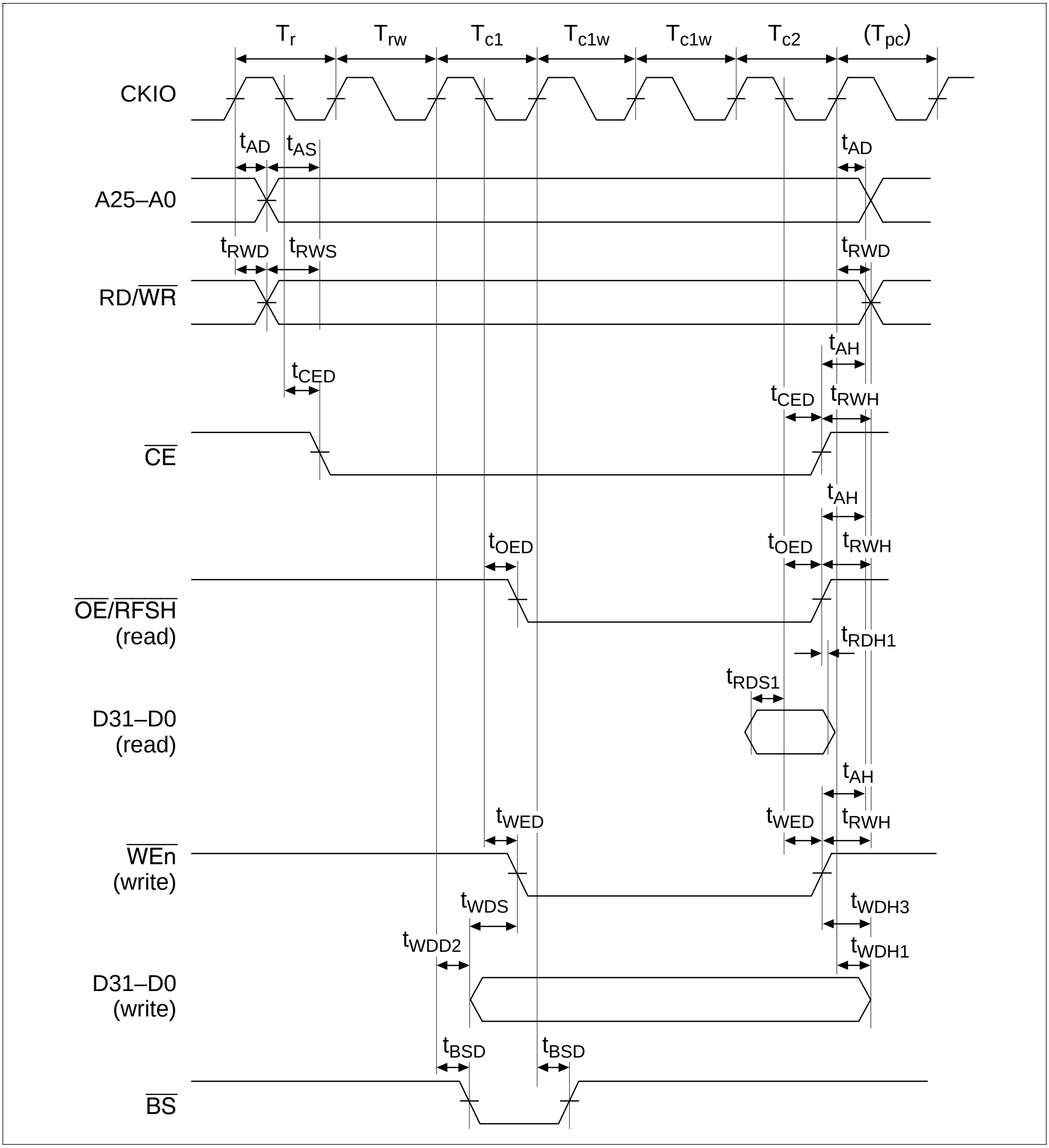


Figure 17.46 Pseudo-SRAM Read Cycle (RCD = 1, A3W = 3, TPC = 0)

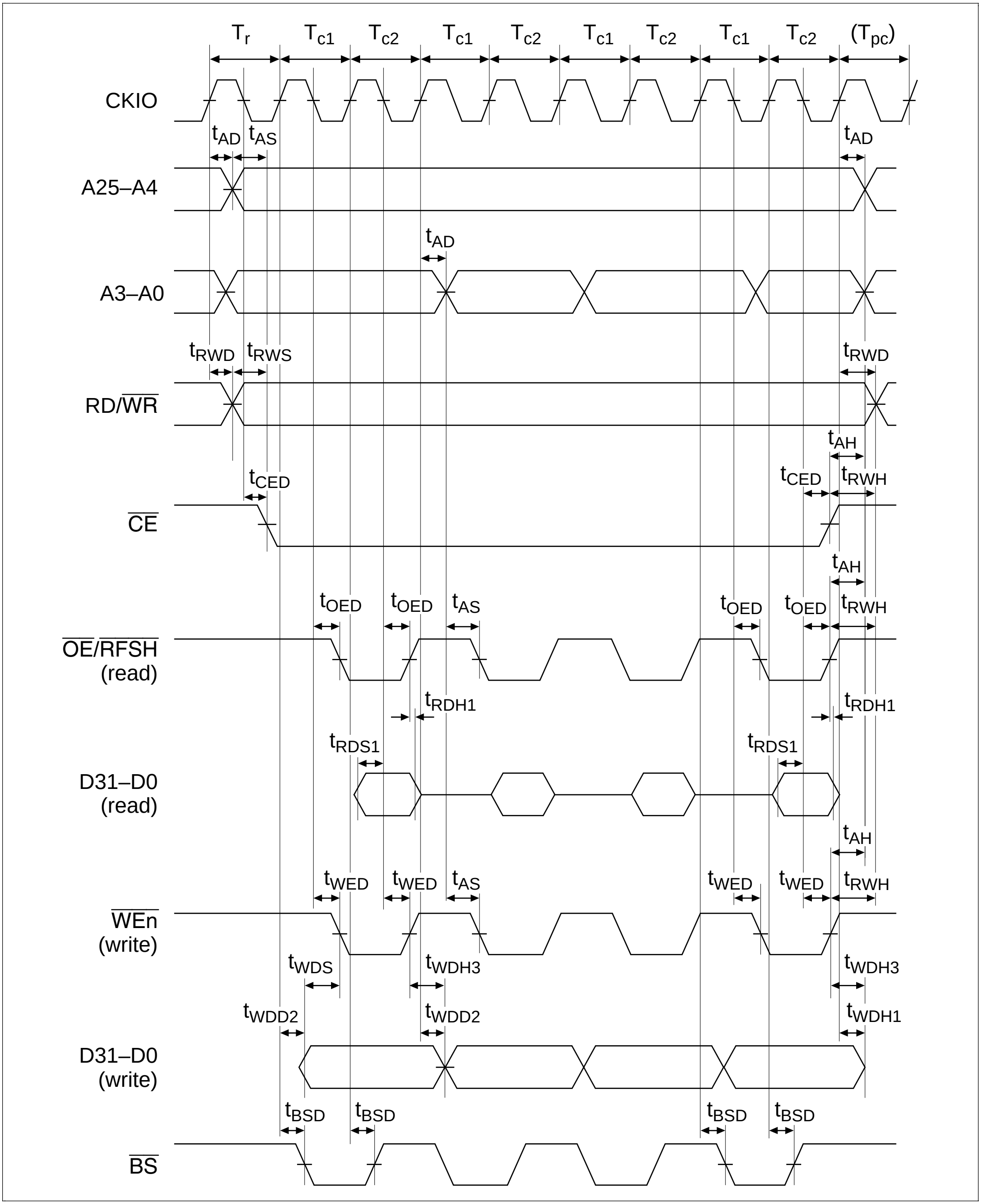


Figure 17.47 Pseudo-SRAM Bus Cycle (Static Column Mode, RCD = 0, A3W = 1, TPC = 0)

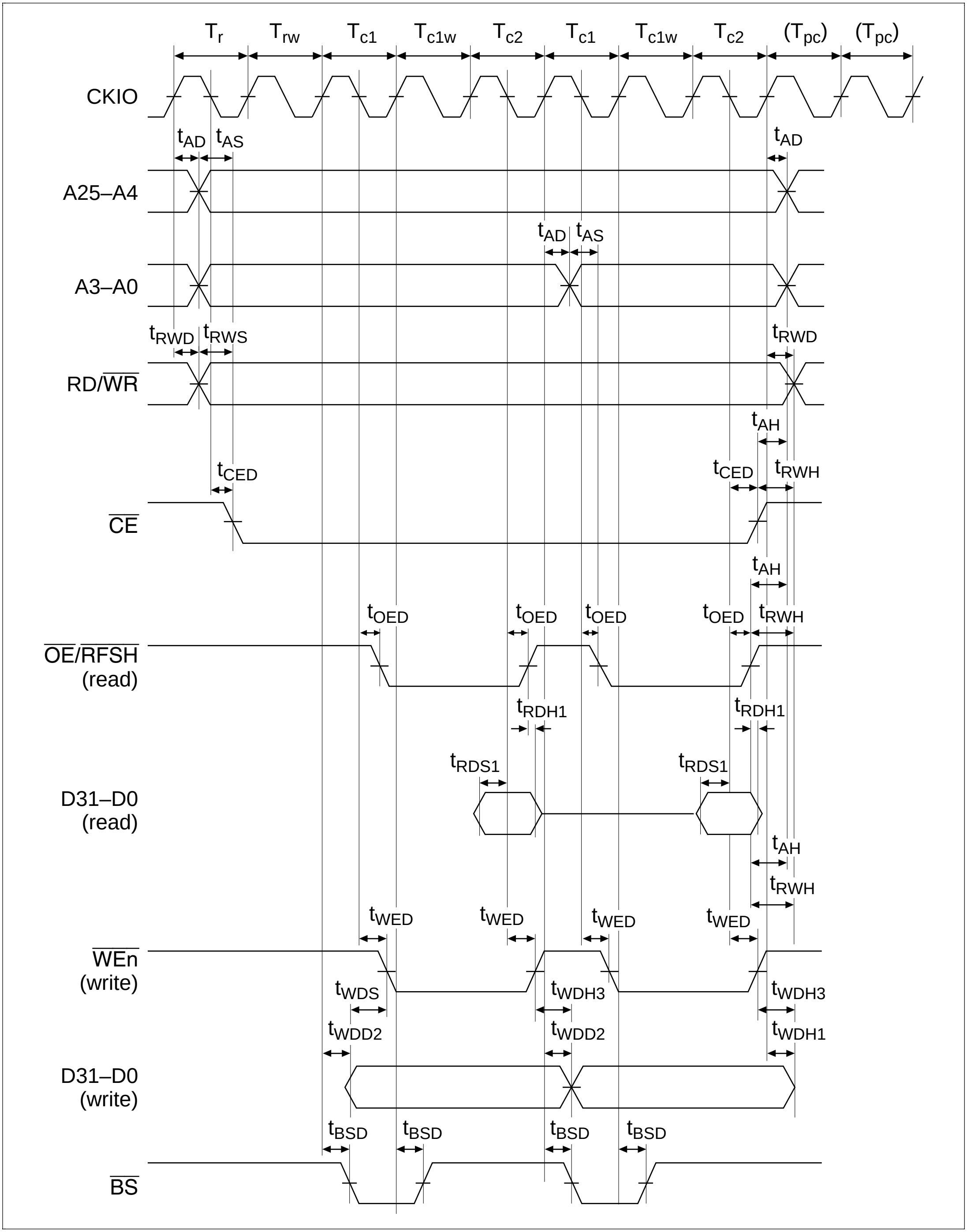


Figure 17.48 Pseudo-SRAM Bus Cycle (Static Column Mode, RCD = 1, A3W = 2, TPC = 1)

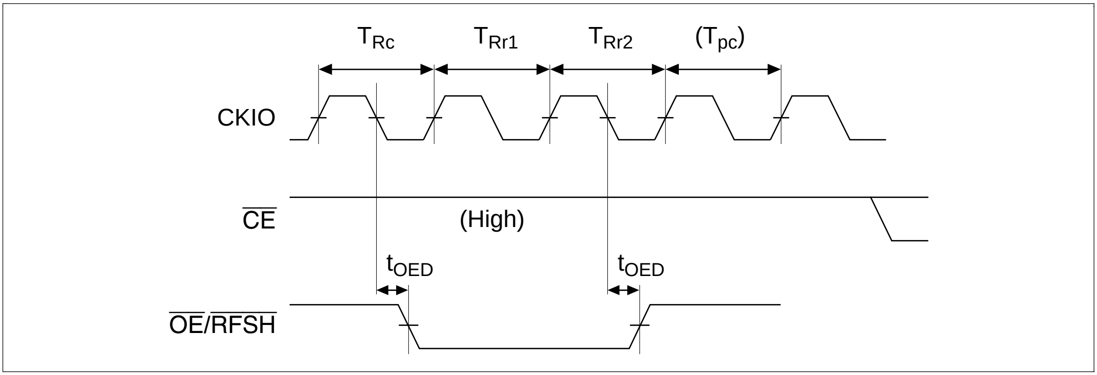


Figure 17.49 Pseudo-SRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

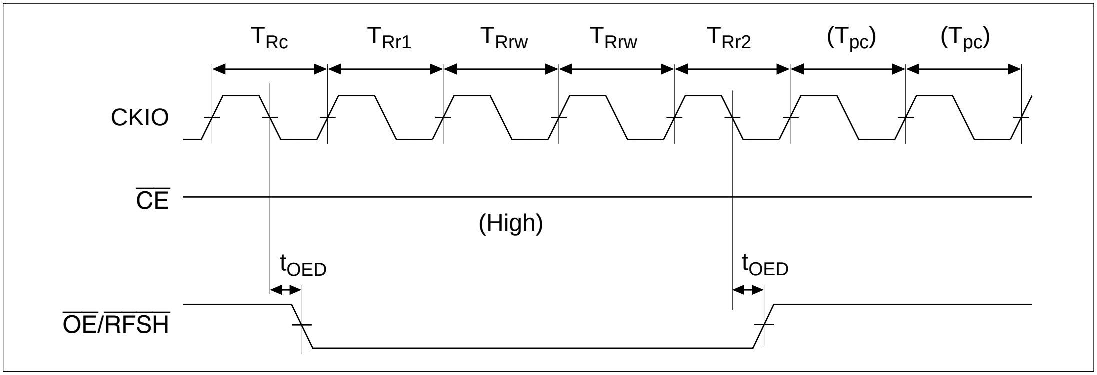


Figure 17.50 Pseudo-SRAM Auto-Refresh Cycle (TRAS = 2, TPC = 1)

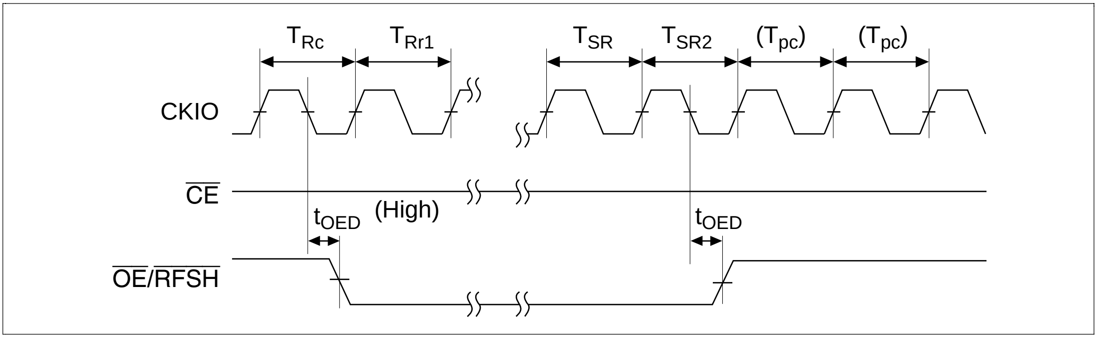


Figure 17.51 Pseudo-SRAM Self-Refresh Cycle (TPC = 0)

17.3.9 PCMCIA Timing

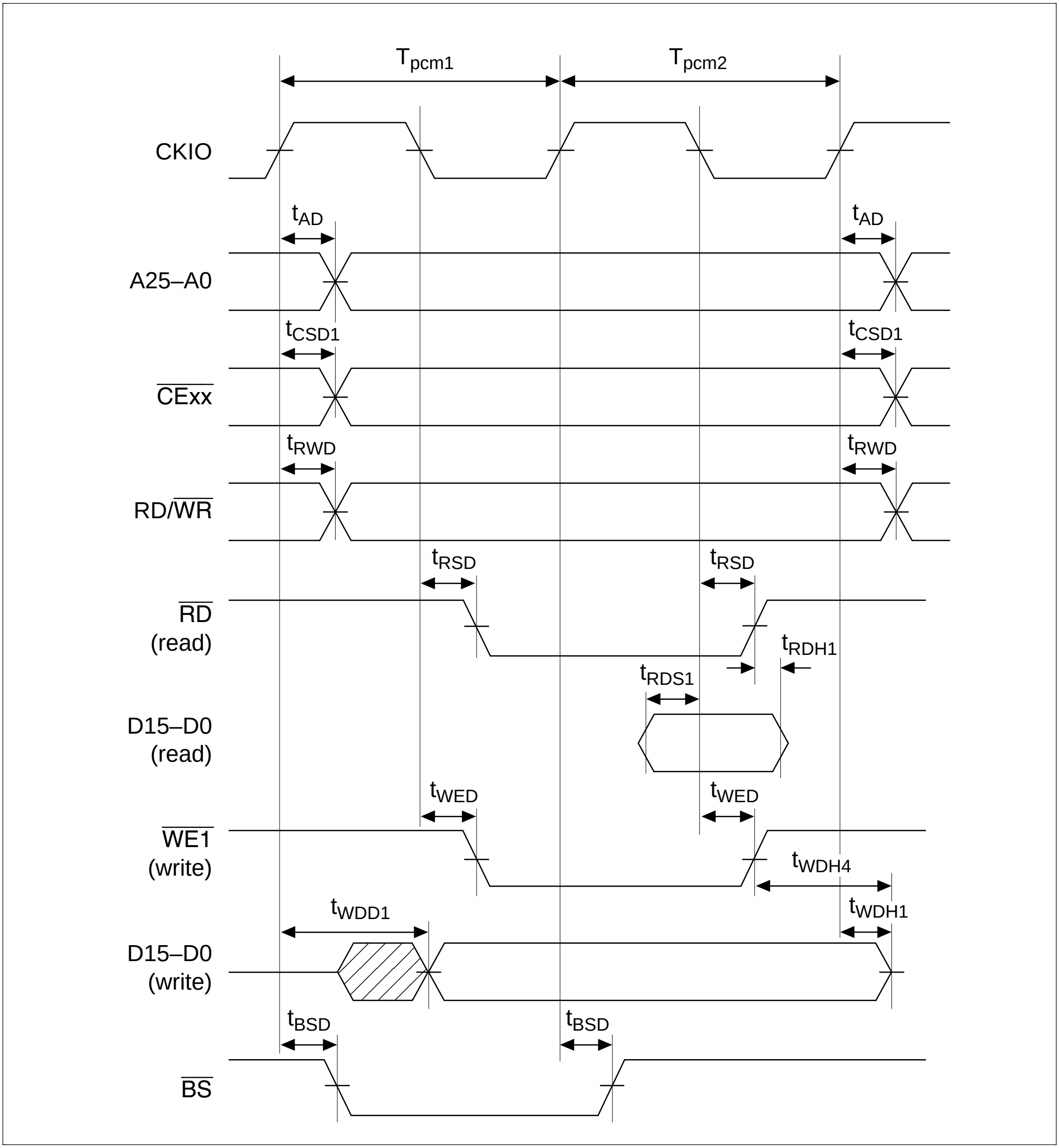


Figure 17.52 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

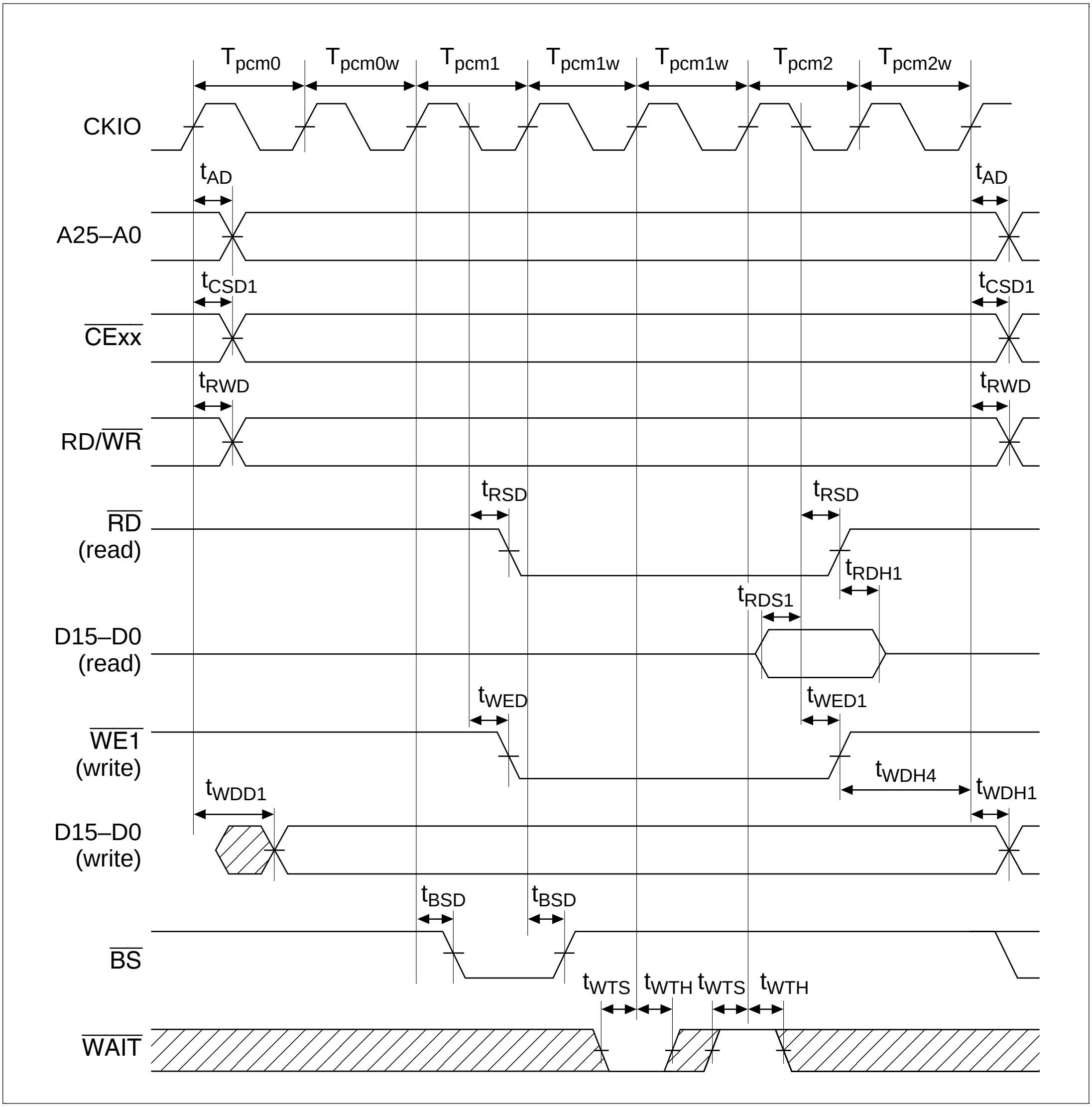
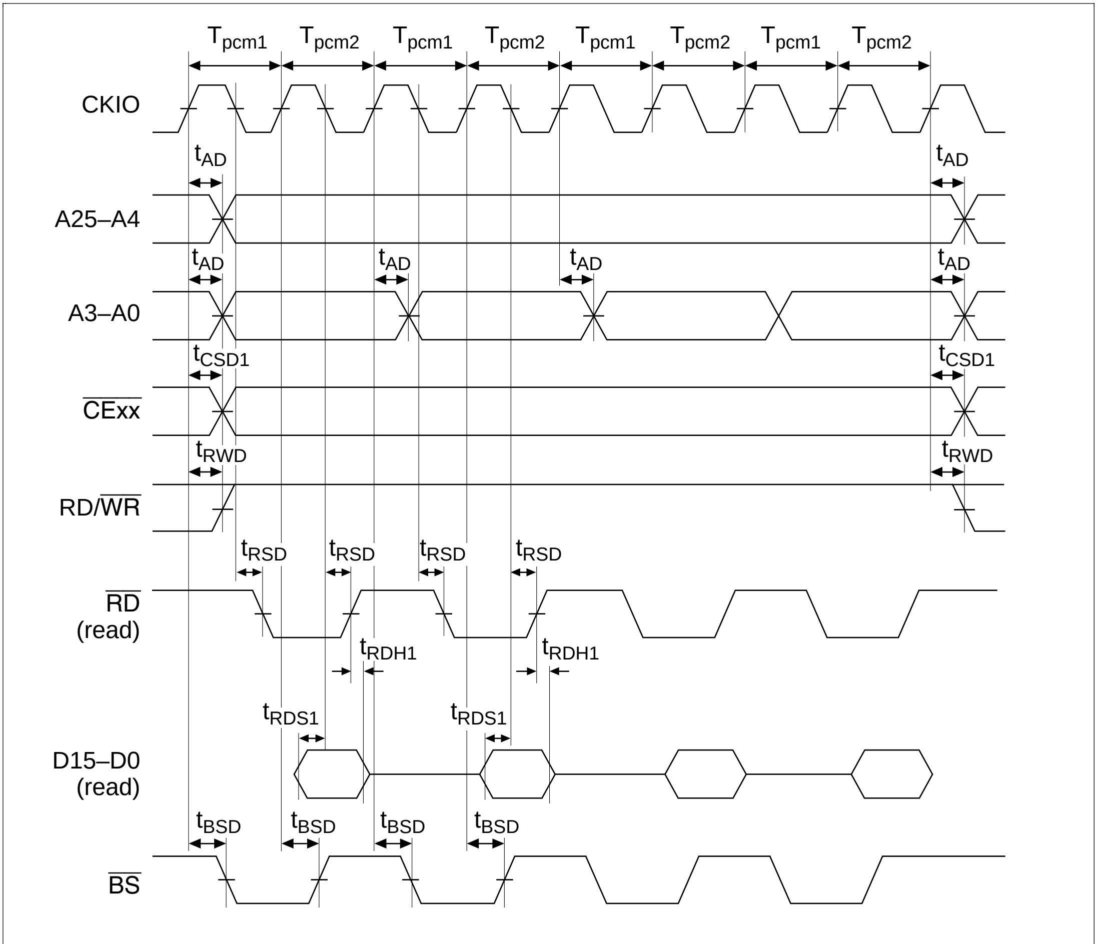
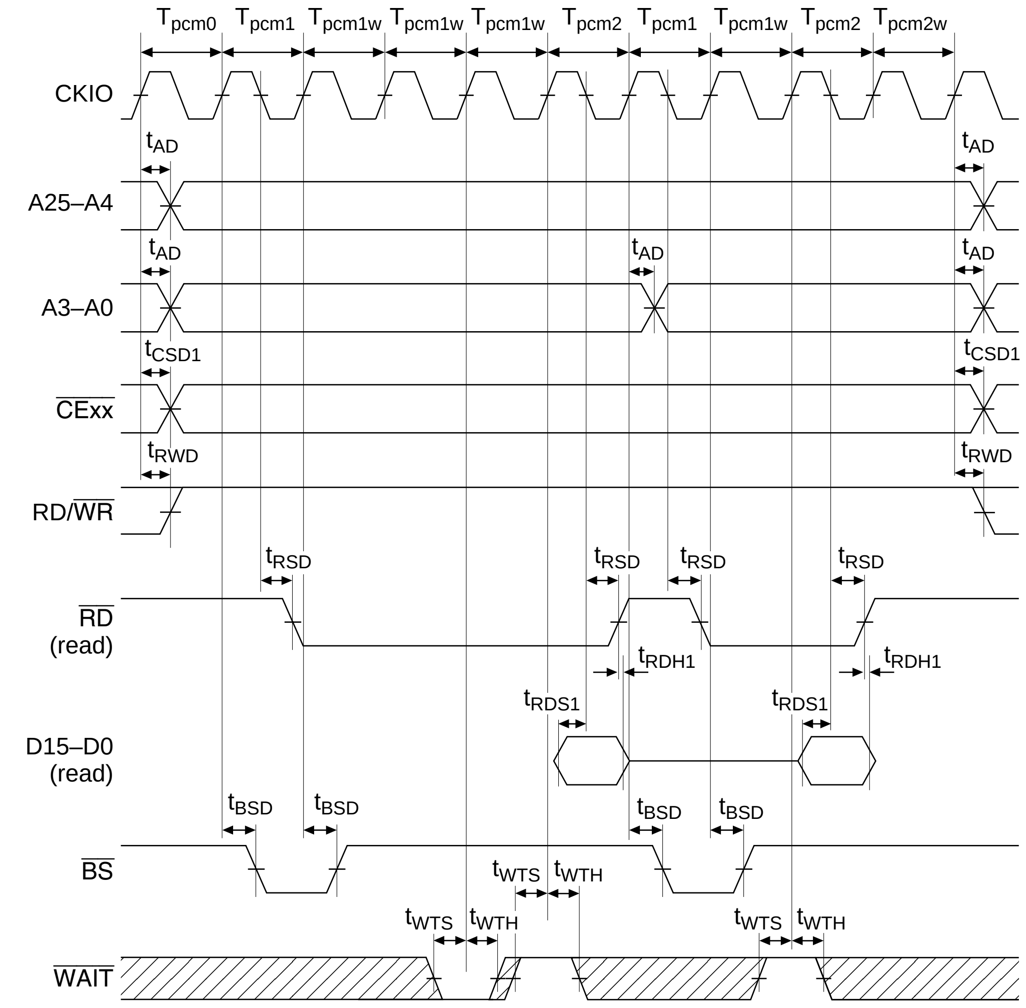


Figure 17.53 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, 1 Wait, External Wait)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 17.54 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 17.55 PCMCIA Memory Bus Cycle (Burst Read, TED = 1, TEH = 1, 2 Waits, Burst Pitch = 3)

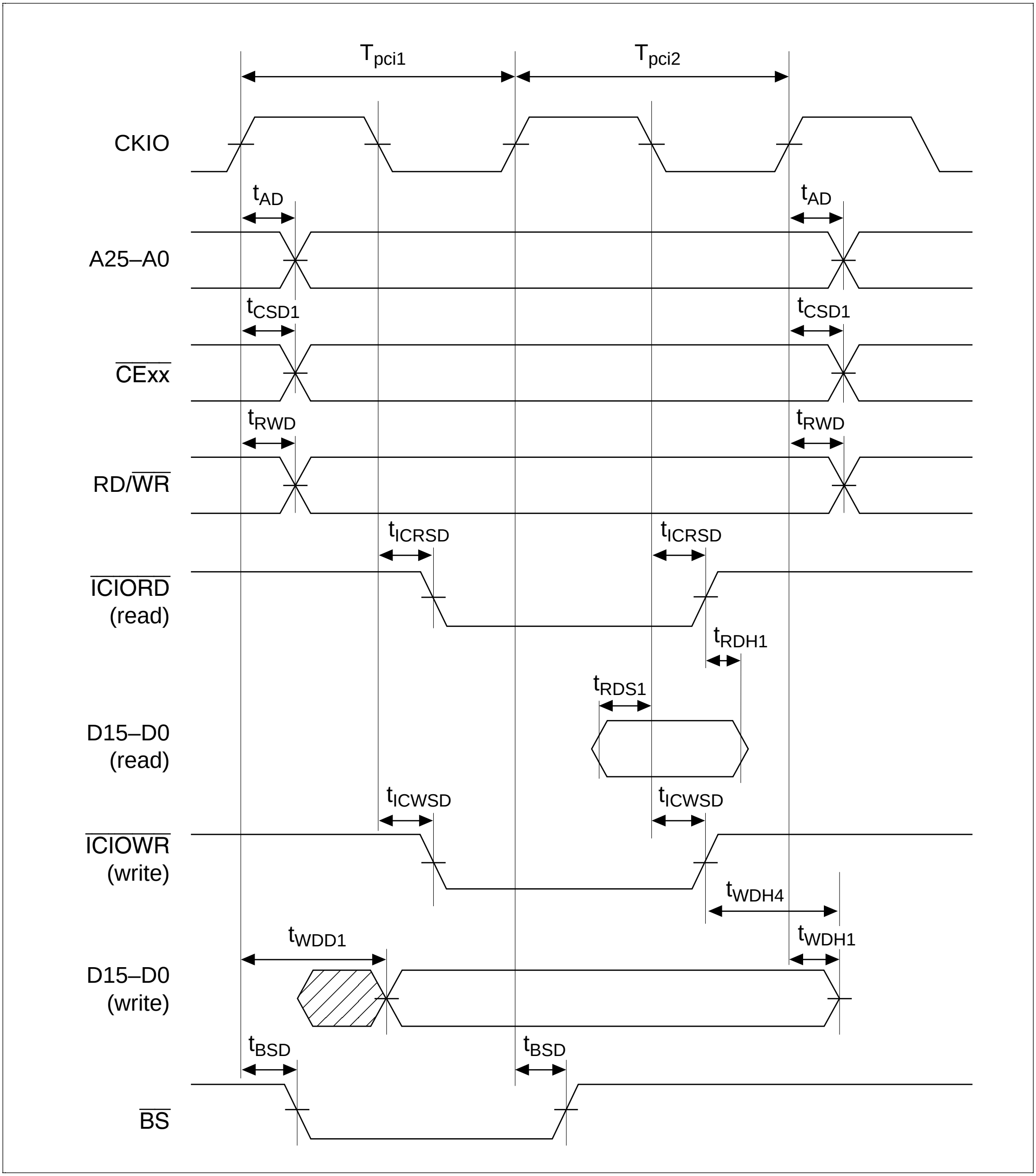


Figure 17.56 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

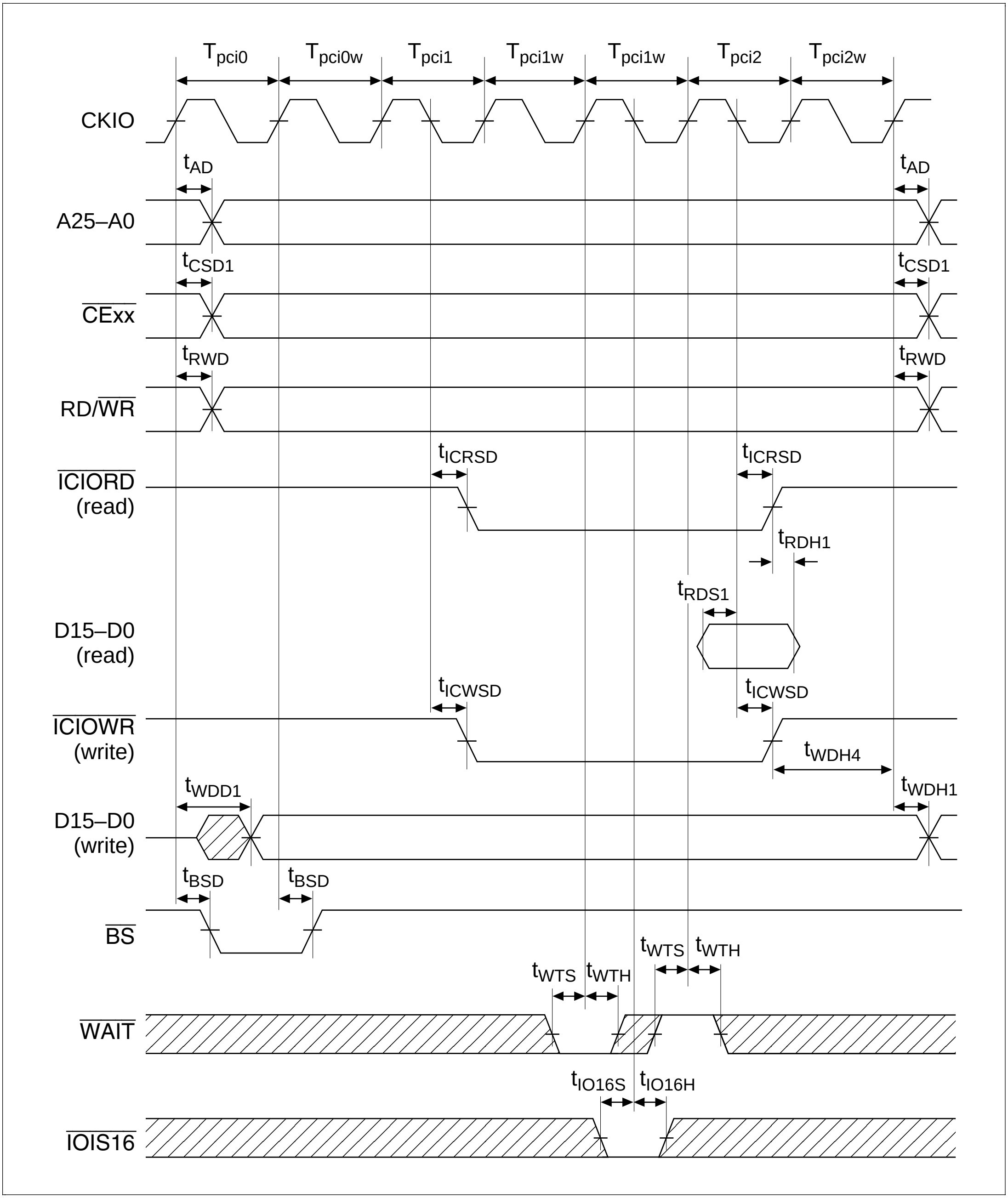


Figure 17.57 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, 1 Wait, External Wait)

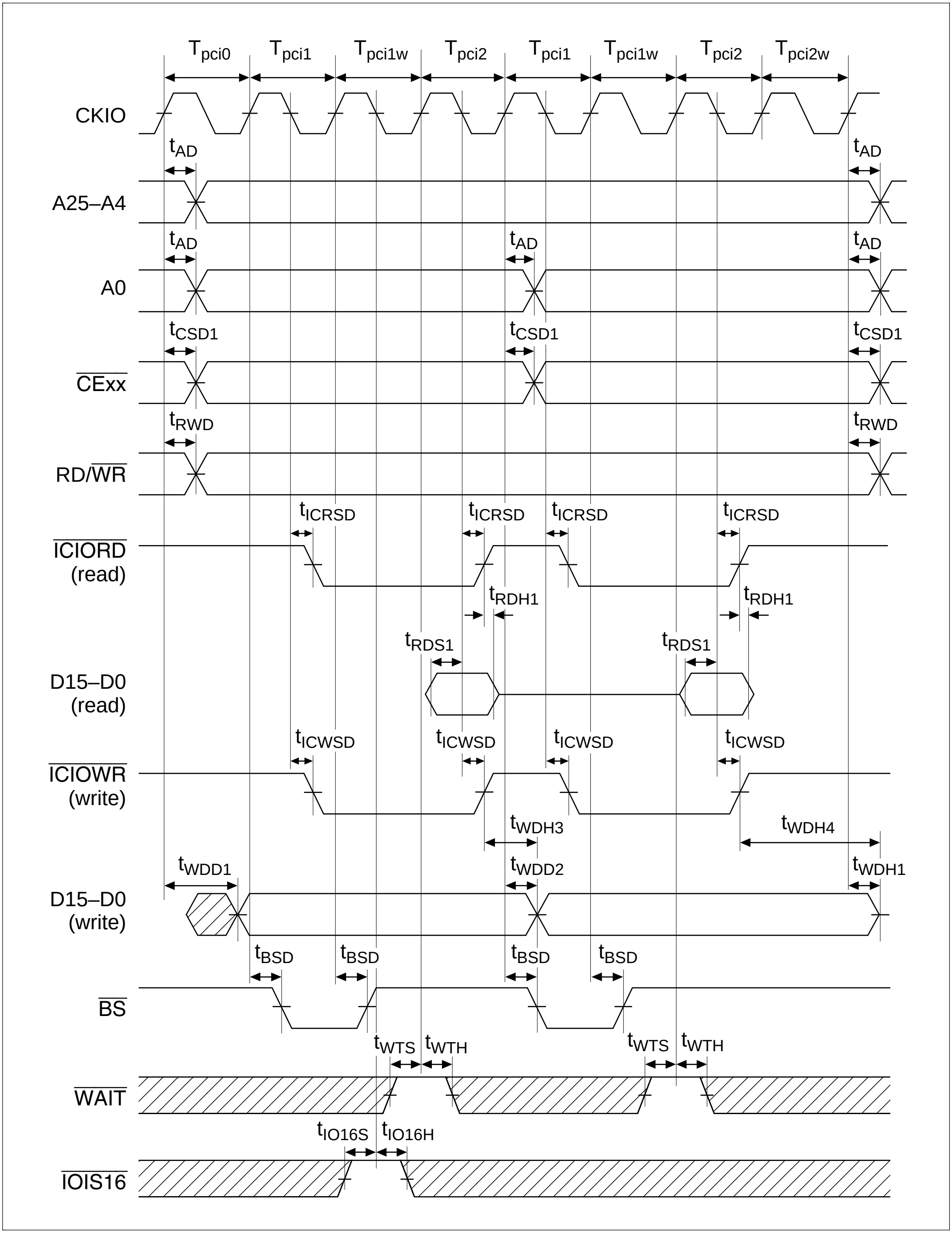


Figure 17.58 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, 1 Wait, Bus Sizing)

17.3.10 Peripheral Module Signal Timing

Table 17.9 Peripheral Module Signal Timing (Conditions: V_{CC} = 3.15–3.6 V, Ta = –20 to 75°C)

				–60			
Module	**Item**		**Symbol**	**Min**	**Max**	**Unit**	**Figure**
TMU, RTC	Timer input setup time		t_{CLKS}	12	—	ns	17.59
	Timer clock input setup time		t_{CKS}	12	—	ns	17.60
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	tcyc	
		Both edges	t_{TCKWL}	2.5	—	tcyc	
	Oscillation settling time		t_{ROSC}	—	3	S	17.61
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	tcyc	17.62–17.63
		synchronous		6	—	tcyc	
	Input clock rise time		t_{SCKr}	—	1.5	tcyc	17.62
	Input clock fall time		t_{SCKf}	—	1.5	tcyc	
	Input clock pulse width		t_{SCKw}	0.4	0.6	tscyc	
	Transmit data delay time		t_{TXD}	—	100	ns	17.63
	Receive data setup time (synchronous)		t_{RXS}	100	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	100	—	ns	
Port	Output data delay time		t_{PORTD}	—	15	ns	17.64
	Input data setup time		t_{PORTS}	12	—	ns	
	Input data hold time		t_{PORTH}	5	—	ns	

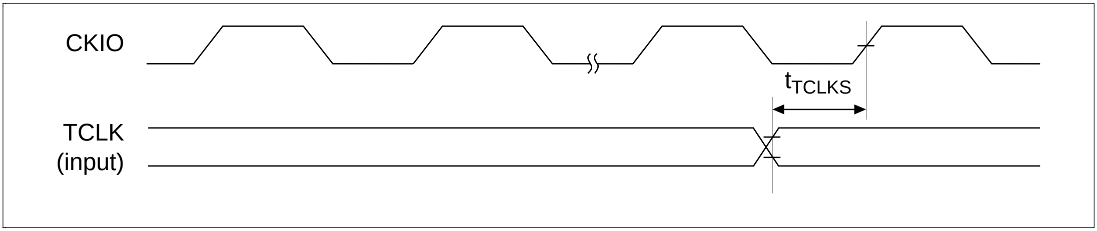


Figure 17.59 TCLK Input Timing (1)

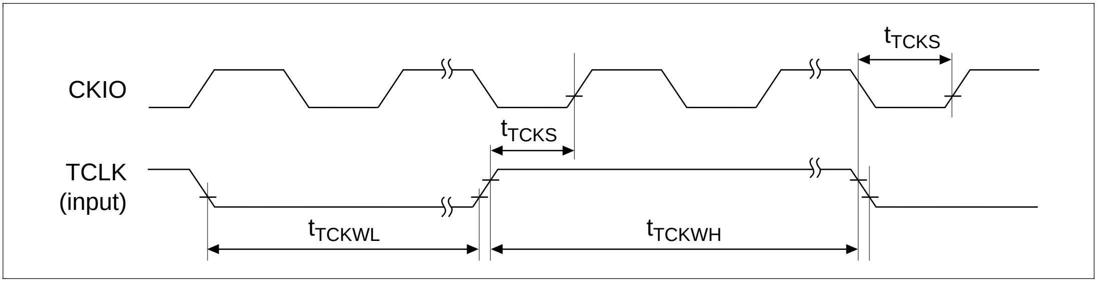


Figure 17.60 TCLK Input Timing (2)

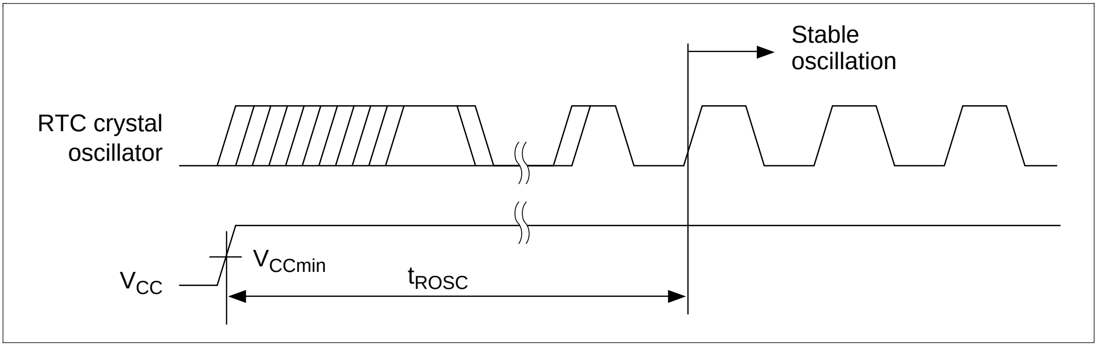


Figure 17.61 RTC Crystal Oscillator Power-On Oscillation Settling Time

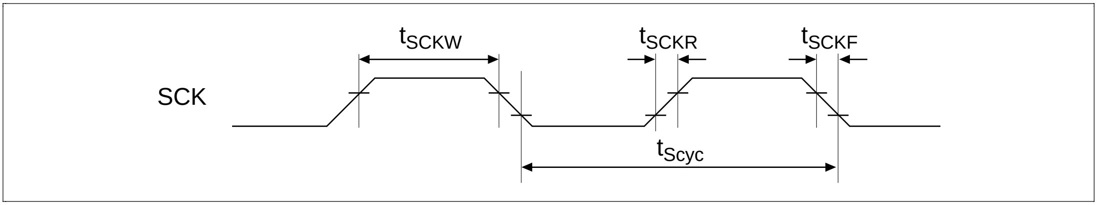


Figure 17.62 SCK Input Clock Timing

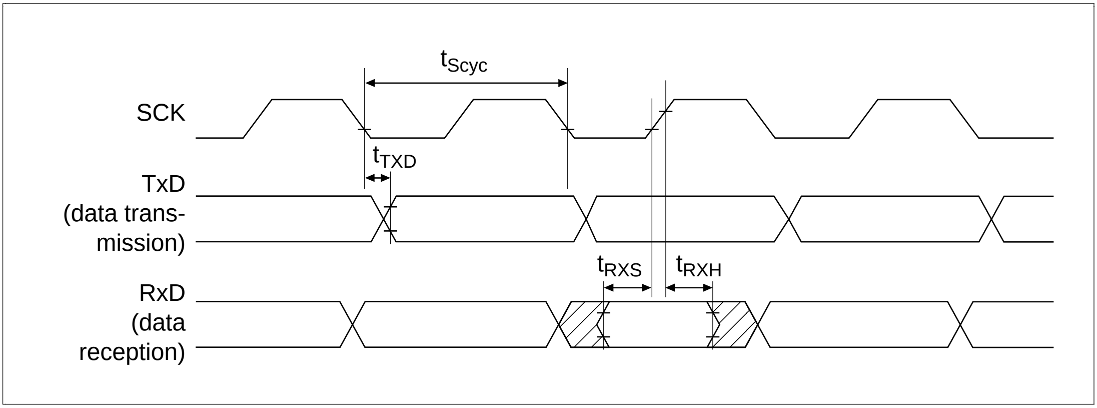


Figure 17.63 Synchronous Mode SCI Input/Output Timing

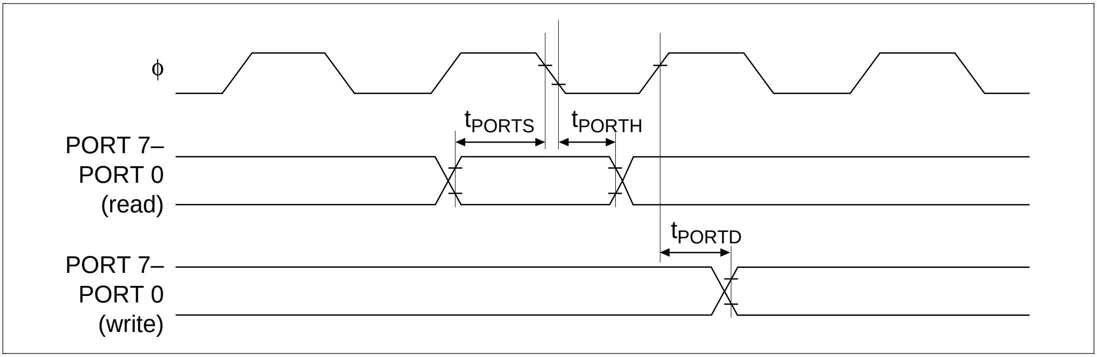
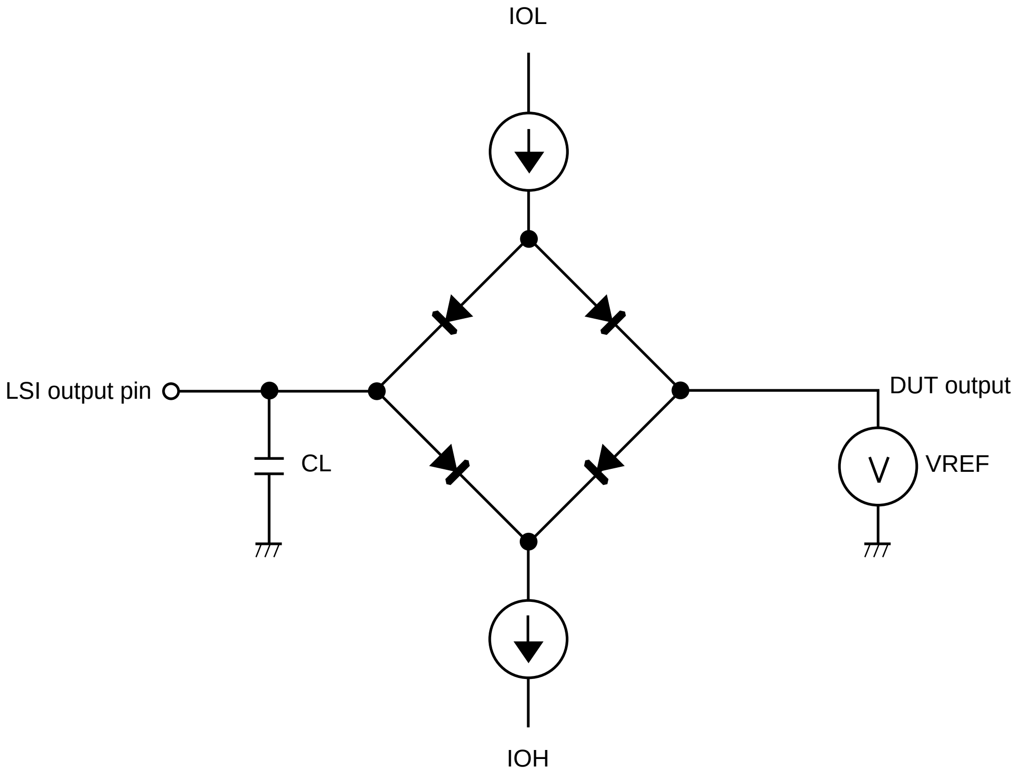


Figure 17.64 I/O Port Input/Output Timing

17.3.11 AC Characteristics Test Conditions

- Input/output signal reference level: 1.5 V (V_{CC} = 3.3–3.6V).
- Input pulse level: V_{SS} to 3.0 V (when $\overline{\text{RESET}}$, $\overline{\text{BREQ}}$, NMI, $\overline{\text{IRL3}}$–$\overline{\text{IRL0}}$, CKIO, MD5–MD0 are V_{SS} to V_{CC}).
- Input rise/fall time: 1 ns



Notes: 1. C_L is the total capacitance including the test jig probe, as shown below:

2. 30 pF: CKIO, $\overline{\text{RAS}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$–$\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
I_{OL} and I_{OH} are the values shown in table 17.3

Figure 17.65 Output Load Circuit

Appendix A Pin Functions

A.1 Pin States

Table A.1 shows pin states during resets, power-down states, and the bus-released state.

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State

Category	Pin	Power-On Reset	Manual Reset	Power-Down State		Bus-Released State
				Standby	Sleep	
Clock	CKIO	IO*1	IO*1	IO*1	IO*1	IO*1
	EXTAL	I*1	I*1	I*1	I*1	I*1
	XTAL	O*1	O*1	O*1	O*1	O*1
	EXTAL2	I	I	I	I	I
	XTAL2	O	O	O	O	O
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
	CA	I	I	I	I	I
	STATUS0, STATUS1	O	O	O	O	O
	MD0/SCK	I	I	I	IO*2	IO*2
	MD1/TXD	I	I	I	IO*3	IO*3
	MD2/RXD	I	I	I	I	I
	MD3/$\overline{\text{CE2A}}$	I	IH*4	IZH*7	IH*4	IZ*4
	MD4/$\overline{\text{CE2B}}$	I	IH*4	IZH*7	IH*4	IZ*4
	MD5/$\overline{\text{RAS2}}$	I	IO*5	IZO*6	IO*5	IZO*6
Interrupt	NMI	I	I	I	I	I
	$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	I	I	I	I	I
	$\overline{\text{IRQOUT}}$	O	O	O	O	O

Table A.1 Pin States during Resets, Power-Down States, and Bus-Released State (cont)

Category	Pin	Power-On Reset	Manual Reset	Power-Down State		Bus-Released State
				Standby	Sleep	
Address Bus	A25 to A0	O	O	Z	O	Z
Data Bus	D31 to D30	Z	ZO[*8]	ZO[*8]	ZO[*8]	ZO[*8]
	D29 to D24	Z	Z	Z	Z	Z
	D23 to D16/ PORT7 to PORT0	Z	ZK[*10]	ZK[*10]	ZK[*10]	ZK[*10]
	D15 to D0	Z	I	Z	I	Z
Bus Control	$\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS4}}$	H	O	ZH[*13]	H	Z
	$\overline{\mathrm{CS5}}$/$\overline{\mathrm{CE1A}}$	H	O	ZH[*13]	H	Z
	$\overline{\mathrm{CS6}}$/$\overline{\mathrm{CE1B}}$	H	O	ZH[*13]	H	Z
	$\overline{\mathrm{BS}}$	H	O	ZH[*13]	H	Z
	$\overline{\mathrm{RAS}}$/$\overline{\mathrm{CE}}$	H	O	ZO[*11]	O	ZO[*11]
	$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{CAS}}$/$\overline{\mathrm{OE}}$	H	O	ZO[*11]	O	ZO[*11]
	$\overline{\mathrm{CASLH}}$	H	O	ZO[*11]	O	ZO[*11]
	$\overline{\mathrm{CASHL}}$/$\overline{\mathrm{CAS2L}}$	H	O	ZO[*11]	O	ZO[*11]
	$\overline{\mathrm{CASHH}}$/$\overline{\mathrm{CAS2H}}$	H	O	ZO[*11]	O	ZO[*11]
	DQMLL/$\overline{\mathrm{WE0}}$	H	O	ZH[*13]	H	Z
	DQMLU/$\overline{\mathrm{WE1}}$	H	O	ZH[*13]	H	Z
	DQMUL/$\overline{\mathrm{WE2}}$/ $\overline{\mathrm{ICIORD}}$	H	O	ZH[*13]	H	Z
	DQMUU/$\overline{\mathrm{WE3}}$/ $\overline{\mathrm{ICIOWR}}$	H	O	ZH[*13]	H	Z
	RD/$\overline{\mathrm{WR}}$	H	O	ZH[*13]	H	Z
	$\overline{\mathrm{RD}}$	H	O	ZH[*13]	H	Z
	CKE	H	O	O	O	O
	$\overline{\mathrm{WAIT}}$	Z	I	Z	I	Z
	$\overline{\mathrm{IOIS16}}$	Z	I	Z	I	Z
TMU/RTC	TCLK	Z	Z	IO[*14]	IO[*12]	IO[*12]
PLL	CAP1, CAP2	IO	IO	IO	IO	IO

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High impedance

K: Input pin is high impedance, output pin holds its state

Notes: 1. Dependent on the clock mode (MD2–MD0 setting).

2. When SCI and port are not used, I. When used, I or O depending on register setting.

3. When SCI and port are not used, I. When used, O.

4. When PCMCIA is not used, I. When used, H or Z.

5. When area 2 DRAM is not used, I. When used, O.

6. When area 2 DRAM is not used, I. When used, Z or O depending on register setting.

7. When PCMCIA is not used, I. When used, Z or H depending on register setting.

8. O when the port function is used.

9. Z when the port function is used.

10. When the port function is used, K depending on register setting.

11. Z or O depending on register setting.

12. I or O depending on register setting.

13. Z or H depending on register setting.

14. In standby mode, I or O depending on register setting. In hardware standby mode, I or L depending on register setting.

A.2 Pin Specifications

Table A.2 shows the pin specifications.

Table A.2 Pin Specifications

Pin	Pin No.	I/O	Function
MD5/$\overline{\text{RAS2}}$	130	I/O	Operating mode pin (endian switching)/RAS (for DRAM). MD signal is fetched in at power-on reset. Switched to $\overline{\text{RAS2}}$ on area 2 DRAM enabling by register.
MD4/$\overline{\text{CE2B}}$	103	I/O	Operating mode pin (area 0 bus width)/PCMCIA $\overline{\text{CE}}$ pin. MD signal is fetched in at power-on reset. Switched to $\overline{\text{CE2B}}$ on area 6 PCMCIA enabling by register.
MD3/$\overline{\text{CE2A}}$	104	I/O	Operating mode pin (area 0 bus width)/PCMCIA $\overline{\text{CE}}$ pin. MD signal is fetched in at power-on reset. Switched to $\overline{\text{CE2A}}$ on area 5 PCMCIA enabling by register.
MD2/RXD	84	I	Operating mode pin/serial data input. MD signal is fetched in at power-on reset. Switched to RXD on SCI enabling by register.
MD1/TXD	85	I/O	Operating mode pin/serial data output. MD signal is fetched in at power-on reset. Switched to TXD on SCI enabling by register.
MD0/SCK	86	I/O	Operating mode pin/serial clock. MD signal is fetched in at power-on reset. Switched to SCK on SCI enabling by register.
STATUS1	97	O	Processor status
STATUS0	98	O	Processor status
A25 to A0	72 to 70, 67 to 61, 58 to 56, 53 to 51, 48 to 43, 40 to 37	O	Address bus
D31 to D24	140 to 143, 1 to 4	I/O	Data bus
D23 to D16/ Port 7 to Port 0	5, 8 to 14	I/O	Data bus / I/O port

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
D15 to D0	15 to 16, 21 to 29, 32 to 36	I/O	Data bus
$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	108	O	Chip select 6/PCMCIA $\overline{\text{CE}}$
$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	109	O	Chip select 5/PCMCIA $\overline{\text{CE}}$
$\overline{\text{CS4}}$ to $\overline{\text{CS0}}$	110 to 114	O	Chip select 4—chip select 0
$\overline{\text{BS}}$	105	O	Bus cycle start
$\overline{\text{RAS}}$/$\overline{\text{CE}}$	129	O	DRAM, synchronous DRAM $\overline{\text{RAS}}$/pseudo-SRAM $\overline{\text{CE}}$
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$	119	O	D31–D24 (DRAM $\overline{\text{CAS}}$)/D15–D8 (area 2 DRAM $\overline{\text{CAS}}$) select signal
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$	120	O	D23-D16 (DRAM $\overline{\text{CAS}}$)/D7–D0 (area 2 DRAM $\overline{\text{CAS}}$) select signal
$\overline{\text{CASLH}}$	125	O	D15–D8 select signal (DRAM $\overline{\text{CAS}}$)
$\overline{\text{CASLL}}$/$\overline{\text{CAS}}$/$\overline{\text{OE}}$	126	O	D7–D0 select (DRAM $\overline{\text{CAS}}$)/memory select signal (synchronous DRAM $\overline{\text{CAS}}$/pseudo-SRAM $\overline{\text{OE}}$)
$\overline{\text{WE3}}$/DQMUU/ $\overline{\text{ICIOWR}}$	117	O	D31–D24 select signal (normal memory, pseudo-SRAM $\overline{\text{WE}}$/synchronous DRAM DQM)/IO write (PCMCIA, PCMCIB)
$\overline{\text{WE2/DQMUL/}}$ $\overline{\text{ICIORD}}$	118	O	D23–D16 select signal (normal memory, pseudo-SRAM $\overline{\text{WE}}$/synchronous DRAM DQM)/IO write (PCMCIA, PCMCIB)
$\overline{\text{WE1}}$/DQMLU	123	O	D15–D8 select signal (normal memory, pseudo-SRAM WE/synchronous DRAM DQM)
$\overline{\text{WE0}}$/DQMLL	124	O	D7–D0 select signal (normal memory, pseudo-SRAM WE/synchronous DRAM DQM)
RD/$\overline{\text{WR}}$	106	O	Read/write (synchronous DRAM/DRAM/PCMCIA)
$\overline{\text{RD}}$	107	O	Read pulse (PCMCIA/normal memory)
$\overline{\text{WAIT}}$	132	I	Hardware wait request
$\overline{\text{IOIS16}}$	94	I	IO16 bit indication (PCMCIA IO area)
$\overline{\text{BREQ}}$	87	I	Bus request
$\overline{\text{BACK}}$	96	O	Bus acknowledge
$\overline{\text{IRQOUT}}$	95	O	Interrupt request notification
$\overline{\text{RESET}}$	88	I	Reset
CA	81	I	Chip active Causes a transition to hardware standby mode when low. Drive high in a power-on reset.

Table A.2 Pin Specifications (cont)

Pin	Pin No.	I/O	Function
NMI	89	I	Nonmaskable interrupt request
IRL3 to IRL0	90 to 93	I	External interrupt source input
TCLK	134	I/O	Timer external clock input/RTC clock output
EXTAL	79	I	External clock/crystal resonator pin
XTAL	80	O	Crystal resonator pin
CAP1	74	O	External capacitance pin (for PLL1)
CAP2	77	O	External capacitance pin (for PLL1)
CKIO	101	I/O	System clock input/output
CKE	131	O	Clock enable control (for synchronous DRAM)
XTAL2	136	O	Crystal resonator pin (for on-chip RTC)
EXTAL2	137	I	Crystal resonator pin (for on-chip RTC)
NC	99	O	Leave unconnected
V_{CC}	7, 18, 20, 31, 42, 50, 55, 60, 69, 81, 83, 102, 116, 122, 128, 139	Power supply	Power supply (3.3 V)
V_{CC} (RTC)	135	Power supply	RTC oscillator power supply (3.3 V)
V_{CC} (PLL)	75, 78	Power supply	PLL power supply (3.3 V)
V_{SS}	6, 17, 19, 30, 41, 49, 54, 59, 68, 82, 100, 115, 121, 127, 133, 144	Power supply	Power supply (0 V)
V_{SS} (RTC)	138	Power supply	RTC oscillator power supply (0 V)
V_{SS} (PLL)	73, 76	Power supply	PLL power supply (0 V)

Note: Except in hardware standby mode, power must be supplied constantly to all power supply pins. In hardware standby mode, power should be supplied at least to the RTC power supply pins.

A.3 Handling of Unused Pins

- When RTC is not used
  - — EXTAL2: Pull up
  - — XTAL2: Leave unconnected
  - — V_{CC} (RTC): Power supply (3.3 V)
  - — V_{SS} (RTC): 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

A.4 Pin States in Access to Each Address Space

Table A.3 Pin States (Normal Memory/Little-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High
	W	Low	Low	High	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High
	W	High	High	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High
	W	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High
	W	High	High	High	High
MD3/$\overline{\text{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]
D31 to D24		High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]

Table A.3 Pin States (Normal Memory/Little-Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}
MD4/$\overline{\text{CE2B}}$		High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}
MD5/$\overline{\text{RAS2}}$		High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.3 Pin States (Normal Memory/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.

4. When WCR2 register wait setting is 0, disabled.

5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.

6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.4 Pin States (Normal Memory/Big-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High
	W	Low	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High
	W	High	Low	High	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High
	W	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High
	W	High	High	High	High
MD3/$\overline{\text{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\text{CE2B}}$		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/$\overline{\text{RAS2}}$		High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*4	Enabled*4	Enabled*4	Enabled*4
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z*5	High-Z*5	High-Z*5	High-Z*5
D31 to D24		High-Z*6	High-Z*6	High-Z*6	High-Z*6

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
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\text{CE2B}}$		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/$\overline{\text{RAS2}}$		High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.4 Pin States (Normal Memory/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.

4. When WCR2 register wait setting is 0, disabled.

5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register .

6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

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
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High
	W	—	—	—	—
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High
	W	—	—	—	—
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High
	W	—	—	—	—
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High
	W	—	—	—	—
MD3/$\overline{\text{CE2A}}$		High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}
MD4/$\overline{\text{CE2B}}$		High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}
MD5/$\overline{\text{RAS2}}$		High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z^{*5}	High-Z^{*5}	High-Z^{*5}	High-Z^{*5}
D31 to D24		High-Z^{*6}	High-Z^{*6}	High-Z^{*6}	High-Z^{*6}

Table A.5 Pin States (Burst ROM/Little-Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
MD3/$\overline{\text{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.5 Pin States (Burst ROM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes:
1. When BCR1.A5PCM = 0, high-Z.
2. When BCR1.A6PCM = 0, high-Z.
3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.
4. When WCR2 register wait setting is 0, disabled.
5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.
6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.6 Pin States (Burst ROM/Big-Endian)

		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
CS6 to CS0		Enabled	Enabled	Enabled	Enabled
RD	R	Low	Low	Low	Low
	W	—	—	—	—
RD/WR	R	High	High	High	High
	W	—	—	—	—
BS		Enabled	Enabled	Enabled	Enabled
RAS/CE		High	High	High	High
CAS/CASLL/OE		High	High	High	High
CASLH		High	High	High	High
CASHL/CAS2L		High	High	High	High
CASHH/CAS2H		High	High	High	High
DQMLL/WE0	R	High	High	High	High
	W	—	—	—	—
DQMLU/WE1	R	High	High	High	High
	W	—	—	—	—
DQMUL/WE2/ICIORD	R	High	High	High	High
	W	—	—	—	—
DQMUU/WE3/ICIOWR	R	High	High	High	High
	W	—	—	—	—
MD3/CE2A		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/CE2B		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/RAS2		High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3
CKE		Disabled	Disabled	Disabled	Disabled
WAIT		Enabled*4	Enabled*4	Enabled*4	Enabled*4
IOIS16		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z*5	High-Z*5	High-Z*5	High-Z*5
D31 to D24		High-Z*6	High-Z*6	High-Z*6	High-Z*6

Table A.6 Pin States (Burst ROM/Big-Endian) (cont)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
MD3/$\overline{\text{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.6 Pin States (Burst ROM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.
3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.
4. When WCR2 register wait setting is 0, disabled.
5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register .
6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.7 Pin States (DRAM/Little-Endian)

		16-Bit Bus Width (Area 3)			16-Bit Bus Width (Area 2)		
Pin		Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		Low	Low	Low	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		Low	High	Low	High	High	High
$\overline{\text{CASLH}}$		High	Low	Low	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	Low	High	Low
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	Low	Low
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
MD3/$\overline{\text{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\text{CE2B}}$		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/$\overline{\text{RAS2}}$		High-Z or high*3	High-Z or high*3	High-Z or high*3	Low	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data

Table A.7 Pin States (DRAM/Little-Endian) (cont)

	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
D23 to D16/ PORT7 to PORT0	High-Z*4	High-Z*4	High-Z*4	High-Z*4	High-Z*4	High-Z*4
D31 to D24	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5

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
$\overline{\mathrm{CS6}}$ to $\overline{\mathrm{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\mathrm{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RAS}}$/$\overline{\mathrm{CE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{CAS}}$/$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{OE}}$		Low	High	High	High	Low	High	Low
$\overline{\mathrm{CASLH}}$		High	Low	High	High	Low	High	Low
$\overline{\mathrm{CASHL}}$/$\overline{\mathrm{CAS2L}}$		High	High	Low	High	High	Low	Low
$\overline{\mathrm{CASHH}}$/$\overline{\mathrm{CAS2H}}$		High	High	High	Low	High	Low	Low
DQMLL/$\overline{\mathrm{WE0}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMLU/$\overline{\mathrm{WE1}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMUL/$\overline{\mathrm{WE2}}$/$\overline{\mathrm{ICIORD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMUU/$\overline{\mathrm{WE3}}$/$\overline{\mathrm{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
MD3/$\overline{\mathrm{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\mathrm{CE2B}}$		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/$\overline{\mathrm{RAS2}}$		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.7 Pin States (DRAM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.
3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.
4. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.
5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

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
CS6 to CS0		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RD	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
RD/WR	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RAS/CE		Low	Low	Low	High	High	High
CAS/CASLL/OE		High	Low	Low	High	High	High
CASLH		Low	High	Low	High	High	High
CASHL/CAS2L		High	High	High	High	Low	Low
CASHH/CAS2H		High	High	High	Low	High	Low
DQMLL/WE0	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMLU/WE1	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMUL/WE2/ICIORD	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
DQMUU/WE3/ICIOWR	R	High	High	High	High	High	High
	W	High	High	High	High	High	High
MD3/CE2A		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/CE2B		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/RAS2		High-Z or high*3	High-Z or high*3	High-Z or high*3	Low	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
WAIT		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
IOIS16		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		Valid data	Invalid data	Valid data	Valid data	Invalid data	Valid data

Table A.8 Pin States (DRAM/Big-Endian) (cont)

	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**
D23 to D16/ PORT7 to PORT0	High-Z[*4]	High-Z[*4]	High-Z[*4]	High-Z[*4]	High-Z[*4]	High-Z[*4]
D31 to D24	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]

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
CS6 to CS0		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RD	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/WR	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RAS/CE		Low	Low	Low	Low	Low	Low	Low
CAS/CASLL/OE		High	High	High	Low	High	Low	Low
CASLH		High	High	Low	High	High	Low	Low
CASHL/CAS2L		High	Low	High	High	Low	High	Low
CASHH/CAS2H		Low	High	High	High	Low	High	Low
DQMLL/WE0	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMLU/WE1	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMUL/WE2/ICIORD	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
DQMUU/WE3/ICIOWR	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
MD3/CE2A		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/CE2B		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/RAS2		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
WAIT		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
IOIS16		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.8 Pin States (DRAM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes:
1. When BCR1.A5PCM = 0, high-Z.
2. When BCR1.A6PCM = 0, high-Z.
3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.
4. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.
5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.9 Pin States (Synchronous DRAM/Little-Endian)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
DQMLU/$\overline{\text{WE1}}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\text{CE2B}}$		High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2	High-Z or high*2
MD5/$\overline{\text{RAS2}}$		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		High*3	High*3	High*3	High*3	High*3	High*3	High*3
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address, command	Address, command	Address, command	Address, command	Address, command	Address, command	Address, command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.9 Pin States (Synchronous DRAM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. Normally high. Low in self-refreshing.

Table A.10 Pin States (Synchronous DRAM/Big-Endian)

		32-Bit Bus Width						
Pin		**Byte Access (Address 4n)**	**Byte Access (Address 4n + 1)**	**Byte Access (Address 4n + 2)**	**Byte Access (Address 4n + 3)**	**Word Access (Address 4n)**	**Word Access (Address 4n + 2)**	**Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}
MD4/$\overline{\text{CE2B}}$		High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}	High-Z or high^{*2}
MD5/$\overline{\text{RAS2}}$		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		High^{*3}	High^{*3}	High^{*3}	High^{*3}	High^{*3}	High^{*3}	High^{*3}
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address, command	Address, command	Address, command	Address, command	Address, command	Address, command	Address, command
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.10 Pin States (Synchronous DRAM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. Normally high. Low in self-refreshing.

Table A.11 Pin States (Pseudo-SRAM/Little-Endian)

		16-Bit Bus Width		
Pin		**Byte Access (Address 2n)**	**Byte Access (Address 2n + 1)**	**Word/Longword Access**
$\overline{\mathrm{CS6}}$ to $\overline{\mathrm{CS0}}$		Enabled	Enabled	Enabled
$\overline{\mathrm{RD}}$	R	High	High	High
	W	High	High	High
RD/$\overline{\mathrm{WR}}$	R	High	High	High
	W	Low	Low	Low
$\overline{\mathrm{BS}}$		Enabled	Enabled	Enabled
$\overline{\mathrm{RAS}}$/$\overline{\mathrm{CE}}$		Low	Low	Low
$\overline{\mathrm{CAS}}$/$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{OE}}$		Low	Low	Low
$\overline{\mathrm{CASLH}}$		High	High	High
$\overline{\mathrm{CASHL}}$/$\overline{\mathrm{CAS2L}}$		High	High	High
$\overline{\mathrm{CASHH}}$/$\overline{\mathrm{CAS2H}}$		High	High	High
DQMLL/$\overline{\mathrm{WE0}}$	R	High	High	High
	W	Low	High	Low
DQMLU/$\overline{\mathrm{WE1}}$	R	High	High	High
	W	High	Low	Low
DQMUL/$\overline{\mathrm{WE2}}$/$\overline{\mathrm{ICIORD}}$	R	High	High	High
	W	High	High	High
DQMUU/$\overline{\mathrm{WE3}}$/$\overline{\mathrm{ICIOWR}}$	R	High	High	High
	W	High	High	High
MD3/$\overline{\mathrm{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\mathrm{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\mathrm{RAS2}}$		High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled
$\overline{\mathrm{WAIT}}$		Disabled	Disabled	Disabled
$\overline{\mathrm{IOIS16}}$		Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z[*3]	High-Z[*3]	High-Z[*3]
D31 to D24		High-Z[*4]	High-Z[*4]	High-Z[*4]

Table A.11 Pin States (Pseudo-SRAM/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**
$\overline{\mathrm{CS6}}$ to $\overline{\mathrm{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\mathrm{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\mathrm{RAS}}$/$\overline{\mathrm{CE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{CAS}}$/$\overline{\mathrm{CASLL}}$/$\overline{\mathrm{OE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\mathrm{CASLH}}$		High	High	High	High	High	High	High
$\overline{\mathrm{CASHL}}$/$\overline{\mathrm{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\mathrm{CASHH}}$/$\overline{\mathrm{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\mathrm{WE0}}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
DQMLU/$\overline{\mathrm{WE1}}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
DQMUL/$\overline{\mathrm{WE2}}$/$\overline{\mathrm{ICIORD}}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
DQMUU/$\overline{\mathrm{WE3}}$/$\overline{\mathrm{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
MD3/$\overline{\mathrm{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\mathrm{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\mathrm{RAS2}}$		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\mathrm{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Table A.11 Pin States (Pseudo-SRAM/Little-Endian) (cont)

	32-Bit Bus Width						
Pin	Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
D15 to D8	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.

4. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.12 Pin States (Pseudo-SRAM/Big-Endian)

		16-Bit Bus Width		
Pin		**Byte Access (Address 2n)**	**Byte Access (Address 2n + 1)**	**Word/Longword Access**
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High
	W	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High
	W	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		Low	Low	Low
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		Low	Low	Low
$\overline{\text{CASLH}}$		High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High
	W	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High
	W	Low	High	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High
	W	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High
	W	High	High	High
MD3/$\overline{\text{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address
D7 to D0		Invalid data	Valid data	Valid data
D15 to D8		Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z[*3]	High-Z[*3]	High-Z[*3]
D31 to D24		High-Z[*4]	High-Z[*4]	High-Z[*4]

Table A.12 Pin States (Pseudo-SRAM/Big-Endian) (cont)

		32-Bit Bus Width						
Pin		Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RD}}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		Low	Low	Low	Low	Low	Low	Low
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]	High-Z or high[*1]
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Table A.12 Pin States (Pseudo-SRAM/Big-Endian) (cont)

	32-Bit Bus Width						
Pin	Byte Access (Address 4n)	Byte Access (Address 4n + 1)	Byte Access (Address 4n + 2)	Byte Access (Address 4n + 3)	Word Access (Address 4n)	Word Access (Address 4n + 2)	Longword Access
D15 to D8	Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24	Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.

4. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.13 Pin States (PCMCIA/Little-Endian)

		PCMCIA Memory Interface (Area 5)			
		8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/Word/Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High
	W	High	High	High	High
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High
	W	Low	Low	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High
	W	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High
	W	High	High	High	High
MD3/$\overline{\text{CE2A}}$		High	High	Low	Low
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]
D31 to D24		High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]

Table A.13 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/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1	High-Z or high*1
MD4/$\overline{\text{CE2B}}$		High	High	Low	Low	High	High	Low	Low
MD5/$\overline{\text{RAS2}}$		High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3	High-Z or high*3
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.13 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/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access	Byte/Word/Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/Longword Access
D7 to D0	Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8	High-Z	Invalid data	Valid data	Valid data	High-Z	Invalid data	Valid data	Valid data
D23 to D16/ PORT7 to PORT0	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]
D31 to D24	High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]

Notes:
1. When BCR1.A5PCM = 0, high-Z.
2. When BCR1.A6PCM = 0, high-Z.
3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.
4. When WCR2 register wait setting is 0, disabled.
5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.
6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

Table A.14 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
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High
	W	High	High	High	High
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High
	W	Low	Low	Low	Low
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High
	W	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High
	W	High	High	High	High
MD3/$\overline{\text{CE2A}}$		High	High	Low	Low
MD4/$\overline{\text{CE2B}}$		High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]	High-Z or high[*2]
MD5/$\overline{\text{RAS2}}$		High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]	High-Z or high[*3]
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0		High-Z[*5]	High-Z[*5]	High-Z[*5]	High-Z[*5]
D31 to D24		High-Z[*6]	High-Z[*6]	High-Z[*6]	High-Z[*6]

Table A.14 Pin States (PCMCIA/BIG-Endian) (cont)

		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
Pin		Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access	Byte/ Word/ Long-word Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Long-word Access
$\overline{\text{CS6}}$ to $\overline{\text{CS0}}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{\text{RD}}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{\text{WR}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{\text{BS}}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{\text{RAS}}$/$\overline{\text{CE}}$		High	High	High	High	High	High	High	High
$\overline{\text{CAS}}$/$\overline{\text{CASLL}}$/$\overline{\text{OE}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASLH}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASHL}}$/$\overline{\text{CAS2L}}$		High	High	High	High	High	High	High	High
$\overline{\text{CASHH}}$/$\overline{\text{CAS2H}}$		High	High	High	High	High	High	High	High
DQMLL/$\overline{\text{WE0}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
DQMLU/$\overline{\text{WE1}}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
DQMUL/$\overline{\text{WE2}}$/$\overline{\text{ICIORD}}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
DQMUU/$\overline{\text{WE3}}$/$\overline{\text{ICIOWR}}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
MD3/$\overline{\text{CE2A}}$		High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}	High-Z or high^{*1}
MD4/$\overline{\text{CE2B}}$		High	High	Low	Low	High	High	Low	Low
MD5/$\overline{\text{RAS2}}$		High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}	High-Z or high^{*3}
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{\text{WAIT}}$		Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}	Enabled^{*4}
$\overline{\text{IOIS16}}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address

Table A.14 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/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n + 1)	Word/ Longword Access
D7 to D0	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8	High-Z	Valid data	Invalid data	Valid data	High-Z	Valid data	Invalid data	Valid data
D23 to D16/ PORT7 to PORT0	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5	High-Z*5
D31 to D24	High-Z*6	High-Z*6	High-Z*6	High-Z*6	High-Z*6	High-Z*6	High-Z*6	High-Z*6

Notes: 1. When BCR1.A5PCM = 0, high-Z.

2. When BCR1.A6PCM = 0, high-Z.

3. When BCR1.DRAMTP (2–0) ≠ 101, high-Z.

4. When WCR2 register wait setting is 0, disabled.

5. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, dependent on PCTR register.

6. When BCR2.PORTEN = 0, high-Z. When BCR2.PORTEN = 1, D31 and D30 only data output.

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:	Realtime clock
INTC:	Interrupt controller
TMU:	Timer unit
SCI:	Serial communication interface controller
CAC:	Cache

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 and cache are connected.
P: Peripheral bus, to which the BSC and peripheral modules (RTC, INTC, TMU, and SCI) 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.

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	CAC	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	R: 8, W: 16
Watchdog timer control/status register	WTCSR	CPG	S	H'FFFFFF86	8	R: 8, W: 16
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 counter	RTCOR	BSC	C	H'FFFFFF72	16	16
Refresh count register	RFCR	BSC	C	H'FFFFFF74	16	16
Port control register	PCTR	BSC	C	H'FFFFFF76	16	16
Port data register	PDTR	BSC	C	H'FFFFFF78	8	8
Serial port register	SCSPTR	SCI	P	H'FFFFFF7C	8	8
SDRAM mode register	SDMR	BSC	C	H'FFFFD000	8	8
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
Smartcard mode register	SCSCMR	SCI	P	H'FFFFFE8C	8	8

B.2 Register Bit List

A register bit list is shown in table B.2

Table B.2 Register Bit List

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SDMR	—	—	—	—					BSC
SCSMR	C/$\overline{\text{A}}$	CHR	PE	O/$\overline{\text{E}}$	STOP	MP	CKS1	CKS0	SCI
SCBRR									SCI
SCSCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI
SCTSR									SCI
SCTDR									SCI
SCSSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI
SCRSR									SCI
SCRDR									SCI
SCSCMR	—	—	—	—	SDIR	SINV	—	SMIF	SCI
TOCR	—	—	—	—	—	—	—	TCOE	TMU
TSTR	—	—	—	—	—	STR2	STR1	STR0	TMU
TCOR0									TMU
TCNT0									TMU
TCR0	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR1									TMU
TCNT1									TMU

Table B.2 Register Bit List (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR1	—	—	—	—	—	—	—	UNF	TMU
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR2									TMU
TCNT2									TMU
TCR2	—	—	—	—	—	—	ICPF	UNF	TMU
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCPR2									TMU
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	RTC
RSECCNT	—	10 seconds	10 seconds	10 seconds	1 second	1 second	1 second	1 second	RTC
RMINCNT	—	10 minutes	10 minutes	10 minutes	1 minute	1 minute	1 minute	1 minute	RTC
RHRCNT	—	—	10 hours	10 hours	1 hours	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 seconds	10 seconds	10 seconds	1 second	1 second	1 second	1 second	RTC
RMINAR	ENB	10 minutes	10 minutes	10 minutes	1 minute	1 minute	1 minute	1 minute	RTC
RHRAR	ENB	—	10 hours	10 hours	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

Table B.2 Register Bit List (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RCR1	CF	—	—	CIE	AIE	—	—	AF	RTC
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	RTC
ICR	NMIL	—	—	—	—	—	—	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	—	—	—	—	
BCR1	—	—	HIZMEM*	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	A1SZ1	A1SZ0	—	PORTEN	
WCR1	—	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0	BSC
	A3IW1	A3IW0	A2IW1	A2IW0	A1IW1	A1IW0	A0IW1	A0IW0	
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	BSC
	A4W0	A3W1	A3W0	A1-2W1	A1-2W0	A0W2	A0W1	A0W0	
MCR	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	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
PCTR	PB7PUP	PB7IO	PB6PUP	PB6IO	PB5PUP	PB5IO	PB4PUP	PB4IO	I/O
	PB3PUP	PB3IO	PB2PUP	PB2IO	PB1PUP	PB1IO	PB0PUP	PB0IO	
PDTR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	I/O
SCSPTR	—	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	I/O

Table B.2 Register Bit List (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FRQCR	STC2	IFC2	PFC2	—	—	—	—	CKOEN	CPG
	PLLEN	PSTBY	STC1	STC0	IFC1	IFC0	PFC1	PFC0	
STBCR	STBY	—	—	—	—	MSTP2	MSTP1	MSTP0	Power-down states
WTCNT									CPG
WTCSR	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	CPG
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24	UBC
	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16	
	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8	
	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0	
BRCR	CMFA	CMFB	—	—	—	PCBA	—	—	UBC
	DBEB	PCBB	—	—	SEQ	—	—	—	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	UBC
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	—	—	—	—	—	BASMB	BAMB1	BAMB0	UBC
BBRB	—	—	—	—	—	—	—	—	UBC
	—	—	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	UBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	—	—	—	—	—	BASMA	BAMA1	BAMA0	UBC
BBRA	—	—	—	—	—	—	—	—	UBC
	—	—	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	

Table B.2 Register Bit List (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TRA	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—			
							—	—	
EXPEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
INTEVT	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—					
MMUCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	SV	
	—	—	RC	RC	—	TF	IX	AT	
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0	UBC
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0	UBC
CCR	—	—	—	—	—	—	—	—	CCN
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	RA	0	CF	CB	WT	CE	
PTEH									CCN
							—	—	
PTEL									CCN
							—	V	
	—	PR	PR	SZ	C	D	SH	—	
TTB									CCN

Table B.2 Register Bit List (cont)

Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TEA									CCN

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

B.3 Register States in Reset and Power-Down States

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	ICR	H'8000/H'0000*3	H'8000/H'0000*3	Held	Held
	IPRA	H'0000	H'0000	Held	Held
	IPRB	H'0000	H'0000	Held	Held
	TRA	Undefined	Undefined	Held	Held
	EXPEVT	H'00000000	H'00000020	Held	Held
	INTEVT	Undefined	Undefined	Held	Held
UBC	BARA	Undefined	Held	Held	Held
	BASRA	Undefined	Held	Held	Held
	BAMRA	Undefined	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
UBC	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
	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'3FFC	Held	Held	Held
	WCR1	H'3FFF	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
	RTCNT	H'0000	Runs	Held	Runs
	RTCOR	H'0000	Held	Held	Held
	RFCR	H'0000	Runs	Held	Runs
	PCTR	H'0000	Held	Held	Held
	PDTR	Undefined	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**
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
	SCSPTR	Initialized*9	Held	Held	Held
	SCSCMR	Initialized*11	Initialized*11	Initialized*11	Initialized*11

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. Bits 2 and 0 are undefined, other bits = 0
10. Held when SCI is operating, initialized when SCI is not used.
11. Bits 0, 2, and 3 are cleared, other bits are undefined.

Appendix C Delay Time Variation Due to Load Capacitance

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated is connected to the SH7708 Series' pins. The graph shown in figure C.1 should be taken into consideration if the stipulated capacitance is exceeded in connecting an external device.

The graph will not be linear if the connected load capacitance exceeds the range shown in figure C.1.

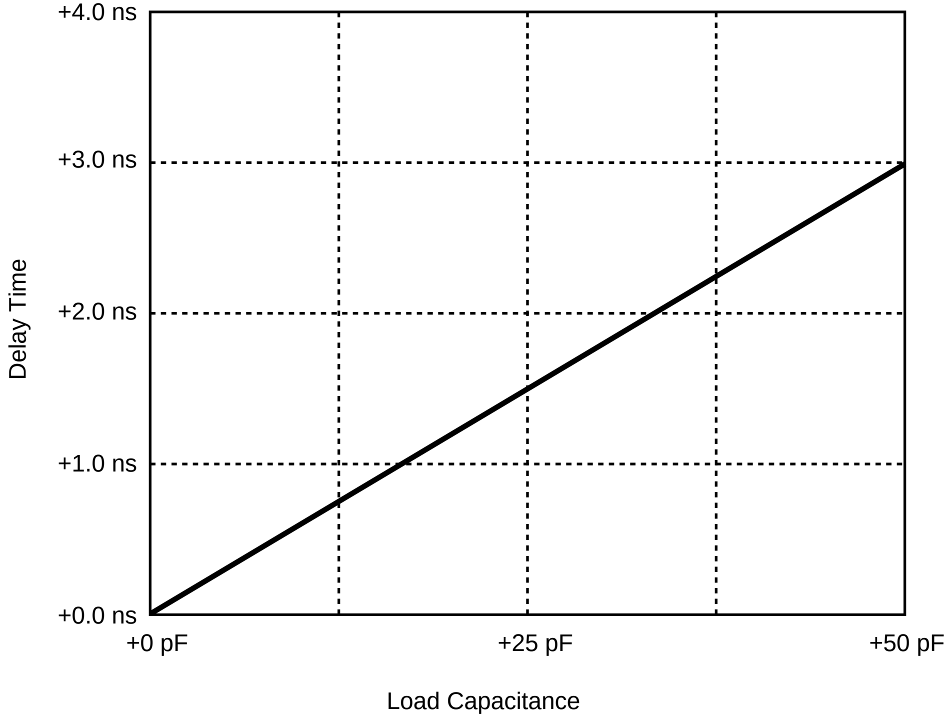
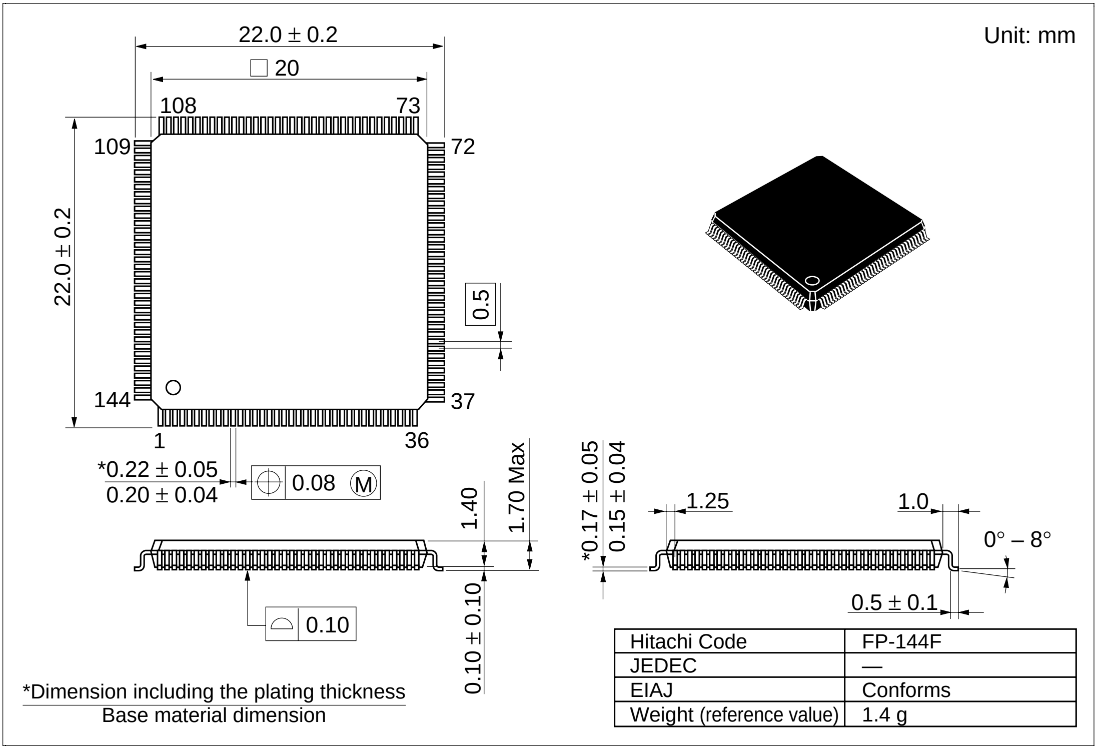


Figure C.1 Load Capacitance vs. Delay Time

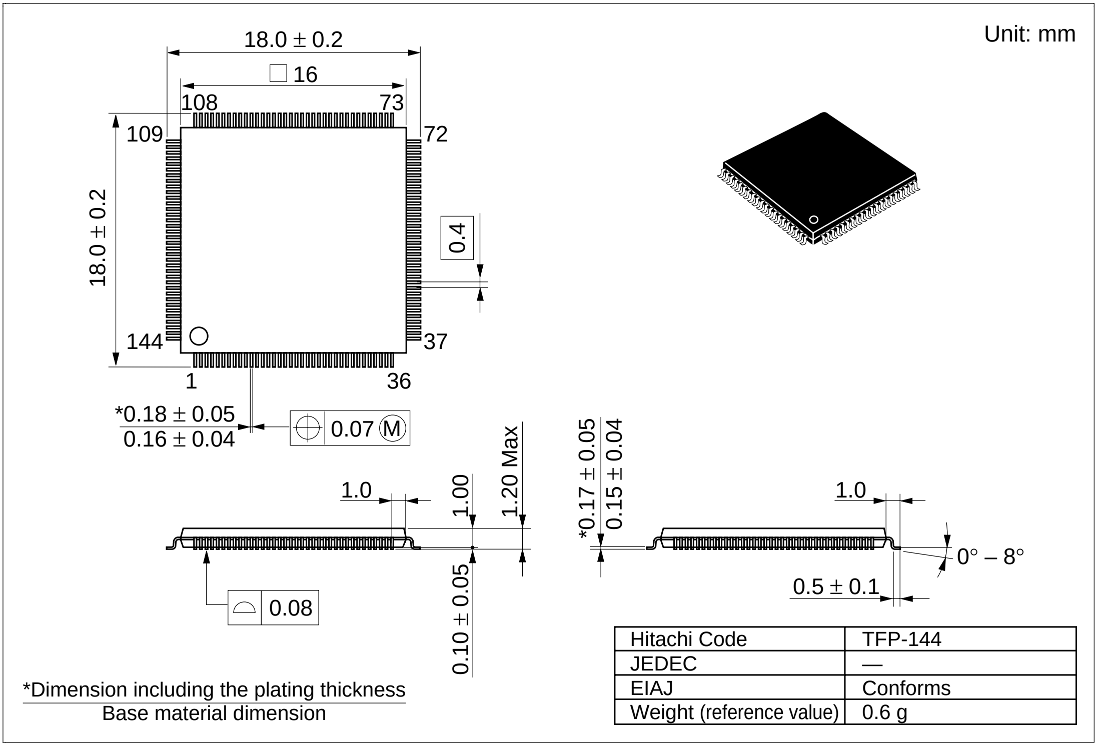
Appendix D   Package Dimensions



Hitachi Code	FP-144F
JEDEC	—
EIAJ	Conforms
Weight (reference value)	1.4 g

*Dimension including the plating thickness
Base material dimension

Figure D.1   Package Dimensions (FP-144F: SH7708, SH7708S, SH7708R)



*Dimension including the plating thickness
Base material dimension

Hitachi Code	TFP-144
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.6 g

Figure D.2 Package Dimensions (TFP-144: SH7708S)

SH7708 Series Hardware Manual

Publication Date: 1st Edition, November 1996
9th Edition, September 2000
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.